

**PEMODELAN SISTEM PAKAR UNTUK IDENTIFIKASI
KERUSAKAN *PRINTER* MENGGUNAKAN METODE
*DEMPSTER - SHAFER***

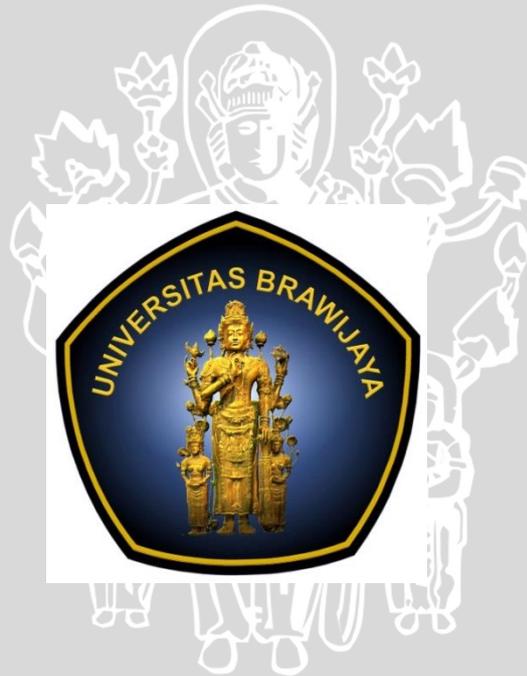
SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:

Idris Mukhni

NIM: 105060800111106



PROGRAM STUDI INFORMATIKA / ILMU KOMPUTER
PROGRAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA

MALANG

2016

PENGESAHAN

PEMODELAN SISTEM PAKAR UNTUK IDENTIFIKASI KERUSAKAN *PRINTER*
MENGUNAKAN METODE *DEMPSTER - SHAFER*

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :

Idris Mukhni

NIM: 105060800111106

Skripsi ini telah diuji dan dinyatakan lulus pada
21 Januari 2016

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II

Nurul Hidayat, S.Pd., M.Sc.

NIP: 19680430 200212 1 001

/*jika terdapat NIK saat ini

Drs. Marji, M.T.

NIP: 19670801 199203 1 001

/*jika terdapat NIP, NIK, atau
lainnya*/

Mengetahui

Ketua Program Studi Informatika/Ilmu Komputer

Drs. Marji, M.T.

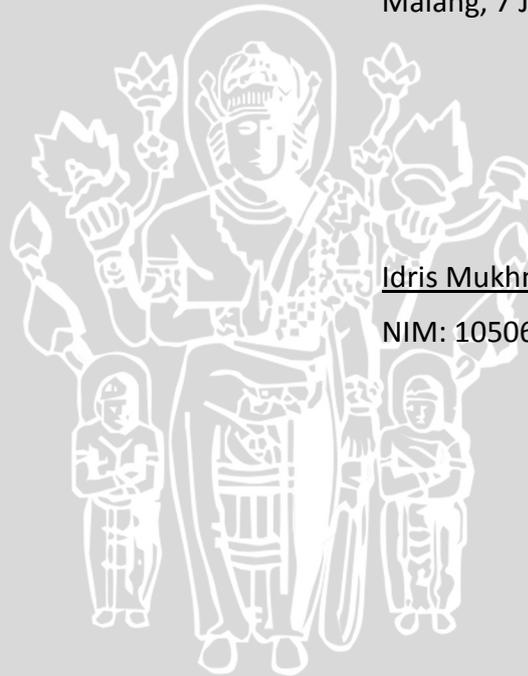
NIP: 19670801 199203 1 001

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 7 Januari 2016



Idris Mukhni

NIM: 105060800111106

KATA PENGANTAR

Puji syukur kehadiran Allah SWT karena atas segala rahmat dan karunia-Nya, penulis dapat menyelesaikan penyusunan skripsi sebagai syarat untuk memperoleh gelar sarjana dengan judul **“Pemodelan Sistem Pakar Untuk Identifikasi Kerusakan *Printer* Menggunakan Metode *Dempster-Shafer*”**.

Penulis menyadari bahwa skripsi ini tidak akan dapat diselesaikan dengan baik tanpa keterlibatan dari berbagai pihak. Untuk itu, penulis menyampaikan ucapan terima kasih yang sebesar-besarnya kepada :

1. Nurul Hidayat,S.Pd.,MSc selaku dosen pembimbing pertama yang telah memberikan bimbingan dan masukan dalam penulisan skripsi ini.
2. Drs.Marji,MT., selaku dosen prmbimbing dua Ketua Program Studi Informatika/Ilmu Komputer Universitas Brawijaya
3. Kedua orang tua atas segala doa, nasihat, dan perhatiannya dalam mendampingi dan memberikan dukungan moral kepada penulis.
4. Segenap Bapak dan Ibu dosen atas kesediaan membagi ilmu kepada penulis selama menempuh pendidikan di Program Studi Informatika dan Ilmu Komputer Universitas Brawijaya.
5. Bapak Nanang Muis selaku teknisi *printer* Maskomlaptop.com yang telah memberikan pengetahuan seputar *printer* terutama memberikan data dan penjelasan mengenai kerusakan *printer* untuk menyelesaikan skripsi ini.
6. Teman-teman seperjuangan, Pitaloka, Dini, Emon, Broto, Nico, Fikar, Rani, Umar, Sulton, Edwin, Iqbal, Alfin, Fendi, Putu, Wawan dan Abi yang tak henti-hentinya selalu berbagi semangatnya dan saling berbagi ilmu untuk menyelesaikan skripsi.
7. Teman-teman TIF angkatan 2009,2010, dan 2011 yang telah memberikan dorongan dan bantuan dalam berbagai hal selama penulis menempuh pendidikan di Universitas Brawijaya.

Semoga segala pertolongan dan kebaikan semuanya mendapatkan berkah dan balasan dari Allah SWT. Penulis menyadari bahwa skripsi ini masih jauh dari kata sempurna, oleh karena itu kritik dan saran yang sifatnya membangun sangat diperlukan untuk memperbaiki mutu penulisan selanjutnya dan juga kebaikan penulis secara pribadi. Akhir kata, penulis berharap semoga skripsi ini dapat bermanfaat bagi semua pihak.

Malang, 7 Januari 2016

Penulis
mughni7@gmail.com

ABSTRAK

Idris Mukhni, 2016. Pemodelan Sistem Pakar Untuk Identifikasi Kerusakan Printer Menggunakan Metode Dempster-Shafer. Skripsi Program Studi Informatika/ Ilmu Komputer, Fakultas Ilmu Komputer Universitas Brawijaya. Pembimbing : Nurul Hidayat, S.Pd., M.Sc. dan Drs. Marji, M.T.

Printer merupakan salah satu jenis perangkat output pada sebuah komputer dimana untuk menghasilkan sebuah catatan dalam bentuk *hardcopy*. Namun fenomena yang terjadi saat ini sering ditemukan masalah-masalah pada *printer* terutama pada *printer* jenis *inkjet*. Masalah-masalah yang sering terjadi antara lain adalah *printer* tidak bisa mencetak, hasil cetakan tidak bagus, dan lain-lain. Dalam penanganan masalah tersebut perlu adanya *diagnosis* awal pada *printer* yang bermasalah. Namun pada umumnya sistem *diagnosis* kerusakan pada *printer* masih harus melibatkan seorang pakar secara langsung (dalam hal ini teknisi) dan selain itu masih bersifat manual. Dari kondisi tersebut penulis berupaya membuat alat bantu berupa pemodelan sistem dengan menggunakan metode *dempster-shafer* dalam menentukan identifikasinya. Setiap gejala kerusakan akan diberi nilai densitas yang didapat dari pengetahuan seorang pakar. Nilai densitas gejala akan diproses dengan rumus *dempster-shafer*. Berdasarkan hasil pengujian fungsionalitas sistem dapat berjalan sesuai kebutuhan dengan prosentase 100%. Sedangkan tingkat akurasi sistem sebesar 84% yang diujikan pada 25 kasus dimana terdapat 2 kasus yang memiliki hasil deteksi berbeda antara sistem dan pakar. Hal ini menunjukkan pemodelan sistem pakar sudah cukup baik dan dapat diterapkan.

Kata kunci: Pemodelan sistem pakar, *Dempster-shafer*, *printer*.

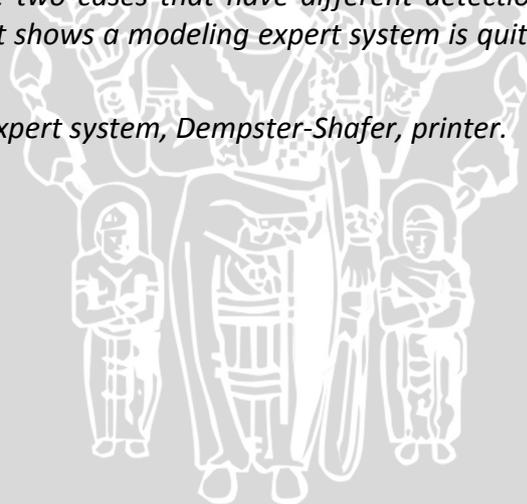


ABSTRACT

Idris Mukhni, 2016. *Modeling Expert System For Printer Damage Identification Using Dempster-Shafer Method. Thesis Study Program Informatics / Computer Science, Faculty of Computer Science Brawijaya University*
Advisor : Nurul Hidayat, S.Pd., M.Sc. and Drs. Marji, M.T.

Printer is a type of output device on a computer in which to make a printout. But the phenomenon is happening today is often found problems with the printer, especially on the types of printer inkjets. The problems often occur include the printer can not print, the print is not good, and others. In handling these issues should the need for early diagnosis of the troubled printer. But in general, the diagnosis of damage to the printer system still have to involve an expert directly (in this case the technician) and besides it is still manual. Of these conditions the authors attempt to create tools in the form of a system using Dempster-Shafer method to determine identification. Each of symptoms will be assigned density values obtained from the knowledge of an expert. The density of symptoms will be processed by the formula Dempster-Shafer. Based on the results of testing the functionality of the system can be run in accordance with the percentage requirement of 100%. While the level of system accuracy of 84% were tested on 25 cases where there are two cases that have different detection results between systems and experts. It shows a modeling expert system is quite good and can be applied.

Keywords: Modeling expert system, Dempster-Shafer, printer.



DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	v
ABSTRACT	vi
DAFTAR ISI.....	vii
DAFTAR GAMBAR.....	xi
DAFTAR TABEL.....	xi
DAFTAR PERSAMAAN.....	xvi
BAB 1 PENDAHULUAN.....	1
1.1 Latar belakang.....	1
1.2 Rumusan masalah.....	2
1.3 Tujuan.....	2
1.4 Manfaat.....	2
1.5 Batasan masalah.....	2
1.6 Sistematika pembahasan.....	3
BAB 2 LANDASAN KEPUSTAKAAN	5
2.1 Kajian Pustaka.....	5
2.2 Pemodelan Sistem.....	5
2.2.1 Prinsip Pemodelan Sistem.....	6
2.2.2 Perangkat Pemodelan Sistem	6
2.3 Sistem Pakar.....	13
2.3.1 Konsep Dasar Sistem Pakar	13
2.3.2 Tujuan Sistem Pakar	13
2.3.3 Bentuk Sistem Pakar	14
2.3.4 Ciri-Ciri Sistem Pakar	14
2.3.5 Keuntungan Sistem Pakar	15
2.3.6 Kelemahan Sistem Pakar	15
2.3.7 Struktur Sistem Pakar.....	15
2.3.8 Representasi Pengetahuan	17



2.3.9 Basis Pengetahuan	18
2.3.10 Mesin Inferensi	18
2.4 Ketidakpastian.....	19
2.5 <i>Printer Inkjet</i>	20
2.5.1 Kelebihan dan Kekurangan <i>Printer Inkjet</i>	20
2.5.2 Komponen <i>Printer inkjet</i>	21
2.6 Teori <i>Dempster-shafer</i>	24
2.7 Pengujian Sistem.....	27
2.7.1 Pengujian Validasi (<i>Black Box</i>)	27
2.7.2 Pengujian Akurasi.....	27
BAB 3 METODOLOGI PENELITIAN	29
3.1 Studi Literatur	29
3.2 Pengumpulan Data.....	30
3.3 Analisis Kebutuhan.....	30
3.4 Perancangan Sistem	30
3.5 Implementasi Sistem.....	31
3.6 Pengujian dan Analisis Sistem.....	32
BAB 4 PERANCANGAN.....	33
4.1 Analisis Kebutuhan Perangkat Lunak.....	34
4.1.1 Identifikasi Pengguna	34
4.1.2 Analisis Kebutuhan Masukan	35
4.1.3 Analisis Kebutuhan Proses	35
4.1.4 Analisis Kebutuhan Keluaran	35
4.2 Perancangan Sistem Pakar	35
4.2.1 Akuisisi Pengetahuan	36
4.2.2 Basis Pengetahuan	39
2.3.4 Mesin Inferensi.....	44
4.5.3 <i>Blackboard</i>	48
4.5.4 Fasilitas Penjelas	48
4.5.6 Perhitungan Manual.....	49
4.5.7 Antarmuka.....	52
4.6 Perancangan Perangkat Lunak Pemodelan Sistem Pakar Untuk Identifikasi Kerusakan <i>Printer</i>	59



4.6.1 Entity Relationship Diagram (ERD)	59
4.6.2 Physical Diagram	60
4.6.3 Perancangan Data Flow Diagram (DFD)	61
BAB 5 IMPLEMENTASI	66
5.1 Spesifikasi Sistem.....	67
5.1.1 Spesifikasi Perangkat Keras.....	67
5.1.2 Spesifikasi Perangkat Lunak	67
5.2 Batasan – Batasan Implementasi.....	67
5.3 Implementasi Algoritma <i>Dempster-shafer</i>	68
5.4 Implementasi Antarmuka.....	71
5.4.1 Implementasi Halaman Utama	72
5.4.2 Implementasi Halaman Identifikasi dan Hasil Identifikasi.....	72
5.4.3 Implementasi Halaman <i>About</i>	73
5.4.4 Implementasi Halaman Bantuan	74
5.4.5 Implementasi Halaman <i>Login</i>	74
5.4.7 Implementasi Halaman Data Kerusakan	75
5.4.7 Implementasi Halaman Data Gejala	76
5.4.8 Implementasi Halaman Basis Pengetahuan	76
5.4.9 Implementasi Halaman Ganti <i>Password</i> Admin	77
5.4.10 Implementasi Halaman Tambah/Ubah Data Kerusakan.....	77
5.4.11 Implementasi Halaman Tambah/Ubah Data Gejala.....	78
5.4.12 Implementasi Halaman Tambah/Ubah Basis Pengetahuan....	78
BAB 6 PENGUJIAN	79
6.1 Pengujian.....	80
6.1.1 Pengujian Validasi	80
6.1.2 Pengujian Akurasi.....	96
6.2 Analisis.....	101
6.2.1 Analisis Hasil Pengujian Validasi.....	101
6.2.2 Analisis Hasil Pengujian Akurasi.....	101
BAB 7 PENUTUP.....	102
7.1 Kesimpulan.....	102

7.2 Saran..... 102
DAFTAR PUSTAKA.....103



DAFTAR GAMBAR

Gambar 2.1 Terminator.....	8
Gambar 2.2 Proses.....	8
Gambar 2.3 Arus Data.....	9
Gambar 2.4 <i>Data Store</i>	9
Gambar 2.5 Simbol Entity.....	9
Gambar 2.6 Simbol Relationship.....	10
Gambar 2.7 Simbol Atribut.....	10
Gambar 2.8 Struktur Sistem Pakar.....	16
Gambar 2.9 Alur Metode <i>Forward Chaining</i>	19
Gambar 2.9 Alur Metode <i>Backward Chaining</i>	19
Gambar 2.11 <i>Power supply</i>	21
Gambar 2.12 <i>Mainboard</i>	21
Gambar 2.13 <i>Cartridge</i>	22
Gambar 2.14 Sensor ASF Penarik Kertas.....	22
Gambar 2.15 <i>Roll</i> ASF Penarik Kertas.....	23
Gambar 2.16 Rumah <i>Cartridge</i>	23
Gambar 2.17 Sensor Pita <i>Encoder</i>	23
Gambar 2.18 Sensor <i>Timing Disk</i>	24
Gambar 2.19 <i>Scanner</i>	24
Gambar 3.1 Alur Metodologi Penelitian.....	29
Gambar 3.2 Desain Umum Perancangan Sistem.....	31
Gambar 4.1 Diagram Perancangan.....	33
Gambar 4.2 Kerangka Konsep Arsitektur Sistem Pakar Untuk Identifikasi Kerusakan <i>Printer</i>	36
Gambar 4.3 Mesin Inferensi <i>Forward Chaining</i> Dengan Metode <i>Dempster-Shafer</i>	45
Gambar 4.4 Diagram Alir Proses Sistem Identifikasi Kerusakan <i>Printer</i>	46
Gambar 4.5 Diagram Alir Inferensi <i>Dempster-shafer</i>	47
Gambar 4.6 Rancangan Algoritma Proses Identifikasi Kerusakan <i>Printer</i>	48
Gambar 4.7 Rancangan Antarmuka Halaman Utama.....	52
Gambar 4.8 Rancangan Antarmuka Halaman Identifikasi.....	53

Gambar 4.9 Rancangan Antarmuka Hasil Identifikasi.....	53
Gambar 4.10 Rancangan Antarmuka Halaman <i>about</i>	54
Gambar 4.11 Rancangan Antarmuka Halaman <i>Login Admin</i>	54
Gambar 4.12 Rancangan Antarmuka Halaman Bantuan.....	55
Gambar 4.13 Rancangan Antarmuka Halaman Admin.....	55
Gambar 4.14 Rancangan Antarmuka Halaman Data Gejala Kerusakan.....	56
Gambar 4.15 Rancangan Antarmuka Halaman Data Kerusakan.....	56
Gambar 4.16 Rancangan Antarmuka Halaman Basis Pengetahuan.....	57
Gambar 4.17 Rancangan Antarmuka Halaman Ganti <i>Password Admin</i>	57
Gambar 4.18 Rancangan Antarmuka Halaman Tambah/Ubah Data Kerusakan.....	58
Gambar 4.19 Rancangan Antarmuka Halaman Tambah/Ubah Data Gejala.....	58
Gambar 4.20 Rancangan Antarmuka Halaman Tambah/Ubah Data Kerusakan.....	59
Gambar 4.21 Rancangan <i>Entity Relationship Diagram</i> sistem.....	60
Gambar 4.22 Rancangan <i>Physical Diagram</i>	60
Gambar 4.23 <i>Data Flow Diagram</i> Level 0 Sistem Identifikasi Kerusakan <i>Printer</i>	61
Gambar 4.24 <i>Data Flow Diagram</i> Level 1 Sistem Identifikasi Kerusakan <i>Printer</i>	62
Gambar 4.25 <i>Data Flow Diagram</i> Level 2 Sub Proses Gejala.....	63
Gambar 4.26 <i>Data Flow Diagram</i> Level 2 Sub Proses Kerusakan.....	63
Gambar 4.27 <i>Data Flow Diagram</i> Level 2 Sub Proses Basis Pengetahuan.....	64
Gambar 4.28 <i>Data Flow Diagram</i> Level 2 Sub Proses Login.....	64
Gambar 4.29 <i>Data Flow Diagram</i> Level 2 Sub Proses Identifikasi.....	65
Gambar 5.1 Diagram Implementasi.....	66
Gambar 5.2 Implementasi Algoritma Proses Identifikasi.....	71
Gambar 5.3 Antarmuka Halaman Utama.....	72
Gambar 5.4 Antarmuka Halaman Identifikasi.....	72
Gambar 5.5 Antarmuka Hasil Identifikasi.....	73
Gambar 5.6 Antarmuka Halaman <i>about</i>	73
Gambar 5.7 Antarmuka Halaman Bantuan.....	74
Gambar 5.8 Antarmuka Halaman <i>Login Admin</i>	74

Gambar 5.9 Antarmuka Halaman Admin.....75

Gambar 5.10 Antarmuka Halaman Data Kerusakan75

Gambar 5.11 Antarmuka Halaman Data Kerusakan.....76

Gambar 5.12 Antarmuka Halaman Basis Pengetahuan.....76

Gambar 5.13 Antarmuka Halaman Ganti *Password* Admin.....77

Gambar 5.14 Antarmuka Halaman Tambah/Ubah Data Kerusakan.....77

Gambar 5.15 Antarmuka Halaman Tambah/Ubah Data Gejala.....78

Gambar 5.16 Antarmuka Halaman Tambah/Ubah Data Kerusakan78

Gambar 6.1 Diagram Pengujian dan Analisis.....79



DAFTAR TABEL

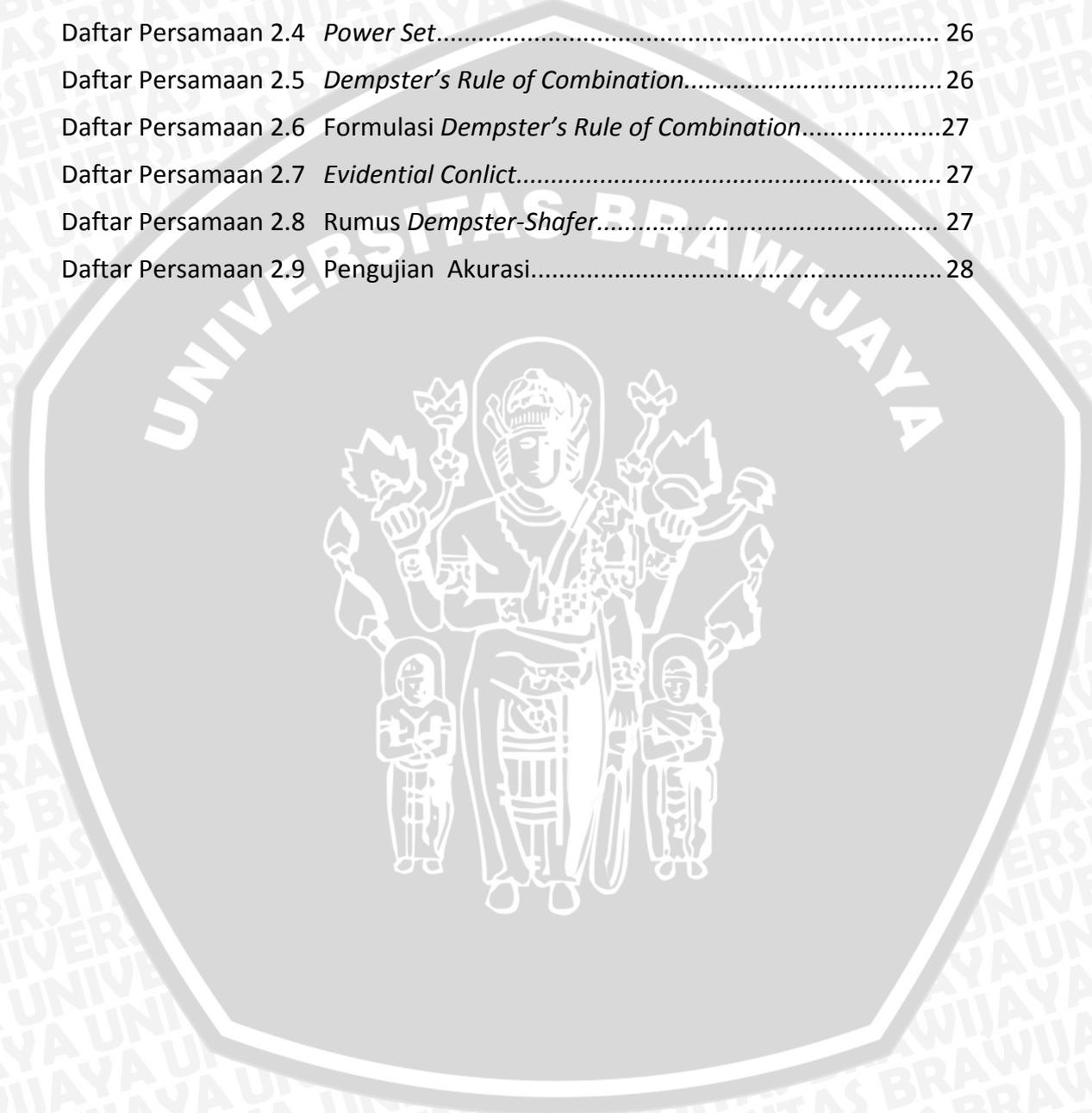
Tabel 2.1 Simbol-Simbol Diagram Konteks.....	7
Tabel 2.2 Simbol-Simbol <i>Flowchart</i>	11
Tabel 2.1 <i>Range Belief</i> dan <i>Plausibility</i>	25
Tabel 4.1 Deskripsi Pengguna.....	34
Tabel 4.2 Akuisisi Pengetahuan Gejala Kerusakan <i>Printer</i> dan Nilai Densitas.....	37
Tabel 4.3 Akuisisi Nilai Densitas Gejala Kerusakan <i>Printer</i>	40
Tabel 4.4 Aturan kerusakan	43
Tabel 4.5 Aturan Kombinasi m2.....	49
Tabel 4.6 Aturan Kombinasi m2.....	51
Tabel 4.7 Aturan Kombinasi m4.....	51
Tabel 5.1 Spesifikasi Perangkat Keras.....	67
Tabel 5.2 Spesifikasi Perangkat Lunak.....	67
Tabel 6.1 Kasus Uji untuk Pengujian Validasi Login Sah.....	80
Tabel 6.2 Kasus Uji untuk Pengujian Validasi <i>Login</i> Tidak Sah.....	81
Tabel 6.3 Kasus Uji untuk Pengujian Validasi Tambah Data Kerusakan Sah.....	81
Tabel 6.4 Kasus Uji untuk Pengujian Validasi Tambah Data Kerusakan Tidak Sah.....	82
Tabel 6.5 Kasus Uji untuk Pengujian Validasi Edit Data Kerusakan.....	83
Tabel 6.6 Kasus Uji untuk Pengujian Validasi Hapus Data Kerusakan.....	83
Tabel 6.7 Kasus Uji untuk Pengujian Validasi Tambah Data Gejala Sah.....	84
Tabel 6.8 Kasus Uji untuk Pengujian Validasi Tambah Data Gejala Tidak Sah.....	84
Tabel 6.9 Kasus Uji untuk Pengujian Validasi Edit Data Gejala.....	85
Tabel 6.10 Kasus Uji untuk Pengujian Validasi Hapus Data Gejala.....	85
Tabel 6.11 Kasus Uji untuk Pengujian Validasi Tambah Data Basis Pengetahuan Sah.....	86
Tabel 6.12 Kasus Uji untuk Pengujian Validasi Tambah Data Basis Pengetahuan Tidak Sah.....	86
Tabel 6.13 Kasus Uji untuk Pengujian Validasi Edit Basis Pengetahuan.....	87
Tabel 6.14 Kasus Uji untuk Pengujian Validasi Hapus Data Basis Pengetahuan.....	88

Tabel 6.15 Kasus Uji untuk Pengujian Ganti <i>Password</i>	88
Tabel 6.16 Kasus Uji untuk Proses <i>Logout</i>	89
Tabel 6.17 Kasus Uji untuk Menampilkan Profil Admin.....	89
Tabel 6.18 Kasus Uji untuk <i>Input</i> Data Fakta Gejala.....	89
Tabel 6.19 Kasus Uji untuk Informasi HasilIdentifikasi.....	90
Tabel 6.20 Hasil Pengujian Validasi.....	90
Tabel 6.21 Hasil Pengujian Akurasi hasil identifikasi sistem dengan pakar.....	96



DAFTAR PERSAMAAN

Daftar Persamaan 2.1 Fungsi <i>Belief</i>	25
Daftar Persamaan 2.2 Rumus <i>Plausibility</i>	25
Daftar Persamaan 2.3 <i>Frame of Discernment</i>	26
Daftar Persamaan 2.4 <i>Power Set</i>	26
Daftar Persamaan 2.5 <i>Dempster's Rule of Combination</i>	26
Daftar Persamaan 2.6 Formulasi <i>Dempster's Rule of Combination</i>	27
Daftar Persamaan 2.7 <i>Evidential Conlict</i>	27
Daftar Persamaan 2.8 Rumus <i>Dempster-Shafer</i>	27
Daftar Persamaan 2.9 Pengujian Akurasi.....	28



BAB 1 PENDAHULUAN

Pada bab ini berisi pembahasan tentang hal yang menjadi latar belakang topik, permasalahan yang akan dibahas dari topik, batasan dari analisis, tujuan dan manfaat dari topik yang akan dibahas, serta sistematika dalam penulisan skripsi.

1.1 Latar belakang

Saat ini *printer* merupakan kebutuhan utama masyarakat untuk melakukan pencetakan suatu dokumen digital baik berupa teks maupun gambar. Sehingga *printer* merupakan alat yang lazim digunakan masyarakat, sama halnya dalam penggunaan laptop atau komputer. *Printer* merupakan salah satu jenis perangkat *output* pada sebuah komputer dimana untuk menghasilkan sebuah catatan dalam bentuk *hardcopy*. Selain itu adanya *print out* ini, menjadikan tanda jika ada sebuah proses yang saling terhubung antara keyboard yang ada pada PC dengan perangkat CPU (McLoughlin,2011). Saat ini semakin banyaknya pengguna *printer* tentunya mempengaruhi industri *printer* untuk berlomba lomba membuat *printer* dengan variasi dan keunggulan yang menarik. Namun semua itu tak luput dari adanya kendala atau kerusakan yang terjadi pada *printer*. Dan juga tingginya ketergantungan sehari hari dengan *printer* tidak diimbangi dengan kemampuan masyarakat untuk menganalisis sejak dini permasalahan yang ada pada *printer*. dikarenakan banyaknya komponen komponen yang ada pada *printer* sehingga pengguna mengalami kesulitan dalam mengidentifikasi kerusakan *printer*. Keadaan tersebut mengakibatkan pengguna *printer* memiliki ketergantungan yang tinggi terhadap pakar atau teknisi *printer*. Padahal tidak semua kerusakan harus dibawa ke pakar atau teknisi *printer*, sehingga menghabiskan biaya dan waktu yang tidak sedikit. Sistem ini nantinya akan mengidentifikasi kerusakan *printer* secara lebih mendalam, dan akan menghasilkan *output* berupa kerusakan yang lebih spesifik. Tidak seperti *troubleshooter* yang telah dikeluarkan oleh produsen printer, yang hanya mengidentifikasi kerusakan secara sekilas dan tidak spesifik.

Adapun penelitian sebelumnya yang mengangkat topik yang sama yaitu *mediagnosis* kerusakan pada sebuah *printer*. Penelitiannya berjudul “Aplikasi Sistem Pakar Pendeteksi Kerusakan *Printer* Dengan *Case Based Reasoning*” oleh Suriyanti, pada penelitian ini dibangunnya sebuah sistem pakar dengan menggunakan bahasa pemrograman Visual Basic dimana menerapkan metode *Case Based Reasoning* yaitu metode yang menerapkan 4 tahapan proses seperti *retrieve, reuse, revise, dan retain*. Cara kerja sistem ini menganalisis kemiripan diagnosis kerusakan *printer* melalui riwayat diagnosis sebelumnya sehingga menghasilkan diagnosis baru yang lebih akurat. Sistem ini berbasis konsultasi dimana pengguna memiliki sebuah form gejala yang kemudian diproses hingga ditemukan sebuah diagnosis (Suriyanti,2013)

Metode *Dempster-Shafer* merupakan metode penalaran *non monotonis* yang digunakan untuk mencari ketidakkonsistenan akibat adanya penambahan maupun pengurangan fakta baru yang akan merubah aturan yang ada, sehingga metode *Dempster-Shafer* memungkinkan seseorang aman dalam melakukan pekerjaan seorang pakar (Wahyuni,2013). Adapun penelitian yang menggunakan

metode *dempster-shafer* berjudul “Implementasi Metode *Dempster-shafer* Pada Sistem Pakar Untuk *Diagnosis* Jenis-jenis Penyakit *Diabetes Melitus*” oleh Dewi Pratama Kurniawati. Kesimpulan yang didapatkan yaitu metode *Dempster-shafer* telah berhasil digunakan untuk mendiagnosis jenis-jenis penyakit *diabetes mellitus* dengan masukan berupa gejala-gejala yang dimasukkan oleh pengguna. Pada penelitian ini, hasil pengujian menunjukkan uji akurasi sebesar 96,67% dengan data uji sebanyak 30 kasus (Kurniawati,2014).

Berdasarkan pemaparan yang dijelaskan diatas, maka akan dirancanglah sebuah sistem pakar untuk mendiagnosis kerusakan *printer* dengan metode *dempster-shafer*. Dimana sistem berbasis konsultasi dengan adanya form konsul untuk pengguna, yang kemudian hasil output sistem berupa *diagnosis* kerusakan pada *printer*.

1.2 Rumusan masalah

Berdasarkan uraian pada latar belakang, maka didapatkan rumusan masalah sebagai berikut :

1. Bagaimana memodelkan sistem pakar untuk identifikasi kerusakan sebuah *printer* menggunakan metode *Dempster-Shafer*.
2. Bagaimana hasil pengujian sistem identifikasi kerusakan pada *printer* menggunakan metode *Dempster-Shafer*.

1.3 Tujuan

Tujuan yang ingin dicapai dalam penelitian ini adalah :

1. Memodelkan sistem pakar untuk identifikasi kerusakan *printer* menggunakan metode *Dempster-Shafer*.
2. Menguji pemodelan sistem pakar yang dirancang menggunakan metode *Dempster-Shafer* dalam mendeteksi kerusakan *printer*.

1.4 Manfaat

Manfaat yang diharapkan dari penelitian skripsi ini adalah membantu masyarakat awam untuk mengenali dan mengidentifikasi kerusakan pada *printer* masing – masing yang mampu memberikan solusi mengenai cara mencegah dan cara mengatasinya .

1.5 Batasan masalah

Beberapa ruang lingkup masalah untuk memfokuskan penelitian, diantaranya yaitu :

1. Metode inferensi yang digunakan untuk menghasilkan informasi dari fakta yang diketahui adalah *Forward Chaining*.
2. Data yang digunakan sebagai dasar deteksi berupa jenis-jenis kerusakan beserta gejala pada setiap kerusakan *printer* dalam penelitian ini diperoleh dari seorang teknisi *printer* sebuah tempat servis *printer* yaitu MaskomLaptop.com yang beralamat di Jl. Gajayana No. 20 A3 Malang.
3. Data kerusakan yang digunakan dalam penelitian ini adalah kerusakan yang terjadi pada *printer Inkjet* dengan merk Canon. Terdapat sebanyak 30 gejala kerusakan dan 13 jenis kerusakan *printer*.

4. Keluaran yang dihasilkan aplikasi ini yaitu mendeteksi kerusakan beserta cara menanggulangnya.
5. Pengujian yang dilakukan dengan cara *black box testing* dan pengujian tingkat akurasi.
6. Bahasa pemrograman yang digunakan dalam implementasi yaitu bahasa pemrograman PHP.
7. *Database* yang digunakan untuk penyimpanan data gejala, kerusakan dan nilai densitas adalah MySQL.

1.6 Sistematika pembahasan

Sistematika Penelitian ini menggunakan kerangka penulisan sebagai berikut:

BAB I

PENDAHULUAN

Bab ini berisi pembahasan masalah umum yang meliputi latar belakang pemilihan judul skripsi, rumusan masalah, tujuan penelitian, manfaat penelitian, batasan masalah, sistematika penulisan. Bab I menjadi dasar dari keseluruhan pelaksanaan penelitian mengenai pemodelan sistem pakar untuk identifikasi kerusakan *printer* menggunakan metode *Dempster-Shafer*.

BAB II

KAJIAN PUSTAKA DAN DASAR TEORI

Bab II membahas kajian pustaka yang menjadi dasar acuan dilakukannya penelitian mengenai pemodelan sistem pakar untuk identifikasi kerusakan *printer* menggunakan metode *Dempster-Shafer* serta membahas teori-teori yang berkaitan dan menunjang dalam penyelesaian penelitian. Dasar teori yang diambil berasal dari jurnal, buku dan sumber referensi lainnya yang berhubungan dengan topik yang diteliti.

BAB III

METODOLOGI PENELITIAN

Bab III menjelaskan metode yang digunakan dan langkah kerja yang dilakukan dalam penulisan yang terdiri dari studi literatur, pengumpulan data, analisis kebutuhan, perancangan sistem, implementasi sistem, pengujian dan analisis sistem dan pengambilan kesimpulan.

BAB IV

PERANCANGAN

Bab IV menjelaskan tentang perancangan desain sistem secara detail tentang pemodelan sistem pakar untuk identifikasi kerusakan *printer* menggunakan metode *Dempster-Shafer*

BAB V

IMPLEMENTASI

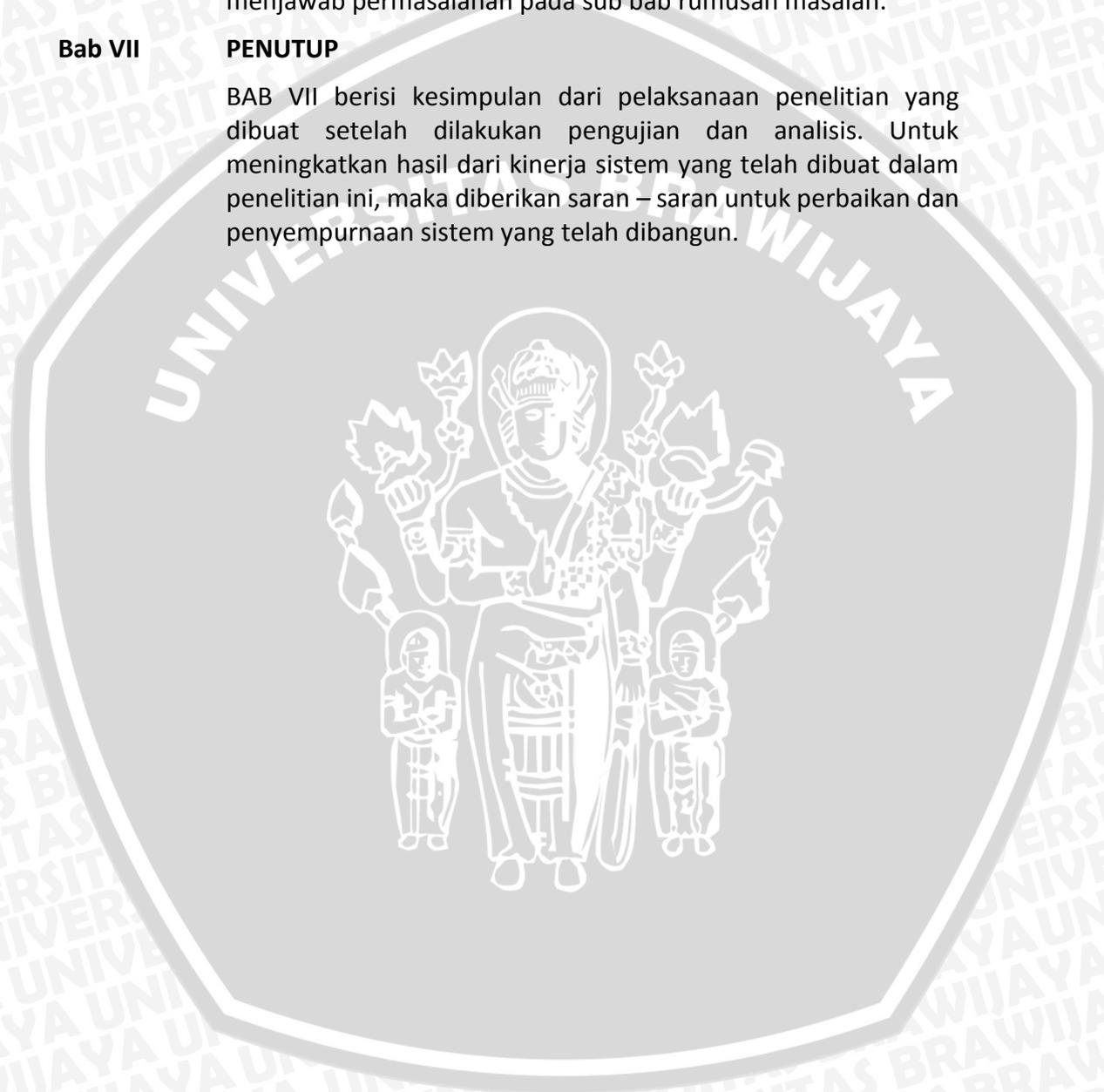
BAB V membahas implementasi perangkat lunak aplikasi pemodelan sistem pakar untuk identifikasi kerusakan *printer* menggunakan metode *Dempster-Shafer*.

Bab VI PENGUJIAN DAN ANALISIS

Bab VI membahas tentang pengujian pada sistem yang telah dibangun. Pengujian ini dilakukan secara bertahap sesuai dengan penjelasan pada Bab III. Dan juga dilakukan analisis atas hasil pengujian terhadap sistem yang telah direalisasikan sehingga menjawab permasalahan pada sub bab rumusan masalah.

Bab VII PENUTUP

BAB VII berisi kesimpulan dari pelaksanaan penelitian yang dibuat setelah dilakukan pengujian dan analisis. Untuk meningkatkan hasil dari kinerja sistem yang telah dibuat dalam penelitian ini, maka diberikan saran – saran untuk perbaikan dan penyempurnaan sistem yang telah dibangun.



BAB 2 LANDASAN KEPUSTAKAAN

Bab ini berisi kajian pustaka dan pembahasan tentang dasar teori yang diperlukan untuk penelitian. Kajian pustaka membahas penelitian yang telah ada sebelumnya. Dasar teori membahas teori yang diperlukan untuk menyusun penelitian yang diusulkan.

2.1 Kajian Pustaka

Penelitian mengenai sistem pakar untuk identifikasi kerusakan *printer* bukanlah baru pertama kali ini dilakukan. Sudah ada penelitian terdahulu tentang penerapan metode *Dempster-Shafer* tersebut.

Penelitian pertama berjudul “Aplikasi Sistem Pakar Pendeteksi Kerusakan *Printer* Dengan *Case Based Reasoning*” oleh Suriyanti, pada penelitian ini dibangunnya sebuah sistem pakar dengan menggunakan bahasa pemrograman Visual Basic dimana menerapkan metode *Case Based Reasoning* yaitu metode yang menerapkan 4 tahapan proses seperti *retrieve*, *reuse*, *revise*, dan *retain*. Cara kerja sistem ini menganalisis kemiripan diagnosis kerusakan *printer* melalui histori diagnosis sebelumnya sehingga menghasilkan diagnosis baru yang lebih akurat. Sistem ini berbasis konsultasi dimana pengguna memiliki sebuah form gejala yang kemudian diproses hingga ditemukan sebuah diagnosis. (Suriyanti, 2013)

Penelitian kedua berjudul “Implementasi Metode *Dempster-shafer* Pada Sistem Pakar Untuk Diagnosis Jenis-jenis Penyakit *Diabetes Mellitus*” oleh Dewi Pratama Kurniawati. Kesimpulan yang didapatkan yaitu metode *Dempster-shafer* telah berhasil digunakan untuk mendiagnosis jenis-jenis penyakit *diabetes mellitus* dengan masukan berupa gejala-gejala yang dimasukkan oleh pengguna. Pada penelitian ini, hasil pengujian menunjukkan uji akurasi sebesar 96,67% dengan data uji sebanyak 30 kasus. Ini menunjukkan bahwa metode *Dempster-shafer* berfungsi dengan baik sesuai diagnosis pakar (Dewi,2014).

Penelitian ketiga berjudul “Sistem Pakar Pendeteksi Hama Dan Penyakit Pada Tanaman Cabai Dengan Metode *Dempster-Shafer*” yang dibuat oleh Innes Yunia Lufitasari (2015). Sistem pakar pendeteksi hama penyakit tanaman cabai ini dapat mendeteksi 16 hama penyakit tanaman cabai dengan 49 gejala sebagai masukan yang selanjutnya diolah dengan metode *Dempster-shafer*. Keluaran sistem berupa hasil deteksi hama penyakit tanaman cabai beserta cara penanganannya. Hasil pengujian menunjukkan uji akurasi sebesar 85% dengan data uji sebanyak 25 kasus yang menunjukkan bahwa sistem pakar dapat berfungsi dengan baik dan memiliki hasil sesuai dengan diagnosis pakar. (Lufitasari,2015)

2.2 Pemodelan Sistem

Model adalah adalah rencana, representasi, atau deskripsi yang menjelaskan suatu objek, sistem, atau konsep, yang seringkali berupa penyederhanaan atau idealisasi. Bentuknya dapat berupa model fisik (maket, bentuk prototipe), model citra (gambar, komputerisasi, grafis), atau rumusan

matematis. Sedangkan sistem adalah suatu kesatuan yang terdiri komponen atau elemen yang dihubungkan bersama untuk memudahkan aliran informasi, materi atau energi.

Pemodelan Sistem adalah suatu bentuk penyederhanaan dari sebuah elemen dan komponen yang sangat kompleks untuk memudahkan pemahaman dari informasi yang dibutuhkan (Wijaya,2010).

Karakteristik dari Pemodelan Sistem adalah sebagai berikut (Wijaya,2010).

1. Dibuat dalam bentuk grafis dan tambahan keterangan secara tekstual.
 2. Dapat diamati dengan pola *top-down* dan *partitioned*.
 3. Memenuhi persyaratan minimal *redundancy*.
 4. Dapat mempresentasikan tingkah laku sistem dengan cara yang transparan.
- Pemodelan sistem dibuat dalam bentuk grafis atau bergambar sehingga dapat memudahkan *customer* dan dilengkapi juga dengan keterangan dari gambar atau grafis tersebut. Alur dari proses model tersebut dapat di lihat dan diamati dan dapat mempresentasikan proses dari pada sistem yang dibuat dan dapat di pahami oleh *customer*.

2.2 1 Prinsip Pemodelan Sistem

Prinsip dari Pemodelan sistem menurut Grady Booch, James Rumbaugh dan Ivar Jacobson adalah:

1. Memilih model apa yang digunakan, bagaimana masalahnya dan bagaimana juga dengan solusinya.
2. Setiap model dapat dinyatakan dalam tingkatan yang berbeda
3. Model yang terbaik adalah yang berhubungan dengan realitas.
4. Tidak pernah ada model tunggal yang cukup baik, setiap sistem yang baik memiliki serangkaian model kecil yang independen.

Prinsip pemodelan sistem tidak terlalu menitik beratkan pada bentuk model apa untuk merancang sebuah sistem. Bentuk model ini bebas, sesuai dengan keinginan kita. Contohnya berupa narasi, *prototype*, maupun gambar. Yang terpenting adalah harus mampu merepresentasikan visualisasi bentuk sistem yang diinginkan oleh pengguna, karena sistem akhir yang dibuat bagi pengguna akan diturunkan dari hasil model tersebut (Fahraini,2014).

2.2.2 Perangkat Pemodelan Sistem

Dalam perancangan sebuah sistem langkah awal yang harus dilakukan adalah memodelkan sistem. Hal ini dilakukan untuk memfokuskan perhatian pada hal-hal penting dalam sistem tanpa terlibat terlalu jauh.

Ada beberapa macam perangkat pemodelan sistem yang dapat digunakan untuk memodelkan sistem, akan tetapi tidak mutlak untuk menggunakan semua perangkat pemodelan yang ada, artinya boleh menggunakan sebagian perangkat pemodelan dari beberapa perangkat yang ada. Perangkat yang digunakan untuk memodelkan sistem, diantaranya adalah (Wijaya,2010):

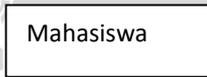
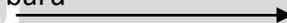
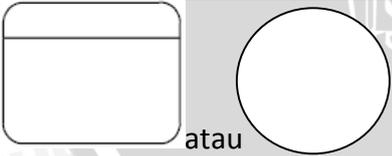
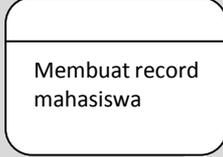
2.2.2.1 Context Diagram (Diagram Konteks)

Diagram konteks merupakan kejadian tersendiri dari suatu diagram alir data. Dimana satu lingkaran merepresentasikan seluruh sistem. Diagram konteks ini harus berupa suatu pandangan, yang mencakup masukan-masukan dasar, sistem-sistem dan keluaran.

Diagram konteks dimulai dengan penggambaran *terminator*, aliran data, aliran kontrol penyimpanan, dan proses tunggal yang menunjukkan keseluruhan sistem. Bagian termudah adalah menetapkan proses (yang hanya terdiri dari satu lingkaran) dan diberi nama yang mewakili sistem.

Nama dalam hal ini dapat menjelaskan proses atau pekerjaan berupa nama perusahaan yang dalam hal ini mewakili proses yang dilakukan keseluruhan organisasi. Terminator ditunjukkan dalam bentuk persegi panjang dan berkomunikasi langsung dengan sistem melalui aliran data atau penyimpanan eksternal. Antar terminator tidak diperbolehkan komunikasi langsung. Pada kenyataannya hubungan antar terminator dilakukan, tetapi secara definitif karena terminator adalah bagian dari lingkungan, maka tidak relevan jika dibahas dalam diagram konteks (Wijaya,2010). Simbol-simbol konteks diagram dapat dilihat pada Tabel 2.1.

Tabel 2.1 Simbol-Simbol Diagram Konteks

Simbol	Arti	Contoh
	Terminator	
	Aliran Data/ <i>Data Flow</i>	
	Proses	

Sumber: (Wijaya,2010)

2.2.2.2 Data Flow Diagram (DFD)

Data Flow Diagram (DFD) adalah alat pembuatan model yang memungkinkan profesional sistem untuk menggambarkan sistem sebagai suatu jaringan proses fungsional yang dihubungkan satu sama lain dengan alur data, baik secara manual maupun komputerisasi. DFD ini sering disebut juga dengan nama

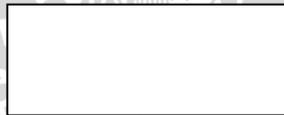
Bubble chart, Bubble diagram, model proses, diagram alur kerja, atau model fungsi.

DFD adalah salah satu alat pembuatan model yang sering digunakan, khususnya bila fungsi-fungsi sistem merupakan bagian yang lebih penting dan kompleks dari pada data yang dimanipulasi oleh sistem. Dengan kata lain, DFD adalah alat pembuatan model yang memberikan penekanan hanya pada fungsi sistem.

Adapun komponen-komponen yang digunakan dalam penggambaran diagram alur data adalah sebagai berikut (Wijaya,2010):

1. Terminator (Sumber)

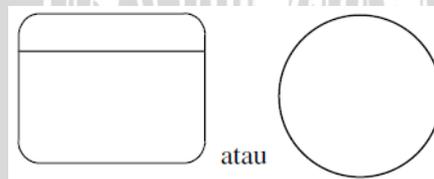
Setiap sistem akan mempunyai batas sistem yang memisahkan suatu sistem dengan lingkungan luarnya. Terminator merupakan kesatuan atau (*entity*) sumber atau tujuan di dilingkungan luar sistem yang dapat berupa orang, organisasi atau sistem lainnya yang berada dilingkungan luarnya yang akan memberikan input atau menerima output dari sistem. Simbol *context diagram* terminator dapat dilihat pada Gambar 2.1.



Gambar 2.1 Terminator

2. Proses

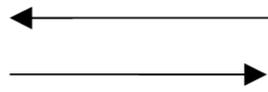
Suatu proses adalah kegiatan yang dilakukan atau komputer dari hasil suatu arus data yang masuk kedalam proses untuk dihasilkan arus data yang akan keluar dari proses. Simbol proses pada *context diagram* dapat dilihat pada Gambar 2.2.



Gambar 2.2 Proses

3. Arus Data (*Data Flow*)

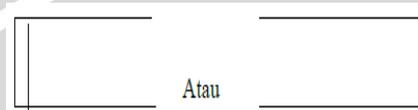
Suatu proses adalah kegiatan yang dilakukan atau komputer dari hasil suatu arus data yang masuk kedalam proses untuk dihasilkan arus data yang akan keluar Arus data menunjukkan arus dari data yang dapat berupa masukan untuk sistem atau hasil dari proses sistem. Arus data ini mengalir antara proses, simpanan data, dan kesatuan luar dari proses. Gambar 2.3 merupakan simbol arus data untuk *context diagram*.



Gambar 2.3 Arus Data

4. Simpanan Data (*Data Store*)

Simpanan data (*data store*) merupakan simpanan dari data yang dapat berupa *file*, arsip, tabel, dan agenda. *Data store* pada *context diagram* dapat dilihat pada Gambar 2.4.



Gambar 2.4 Data Store

2.2.2.3 Entity Relationship Diagram (ERD)

Entity Relationship Diagram (ERD) adalah sebuah diagram yang menggambarkan hubungan atau relasi antar entitas (*entity*), setiap entitas terdiri atas satu atau lebih atribut yang merepresentasikan seluruh kondisi atau fakta dari dunia nyata yang ditinjau. Dengan ERD dapat mentransformasikan keadaan dari dunia nyata ke dalam bentuk basis data.

Ada tiga macam komponen- komponen ERD yang digunakan yaitu:

1. *Entity*/objek data

Entity adalah kumpulan objek atau suatu yang dapat dibedakan atau dapat diidentifikasi secara unik, kumpulan entitas yang sejenis disebut *entity set*. Penggambaran entitas pada ERD menggunakan simbol persegi panjang. Simbol *Entity*/objek data dapat dilihat pada Gambar 2.5.

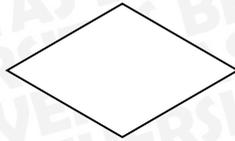


Gambar 2.5 Simbol Entity

2. *Relationship*

Relationship adalah Hubungan yang terjadi antara satu entitas atau lebih, kumpulan *relationship* yang sejenis disebut *Relationship set*. Hubungan digambarkan dengan bentuk belah ketupat, tiap belah ketupat diberi label kata kerja. Simbol ERD *Relationship* dapat ditunjukkan pada Gambar 2.6.





Gambar 2.6 Simbol Relationship

3. Atribut

Atribut merupakan sifat atau karakteristik suatu entitas yang menyediakan penjelasan detail tentang entitas tersebut. Simbol Atribut dapat ditunjukkan pada Gambar 2.7.



Gambar 2.7 Simbol Atribut

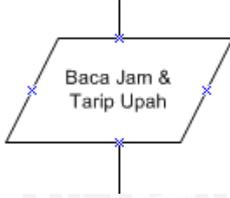
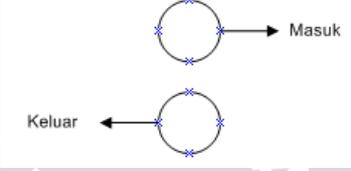
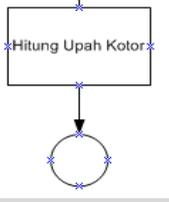
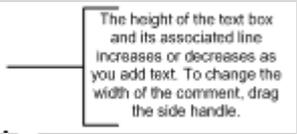
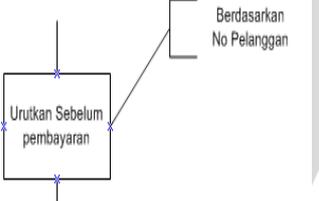
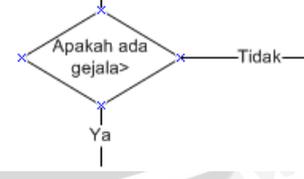
2.2.2.4 Flowchart (Diagram Alir)

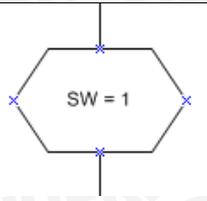
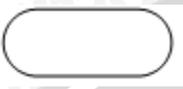
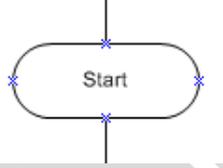
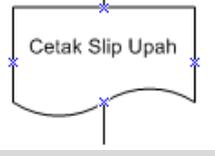
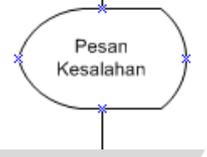
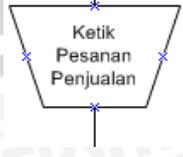
Flowchart adalah bagian (*chart*) yang menunjukkan alir (*flow*) didalam program atau prosedur sistem atau logika. *Flowchart* digunakan terutama untuk alat bantu komunikasi dan untuk dokumentasi. *Flowchart* dibagi menjadi lima jenis yaitu (Supriyanto,2008):

1. Bagan Alir sistem (*system flowchart*) merupakan bagan yang menunjukkan arus pekerjaan secara keseluruhan dari sistem. Bagan ini menjelaskan urutan dari prosedur yang ada di dalam sistem tersebut. Bagan alir sistem mewujudkan apa yang dikerjakan di sistem.
2. Bagan Alir Dokumen (*document flowchart*) merupakan bagan yang menunjukkan arus dari laporan dan formulir termasuk tembusan-tembusannya.
3. Bagan Alir Skematik (*schematic flowchart*) merupakan bagan alir yang mirip dengan bagan alir sistem. Perbedaannya adalah bagan alir skematik selain menggunakan symbol-simbol bagan alir sistem juga menggunakan gambar-gambar komputer dan peralatan lain yang digunakan. Maksud menggunakan gambar-gambar ini adalah untuk memudahkan komunikasi kepada orang yang kurang paham dengan simbol-simbol bagan alir.
4. Bagan Alir Program (*program flowchart*) merupakan bagan yang menjelaskan secara rinci langkah-langkah dari proses program.
5. Bagan Alir Proses (*process flowchart*) merupakan bagan alir yang banyak digunakan oleh teknik industri. Bagan alir ini juga berguna bagi analisis sistem untuk menggambarkan proses dalam suatu prosedur

Pada Tabel 2.2 merupakan simbol-simbol *flowchart* yang biasanya digunakan adalah simbol-simbol standar yang dikeluarkan oleh ANSI dan ISO (Supriyanto,2008).

Tabel 2.2 Simbol-Symbol *Flowchart*

Simbol	Arti	Contoh
<p><i>Input/ Output</i></p> 	Merepresentasikan <i>Input</i> data atau <i>Output</i> data yang diproses atau Informasi.	
<p>Proses</p> 	Mempresentasikan operasi	
<p>Penghubung</p> 	Keluar atau masuk dari bagian lain <i>flowchart</i> yang sama	
<p>Anak Panah</p> 	Mempresentasikan alur kerja	
<p>Penjelasan</p> 	Digunakan untuk komentar tambahan	
<p>Keputusan</p> 	Keputusan dalam program	
<p><i>Predefined Process</i></p> 	Rincian operasi berada di tempat lain	

Simbol	Arti	Contoh
<p><i>Preparation</i></p> 	Pemberian harga awal	
<p><i>Terminal Points</i></p> 	Awal/ akhir program	
<p><i>Punched Card</i></p> 	Input/ output yang menggunakan kartu berlubang.	
<p>Dokumen</p> 	I/ O dalam format dicetak	
<p><i>Display</i></p> 	Output yang ditampilkan pada terminal	
<p><i>Manual Operation</i></p> 	Operasi manual	

Sumber: (Supriyanto,2008)

2.3 Sistem Pakar

Sistem pakar adalah sistem yang berusaha mengadopsi pengetahuan manusia ke komputer agar komputer dapat menyelesaikan masalah seperti yang biasa dilakukan oleh para ahli. Sistem pakar yang baik dirancang agar dapat menyelesaikan suatu permasalahan tertentu dengan meniru kerja dari para ahli. Bagi para ahli sistem pakar ini juga akan membantu aktivitasnya sebagai asisten yang sangat berpengalaman (Kusumadewi,2003).

Ada beberapa definisi tentang sistem pakar antara lain (Kusumadewi,2003):

- Menurut Durkin : Sistem pakar adalah suatu program komputer yang dirancang untuk memodelkan kemampuan penyelesaian masalah yang dilakukan oleh seorang pakar.
- Menurut Ignizio : Sistem pakar adalah suatu model dan prosedur yang berkaitan dalam suatu domain tertentu yang mana tingkat keahliannya dapat dibandingkan dengan seorang pakar.
- Menurut Giarratano dan Riley : Sistem pakar adalah suatu sistem komputer yang bisa menyamai atau meniru kemampuan seorang pakar.

2.3.1 Konsep Dasar Sistem Pakar

Konsep dasar dari sistem pakar meliputi keahlian (*expertise*), ahli (*experts*), pemindahan keahlian (*transferring expertise*), inferensi (*inferencing*), aturan (*rules*) dan kemampuan memberikan penjelasan (*explanation capability*) (Hidayati,2010).

Keahlian (*expertise*) adalah pengetahuan yang mendalam tentang suatu masalah tertentu, dimana keahlian bisa diperoleh dari pelatihan/ pendidikan, membaca dan pengalaman dunia nyata. Ada dua macam pengetahuan yaitu pengetahuan dari sumber yang ahli dan pengetahuan dari sumber yang tidak ahli. Pengetahuan dari sumber yang ahli dapat digunakan untuk mengambil keputusan dengan cepat dan tepat.

Ahli (*experts*) adalah seorang yang memiliki ahli tentang suatu hal dalam tingkatan tertentu. Ahli dapat menggunakan suatu permasalahan yang ditetapkan dengan beberapa cara yang berubah-ubah dan merubahnya ke dalam bentuk yang dapat dipergunakan oleh dirinya sendiri dengan cepat dan cara pemecahan yang mengesankan. Ahli seharusnya dapat untuk menjelaskan hasil yang diperoleh, mempelajari sesuatu yang baru tentang domain masalah, merestrukturisasi pengetahuan kapan saja yang diperlukan dan menentukan apakah keahlian mereka relevan atau saling berhubungan (Hidayati,2010).

2.3.2 Tujuan Sistem Pakar

Tujuan dari sistem pakar adalah untuk memindahkan kemampuan (*transferring expertise*) dari seorang ahli atau sumber keahlian yang lain ke sumber komputer dan kemudian memindahkannya dari komputer kepada pemakai yang tidak ahli (bukan pakar). Proses ini meliputi empat aktivitas yaitu (Hidayati.2010):

1. Akuisisi pengetahuan (*knowledge acquisition*) yaitu kegiatan mencari dan mengumpulkan pengetahuan dari para ahli atau sumber keahlian yang lain.
2. Representasi pengetahuan (*knowledge representation*) adalah kegiatan menyimpan dan mengatur penyimpanan pengetahuan yang diperoleh dalam komputer. Pengetahuan berupa fakta dan aturan disimpan dalam komputer sebagai sebuah komponen yang disebut basis pengetahuan.
3. Inferensi pengetahuan (*knowledge inferencing*) adalah kegiatan melakukan inferensi berdasarkan pengetahuan yang telah disimpan didalam komputer.
4. Pemindahan pengetahuan (*knowledge transfer*) adalah kegiatan pemindahan pengetahuan dari komputer ke pemakai yang tidak ahli.

2.3.3 Bentuk Sistem Pakar

Sistem pakar dikelompokkan ke dalam empat bentuk yaitu (Hidayati,2010):

1. Berdiri sendiri. Sistem pakar jenis ini merupakan *software* yang berdiri sendiri tidak tergabung dengan *software* yang lainnya.
2. Tergabung. Sistem pakar jenis ini merupakan bagian program yang terkandung di dalam suatu algoritma (konvensional), atau merupakan program dimana didalamnya memanggil algoritma subrutin lain (konvensional).
3. Menghubungkan ke *software* lain. Bentuk ini biasanya merupakan sistem pakar yang menghubungkan ke suatu paket program tertentu misalnya DBMS.
4. Sistem Mengabdikan. Sistem pakar merupakan bagian dari komputer khusus yang dihubungkan dengan suatu fungsi tertentu. Misalnya sistem pakar yang digunakan untuk menganalisis data radar.

2.3.4 Ciri-Ciri Sistem Pakar

Sistem pakar mempunyai ciri-ciri, diantaranya dapat dijabarkan sebagai berikut (Sulistiyohati,2008):

1. Terbatas pada bidang yang spesifik.
2. Dapat memberikan penalaran untuk data-data yang tidak lengkap atau tidak pasti
3. Dapat mengemukakan rangkaian alasan yang diberikan dengan cara yang dipahami.
4. Berdasarkan pada *rule* atau kaidah tertentu.
5. Dirancang untuk dapat dikembangkan secara bertahap.
6. *Output* bersifat anjuran.
7. *Output* tergantung dari dialog dengan pengguna.
8. Basis pengetahuan dan mesin inferensi terpisah.

2.3.5 Keuntungan Sistem Pakar

Sistem pakar memiliki keuntungan, diantaranya adalah (Hidayati,2010):

1. Membantu seseorang yang awam dapat bekerja seperti layaknya seorang pakar.
2. Dapat bekerja sam dengan informasi yang tidak lengkap atau tidak pasti.
3. *Expert System* menyediakan nasehat yang konsisten dan dapat mengurangi tingkat kesalahan.
4. Membuat peralatan yang kompleks lebih mudah dioperasikan karena *expert system* dapat melatih pekerja yang tidak berpengalaman.
5. *Expert System* tidak dapat lelah atau bosan, juga konsisten dalam memberi jawaban dan selalu memberi perhatian penuh.
6. Memiliki kemampuan untuk memecahkan masalah yang kompleks.
7. Memungkinkan pemindahan pengetahuan ke lokasi yang jauh serta memperluas jangkauan seorang pakar, dapat diperoleh dan dipakai dimana saja.

2.3.6 Kelemahan Sistem Pakar

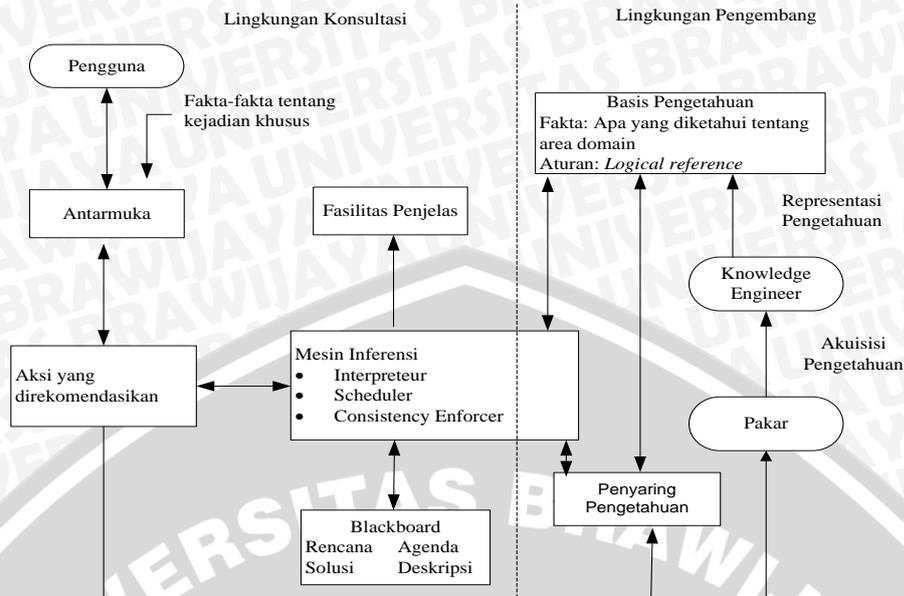
Sistem pakar seperti halnya sistem lainnya yang juga memiliki kelemahan, diantaranya adalah (Sulistyohati,2008) :

1. Masalah dalam mendapatkan pengetahuan dimana pengetahuan tidak selalu bisa didapatkan dengan mudah, karena kadang kala pakar dari masalah yang dibuat tidak ada dan walaupun ada kadang-kadang pendekatan yang dimiliki oleh pakar berbeda-beda.
2. Untuk membuat sistem pakar yang benar-benar berkualitas tinggi sangatlah sulit dan memerlukan biaya yang besar untuk pemeliharaan dan pengembangannya.
3. Boleh jadi sistem tidak dapat membuat keputusan.
4. Sistem pakar tidak 100% menguntungkan, walaupun seorang tetap tidak sempurna atau tidak selalu benar. Dalam hal ini peran manusia tetap merupakan faktor domain.

2.3.7 Struktur Sistem Pakar

Sistem pakar disusun oleh dua bagian utama, yaitu lingkungan pengembang (*development environment*) dan lingkungan konsultasi (*consultation environment*). Lingkungan sistem pakar pengembang digunakan untuk memasukkan pengetahuan pakar ke dalam lingkungan sistem pakar, sedangkan lingkungan konsultasi digunakan oleh pengguna yang bukan pakar guna memperoleh pengetahuan pakar.

Komponen-komponen sistem pakar dalam kedua bagian tersebut dapat dilihat pada Gambar 2.8 (Sulistyohati,2008) .



Gambar 2.8 Struktur Sistem Pakar

Sumber: (Sulistyohati,2008)

1. **Pengguna (User)**
 Pada umumnya pengguna sistem pakar adalah orang awam yang membutuhkan solusi, saran, atau pelatihan (*training*) dari berbagai permasalahan yang ada.
2. **Antarmuka Pengguna (User Interface)**
 Merupakan mekanisme yang digunakan oleh pengguna dan sistem pakar untuk berkomunikasi. Antarmuka menerima informasi dari pemakai dan mengolah ke dalam bentuk yang dapat diterima oleh sistem. Selain itu antarmuka menerima informasi dari sistem dan menampilkan ke dalam bentuk yang dapat dimengerti oleh pemakai.
3. **Akuisisi pengetahuan atau Penambahan Pengetahuan**
 Subsistem ini digunakan untuk memasukkan pengetahuan dari seorang pakar dengan cara merekayasa pengetahuan agar bisa diproses oleh komputer dan menaruhnya dalam basis pengetahuan dengan format tertentu (dalam bentuk representasi pengetahuan).
4. **Basis pengetahuan (Knowledge Base)**
 Basis pengetahuan mengandung pengetahuan yang diperlukan untuk memformulasikan, memahami dan menyelesaikan masalah. Basis pengetahuan terdiri dari dua elemen dasar yaitu fakta dan aturan.
5. **Mesin Inferensi (Inference Engine)**
 Sebuah program yang berfungsi untuk memandu proses penalaran terhadap suatu kondisi berdasarkan basis pengetahuan yang ada, manipulasi dan mengarahkan kaidah, model dan fakta yang disimpan hingga dicapai suatu kesimpulan.
6. **Daerah Kerja (Blackboard)**

Merekam hasil sementara untuk dijadikan keputusan dan menjelaskan masalah yang terjadi. Tiga tipe keputusan yang direkam pada *blackboard* meliputi rencana, agenda dan solusi.

7. Fasilitas Penjelasan (*Explanation Subsystem*)
Fasilitas penjelasan adalah komponen tambahan yang akan meningkatkan kemampuan sistem pakar. Komponen ini menggambarkan penalaran sistem kepada pemakai. Fasilitas penjas dapat menjelaskan perilaku sistem pakar dengan menjawab beberapa pertanyaan sebagai berikut:
 - a. Mengapa pertanyaan tertentu ditanyakan oleh sistem pakar.
 - b. Bagaimana kesimpulan tertentu diperoleh.
 - c. Mengapa alternatif tertentu ditolak.
 - d. Apa rencana untuk memperoleh penyelesaian.
8. Sistem Perbaikan Pengetahuan (*Knowledge Refining System*)
Memiliki kemampuan menganalisa pengetahuan yang yang diperlukan dari seorang pakar dan juga untuk mengevaluasi diri sehingga mengetahui alasan kesuksesan dan kegagalan dalam mengambil keputusan.

2.3.8 Representasi Pengetahuan

Representasi pengetahuan merupakan metode yang digunakan untuk mengkodekan pengetahuan dalam sebuah sistem pakar (Sulistyohati,2008). Metode representasi yang cocok untuk pengetahuan bersifat deklaratif adalah:

1. Logika (*Logic*)
Logika merupakan suatu pengkajian ilmiah tentang serangkaian penalaran, sistem kaidah dan prosedur yang membantu proses penalaran.
2. Jaringan Semantik (*Semantic Nets*)
Jaringan semantik merupakan teknik representasi kecerdasan buatan klasik yang digunakan untuk informasi proporsional. Yang dimaksud dengan informasi proporsional adalah pernyataan yang mempunyai nilai benar atau salah.
3. Bingkai (*Frame*)
Bingkai merupakan ruang (*slots*) yang berisi atribut untuk mendeskripsikan pengetahuan. Pengetahuan yang termuat dalam *slots* dapat berupa kejadian, lokasi, situasi atau elemen lainnya.

Representasi yang tepat untuk pengetahuan prosedural (ada aksi dan reaksi) adalah kaidah produksi (*Production Rule*). Dimana kaidah produksi adalah kaidah yang menyediakan cara formal untuk mempresentasikan rekomendasi, arahan, atau strategi. Kaidah produksi dituliskan dalam bentuk jika-maka (*if-then*). Kaidah *if-then* menghubungkan antiseden (*antecedent*) dengan konsekuensi yang diakibatkannya.

2.3.9 Basis Pengetahuan

Basis pengetahuan berisi pengetahuan-pengetahuan dalam menyelesaikan masalah didalam domain tertentu. Ada dua bentuk pendekatan basis pengetahuan yang sangat umum digunakan, yaitu (Sulistyohati,2008) :

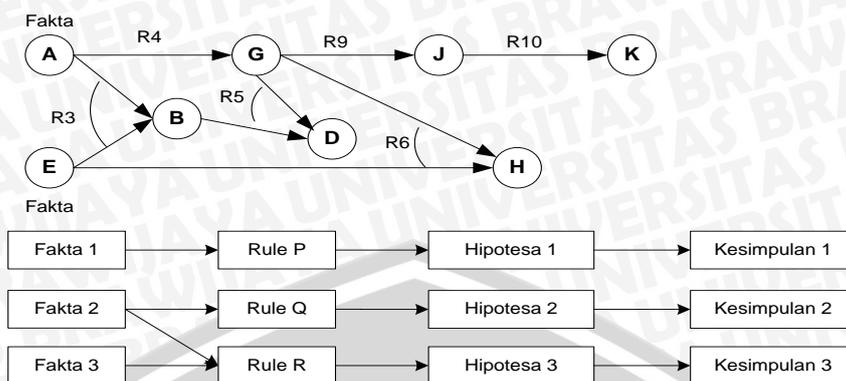
1. **Penalaran Berbasis Aturan (*Rule-Based Reasoning*)**
Pada penalaran berbasis aturan, pengetahuan direpresentasikan dengan menggunakan aturan berbentuk *if-then*. Bentuk ini digunakan apabila memiliki sejumlah pengetahuan pakar pada suatu permasalahan tertentu dan pakar dapat menyelesaikan masalah tersebut secara berurutan. Disamping itu, juga dibutuhkan penjelasan tentang jejak (langkah-langkah) pencapaian solusi.
2. **Penalaran Berbasis Kasus (*Case-Based Reasoning*)**
Pada penalaran berbasis kasus, basis pengetahuan akan berisi solusi-solusi yang telah dicapai sebelumnya, kemudian akan diturunkan suatu solusi untuk keadaan yang terjadi saat ini (fakta yang ada). Bentuk ini akan digunakan apabila pengguna menginginkan untuk tahu lebih banyak lagi pada kasus-kasus yang hampir sama (mirip). Selain itu, bentuk ini digunakan apabila telah memiliki sejumlah situasi atau kasus tertentu dalam basis pengetahuan.

2.3.10 Mesin Inferensi

Inferensi merupakan proses untuk menghasilkan informasi dari fakta yang diketahui atau diasumsikan. Proses inferensi dalam sistem pakar disebut inferensi. Berikut adalah dua jenis metode inferensi (Sulistyohati,2008) :

2.3.10.1 Foward Chaining

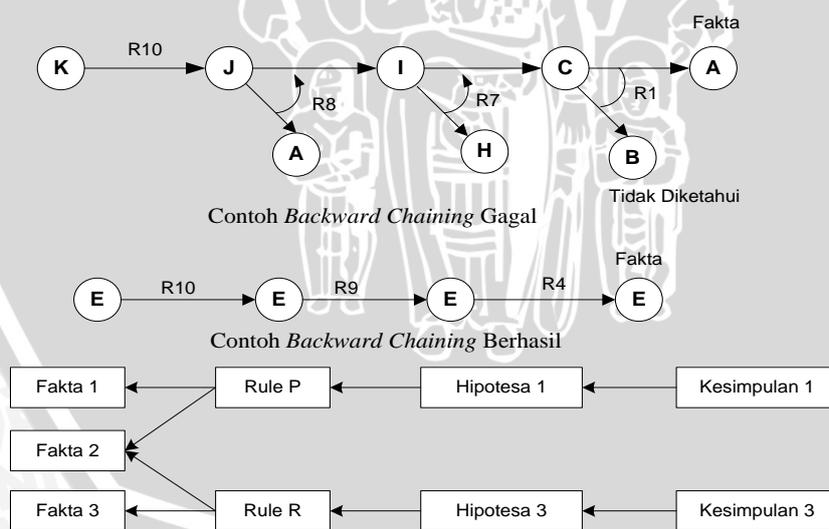
Teknik pencarian yang dimulai dengan fakta yang diketahui, kemudian dicocokkan fakta tersebut dengan bagian *IF* dari aturan *IF-THEN*. Bila ada aturan yang sesuai dengan bagian *IF*, maka aturan tersebut dieksekusi. Bila aturan dieksekusi maka sebuah fakta baru (bagian *THEN*) ditambahkan kedalam basis data. Pencocokan dimulai dari aturan teratas dan setiap aturannya hanya boleh dieksekusi sekali. Alur metode inferensi *Forward Chaining* terlihat seperti Gambar 2.9.



Gambar 2.9 Alur Metode Forward Chaining
 Sumber:(Kusumadewi,2003)

2.3.10.2. Backward Chaining

Metode inferensi yang bekerja mundur ke arah kondisi awal. Proses diawali dari *goal* (yang ada pada bagian *THEN* dari aturan *IF-THEN*), kemudian pencarian mulai dijalankan untuk mencocokkan apakah fakta-fakta yang ada cocok dengan premis-premis dibagian *IF*. Jika cocok, maka aturan dieksekusi, kemudian hipotesis dibagian *THEN* ditempatkan dibasis data sebagai fakta baru. Jika tidak cocok simpan premis dibagian *IF* kedalam *subgoal*. Proses berakhir jika *goal* ditemukan atau tidak ada aturan yang bisa membuktikan kebenaran *subgoal* atau *goal*. Alur metode inferensi *Backward Chaining* terlihat seperti Gambar 2.10.



Gambar 2.10 Alur Metode Backward Chaining
 Sumber :(Kusumadewi,2003)

2.4 Ketidakpastian

Jika sistem kecerdasan buatan yang dikembangkan memiliki pengetahuan yang lengkap tentang permasalahan yang ditanganinya, maka sistem tersebut



dapat dengan mudah memberikan solusi dengan menggunakan pendekatan logika. Akan tetapi, sistem hampir tidak pernah dapat mengakses seluruh fakta yang ada dalam lingkungan permasalahan yang akan ditanganinya, sehingga sistem harus bekerja dalam ketidakpastian atau kesamaran. Untuk itu, sistem harus menggunakan teknik-teknik khusus yang dapat menangani ketidakpastian dan kesamaran dalam menyelesaikan permasalahan yang ditanganinya (Prihatini,2011).

Ada tiga teknik yang dapat digunakan untuk menangani ketidakpastian dan kesamaran pengetahuan, yaitu (Prihatini,2011)

1. Teknik Probabilitas, yang dikembangkan dengan memanfaatkan teorema *Bayes* yang menyajikan hubungan sebab akibat yang terjadi diantara *evidence* yang ada. Pendekatan alternatif lainnya yang dapat digunakan adalah teori *Dempster-shafer*.
2. Faktor Kepastian, merupakan teknik penalaran tertua yang digunakan pada sistem MYCIN. Teknik ini bersifat semi probabilitas, karena tidak sepenuhnya menggunakan notasi probabilitas.
3. *Logika Fuzzy*, merupakan teknik baru yang diperkenalkan oleh Zadeh. Setiap variabel dalam teknik ini memiliki nilai rentang tertentu, yang akan digunakan untuk menghitung nilai fungsi keanggotaannya.

2.5 Printer Inkjet

Printer Inkjet adalah jenis *printer* yang menggunakan teknologi " *Dor On Demand* " yaitu melakukan cetakan dengan cara menyemprotkan titik demi titik (biasa di sebut Dot) kecil tinta pada kertas melalui *nozzle* atau lubang pipa yang sangat kecil. Cara sederhana untuk mengetahui sebuah *printer* itu *inkjet* atau tidak adalah dengan melihat apakah alat cetak itu menggunakan tinta untuk proses mencetak.(Hartono,2004)

2.5.1 Kelebihan dan Kekurangan *Printer Inkjet*

Kelebihan *Printer Inkjet*

- Berkecepatan tinggi dalam melakukan pencetakan
- Hasil cetak foto berkualitas tinggi
- Harga printer relatif murah
- Memiliki kerapatan cetak yang lebih akurat dan halus

Kekurangan *Printer Inkjet*

1. Untuk proses cetak hitam putih masih kalah cepat dengan *printer laserjet*
2. Seringnya terjadi kerusakan pada *cartridge*.
3. Tinta dapat luntur saat belum kering atau terkena cairan. (Alhafiz,2013)

2.5.2 Komponen Printer inkjet

Printer Inkjet memiliki beberapa komponen penting, dimana komponen tersebut antara lain (Aston,2012) :

1. Power Supply

Power supply ini berfungsi untuk menyuplai tegangan listrik agar printer bisa bekerja. Pada Gambar 2.11 adalah power supply printer inkjet merk Canon.



Gambar 2.11 Power supply

2. Mainboard

Merupakan pusat kontrol dari printer, semua perintah kerja mekanik maupun data terjadi di bagian ini. Pada Gambar 2.12 adalah Gambar mainboard printer.



Gambar 2.12 Mainboard

3. Cartridge

Fungsi cartridge sendiri ialah sebagai wadah tinta yang digunakan untuk mencetak. Pada Gambar 2.13 adalah cartridge printer Canon.



Gambar 2.13 Cartridge

4. Sensor ASF penarik kertas

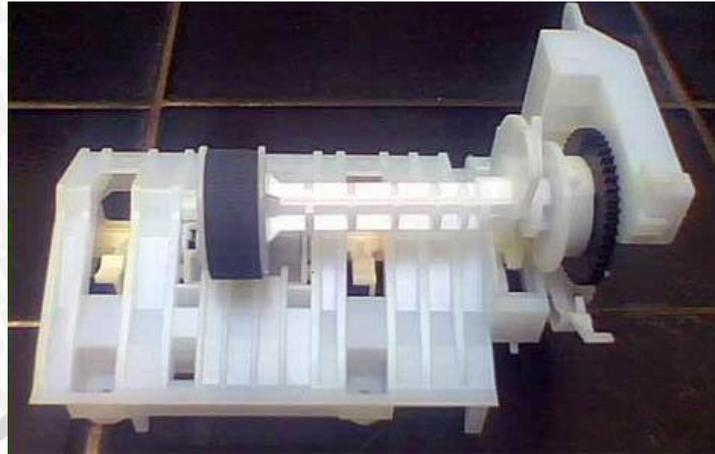
Pada Gambar 2.14 adalah Gambar sensor ASF. Sensor ini bekerja sebagai pendeteksi ada tidaknya kertas untuk dicetak, berfungsi pula untuk mendeteksi panjang pendeknya kertas. biasanya alat ini terdapat di bagian samping atau atas Rol penarik kertas.



Gambar 2.14 Sensor ASF Penarik Kertas

5. Roll ASF penarik kertas

Rol yang tersusun secara mekanik berisi *gear*/gir untuk pemutar yang mengeluarkan hasil *print*/kertas. Pada pencetak *printer standard roll* tersusun dari satu buah gulungan karet cukup besar untuk berfungsi mengeluarkan kertas. Pada Gambar 2.15 adalah gambar *roll* penarik kertas



Gambar 2.15 Roll ASF Penarik Kertas

6. Rumah *cartridge* (*carriage unit*)

Sebuah perangkat yang berguna sebagai tempat *cartridge* di letakkan /di dudukkan , sehingga tersambung antara *chip* dari *cartridge* ke *chip* di bagian rumah *cartridge* . Pada Gambar 2.16 adalah Gambar rumah *cartridge*.



Gambar 2.16 Rumah Cartridge

7. Sensor pita *encoder*

Encoder adalah pita sensor yang terbuat dari plastik bening dengan kode baris di permukaannya. letaknya dibelakang *head cartridge*/rumah *cartridge*. Pada Gambar 2.17 adalah gambar sensor pita *encoder*.



ENCODER

Gambar 2.17 Sensor Pita Encoder

8. Sensor *timing disk*

Komponen yang berfungsi sebagai *timing* (Pewaktu) pada waktu *printer* menarik kertas, terjadi perbedaan berjalannya kertas ketika kita menggunakan kualitas biasa dan kualitas foto / *high*. Pada Gambar 2.18 adalah gambar sensor *timing disk*



Gambar 2.18 Sensor *Timing Disk*

9. *Scanner*

Salah satu bagian tambahan pada fitur *printer* yang memiliki fungsi untuk melakukan *scan* atau pembacaan gambar yang menempel pada area *scan*, untuk di simpan dalam bentuk dokumen ataupun di cetak ulang. Pada Gambar 2.19 adalah gambar *scanner*.



Gambar 2.19 *Scanner*

2.6 Teori *Dempster-shafer*

Metode *Dempster-Shafer* pertama kali diperkenalkan oleh Dempster, yang melakukan percobaan model ketidakpastian dengan *range* probabilitas sebagai

probabilitas tunggal. Kemudian pada tahun 1976 Shafer mempublikasikan teori Dempster tersebut pada sebuah buku yang berjudul *Mathematical Theory of Evident* (Desiani,2006).

Secara umum teori *Dempster-Shafer* ditulis dalam suatu interval:

[Belief,Plausibility]

Belief (Bel) adalah ukuran kekuatan *evidence* dalam mendukung suatu himpunan proposisi. Jika bernilai 0 (nol) maka mengindikasikan bahwa tidak ada *evidence*, dan jika bernilai 1 menunjukkan adanya kepastian. Menurut Giarratano dan Riley fungsi *belief* dapat diformulasikan sebagai persamaan 2.1.

$$Bel(X) = \sum_{Y \subseteq X} m(Y) \tag{2.1}$$

Sedangkan *Plausibility* (Pls) dinotasikan sebagai persamaan 2.2.

$$Pls(X) = 1 - Bel(X') = 1 - \sum_{Y \subseteq X'} m(Y) \tag{2.2}$$

dimana:

$Bel(X) = Belief(X)$

$Pls(X) = Plausibility(X)$

$m(X) = mass\ function\ dari\ (X)$

$m(Y) = mass\ function\ dari\ (Y)$

Plausibility juga bernilai 0 sampai 1, jika kita yakin akan X' maka dapat dikatakan *Belief* (X') = 1 sehingga dari rumus Persamaan 2.2 nilai $Pls(X) = 0$. Beberapa kemungkinan range antara *Belief* dan *Plausibility* dapat dilihat pada Tabel 2.1:

Tabel 2.1 Range Belief dan Plausibility

Kemungkinan	Keterangan
[1,1]	Semua Benar
[0,0]	Semua Salah
[0,1]	Ketidaktepastian
[Bel,1] where $0 < Bel < 1$	Cenderung Mendukung
[0,Pls] where $0 < Pls < 1$	Cenderung Menolak
[Bel,Pls] where $0 < Bel \leq Pls < 1$	Cenderung Mendukung dan Menolak

Pada teori *Dempster-Shafer* juga dikenal adanya *frame of discernment* yang dinotasikan dengan Θ . *Frame of discernment* ini merupakan semesta



pembicaraan dari sekumpulan hipotesis sehingga sering disebut dengan *environment* yang ditunjukkan pada Persamaan 2.3.

$$\Theta = \{\theta_1, \theta_2, \dots, \theta_n\} \quad (2.3)$$

dimana:

Θ = *Frame of discernment* atau *environment*

$\theta_1, \dots, \theta_n$ = Elemen atau unsur bagian dalam *environment*

Environment mengandung elemen-elemen yang menggambarkan kemungkinan sebagai jawaban dan hanya ada satu yang akan sesuai dengan jawaban yang dibutuhkan. Kemungkinan ini dalam teori *Dempster-Shafer* disebut dengan *power set* dan dinotasikan dengan $P(\Theta)$, setiap elemen dalam *power set* ini memiliki nilai interval antara 0 sampai 1.

$$m = P(\Theta) \rightarrow [0,1]$$

sehingga dapat dirumuskan seperti persamaan 2.4.

$$\sum_{X \in P(\Theta)} m(X) = 1 \approx \sum_{X \in P(\Theta)} m(X) = 1 \quad (2.4)$$

dengan :

$P(\Theta)$ = *power set*

$m(X)$ = *mass function* dari (X)

Dalam teori *Dempster-Shafer*, *disbelief* dalam *environment* biasanya dinotasikan $m(\theta)$. Sedangkan *mass function* (m) dalam teori *Dempster-Shafer* adalah tingkat kepercayaan dari suatu *evidence* (gejala), sering disebut dengan *evidence measure* sehingga dinotasikan dengan (m).

Pada aplikasi sistem terdapat sejumlah *evidence* yang akan digunakan pada faktor ketidakpastian dalam pengambilan keputusan untuk *diagnosis* suatu permasalahan. Untuk mengatasi sejumlah *evidence* tersebut pada teori *Dempster-Shafer* menggunakan aturan yang lebih dikenal dengan *Dempster's Rule of Combination*. Dapat dilihat pada persamaan 2.5.

$$m_1 \oplus m_2(Z) = \sum_{X \cap Y = Z} m_1(X) m_2(Y) \quad (2.5)$$

dimana:

$m_1 \oplus m_2(Z)$ = *mass function* dari *evidence* (Z)

$m_1(X)$ = *mass function* dari *evidence* (X)

$m_2(Y)$ = *mass function* dari *evidence* (Y)

\oplus = operator *direct sum*

secara umum formulasi untuk *Dempster's Rule of Combination* adalah seperti yang terdapat pada persamaan 2.6.

$$m1 \oplus m2(Z) = \frac{\sum_{X \cap Y = Z} m1(X)m2(Y)}{1-k} \quad (2.6)$$

dimana:

k = Jumlah *evidential conflict*.

Besarnya jumlah *evidential conflict* (k) dirumuskan dengan persamaan 2.7 [SEN-02].

$$k = \sum_{X \cap Y = \theta} m1(X)m2(Y) \quad (2.7)$$

Sehingga bila Persamaan (2.7) disubstitusikan ke Persamaan (2.6) akan menjadi persamaan 2.8.

$$m1 \oplus m2(Z) = \frac{\sum_{X \cap Y = Z} m1(X)m2(Y)}{1 - \sum_{X \cap Y = Z} m1(X)m2(Y)} \quad (2.8)$$

dimana:

$m1 \oplus m2(Z)$ = *mass function* dari *evidence* (Z)

$m1(X)$ = *mass function* dari *evidence* (X)

$m2(Y)$ = *mass function* dari *evidence* (Y)

k = jumlah *evidential conflict*

2.7 Pengujian Sistem

Proses pengujian dilakukan dengan pengujian validasi dan pengujian akurasi. Pengujian validasi dilakukan untuk mengetahui fungsionalitas sistem yang dibangun sesuai dengan daftar kebutuhan yang ada. Pengujian akurasi dilakukan untuk mengetahui seberapa besar akurasi yang didapat dari penggunaan metode *Dempster-shafer* dalam menyelesaikan masalah pemodelan sistem pakar identifikasi kerusakan pada *printer*.

2.7.1 Pengujian Validasi (*Black Box*)

Pengujian validasi digunakan untuk mengetahui apakah sistem yang dibangun sudah benar sesuai dengan yang dibutuhkan. Pengujian validasi menggunakan metode pengujian *Black Box*, karena tidak difokuskan terhadap alur jalannya algoritma program namun lebih ditekankan untuk menemukan kesesuaian antara kinerja sistem dengan daftar kebutuhan (Fitriani,2012).

2.7.2 Pengujian Akurasi

Akurasi merupakan seberapa dekat suatu angka hasil pengukuran terhadap angka sebenarnya (*true value/reference value*). Dalam penelitian ini pengujian akurasi dilakukan untuk mengetahui performa dari sistem pakar dalam memberikan kesimpulan diagnosis. Pengujian akurasi diagnosis dihitung dari

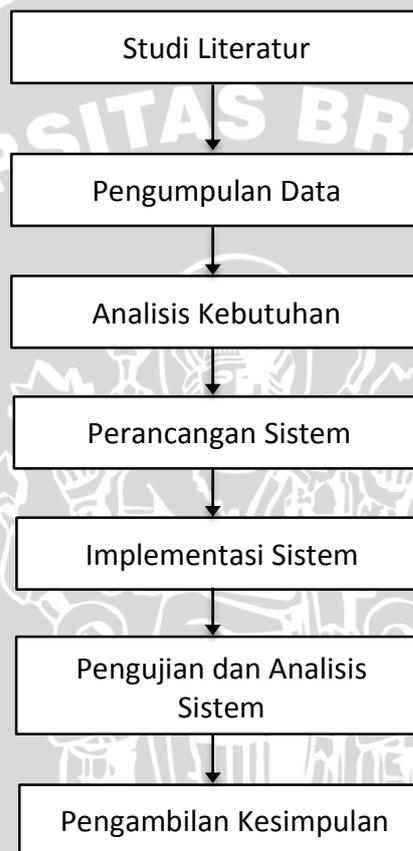
jumlah diagnosis yang tepat dibagi dengan jumlah data. Secara umum perhitungan akurasi seperti pada persamaan 2.4 (Fitriani,2012).

$$\text{Nilai Akurasi} = \frac{\text{Jumlah data akurat}}{\text{Jumlah seluruh data}} \times 100\% \quad (2.9)$$



BAB 3 METODOLOGI

Bab metodologi ini akan membahas langkah-langkah yang dilakukan penulis dalam pembuatan pemodelan sistem untuk identifikasi kerusakan printer dengan metode *Dempster-shafer*. Langkah-langkah tersebut antara lain studi literature, pengumpulan data, analisis kebutuhan, perancangan sistem, implementasi sistem, pengujian sistem dan pengambilan kesimpulan. Langkah-langkah dalam penelitian digambarkan dalam diagram blog pada Gambar 3.1 dibawah ini.



Gambar 3.1 Alur Metodologi Penelitian

3.1 Studi Literatur

Mempelajari literatur dari beberapa bidang ilmu yang berhubungan dengan pembuatan sistem untuk identifikasi kerusakan *printer*, diantaranya :

- Sistem Pakar
- Metode Teori *Dempster-Shafer*
- Berbagai kerusakan pada *printer*

Literatur tersebut diperoleh dari buku, jurnal, *e-book*, pakar dan penelitian sebelumnya.

3.2 Pengumpulan Data

Pada tahap pengumpulan data penelitian yang dibutuhkan adalah definisi kerusakan dan gejala-gejala kerusakan pada *printer inkjet* dengan merk Canon serta nilai densitas tiap gejala untuk perhitungan menggunakan metode *Dempster-shafer*. Sumber data diperoleh dari hasil wawancara yang dilakukan dengan seorang pakar yaitu teknisi *printer* Maskomlaptop.com. Dari hasil wawancara dengan pakar, penulis mendapatkan data pengetahuan tentang gejala – gejala kerusakan *printer*, serta meminta nilai densitas pada tiap gejala untuk perhitungan menggunakan metode *Dempster-shafer*.

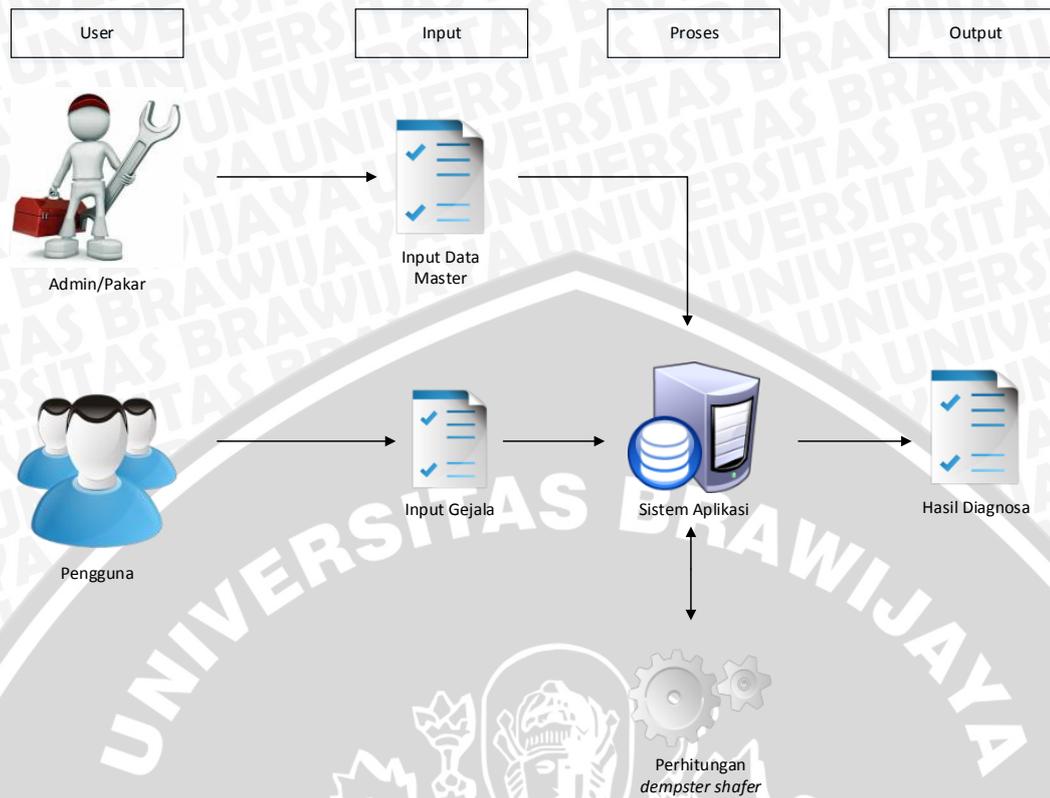
3.3 Analisis Kebutuhan

Analisis kebutuhan berguna untuk mendaftar macam-macam kebutuhan yang diperlukan dalam pembuatan sistem pakar. Berikut ini daftar kebutuhan dalam pembuatan sistem:

- Spesifikasi kebutuhan *hardware*, meliputi:
 - Laptop dengan prosesor core i3
 - *Memory* 2GB
 - *Harddisk* 250 GB
- Spesifikasi kebutuhan *software*, meliputi:
 - Windows 7 32bit sebagai sistem operasi.
 - XAMPP sebagai server localhost, MySQL sebagai *database management sistem* (DBMS).
- Data yang dibutuhkan, meliputi:
 - Data macam-macam kerusakan pada *printer* dan cara mengatasinya.
 - Data gejala pada masing – masing kerusakan yang terjadi pada *printer*.
 - Data nilai densitas gejala kerusakan *printer*.

3.4 Perancangan Sistem

Tahap desain sistem dibutuhkan agar penulis dapat dengan mudah melakukan proses selanjutnya terhadap sistem. Desain sistem memberikan gambaran mengenai proses jalannya sistem yang akan dibuat dalam penelitian ini. Perancangan sistem bertujuan sebagai acuan dalam implementasi sistem dan untuk melakukan analisis kebutuhan yang akan dipergunakan dalam penelitian. Pada Gambar 3.2 dapat dilihat rencana desain sistem yang akan digunakan oleh penulis :



Gambar 3.2 Desain Umum Perancangan Sistem

Pada gambar diatas dijelaskan bagaimana cara aplikasi ini bekerja. Pertama, admin menginput data kerusakan, data solusi, data gejala berikut dengan nilai densitasnya. Setelah data dari pakar disimpan ke dalam *database* maka data tersebut akan dijadikan acuan dari perhitungan identifikasi kerusakan menggunakan metode *Dempster-Shafer* pada sistem. Pengguna dapat melakukan identifikasi kerusakan dengan menginput gejala yang terjadi pada printer ke dalam aplikasi. Sistem kemudian akan memproses nilai densitas berdasarkan gejala-gejala kerusakan yang dimasukkan oleh pengguna melalui proses perhitungan densitas *Dempster-Shafer* yang sudah ada. Hasil identifikasi kerusakan akan didapat melalui perhitungan nilai densitas tertinggi.

3.5 Implementasi Sistem

Implementasi perangkat lunak dilakukan dengan menggunakan bahasa pemrograman *PHP*, untuk *database* menggunakan *MySQL* dan juga aplikasi pendukung lainnya. Implementasi dari sistem meliputi sebagai berikut :

1. Pembuatan antarmuka pengguna berupa halaman – halaman web.
2. Memasukkan data penelitian ke *database MySQL* untuk diolah menjadi informasi yang berguna dan dibutuhkan bagi aplikasi.
3. Melakukan perhitungan untuk memperoleh nilai kepercayaan dari setiap inputan dengan menggunakan metode *Dempster-Shafer*.

3.6 Pengujian dan Analisis Sistem

Pada tahap ini akan dilakukan uji coba identifikasi kerusakan pada *printer* yang bertujuan untuk menunjukkan bahwa sistem dapat menghasilkan identifikasi sesuai dengan yang diharapkan. Pengujian dilakukan dengan cara memeriksa apakah sistem sudah berjalan dengan baik dan tidak ada *error* yang terjadi. Pengujian juga dilakukan dengan cara membandingkan hasil identifikasi dari sistem dengan hasil identifikasi yang dilakukan oleh pakar untuk dapat mengetahui apakah keluaran dari sistem sudah sesuai dengan hasil yang diinginkan atau belum.

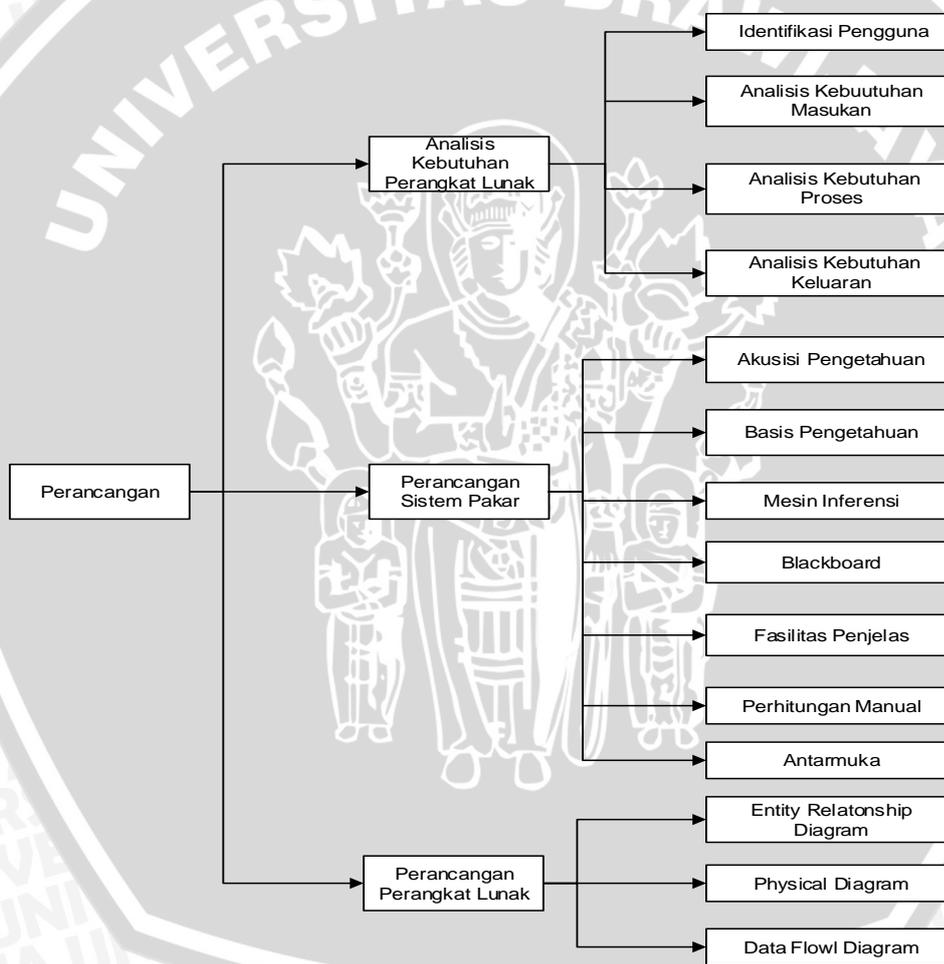
3.7 Pengambilan Kesimpulan

Kesimpulan dibuat setelah semua tahapan perancangan, pengolahan data, implementasi dan pengujian sistem terhadap metode yang digunakan selesai diterapkan. Kesimpulan disusun berdasarkan hasil pengujian dan analisis sistem yang telah dibuat. Isi dari kesimpulan diharapkan dapat menjadi acuan pada penelitian lain untuk mengembangkan sistem identifikasi kerusakan printer. Selain itu, pada akhir penulisan terdapat saran yang bertujuan untuk penyempurnaan penelitian ini.



BAB 4 PERANCANGAN

Bab ini membahas tentang perancangan “Pemodelan Sistem Pakar untuk Identifikasi Kerusakan *Printer* Menggunakan Metode *Dempster-Shafer*”. Pohon perancangan sistem pakar terdiri dari tiga tahapan yaitu analisis kebutuhan perangkat lunak, perancangan sistem pakar dan perancangan perangkat lunak. Analisis kebutuhan perangkat lunak terdiri dari identifikasi aktor, analisis kebutuhan masukan, analisis kebutuhan proses, dan analisis kebutuhan keluaran. Sistem pakar terdiri dari akuisisi pengetahuan, basis pengetahuan, mesin inferensi, *blackboard*, fasilitas penjelas, perancangan *algoritma* dan antarmuka. Perancangan perangkat lunak terdiri dari *entity relationship diagram*, *physical diagram*, dan *data flow diagram*. Pohon perancangan sistem pakar dapat dilihat pada Gambar 4.1.



Gambar 4.1 Diagram Perancangan

4.1 Analisis Kebutuhan Perangkat Lunak

Analisis kebutuhan perangkat lunak terdiri dari beberapa tahapan. Analisis kebutuhan ini bertujuan untuk menjabarkan daftar kebutuhan dalam membangun sistem pakar agar dapat memenuhi kebutuhan pengguna. Berikut ini adalah kebutuhan yang digunakan dalam pembuatan sistem pakar :

1. Kebutuhan *Hardware*, meliputi :
 - Komputer
2. Kebutuhan *Software*, meliputi :
 - Sistem Operasi Windows 7
 - Basisdata MySQL
 - Bahasa Pemrograman PHP
 - Aplikasi *Sublime text 2*
3. Data yang dibutuhkan meliputi :
 - Data jenis kerusakan *printer*.
 - Data nilai densitas setiap gejala kerusakan *printer*.

Berikut ini merupakan penjelasan dari masing-masing tahap analisis kebutuhan perangkat lunak

4.1.1 Identifikasi Pengguna

Tahap ini bertujuan untuk mengidentifikasi aktor-aktor yang berinteraksi dengan sistem. Pada Tabel 4.1 memperlihatkan aktor-aktor yang terlibat dalam sistem beserta penjelasannya masing-masing yang merupakan hasil dari proses identifikasi aktor.

Tabel 4.1 Deskripsi Pengguna

Aktor	Deskripsi Pengguna
Pengguna	Pengguna yang dapat menggunakan pemodelan sistem pakar untuk melihat gejala kerusakan pada <i>printer</i> , dan melakukan deteksi kerusakan berdasarkan gejala yang dialami.
Pakar (Teknisi)	Pengguna yang memiliki pengetahuan di bidang <i>printer</i> . Dari pakar didapatkan data macam-macam kerusakan, solusi mengatasi kerusakan, gejala tiap-tiap kerusakan serta nilai densitas tiap-tiap gejala pada kerusakan <i>printer</i> . Pakar dapat melakukan proses <i>login</i> , mengelola data gejala, kerusakan, solusi, dan merubah nilai densitasnya.
<i>Knowledge Engineer (KE)</i>	Pengguna yang menyerap sumber pengetahuan dari pakar kemudian ditransformasikan ke basis pengetahuan.

4.1.2 Analisis Kebutuhan Masukan

Pakar memberikan masukan berupa :

- Data gejala baru yang belum terdapat dalam sistem. Data gejala meliputi id gejala, nama gejala dan nilai densitasnya.
- Data kerusakan berupa kode kerusakan dan nama kerusakan.
- Data aturan ditambahkan sesuai dengan gejala dan kerusakan yang ditimbulkan.

Dari ketiga masukan pakar yang digunakan sebagai basis pengetahuan dari sistem dalam mendiagnosis kerusakan *printer*.

4.1.3 Analisis Kebutuhan Proses

Proses dari sistem ini adalah proses penalaran. Sistem akan melakukan penalaran untuk menentukan jenis kerusakan yang terjadi pada *printer* sesuai dengan gejala-gejala yang dimasukkan pengguna. Pada sistem telah disediakan aturan basis pengetahuan untuk penelusuran jenis kerusakan yang terjadi pada *printer*.

4.1.4 Analisis Kebutuhan Keluaran

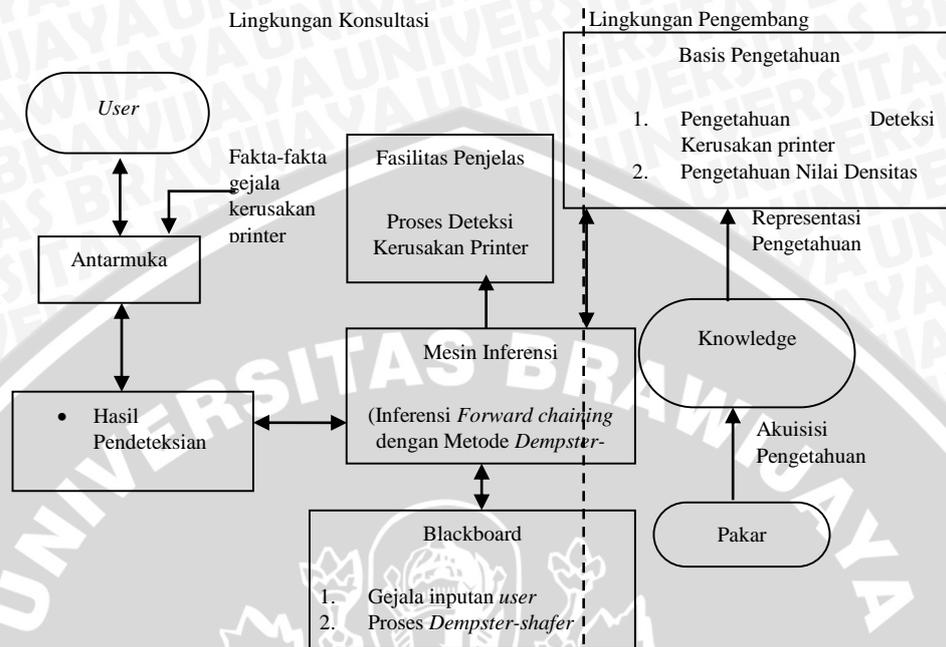
Data keluaran dari sistem ini adalah hasil proses diagnosis menggunakan perhitungan metode *Dempster-shafer*. Hasil diagnosis tersebut berdasarkan fakta gejala pada *printer* yang pengguna masukan saat melakukan diagnosis. Hasil *output* sistem terdiri dari : gejala yang terpilih, nama kerusakan, nilai kepercayaan, dan solusi.

4.2 Perancangan Sistem Pakar

Sistem pakar yang dibangun digunakan untuk mendeteksi kerusakan *printer*. Metode *Dempster-shafer* digunakan untuk proses pengambilan kesimpulan, sedangkan penelusuran jawaban untuk mencari nilai kepercayaan terbesar dari hasil perhitungan metode *Dempster-shafer* menggunakan metode *forward chaining*. Tahap yang biasa dilakukan baik oleh orang awam maupun seorang pakar dalam bidang kerusakan untuk melakukan deteksi adalah dengan melihat gejala yang terjadi pada *printer* tersebut. Semakin spesifik gejala yang diamati, maka semakin besar tingkat kepercayaannya.

Konsep sistem pakar yang akan dibangun dengan metode *Dempster-shafer* merupakan sistem yang melakukan pengambilan kesimpulan berdasarkan data densitas gejala yang sudah disimpan oleh pakar. Sistem menerima masukan dari *user* berupa fakta gejala yang telah diamati pada *printer*. Kemudian hasil dari masukan *user* tersebut digunakan sebagai dasar proses perhitungan dengan menggunakan rumus *Dempster-shafer*. Dari hasil perhitungan tersebut dilakukan perbandingan nilai densitas tertinggi setiap hasil perhitungan kerusakan untuk mendapatkan kesimpulannya. Hasil akhir berupa hasil deteksi kerusakan *printer* beserta nilai kepercayaan dan solusi dari kerusakan.

Pada Gambar 4.2 menjelaskan secara singkat arsitektur sistem pakar yang mengacu pada konsep perancangan yang telah disebutkan sebelumnya.



Gambar 4.2 Kerangka Konsep Arsitektur Sistem Pakar Untuk Identifikasi Kerusakan Printer.

4.2.1 Akuisisi Pengetahuan

Akuisisi pengetahuan merupakan proses mengambil pengetahuan dari pakar atau sumber pengetahuan lainnya ke dalam sistem komputer untuk membangun basis pengetahuan. Pengetahuan pakar yang akan diakuisisi, yaitu gejala tiap-tiap kerusakan *printer* dan analisa hasil identifikasi. Pada bagian ini dilakukan pemindahan pengetahuan pakar mengenai kerusakan, gejala dan solusi serta hasil identifikasi.

Tahap – tahap yang dilakukan dalam akuisisi pengetahuan dari pakar ini antara lain :

1. Wawancara

Wawancara merupakan metode akuisisi yang paling banyak digunakan. Tujuan wawancara ini adalah memperoleh wawasan pakar untuk domain masalah tertentu. Pada wawancara ini berguna untuk mengumpulkan informasi tentang cara identifikasi kerusakan pada *printer* khususnya melalui gejala kerusakan yang dirasakan oleh pengguna. Setelah mengumpulkan informasi mengenai kerusakan pada *printer*, kemudian pakar menjelaskan tentang kerusakan-kerusakan pada *printer* yang sering terjadi pada *printer* pengguna, menjelaskan gejala-gejala yang dirasakan pengguna serta solusi bagaimana mengatasi permasalahan tersebut.

Hasil dari wawancara dengan pakar yang kemudian akan di analisa untuk dijadikan aturan basis pengetahuan .

2. Analisa Protokol (Aturan)

Pada metode akuisisi ini, pakar diminta untuk melakukan suatu pekerjaan dan mengungkapkan proses pemikirannya dengan menggunakan kata-kata. Pekerjaan tersebut dituliskan dan dianalisis pada saat dilakukan wawancara sebelumnya. Kemudian pakar diminta untuk memberikan proses pemikiran yang akan dijadikan sebagai aturan basis pengetahuan tentang mengidentifikasi gejala kerusakan yang terjadi pada *printer* untuk memberikan kesimpulan kerusakannya. Selain itu pakar juga diminta untuk memberikan nilai tingkat kepercayaan pada setiap gejala kerusakan pada *printer* berdasarkan pengetahuan pakar yang dimiliki untuk dijadikan dasar perhitungan metode.

Akuisisi pengetahuan gejala kerusakan *printer*, kerusakan *printer*, dan nilai densitas pada setiap gejala yang diperoleh dari proses wawancara dapat dilihat pada Tabel 4.2.

Tabel 4.2 Akuisisi Pengetahuan Gejala Kerusakan Printer dan Nilai Densitas.

	Gejala	Nilai Densitas
A	<i>Power supply</i> rusak	
1	<i>Printer</i> tidak dapat dinyalakan	0,7
2	<i>Printer</i> tiba tiba mati	0,8
B	<i>Mainboard</i> rusak	
1	<i>Printer</i> tidak dapat dinyalakan	0,5
2	<i>Driver printer</i> tidak terdeteksi di komputer	0,7
3	<i>Printer</i> tidak dapat direset	0,7
4	Lampu indikator <i>power</i> (hijau) dan lampu indikator <i>stop/resume</i> (orange) kedip bergantian 9x	0,8
5	Muncul kode <i>error</i> p10 di panel LCD	0,8
C	<i>Cartridge</i> hitam rusak	
1	<i>Printer</i> tidak dapat mencetak	0,4
2	Hasil cetakan hitam tidak sempurna	0,8
3	Muncul kode <i>error</i> e05 di panel LCD	0,6
4	Lampu indikator <i>stop</i> (orange) kedip 5x	0,6
D	<i>Cartridge</i> warna rusak	
1	<i>Printer</i> tidak mencetak	0,4
2	Hasil cetakan hanya muncul warna tertentu saja	0,8
3	Lampu indikator <i>stop</i> (orange) kedip 5x	0,6
4	Muncul kode <i>error</i> e05 di panel LCD	0,6

	Gejala	Nilai Densitas
E	Cartridge tidak terdeteksi	
1	Printer tidak dapat mencetak	0,5
2	Lampu indikator stop (orange) kedip 5x	0,8
3	Muncul kode error e05 di panel LCD	0,8
F	Sensor ASF penarik kertas rusak	
1	Printer tidak dapat mencetak	0,4
2	Kertas macet	0,6
3	Lampu indikator stop (orange) kedip 3x	0,7
4	Muncul kode error e03 di panel LCD	0,7
G	Roll ASF Penarik kertas rusak	
1	Printer menarik kertas double	0,8
2	Lampu indikator stop (orange) kedip 2x	0,7
3	Muncul kode error e02 pada panel LCD	0,7
4	Printer tidak dapat mencetak	0,4
H	Rumah cartridge rusak	
1	Printer tidak dapat mencetak	0,3
2	Hasil cetakan tidak bagus	0,4
3	Muncul kode error p02 pada panel LCD	0,7
4	Lampu indikator power (hijau) dan lampu indikator stop/resume (orange) kedip bergantian 2x	0,7
5	Muncul kode error e05 di panel LCD	0,3
I	Reset printer	
1	Printer tidak dapat mencetak	0,4
2	Muncul kode error p07/e08 pada panel LCD	0,9
3	Lampu indikator stop (orange) kedip 7x/8x	0,9
4	Muncul peringatan "Ink absorber is full" pada layar monitor	0,9
J	Reset cartridge	
1	Printer tidak dapat mencetak	0,3
2	Muncul kode error e15/e16 pada panel LCD	0,8
3	Lampu indikator stop (orange) kedip 15x/16x	0,8
K	Sensor pita encoder rusak	
1	Hasil cetakan bergaris putih horizontal	0,9
2	Hasil cetakan patah-patah	0,7
L	Sensor timing disk rusak	

	Gejala	Nilai Densitas
1	Hasil cetakan bergaris putih vertikal	0,9
2	hasil cetakan patah-patah	0,6
M	<i>Scanner</i> rusak	
1	<i>Printer</i> tidak dapat mencetak	0,4
2	Muncul kode error p22 pada panel LCD	0,8
3	Hasil <i>scan</i> kurang bagus	0,9

Sumber :Diadaptasi dari Muis (2015)

4.2.2 Basis Pengetahuan

Basis Pengetahuan berisi tentang pengetahuan yang relevan yang diperlukan untuk memahami, merumuskan dan memecahkan persoalan. Basis pengetahuan tersebut mencakup dua elemen dasar yaitu fakta dan aturan khusus yang mengarahkan pengguna pengetahuan untuk memecahkan persoalan khusus dalam domain tertentu. Basis pengetahuan merupakan inti dari sistem ini dimana basis pengetahuan ini merupakan representasi pengetahuan dari seorang pakar. Penalaran berdasarkan pada basis pengetahuan yang ada, memanipulasi dan mengarahkan sesuai kaidah, model dan fakta yang disimpan hingga dicapai suatu kesimpulan.

Tabel pengetahuan gejala kerusakan dan kerusakan *printer* dibedakan menjadi 2 yaitu :

1. Tabel pengetahuan gejala dan nilai densitas kerusakan *printer*.
2. Tabel data aturan kerusakan *printer*.

Pengetahuan gejala dan nilai densitas dari kerusakan *printer* dapat dilihat pada Tabel 4.3, sedangkan data aturan kerusakan *printer* dapat dilihat pada Tabel 4.4.

Tabel 4.3 Akuisisi Nilai Densitas Gejala Kerusakan *Printer*

No	Gejala	Kerusakan												
		A	B	C	D	E	F	G	H	I	J	k	L	M
1	<i>Printer</i> tidak dapat dinyalakan	0,7	0,5											
2	<i>Printer</i> tiba tiba mati	0,8												
3	<i>Driver printer</i> tidak terdeteksi di komputer		0,7											
4	<i>Printer</i> tidak dapat direset		0,7											
5	Lampu indikator <i>power</i> (hijau) dan lampu indikator <i>stop/resume</i> (orange) kedip bergantian 9x		0,8											
6	Muncul kode <i>error p10</i> di panel LCD		0,8											
7	<i>Printer</i> tidak dapat mencetak			0,4	0,4	0,5	0,4	0,4	0,3	0,4	0,3			0,4
8	Hasil cetakan hitam tidak sempurna			0,8										
9	Muncul kode <i>error e05</i> di panel LCD			0,6	0,6	0,8			0,3					
10	Lampu indikator <i>stop</i> (orange) kedip 5x			0,6	0,6	0,8								
11	Hasil cetakan hanya muncul warna tertentu saja				0,8									
12	Muncul peringatan <i>Ink absorber is full</i>									0,9				

No	Gejala	Kerusakan												
		A	B	C	D	E	F	G	H	I	J	k	L	M
13	Kertas macet						0,6							
14	Lampu indikator <i>stop (orange)</i> kedip 3x						0,7							
15	Muncul kode <i>error</i> e03 di panel LCD						0,7							
16	<i>Printer</i> menarik kertas double							0,8						
17	Lampu indikator <i>stop (orange)</i> kedip 2x							0,7						
18	Muncul kode <i>error</i> e02 pada panel LCD							0,7						
19	Hasil cetakan tidak bagus								0,4					
20	Muncul kode <i>error</i> p02 pada panel LCD								0,7					
21	Lampu indikator <i>power</i> (hijau) dan lampu indikator <i>stop/resume (orange)</i> kedip bergantian 2x								0,7					
22	Muncul kode <i>error</i> p07/e08 pada panel LCD									0,9				
23	Lampu indikator <i>stop (orange)</i> kedip 7x/8x									0,9				
24	Muncul kode <i>error</i> e15/e16 pada panel LCD										0,8			
25	Lampu indikator <i>stop (orange)</i> kedip 15x/16x										0,8			

No	Gejala	Kerusakan												
		A	B	C	D	E	F	G	H	I	J	k	L	M
26	Hasil cetakan bergaris putih horizontal											0,9		
27	Hasil cetakan patah-patah											0,7	0,6	
28	Hasil cetakan bergaris putih vertikal												0,9	
29	Muncul kode <i>error</i> p22 pada panel LCD													0,8
30	Hasil <i>scan</i> kurang bagus													0,9

Keterangan

A = *Power supply* rusak.

B = *Mainboard* rusak.

C = *Cartridge* hitam rusak.

D = *Cartridge* warna rusak.

E = *Cartridge* tidak terdeteksi.

F = Sensor ASF penarik kertas rusak.

G = *Roll* ASF Penarik kertas rusak.

H = Rumah *cartridge* rusak.

I = *Reset printer*.

J = *Reset cartridge*.

K = Sensor pita *encoder* rusak.

L = Sensor *timing disk* rusak.

M = *Scanner* rusak.

Tabel 4.4 Aturan kerusakan

Aturan	Kerusakan	Gejala
R1	<i>Power supply</i> rusak	<i>Printer</i> tidak dapat dinyalakan, <i>printer</i> tiba – tiba mati.
R2	<i>Mainboard</i> rusak	<i>Printer</i> tidak dapat dinyalakan, <i>driver printer</i> tidak terdeteksi di komputer, <i>printer</i> tidak dapat <i>direset</i> , lampu indikator <i>stop/resume (orange)</i> kedip bergantian 9x, muncul kode <i>error p10</i> di panel LCD .
R3	<i>Cartridge</i> hitam rusak	<i>Printer</i> tidak dapat mencetak, hasil cetakan hitam tidak sempurna, muncul kode error <i>e05</i> di panel LCD, lampu indikator <i>stop (orange)</i> kedip 5x .
R4	<i>Cartridge</i> warna rusak	<i>Printer</i> tidak dapat mencetak, hasil cetakan hanya muncul warna tertentu saja, lampu indikator <i>stop (orange)</i> kedip 5x, muncul kode <i>error e05</i> di panel LCD.
R5	<i>Cartridge</i> tidak terdeteksi	<i>Printer</i> tidak dapat mencetak, lampu indikator <i>stop (orange)</i> kedip 5x, muncul kode <i>error e05</i> di panel LCD.
R6	Sensor ASF penarik kertas rusak	<i>Printer</i> tidak dapat mencetak, kertas macet, lampu indikator <i>stop (orange)</i> kedip 3x, muncul kode <i>error e03</i> di panel LCD.
R7	<i>Roll</i> ASF Penarik kertas rusak	<i>Printer</i> menarik kertas double, lampu indikator <i>stop (orange)</i> kedip 2x, muncul kode error <i>e02</i> pada panel LCD, <i>printer</i> tidak dapat mencetak.
R8	Rumah <i>cartridge</i> rusak	<i>Printer</i> tidak dapat mencetak, hasil cetakan tidak bagus, muncul kode <i>error p02</i> pada panel LCD, lampu indikator <i>power (hijau)</i> dan lampu indikator <i>stop/resume (orange)</i> kedip bergantian 2x, muncul kode <i>error e05</i> pada panel LCD.
R9	<i>Reset printer</i>	<i>Printer</i> tidak dapat mencetak, muncul kode <i>error p07/e08</i> pada panel LCD, lampu indikator <i>stop (orange)</i> kedip 7x/8x, muncul peringatan “ <i>ink absorber is full</i> ” pada layar monitor.

Aturan	Kerusakan	Gejala
R10	<i>Reset cartridge</i>	<i>Printer</i> tidak dapat mencetak, muncul kode error e15/e16 pada pnel LCD, lampu indikator <i>stop (orange)</i> kedip 15x/16x
R11	Sensor pita <i>encoder</i> rusak	Hasil cetakan bergaris putih horizontal, hasil cetakan patah – patah.
R12	Sensor <i>timing disk</i> rusak	Hasil cetakan bergaris putih vertikal, hasil cetakan patah – patah.
R13	<i>Scanner</i> rusak	<i>Printer</i> tidak dapat mencetak, muncul kode error p22 pada panel LCD, hasil scan kurang bagus.

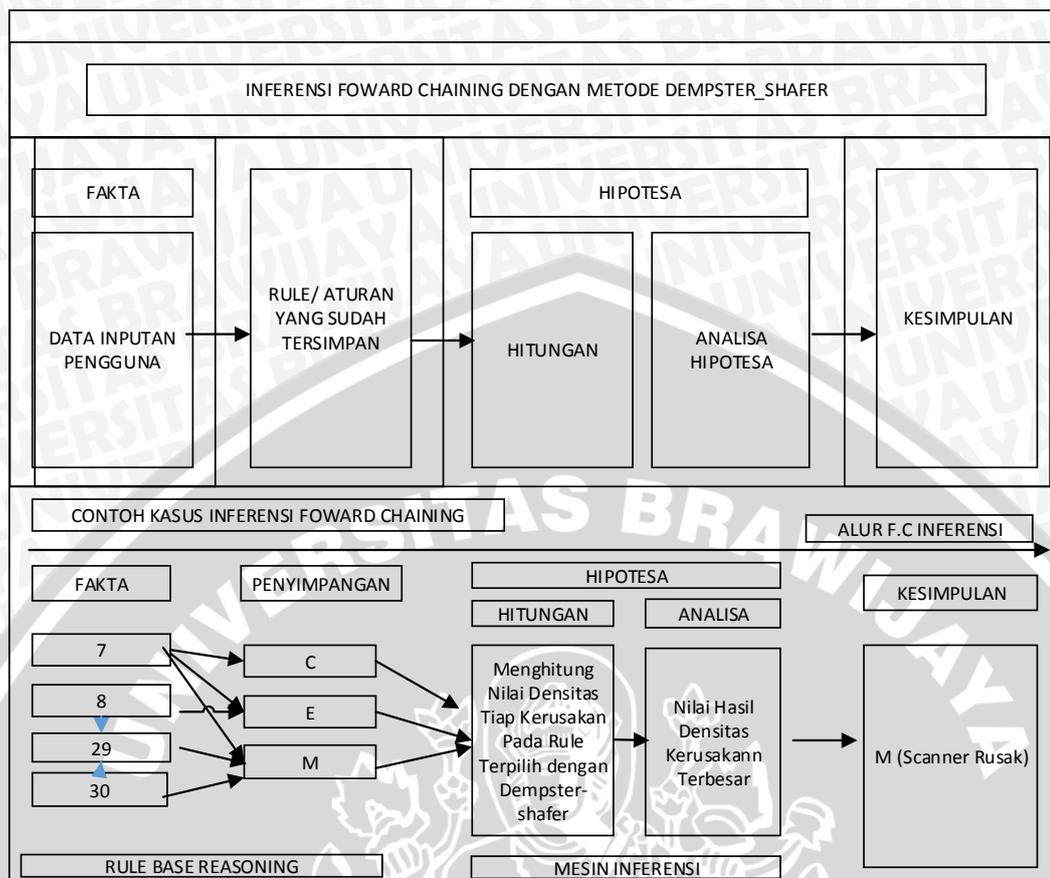
Sumber : Diadaptasi dari Muis (2015)

Data pengetahuan ini selanjutnya disimpan dalam *database* dalam bentuk idpengetahuan, namakerusakan, namagejala. Apabila terdapat gejala *input* sistem akan mencocokkan data masukan dengan *database* kemudian diambil nilai densitasnya dan dihitung dengan metode *Dempster-shafer* untuk mendapatkan kemungkinan kerusakan yang terjadi.

2.3.4 Mesin Inferensi

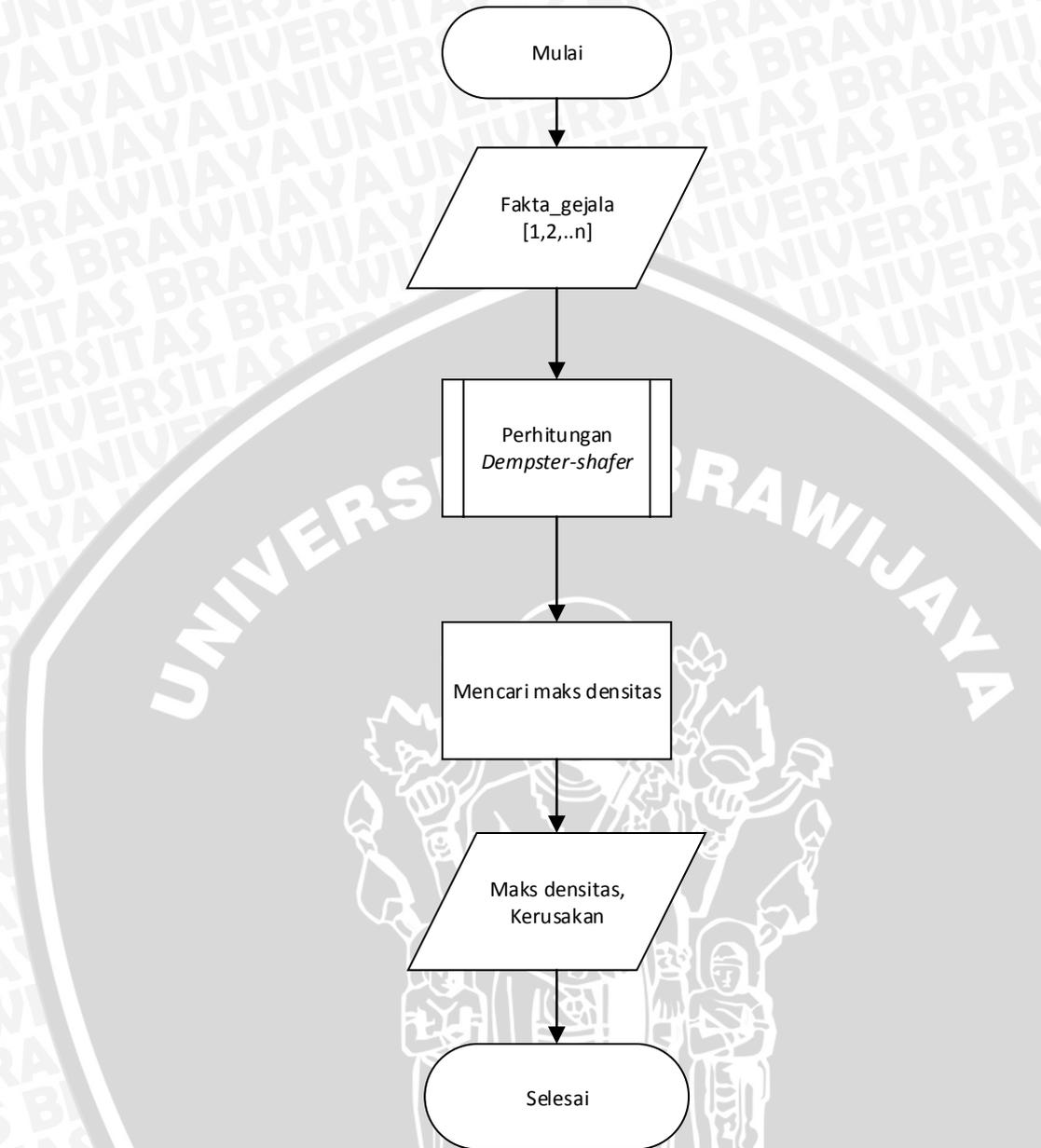
Mesin inferensi berfungsi untuk memandu proses penalaran terhadap suatu kondisi, berdasarkan pada basis pengetahuan yang tersedia. Di dalam mesin inferensi terjadi proses untuk memanipulasi dan mengarahkan kaidah, model, dan fakta yang disimpan dalam basis pengetahuan dalam rangka mencapai solusi atau kesimpulan. Mesin inferensi yang digunakan pada sistem ini adalah *Forward Chaining* dan *Dempster-shafer*. Pada penelitian ini penelusuran dimulai dari premis (gejala) untuk menentukan konklusi (kerusakan). Teknik seperti ini disebut teknik *Forward Chaining*. Hasil penelusuran didapat berdasarkan pada nilai kepastian tiap premis (gejala) yang dihitung menggunakan metode *Dempster-shafer*.

Metode *Forward chaining* yaitu dimulai dari sekumpulan fakta-fakta tentang suatu gejala yang diberikan oleh pengguna sebagai masukan pada sistem. Kemudian dilakukan proses pelacakan dari masukan gejala dengan gejala yang tersimpan pada basis data untuk kemudian diambil nilai kepercayaan tiap kerusakan yang sesuai. Setelah didapat nilai bobotnya maka dilakukan hipotesa yang terdiri dari 2 bagian yaitu proses perhitungan dengan menggunakan *Dempster-shafer* dan analisa hipotesa hasil dari perhitungan akhir yang kemudian dijadikan sebagai kesimpulan. Hipotesa blok diagram alur proses metode inferensi *Foward chaining* dapat dilihat pada Gambar 4.3.



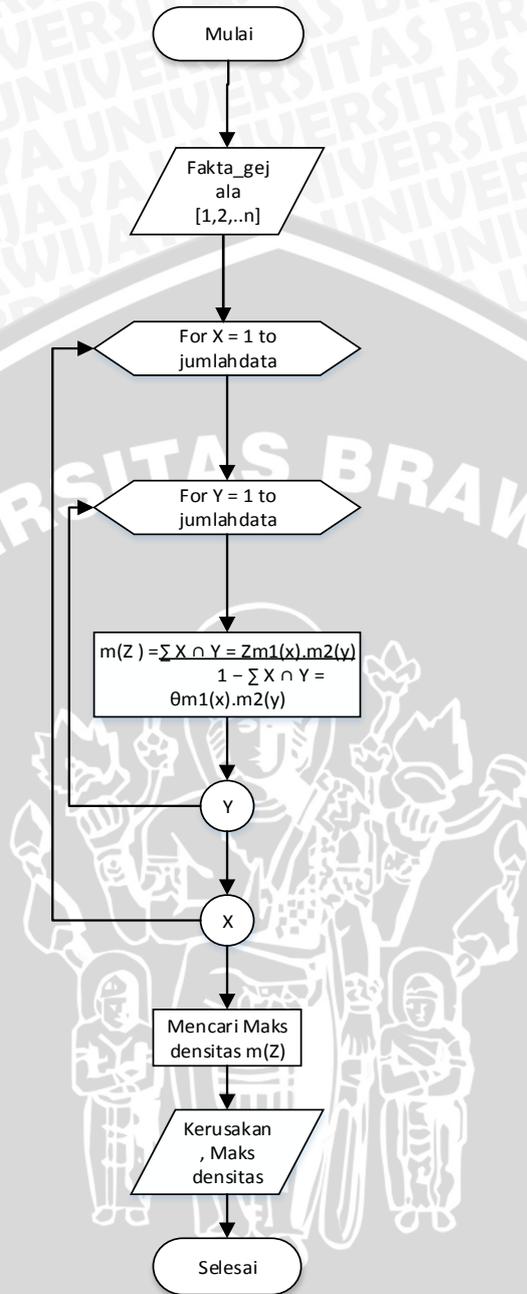
Gambar 4.3 Mesin Inferensi Foward Chaining Dengan Metode Dempster-Shafer

Gambar 4.3 merupakan *flowchart* atau diagram alir gambaran pencarian solusi sistem pakar menggunakan metode *Dempster-shafer* sebagai penarikan kesimpulan. Alur proses sistem identifikasi kerusakan *printer* dapat dilihat seperti pada Gambar 4.4 dan Gambar 4.5.



Gambar 4.4 Diagram Alir Proses Sistem Identifikasi Kerusakan *Printer*

Dempster-Shafer



Gambar 4.5 Diagram Alir Inferensi Dempster-shafer

Proses identifikasi dimulai dengan memasukkan data gejala kemudian akan masuk ke *rule of combination* dari *dempster-shafer* dan mencari nilai densitas terbesarnya lalu proses akan selesai.

Rancangan algoritma proses identifikasi kerusakan dapat dilihat pada Gambar 4.6 .



Nama Algoritma : Identifikasi Kerusakan *Printer*

Deklarasi :

- String : Kerusakan, Gejala, Basis Pengetahuan, Hasil Identifikasi
- Integer : perhitungan, nilai densitas

Deskripsi :

- Input : Gejala
- Proses :
 - a. Mengecek field telah terisi atau belum.
 - b. Mengecek jumlah input.
 - c. Mengambil gejala dari field.
 - d. Data gejala disimpan pada variable array gejala.
 - e. Mencocokkan data array gejala dengan gejala pada database untuk mengambil nilai densitas.
 - f. Mencari hasil nilai densitas tertinggi dari hasil perhitungan Dempster-Shafer yang didapat sebagai jawaban kesimpulan kerusakannya.
- Output : Menampilkan hasil kesimpulan kerusakan dan nilai kepercayaannya.

Gambar 4.6 Rancangan Algoritma Proses Identifikasi Kerusakan *Printer*

4.5.3 Blackboard

Blackboard merupakan area memori yang berfungsi sebagai basis data untuk merekam hasil sementara. *Blackboard* berisi rencana solusi yang berupa data yang digunakan sebagai bahan pertimbangan dalam memberikan kesimpulan akhir. Pada aplikasi identifikasi kerusakan printer ini, data yang disimpan pada area ini adalah data gejala masukan dari *user*, nilai perhitungan *belief* dan *plausibility* tiap gejala, hasil kombinasi nilai densitas dan hasil identifikasi kerusakan.

4.5.4 Fasilitas Penjelas

Pada umumnya, fasilitas penjelas berisi tuntunan penggunaan aplikasi identifikasi kerusakan *printer*. Fasilitas penjelas yang akan diberikan dalam aplikasi ini yaitu menjelaskan proses identifikasi sehingga mendapatkan kesimpulan beserta nilai densitasnya. Fasilitas penjelas ini penting bertujuan memberikan informasi kepada para pengguna mengenai manfaat dari aplikasi dan bagaimana cara penggunaannya.

4.5.6 Perhitungan Manual

Dalam proses perhitungan manualisasi implementasi metode *Dempster-Shafer* terdapat dalam beberapa langkah. Dalam contoh manualisasi akan dibagi menjadi 2 kasus, yaitu kasus 1 dengan perhitungan 2 gejala dimasukkan. Kemudian dilakukan percobaan pada kasus 2, yaitu dengan perhitungan 3 gejala dimasukkan.

a) Kasus 1 (Perhitungan 2 Gejala)

Pada kasus ini akan diberikan contoh dengan memasukkan 2 gejala. Perhitungan ini dimisalkan seorang pengguna merasakan gejala yang aneh dengan *printer*, yaitu *printer* tidak dapat dinyalakan, *printer* tiba – tiba mati.

- **Gejala 1 : *Printer* tidak dapat dinyalakan**

Nilai densitas pada tiap gejala akan diambil yang mempunyai nilai densitas tertinggi. Setelah dilakukan pengamatan *printer* susah dihidupkan adalah sebagai gejala kerusakan *power supply* rusak, dan *mainboard* rusak,

$m\{\text{power supply rusak}\} = 0,70$; $m\{\text{mainboard rusak}\} = 0,50$; Untuk m_0 nilai densitas yang dipilih adalah yang tertinggi, maka :

$$m_0 \{A, B, \} = 0,70$$

$$t_0 \{ \theta \} = 1 - 0,70 = 0,30$$

- **Gejala 2 : *Printer* tiba - tiba mati**

Nilai densitas pada tiap gejala akan diambil yang mempunyai nilai densitas tertinggi. Setelah dilakukan pengamatan *printer* tiba - tiba mati adalah sebagai gejala kerusakan *power supply* rusak :

$m\{\text{power supply rusak}\} = 0,80$. Untuk m_1 nilai densitas yang dipilih adalah yang tertinggi, maka :

$$m_1 \{A \} = 0,80$$

$$t_1 \{ \theta \} = 1 - 0,80 = 0,20$$

Maka dihitung nilai densitas baru untuk beberapa kombinasi dengan fungsi densitas m_2 dapat dilihat pada Tabel 4.5 .

Tabel 4.5 Aturan Kombinasi m_2

	$m_1 \{A\} = 0,80$	$t_1 = 0,20$
$m_0 \{A, B \} = 0,70$	$\{A\} = 0,56$	$\{A, B\} = 0,14$
$t_0 = 0,30$	$\{A\} = 0,24$	$\emptyset = 0,60$



Sehingga dapat dihitung dengan persamaan 2.8:

$$m_2 \{A\} = \frac{0,56+0,24}{1-0} = 0,80$$

$$m_2 \{A, B\} = \frac{0,14}{1-0} = 0,14$$

$$m_2 \{\theta\} = \frac{0,60}{1-0} = 0,60$$

Dari hasil perhitungan dengan metode *Dempster-Shafer*, nilai densitas paling tinggi adalah pada *power supply* rusak. Maka dapat disimpulkan bahwa *printer* pengguna mengalami kerusakan di *power supply*.

b. Kasus 2 (Perhitungan 3 gejala)

Pada kasus ini akan diberikan contoh dengan memasukkan 3 gejala. Perhitungan ini dimisalkan seorang pengguna merasakan gejala yang aneh dengan *printernya*, yaitu hasil cetakan bergaris putih vertikal, hasil cetakan patah-patah, hasil cetakan hitam tidak sempurna.

- **Gejala 1 : Hasil cetakan bergaris putih vertikal**

Nilai densitas pada tiap gejala akan diambil yang mempunyai nilai densitas tertinggi. Setelah dilakukan pengamatan hasil cetakan bergaris putih vertikal adalah sensor timing disk rusak.

$m\{\text{sensor timing disk rusak}\} = 0,90$ sehingga nilai densitas yang dipilih adalah sensor timing disk :

$$m_0 \{L\} = 0,90$$

$$t_0 \{\theta\} = 1 - 0,90 = 0,10$$

- **Gejala 2 : hasil cetakan patah - patah**

Nilai densitas pada tiap gejala akan diambil yang mempunyai nilai densitas tertinggi. Setelah itu dilakukan hasil *print* kabur adalah sebagai gejala kerusakan *cartridge* warna rusak, dan *cartridge* hitam rusak :

$$m\{\text{sensor timing disk rusak}\} = 0,60 ; m\{\text{sensor pita encoder rusak}\} = 0,70 ;$$

Untuk m_1 nilai densitas yang dipilih adalah yang tertinggi, maka :

$$m_1 \{L\} = 0,70$$

$$t_1 \{\theta\} = 1 - 0,70 = 0,30$$

Maka dihitung nilai densitas baru untuk beberapa kombinasi dengan fungsi m_2 dapat dilihat pada Tabel 4.6.

Tabel 4.6 Aturan Kombinasi m2

	$m1\{K,L\} = 0,70$	$t1 = 0,30$
$m0 \{L\} = 0,90$	$\{L\} = 0,63$	$\{L\} = 0,27$
$t0 = 0,10$	$\{K,L\} = 0,07$	$\emptyset = 0,03$

Sehingga dapat dihitung :

$$m2 \{L\} = \frac{0,63+0,27}{1-0} = 0,90$$

$$m2\{K,L\} = \frac{0,07}{1-0} = 0,07$$

$$m2 \{\theta\} = \frac{0,03}{1-0} = 0,03$$

- **Gejala 3 : hasil cetakan hitam tidak sempurna**

Nilai densitas pada tiap gejala akan diambil yang mempunyai nilai densitas tertinggi. Setelah itu dilakukan pengamatan hasil cetakan hitam tidak sempurna adalah sebagai gejala kerusakan cartridge hitam:

$m\{\text{cartridge hitam rusak}\} = 0,80$ maka:

$$m3 \{C\} = 0,80$$

$$t3 \{\theta\} = 1 - 0,80 = 0,20$$

Maka dihitung nilai densitas baru untuk beberapa kombinasi dengan fungsi $m4$ dapat dilihat pada Tabel 4.7 .

Tabel 4.7 Aturan Kombinasi m4

	$m3 \{C\} = 0,80$	$t3 = 0,20$
$m2 \{L\} = 0,90$	$\{L\} = 0,72$	$\{L\} = 0,18$
$m2 \{K,L\} = 0,07$	$\{L\} = 0,056$	$\{K,L\} = 0,014$
$\{\theta\} = 0,03$	$\{C\} = 0,024$	$\emptyset = 0,006$

Sehingga dapat dihitung :

$$m4 \{L\} = \frac{0,18}{1-(0,72+0,056)} = 0,803$$

$$m4 \{K,L\} = \frac{0,014}{1-(0,72+0,056)} = 0,0625$$

$$m4 \{C\} = \frac{0,024}{1-(0,72+0,056)} = 0,107$$

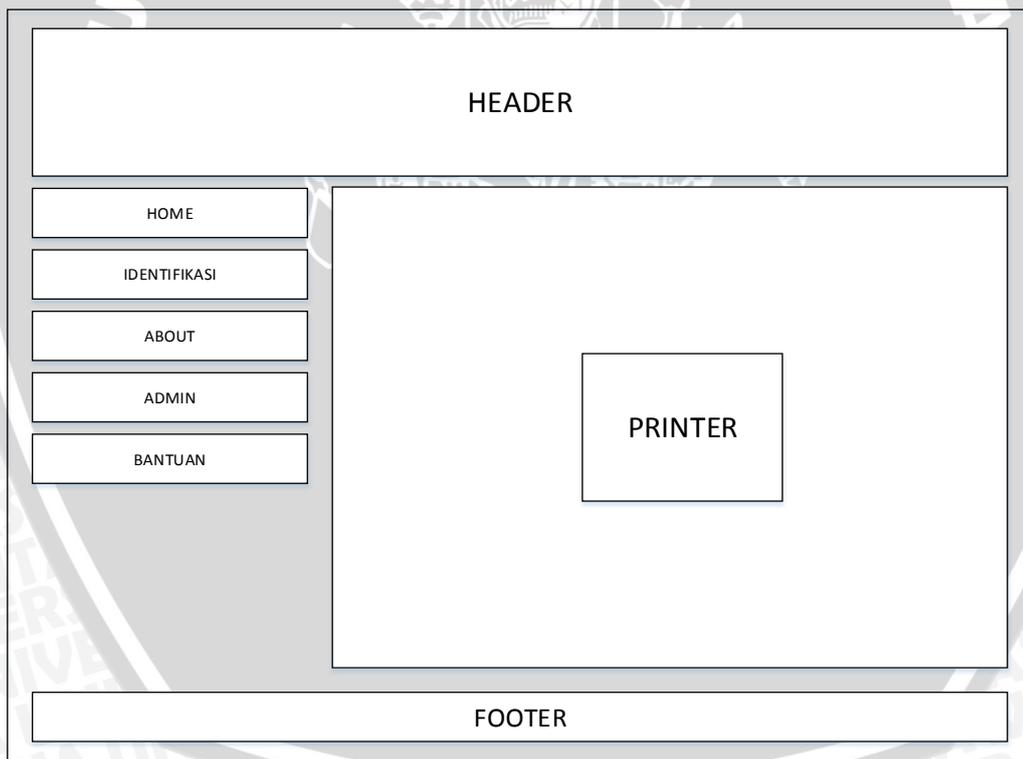
$$m4 \{\emptyset\} = \frac{0,006}{1-(0,72+0,056)} = 0.026$$

Dari hasil perhitungan dengan metode *Dempster-Shafer*, nilai densitas paling tinggi adalah pada kerusakan pada kode kerusakan (L) atau kerusakan pada sensor timing disk dengan nilai kepercayaan sebesar 0,803.

4.5.7 Antarmuka

Antarmuka merupakan mekanisme yang berfungsi untuk menjadi sarana komunikasi pengguna dengan sistem. seperti melihat informasi yang ada dalam sistem, melakukan identifikasi dan lain sebagainya. Program menampilkan gejala-gejala yang nantinya akan dipilih oleh pengguna, kemudian sistem akan melakukan diagnosis sesuai gejala yang telah dipilih.

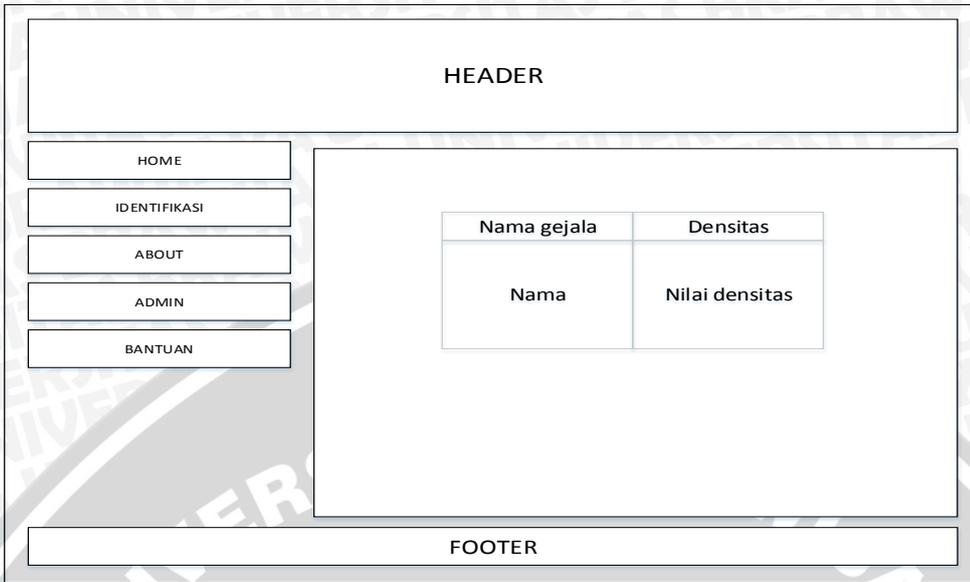
4.5.7.1 Antarmuka Halaman Utama



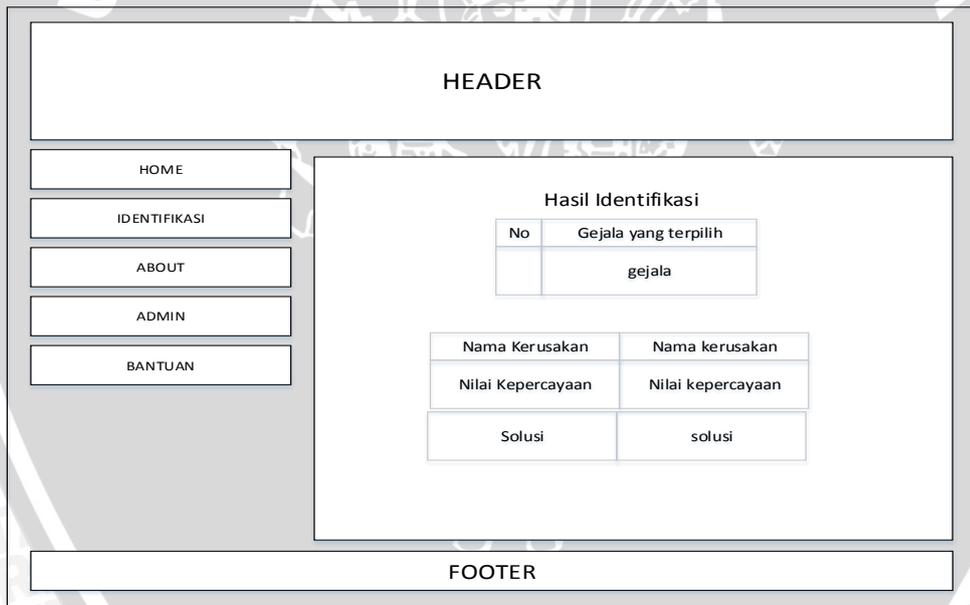
Gambar 4.7 Rancangan Antarmuka Halaman Utama

Pada Gambar 4.7 merupakan rancangan antarmuka halaman utama yang terdiri dari *header*, *footer*, menu *home*, *identifikasi*, *about*, *admin*, dan *bantuan*. Pengguna bisa langsung melakukan identifikasi dengan menekan tombol *identifikasi*.

4.5.7.2 Antarmuka Halaman Identifikasi dan Hasil Identifikasi



Gambar 4.8 Rancangan Antarmuka Halaman Identifikasi

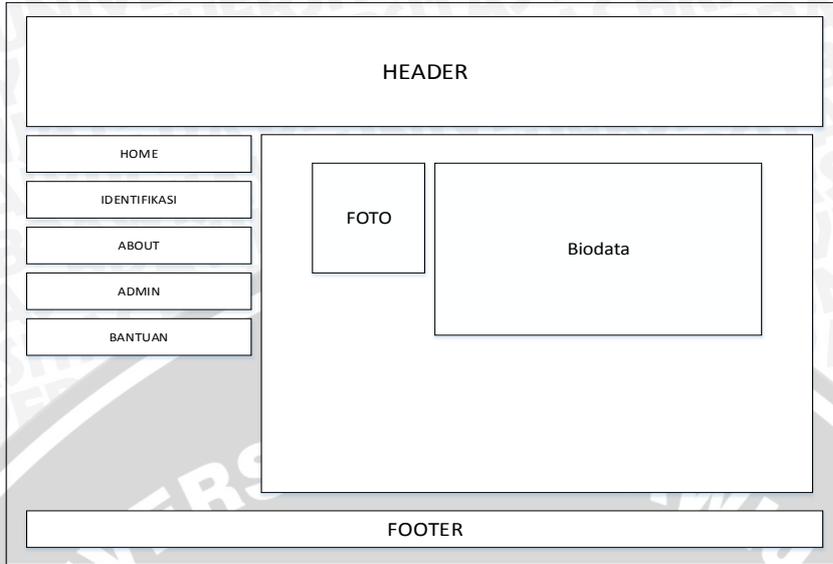


Gambar 4.9 Rancangan Antarmuka Hasil Identifikasi

Pada Gambar 4.8 adalah rancangan halaman identifikasi untuk pengguna. Di halaman tersebut dapat dilihat gejala-gejala yang dapat dipilih oleh pengguna sesuai dengan fakta yang dialami oleh pengguna. Kemudian di Gambar 4.9 adalah rancangan halaman hasil identifikasi, pengguna bisa melihat hasil dari identifikasi sesuai dengan gejala yang dimasukkan.

repository.ub.ac.id

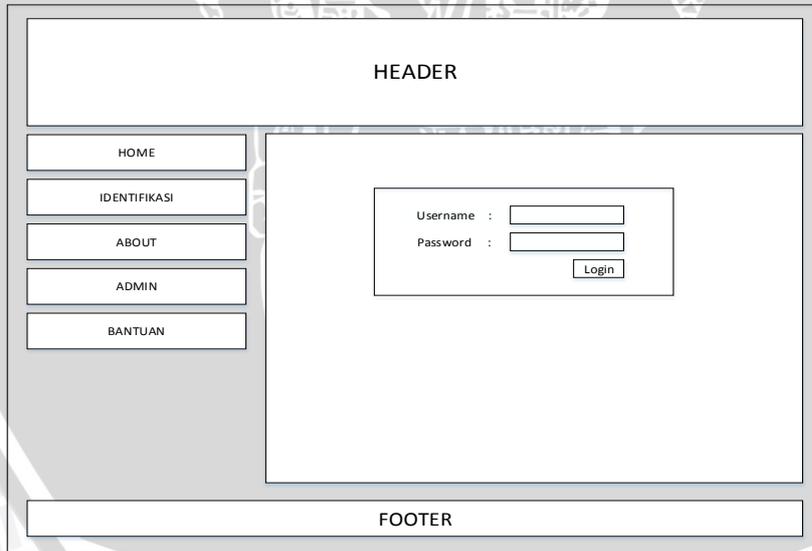
4.5.7.3 Antarmuka Halaman About



Gambar 4.10 Rancangan Antarmuka Halaman about

Pada Gambar 4.10 adalah rancangan halaman yang menerangkan tentang biodata pembuat aplikasi identifikasi kerusakan *printer* menggunakan metode *Dempster-Shafer*.

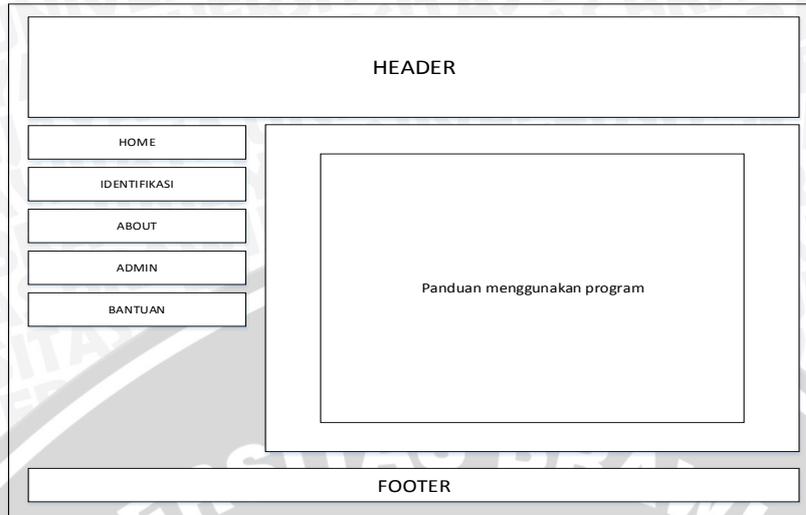
4.5.7.4 Antarmuka Halaman Login Admin



Gambar 4.11 Rancangan Antarmuka Halaman Login Admin

Pada Gambar 4.11 adalah rancangan antarmuka halaman *login* admin. Admin memasukkan *username* dan *password*, kemudian sistem akan mencocokkan dengan *database*.

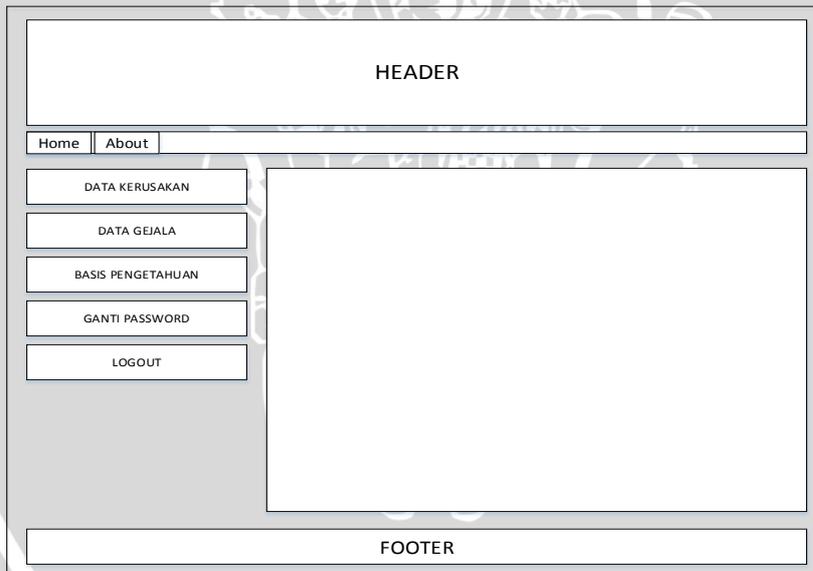
4.5.7.5 Antarmuka Halaman Bantuan



Gambar 4.12 Rancangan Antarmuka Halaman Bantuan

Pada Gambar 4.12 adalah rancangan antarmuka halaman bantuan. Halaman ini berisi informasi cara penggunaan sistem.

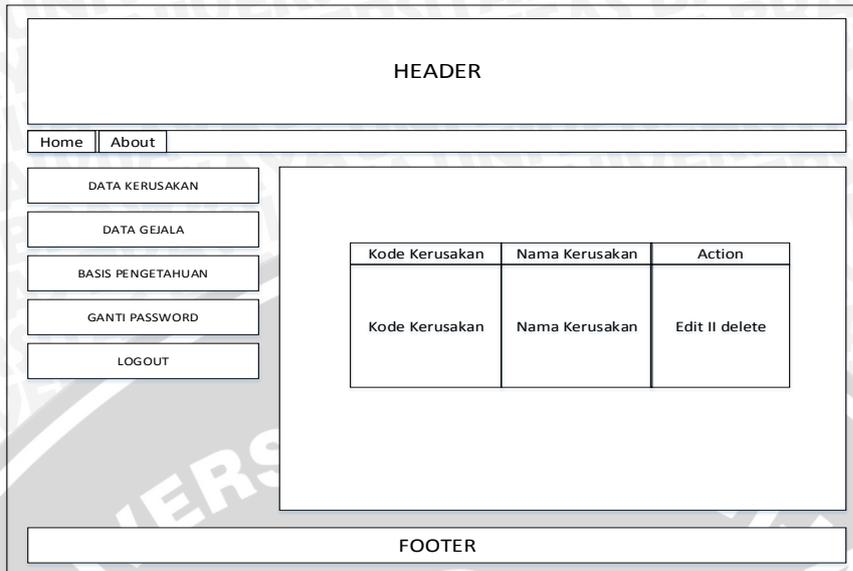
4.5.7.6 Antarmuka Halaman Admin



Gambar 4.13 Rancangan Antarmuka Halaman Admin

Pada Gambar 4.13 adalah rancangan antarmuka halaman Admin. Pada halaman ini terdapat menu data kerusakan, data gejala, basis pengetahuan, ganti password dan logout.

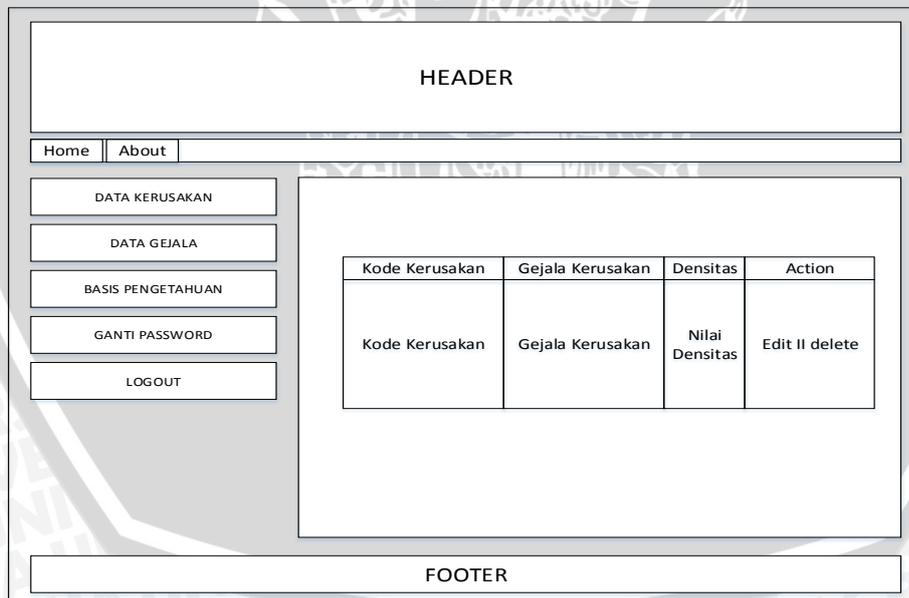
4.5.7.7 Antarmuka Halaman Data Kerusakan



Gambar 4.14 Rancangan Antarmuka Halaman Data Kerusakan

Pada Gambar 4.14 adalah rancangan antarmuka halaman data kerusakan. Pada halaman ini terdapat kode kerusakan, nama kerusakan dan *action* yang dapat dilakukan oleh admin yaitu edit dan *delete* kerusakan.

4.5.7.8 Antarmuka Halaman Data Gejala Kerusakan



Gambar 4.15 Rancangan Antarmuka Halaman Data Kerusakan

Pada Gambar 4.15 adalah rancangan antarmuka halaman data kerusakan. Di halaman ini admin menambah, mengedit dan menghapus data gejala.

4.5.7.9 Antarmuka Halaman Basis Pengetahuan

The wireframe shows a page layout with a header, navigation links, a sidebar menu, a main content area with a table, and a footer.

HEADER			
Home	About		
DATA KERUSAKAN			
DATA GEJALA			
BASIS PENGETAHUAN			
GANTI PASSWORD			
LOGOUT			
Id Pengetahuan	Nama kerusakan	Nama gejala	Action
Id pengetahuan	Nama kerusakan	Gejala kerusakan	Edit delete
FOOTER			

Gambar 4.16 Rancangan Antarmuka Halaman Basis Pengetahuan

Pada Gambar 4.16 adalah rancangan halaman basis pengetahuan, dimana dalam halaman tersebut admin dapat melihat data-data gejala yang sudah dimasukkan sebelumnya, menambah, mengedit dan menghapus data.

4.5.7.10 Antarmuka Halaman Ganti *Password* Admin

The wireframe shows a page layout with a header, navigation links, a sidebar menu, a main content area with a login form, and a footer.

HEADER		
Home	About	
DATA KERUSAKAN		
DATA GEJALA		
BASIS PENGETAHUAN		
GANTI PASSWORD		
LOGOUT		
Username :	<input type="text"/>	
Password :	<input type="password"/>	<input type="submit" value="submit"/>
FOOTER		

Gambar 4.17 Rancangan Antarmuka Halaman Ganti *Password* Admin

Pada Gambar 4.17 adalah rancangan halaman untuk admin. Dimana fungsi dari halaman ini adalah pergantian *password* baru hanya untuk akun admin.

4.5.7.11 Antarmuka Halaman Tambah/Ubah Data Kerusakan

HEADER

Home | About

DATA KERUSAKAN

DATA GEJALA

BASIS PENGETAHUAN

GANTI PASSWORD

LOGOUT

Kode Kerusakan :

Nama Kerusakan :

Solusi :

submit

FOOTER

Gambar 4.18 Rancangan Antarmuka Halaman Tambah/Ubah Data Kerusakan

Pada Gambar 4.18 adalah halaman yang muncul apabila admin menambah atau menubah data kerusakan yang ada pada menu data kerusakan.

4.5.7.12 Antarmuka Halaman Tambah/Ubah Data Gejala

HEADER

Home | About

DATA KERUSAKAN

DATA GEJALA

BASIS PENGETAHUAN

GANTI PASSWORD

LOGOUT

Id Gejala :

Nama Gejala :

Nilai Densitas :

submit

FOOTER

Gambar 4.19 Rancangan Antarmuka Halaman Tambah/Ubah Data Gejala

Pada Gambar 4.19 adalah halaman yang muncul apabila admin menambah atau mengubah data gejala yang ada pada menu data gejala.

4.5.7.13 Antarmuka Halaman Tambah/Ubah Basis Pengetahuan

Gambar 4.20 Rancangan Antarmuka Halaman Tambah/Ubah Data Kerusakan

Pada Gambar 4.20 adalah halaman yang muncul apabila admin menambah atau mengubah data basis pengetahuan yang ada pada menu basis pengetahuan.

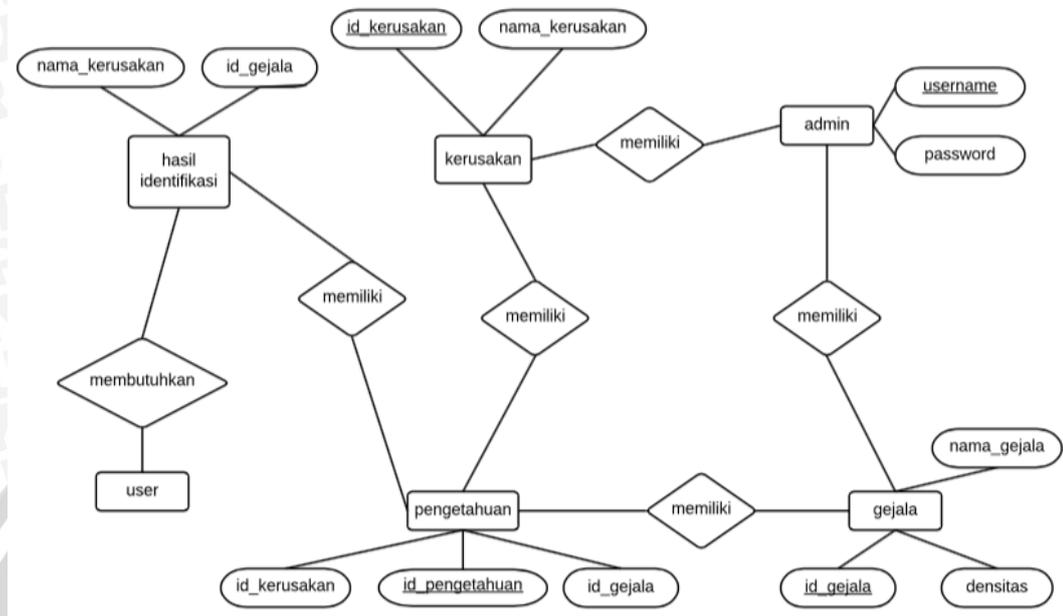
4.6 Perancangan Perangkat Lunak Pemodelan Sistem Pakar Untuk Identifikasi Kerusakan *Printer*

Perancangan perangkat lunak ini menjelaskan mengenai pola hubungan antar komponen-komponen detail sehingga mampuk membentuk sebuah fungsi yang mampu memberikan pelayanan terhadap kebutuhan pengguna. Perancangan perangkat lunak menggunakan *data flow diagram* (DFD) sebagai pemodelan perangkat lunak dan *entity relationship diagram* (ERD) sebagai rancangan databasenya.

4.6.1 Entity Relationship Diagram (ERD)

Entity Relationship Diagram (ERD) adalah diagram yang dipakai untuk mendokumentasikan data dengan mengidentifikasi jenis entitas dan hubungannya. *Entity Relationship Diagram* berisi komponen-komponen himpunan entitas dan himpunan relasi yang masing-masing dilengkapi dengan atribut yang merepresentasikan seluruh fakta yang ditinjau dari keadaan nyata. Pada *Entity Relationship Diagram* aplikasi sistem identifikasi kerusakan *printer* ini terdapat 5 entitas yang digunakan yaitu admin, kerusakan, gejala, basis

pengetahuan dan hasil identifikasi. Untuk entitas pengguna terdapat 2 kategori level *user* yaitu pengguna dan admin. Rancangan Entity Relationship Diagram aplikasi sistem ditunjukkan pada Gambar 4.21.

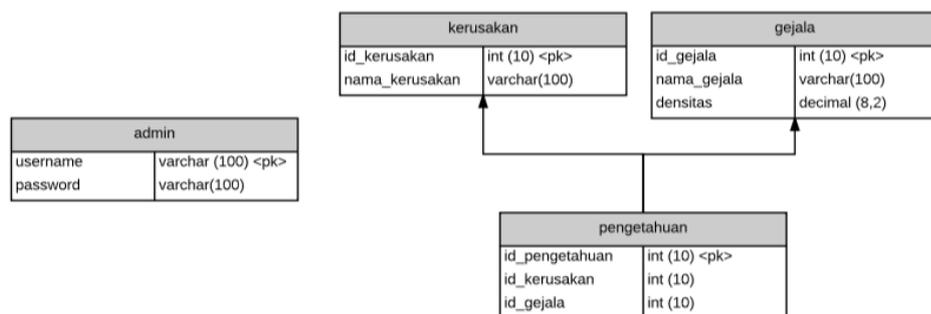


Gambar 4.21 Rancangan Entity Relationship Diagram Sistem

Untuk entitas *user* dibuat sistem kategori dibagi menjadi 2 level, yaitu pengguna dan admin, dimana untuk level pengguna yang dimaksud adalah masyarakat umum dan teknisi. Admin dapat melakukan proses manajemen data kerusakan, data gejala dan data basis pengetahuan yang digunakan untuk proses perhitungan identifikasi.

4.6.2 Physical Diagram

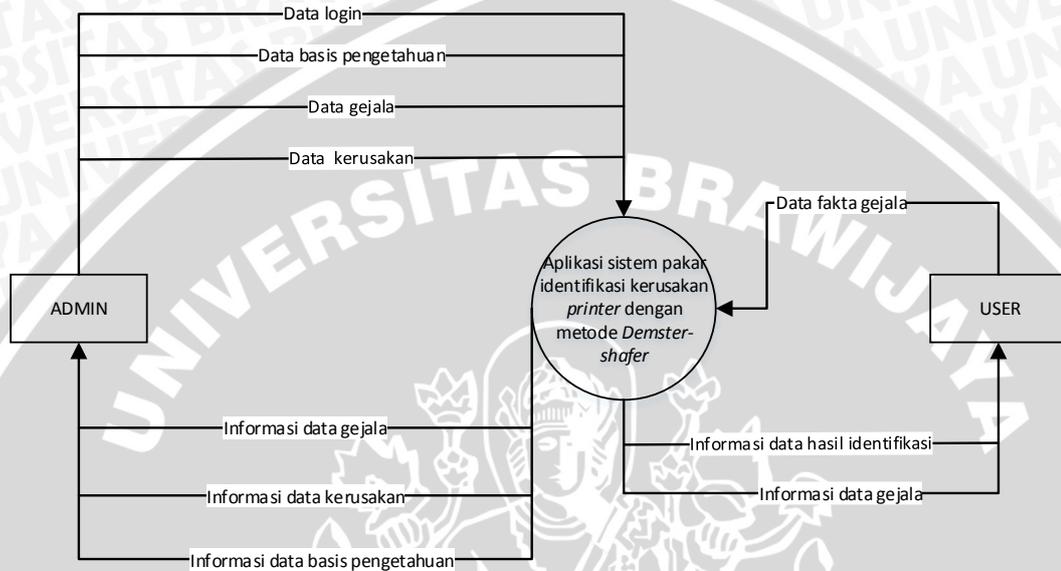
Sistem pakar ini menggunakan MySQL sebagai *Database Management System* (DBMS). *Physical Diagram* merupakan desain nyata (struktur fisik) dari basis data berdasarkan kebutuhan. Sistem pakar pendeteksi kerusakan ini menggunakan 4 tabel yaitu table admin, table gejala, tabel kerusakan, dan tabel pengetahuan. Gambar 4.22 menunjukkan desain *Physical Diagram* dari sistem pakar.



Gambar 4.22 Rancangan Physical Diagram

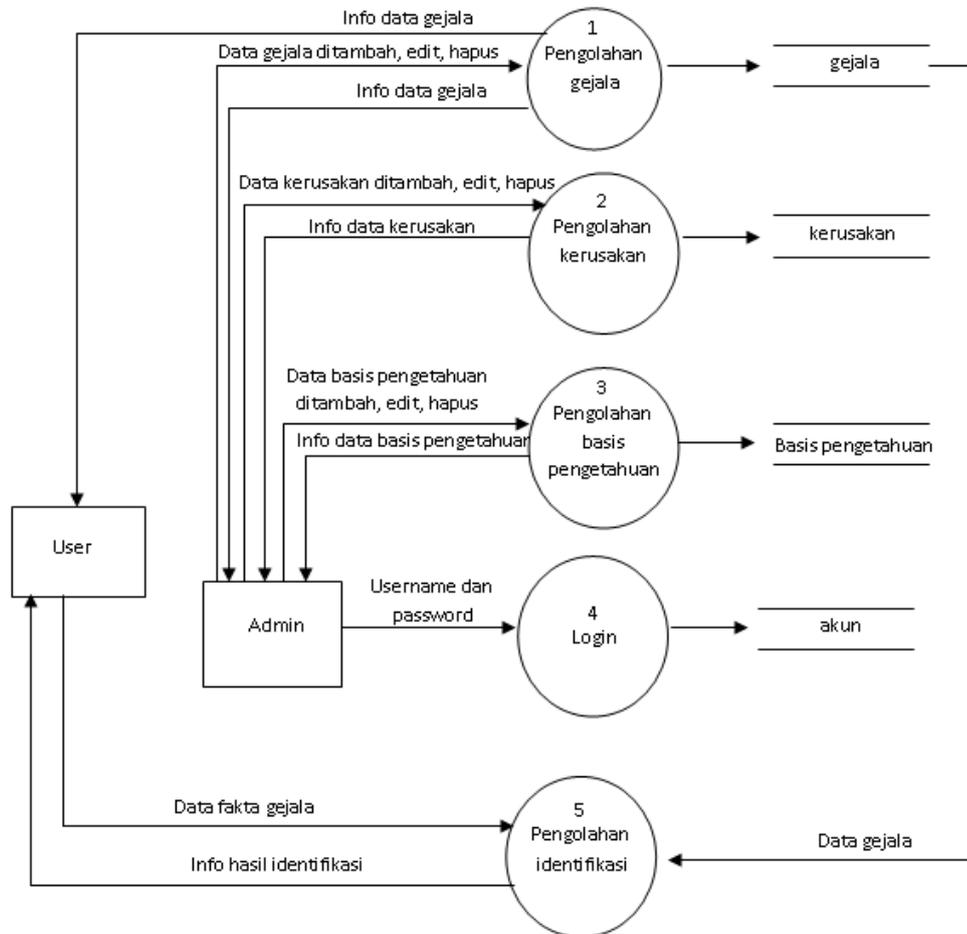
4.6.3 Perancangan *Data Flow Diagram* (DFD)

Data Flow Diagram merupakan gambar yang menggambarkan aliran data pada sebuah sistem informasi. *Data flow diagram* dapat menggambarkan proses-proses di dalam sistem informasi dengan menggunakan sudut pandang data. *Data flow diagram* dapat menunjukkan secara visual bagaimana sistem beroperasi, serta apa penyusun dari sistem dan bagaimana akan diimplementasikan. Pada Gambar 4.23 dapat dilihat *Data Flow Diagram* Sistem Level 0.



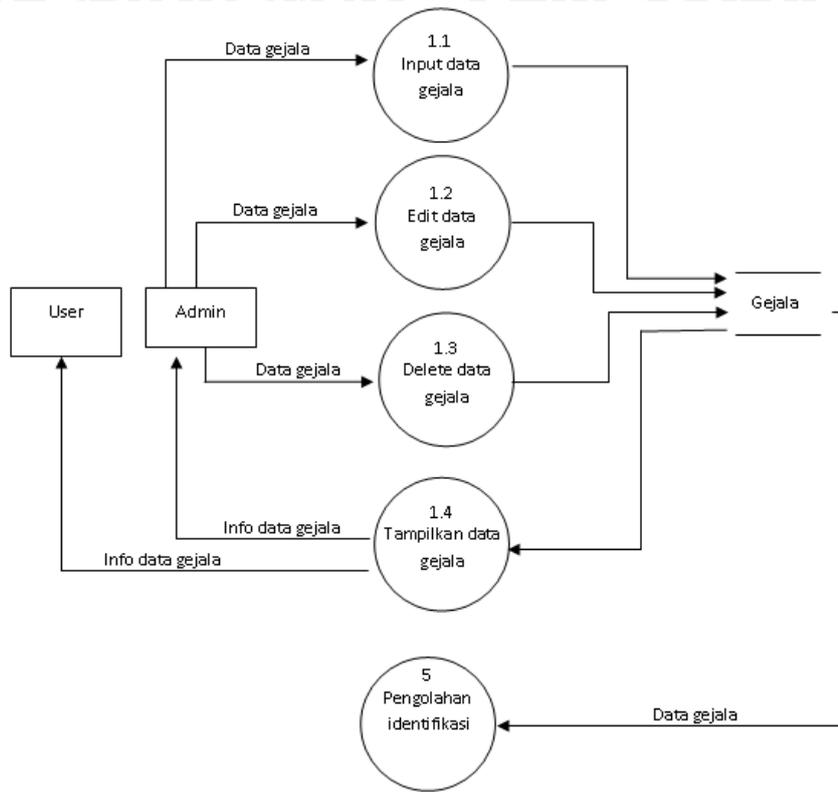
Gambar 4.23 *Data Flow Diagram* Level 0 Sistem Identifikasi Kerusakan Printer

Diagram level 0 akan didekomposisi menjadi beberapa sub proses diagram level selanjutnya yang dinamakan diagram level 1. Terdapat beberapa sub proses pada *diagram flow* selanjutnya antara lain pengolahan gejala, pengolahan kerusakan, pengolahan basis pengetahuan dan pengolahan identifikasi kerusakan. Terdapat beberapa simpanan data diantaranya adalah gejala, kerusakan dan basis pengetahuan. Diagram pada Gambar 4.24 menunjukkan proses diagram level 1.



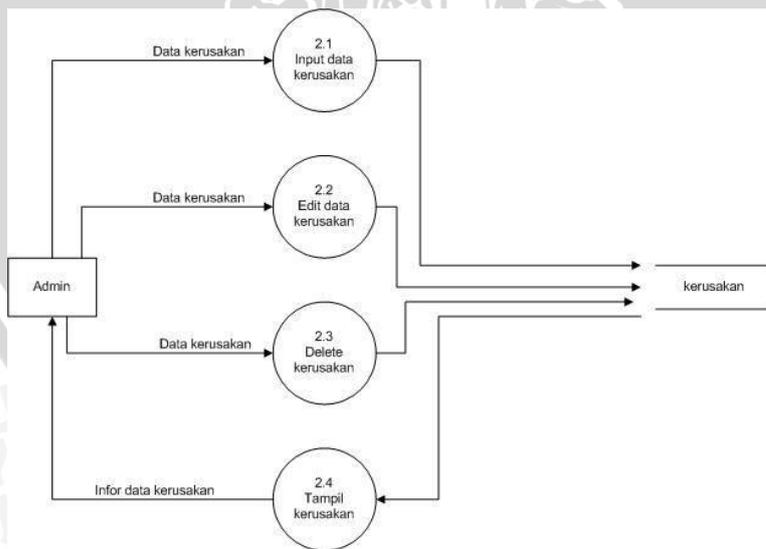
Gambar 4.24 Data Flow Diagram Level 1 Sistem Identifikasi Kerusakan Printer

Setiap proses yang terdapat pada diagram level 1 akan didekomposisi lagi menjadi sub bab proses diagram level selanjutnya. Sub proses pengolahan gejala akan di dekomposisi menjadi 4 proses dan 1 simpanan data. 4 proses terdiri dari *input* data gejala, *edit* data gejala, *delete* data gejala dan *tampil* data gejala. Sedangkan simpanan data yaitu gejala. Gambar 4.25 menunjukkan sub proses gejala.



Gambar 4.25 Data Flow Diagram Level 2 Sub Proses Gejala

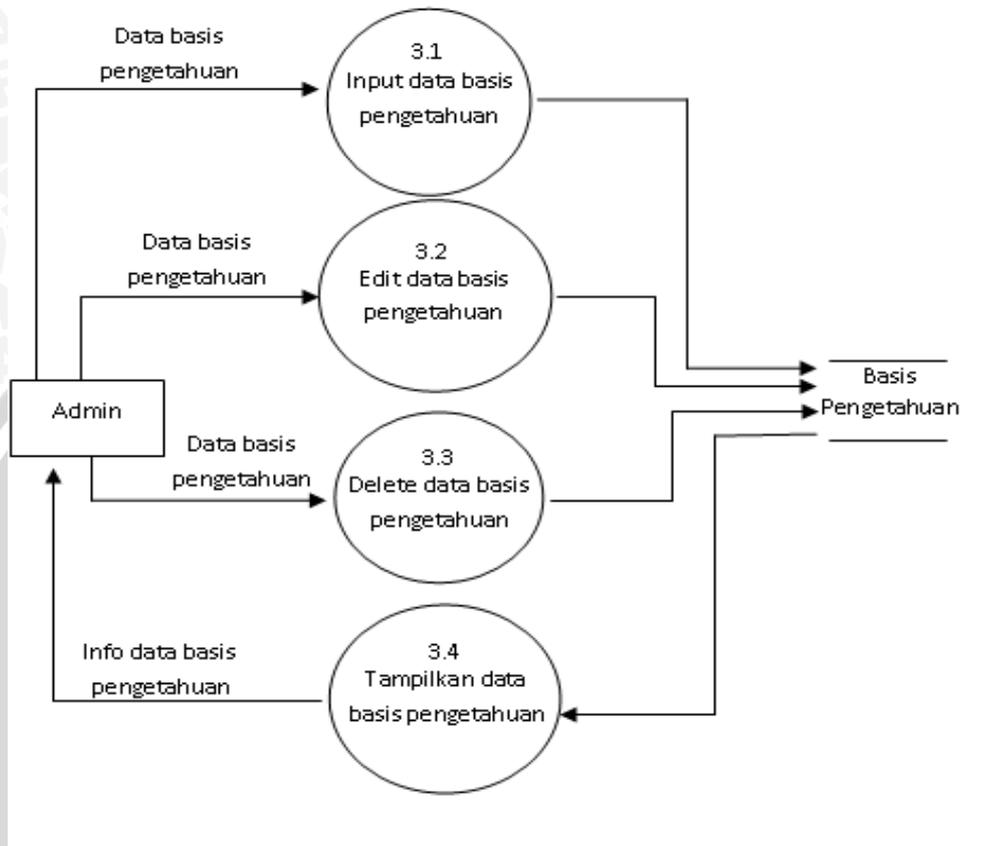
Sub proses pengolahan kerusakan akan di dekomposisi menjadi 4 proses dan 1 simpanan data. 4 proses terdiri dari *input* data kerusakan, *edit* data kerusakan, *delete* data kerusakan dan tampil data kerusakan. Sedangkan simpanan data yaitu kerusakan. Gambar 4.26 menunjukkan sub proses kerusakan.



Gambar 4.26 Data Flow Diagram Level 2 Sub Proses Kerusakan

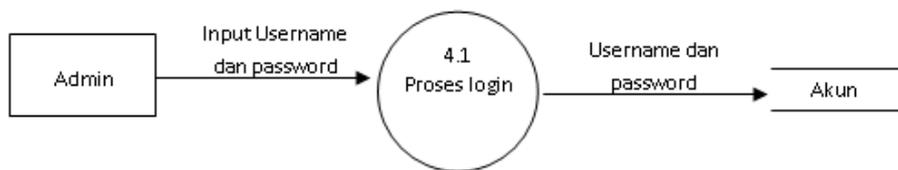
Sub proses pengolahan basis pengetahuan akan di dekomposisi menjadi 4 proses dan 1 simpanan data. 4 proses terdiri dari *input* data pengetahuan, *edit*

data pengetahuan, *delete* data pengetahuan dan tampil data pengetahuan. Sedangkan simpanan data yaitu basis pengetahuan. Gambar 4.27 menunjukkan sub proses basis pengetahuan.



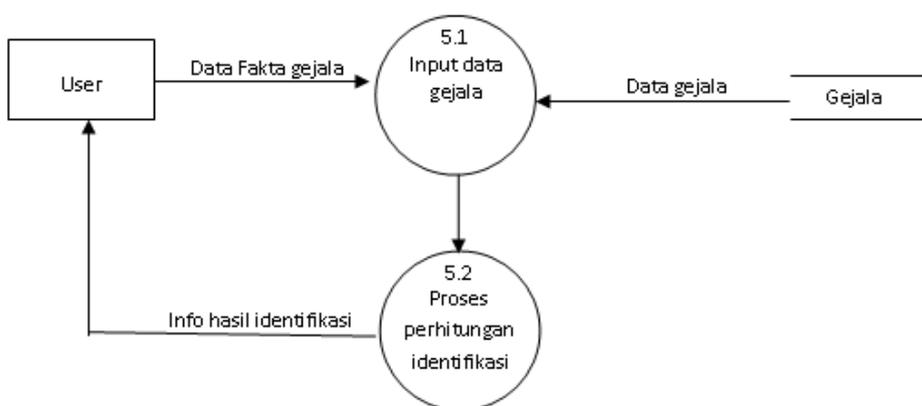
Gambar 4.27 Data Flow Diagram Level 2 Sub Proses Basis Pengetahuan

Sub proses login akan didekomposisi menjadi 1 proses dan 1 simpanan data. Proses terdiri dari proses login, sedangkan simpanan data yaitu akun. Sub proses login ditunjukkan pada Gambar 4.28



Gambar 4.28 Data Flow Diagram Level 2 Sub Proses Login

Sub proses pengolahan identifikasi akan di dekomposisi menjadi 1 proses dan 1 simpanan data. Proses terdiri dari tampil identifikasi. Sedangkan simpanan data yaitu identifikasi. Gambar 4.29 menunjukkan sub proses identifikasi.

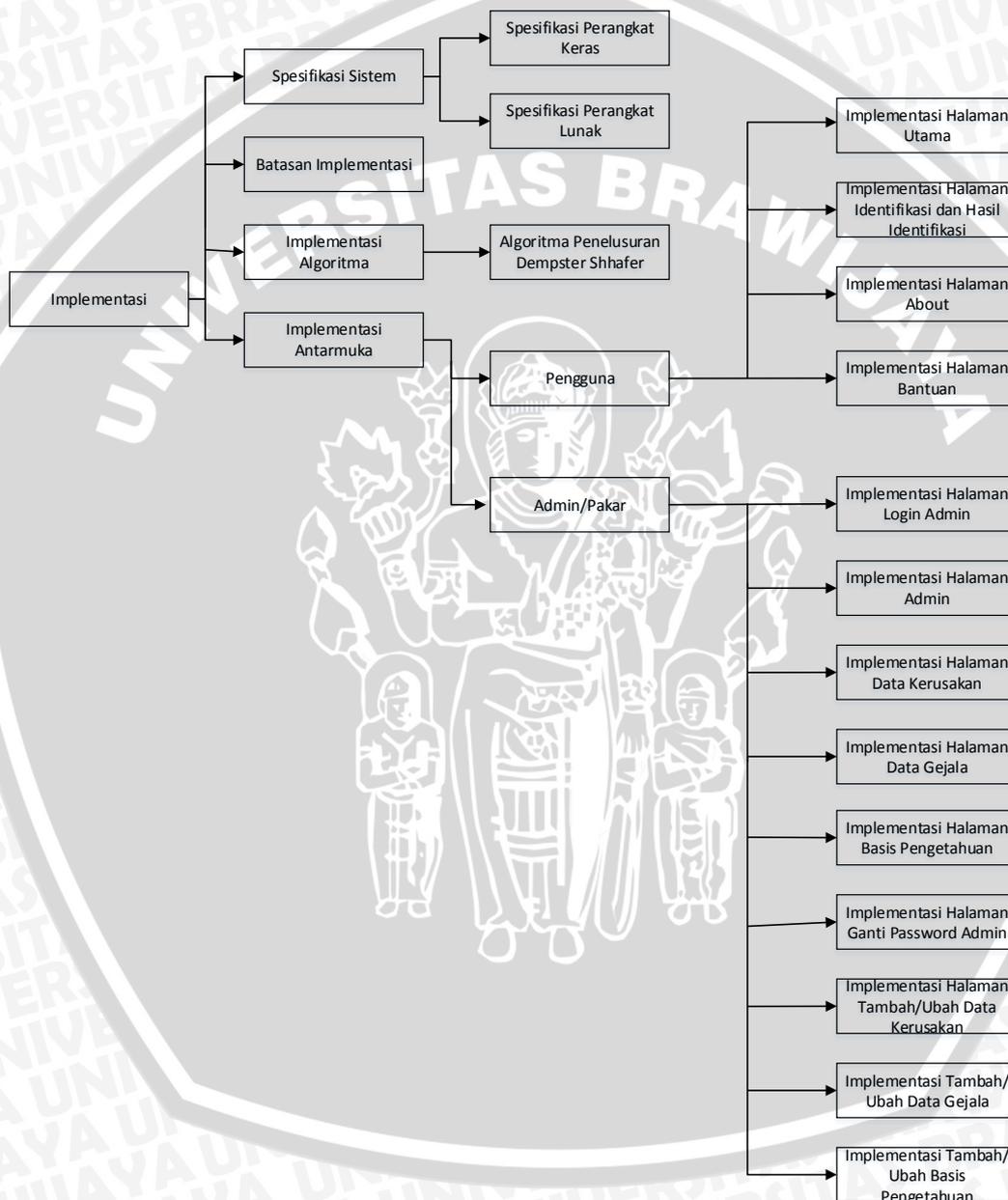


Gambar 4.29 Data Flow Diagram Level 2 Sub Proses Identifikasi



BAB 5 Implementasi

Bab ini membahas mengenai implementasi perangkat lunak berdasarkan hasil yang telah diperoleh dari proses perancangan yang telah dibuat. Pembahasan terdiri dari penjelasan tentang batasan-batasan implementasi, implementasi basis pengetahuan, implementasi mesin inferensi dan implementasi antarmuka. Diagram implementasi identifikasi kerusakan *printer* menggunakan metode *Dempster-Shafer* dapat dilihat pada Gambar 5.1 .



Gambar 5.1 Diagram Implementasi

5.1 Spesifikasi Sistem

Hasil kebutuhan dan perancangan perangkat lunak yang telah dibahas pada bab sebelumnya menjadi acuan untuk melakukan implementasi sebuah sistem yang dapat berfungsi sesuai kebutuhan. Spesifikasi sistem diimplementasi pada spesifikasi perangkat keras dan perangkat lunak.

5.1.1 Spesifikasi Perangkat Keras

Pembuatan Pemodelan Sistem Pakar Untuk Identifikasi Kerusakan *Printer* Dengan Metode *Dempster-shafer* menggunakan sebuah laptop dengan spesifikasi pada Tabel 5.1.

Tabel 5.1 Spesifikasi Perangkat Keras

Nama Komponen	Spesifikasi
Prosesor	Intel(R) Core(TM) i5-2410M CPU @ 2.50GHz ~2.5 GHz
Memori RAM	4.00 GB (2.45 GB usable)
Kartu Grafis	Intel(R) HD Graphics 3000
Hardisk	640 GB HDD

5.1.2 Spesifikasi Perangkat Lunak

Pembuatan Pemodelan Sistem Pakar Untuk Identifikasi Kerusakan *Printer* Dengan Metode *Dempster-shafer* menggunakan sebuah aplikasi perangkat lunak dengan spesifikasi pada Tabel 5.2.

Tabel 5.2 Spesifikasi Perangkat Lunak

Nama Aplikasi	Spesifikasi Perangkat Lunak
Sistem Operasi	Microsoft Windows 7.1 (64-bit)
Bahasa Pemrograman	HTML 5 dan PHP
Tools Pemrograman	Sublime Text 2
<i>Server Localhost</i>	XAMPP
DBMS	MySQL

5.2 Batasan – Batasan Implementasi

Batasan implementasi bertujuan untuk memberikan ruang lingkup yang jelas pada sistem. Beberapa batasan dalam mengimplementasikan Pemodelan Sistem Pakar Identifikasi Kerusakan *Printer* adalah sebagai berikut :

1. Masukan yang diterima oleh sistem adalah berupa inputan gejala-gejala yang dipilih tentang kondisi *printer* yang diinputkan oleh pengguna.
2. Keluaran yang diterima oleh pengguna berupa kerusakan dan nilai kepercayaannya.

3. Aplikasi yang digunakan berbasis web dengan basis data penyimpanan menggunakan MySQL.
4. Metode yang digunakan yaitu *Dempster-Shafer*.
5. Sistem pakar yang dibangun berguna untuk identifikasi kerusakan *printer* tipe *inkjet* dengan merk Canon.
6. Sistem pakar ini dapat mendeteksi 13 jenis kerusakan dengan 30 gejala.
7. Proses manajemen data gejala dan kerusakan hanya dapat dilakukan oleh Pakar dan KE.

5.3 Implementasi Algoritma *Dempster-shafer*

Proses identifikasi ini dengan cara menginputkan fakta-fakta gejala yang terjadi pada *printer* pengguna. Kemudian dilakukan proses identifikasi, implementasi algoritma *Dempster-Shafer* dapat dilihat pada Gambar 5.2.

```

1 function irisan($arr1, $arr2) {
2     $c = -1;
3     unset($arrhsl);
4     $arrhsl = array();
5     for ($i = 0; $i < count($arr1); $i++) {
6         for ($j = 0; $j < count($arr2); $j++) {
7             if ($arr1[$i] == $arr2[$j]) {
8                 $c++;
9                 $arrhsl[$c] = $arr1[$i];
10            }
11        }
12    }
13    return $arrhsl;
14 }
15 function proses($gejala) {
16     session_start();
17     if (count($gejala) >= 0) {
18         if (count($gejala) == 0) {
19             echo "<script>
20                 eval(\"parent.location='hasil.php'\");
21                 alert('Maaf Tidak Dapat di Proses,
22                 Gejala yang Dipilih Harus Lebih Dari Satu');
23             </script>";
24         } else if (count($gejala) == 1) {
25             echo "<script>
26                 eval(\"parent.location='hasil.php'\");
27                 alert('Maaf Tidak Dapat di Proses,
28                 Gejala yang Dipilih Harus Lebih Dari Satu');
29             </script>";
30         } else {
31             $q = mysql_query("select * from gejala ORDER BY
32             id_gejala");
33             $i = 0;
34             while ($r = mysql_fetch_row($q)) {
35                 if ($POST[$r[0]] == true) {
36                     $gejala_terpilih[$i] = $r[0];
37                     $i++;
38                 }
39             }
40             unset($m);
41             $m = array();
42             unset($barishasil);
43             $barishasil = array();
44             for ($k = 1; $k < count($gejala_terpilih); $k++) {
45                 if ($k == 1) {

```

```

46     $gejala = mysql_query("SELECT * FROM gejala
47 WHERE id_gejala = '" . $gejala_terpilih[0] . "'");
48     $datagejala = mysql_fetch_array($gejala);
49     $m[0] = $datagejala['densitas'];
50     $t[0] = 1 - $datagejala['densitas'];
51     $gejala = mysql_query("SELECT * FROM gejala
52 WHERE id_gejala = '" . $gejala_terpilih[$k] . "'");
53     $datagejala = mysql_fetch_array($gejala);
54     $m[$k] = $datagejala['densitas'];
55     $t[$k] = 1 - $datagejala['densitas'];
56     unset($p0);
57     $strp0 = "";
58     $qbasisgejala = mysql_query("SELECT * FROM
59 pengetahuan WHERE id_gejala = '" . $gejala_terpilih[0] . "'");
60     $c = -1;
61     while ($databasisgejala =
62 mysql_fetch_array($qbasisgejala)) {
63         $c++;
64         $p0[$c] = $databasisgejala
65 ['id_kerusakan'];
66         $strp0 = implode(',', $p0);
67     }
68     unset($p1);
69     $strp1 = "";
70     $qbasisgejala = mysql_query("SELECT * FROM
71 pengetahuan WHERE id_gejala = '" . $gejala_terpilih[1] . "'");
72     $c = -1;
73     while ($databasisgejala =
74 mysql_fetch_array($qbasisgejala)) {
75         $c++;
76         $p1[$c] = $databasisgejala
77 ['id_kerusakan'];
78     }
79     $strp1 = implode(',', $p1);
80     unset($pp);
81     $pp = array();
82     $pp = irisan($p0, $p1);
83     $strpp = implode(',', $pp);
84     $barishasil[$k][0] = $strpp;
85     $barisnilai[$k][0] = $m[0] * $m[$k];
86     $barishasil[$k][1] = $strp0;
87     $barisnilai[$k][1] = $m[0] * $t[$k];
88     $barishasil[$k][2] = $strp1;
89     $barisnilai[$k][2] = $m[$k] * $t[0];
90     $baristeta[$k] = $t[0] * $t[$k];
91     $tetapembagi = 0;
92     for ($i = 0; $i < count($barishasil[$k]);
93 $i++) {
94         if ($barishasil[$k][$i] == '') {
95             $tetapembagi += $barisnilai[$k][$i];
96         }
97     }
98     for ($i = 0; $i < count($barishasil[$k]);
99 $i++) {
100         for ($j = $i + 1; $j <
101 count($barishasil[$k]); $j++) {
102             if ($barishasil[$k][$i] == $barishasil[$k][$j]) {
103                 $barisnilai[$k][$i] += $barisnilai[$k][$j];
104                 $barishasil[$k][$j] = '';
105             }
106         }
107     }
108     } else {
109         $gejala = mysql_query("SELECT * FROM gejala
110 WHERE id gejala = '" . $gejala_terpilih[$k] . "'");

```

```

111         $datagejala = mysql_fetch_array($gejala);
112         $m[$k] = $datagejala['densitas'];
113         $t[$k] = 1 - $datagejala['densitas'];
114         unset($p1);
115         $strp1 = "";
116         $qbasisgejala = mysql_query("SELECT * FROM
117 pengetahuan WHERE id_gejala = ' . $gejala_terpilih[$k] . '");
118         $c = -1;
119         while ($databasisgejala =
120 mysql_fetch_array($qbasisgejala)) {
121             $c++;
122             $p1[$c] = $databasisgejala
123 ['id_kerusakan'];
124         }
125         $strp1 = implode(',', $p1);
126         for ($i = 0; $i < count($barishasil[$k - 1]);
127 $i++) {
128             $barisnilai[$k - 1][$i] = $barisnilai[$k -
129 1][$i] / (1 - $tetapembagi);
130         }
131         $baristeta[$k - 1] = $baristeta[$k - 1] / (1 -
132 $tetapembagi);
133         $zz = -1;
134         for ($i = 0; $i < count($barishasil[$k - 1]);
135 $i++) {
136             if ($barishasil[$k - 1][$i] != '') {
137                 unset($pp);
138                 $pp = array();
139                 $arrh = explode(',', $barishasil[$k - 1][$i]);
140                 $pp = irisan($arrh, $p1);
141                 $strpp = implode(',', $pp);
142             }
143             $zz++;
144             $barishasil[$k][$zz] = $strpp;
145             $barisnilai[$k][$zz] = $barisnilai[$k - 1][$i] * $m[$k];
146             $zz++;
147             $barishasil[$k][$zz] = $barishasil[$k - 1][$i];
148             $barisnilai[$k][$zz] = $barisnilai[$k - 1][$i] * $t[$k];
149         }
150     }
151     $zz++;
152     $barishasil[$k][$zz] = $strp1;
153     $barisnilai[$k][$zz] = $m[$k] * $baristeta[$k - 1];
154     $baristeta[$k] = $baristeta[$k - 1] * $t[$k];
155     $tetapembagi = 0;
156     for ($i = 0; $i < count($barishasil[$k]); $i++) {
157         if ($barishasil[$k][$i] == '') {
158             $tetapembagi += $barisnilai[$k][$i];
159         }
160     }
161     for ($i = 0; $i < count($barishasil[$k]); $i++) {
162         for ($j = $i + 1; $j < count($barishasil[$k]); $j++) {
163             if ($barishasil[$k][$i] == $barishasil[$k][$j]) {
164                 $barisnilai[$k][$i] += $barisnilai[$k][$j];
165                 $barishasil[$k][$j] = '';
166             }
167         }
168     }
169 }
170 }
171
172 $strp1 = implode(',', $p1);
173 $arrkerusakanterbesar = array();
174 $strkerusakanterbesar = "";
175 $nilaikerusakanterbesar = -1000000;

```

```

176
177     for ($i = 0; $i < count($barishasil[$k - 1]); $i++) {
178         $barisnilai[$k - 1][$i] = $barisnilai[$k - 1][$i]
179 / (1 - $tetapembagi);
180         if (($nilaikerusakanterbesar < $barisnilai[$k - 1][$i])
181 && ($barishasil[$k - 1][$i] != "")) {
182             $strkerusakanterbesar = $barishasil[$k - 1][$i];
183             $nilaikerusakanterbesar = $barisnilai[$k - 1][$i];
184         }
185     }
186     $baristeta[$k - 1] = $baristeta[$k - 1] / (1 -
187 $tetapembagi);
188     if ($nilaikerusakanterbesar < $baristeta[$k - 1]) {
189         $strkerusakanterbesar = "&oslash;";
190         $nilaikerusakanterbesar = $baristeta[$k - 1];
191     }

```

Gambar 5.2 Implementasi Algoritma Proses Identifikasi

Penjelasan Gambar implementasi algoritma, yaitu

1. Baris 1-14 : membuat irisan antar kode kerusakan.
2. Baris 15-39 : proses untuk inputan 1 gejala.
3. Baris 40-79 : proses mengambil nilai densitas dan *plausibility* dari setiap gejala yang diinput.
4. Baris 80-172 : proses perhitungan tabel kombinasi.
5. Baris 175 : \$nilaikerusakanterbesar = -100000 karena untuk membandingkan \$nilaikerusakanterbesar dengan hasil perhitungan kombinasi yang pasti bernilai 0 – 1. Apabila \$nilaikerusakanterbesar > 0 maka yang akan tampil adalah \$nilaikerusakanterbesar.
6. Baris 173-191 : proses *sorting* pencarian nilai maksimum.

5.4 Implementasi Antarmuka

Antarmuka aplikasi Pemodelan Sistem Pakar Untuk Identifikasi Kerusakan *Printer* ini digunakan oleh pengguna untuk berinteraksi dengan sistem perangkat lunak. Pada implementasi antarmuka perangkat lunak untuk pengguna tidak semua ditampilkan tetapi hanya tertentu saja yaitu halaman utama, halaman identifikasi, halaman hasil identifikasi, halaman about dan halaman bantuan.

Sedangkan bagi admin/pakar bisa menampilkan semua halaman yaitu halaman *login* admin, halaman admin, halaman data kerusakan, halaman gejala, halaman basis pengetahuan, halaman ganti *password* admin, halaman tambah/ubah data kerusakan, halaman tambah/ubah data gejala, dan halaman tambah/ubah data pengetahuan.

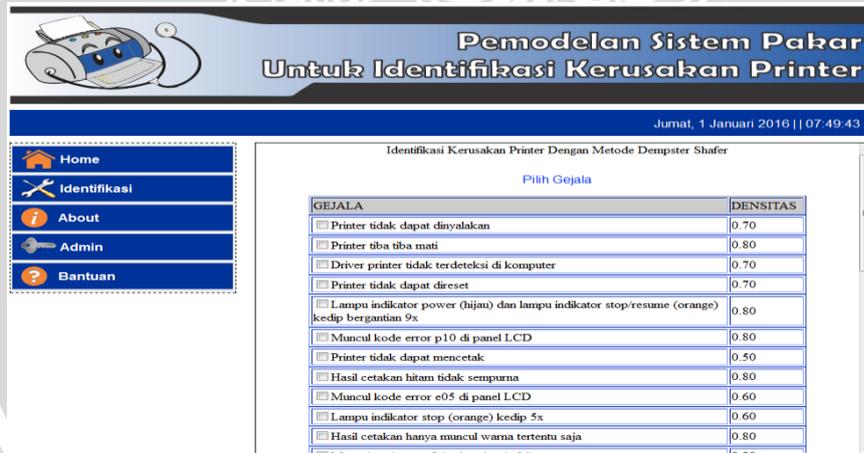
5.4.1 Implementasi Halaman Utama



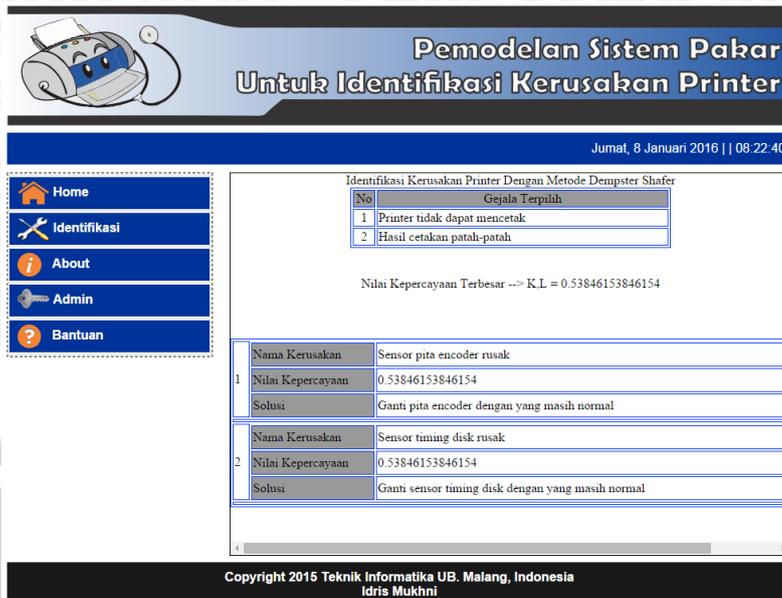
Gambar 5.3 Antarmuka Halaman Utama

Pada Gambar 5.3 merupakan rancangan antarmuka halaman utama yang terdiri dari *header*, *footer*, menu *home*, *identifikasi*, *about*, *admin*, dan *bantuan*. Pengguna bisa langsung melakukan identifikasi dengan menekan tombol identifikasi.

5.4.2 Implementasi Halaman Identifikasi dan Hasil Identifikasi



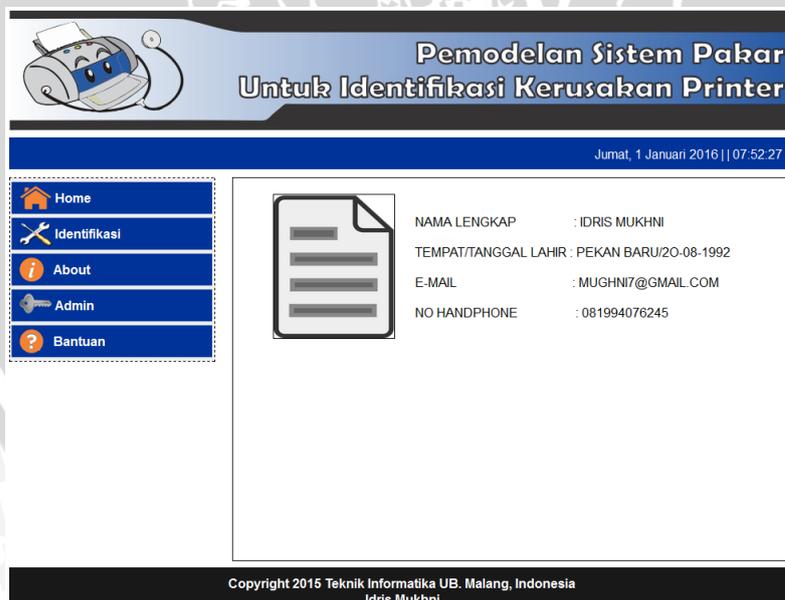
Gambar 5.4 Antarmuka Halaman Identifikasi



Gambar 5.5 Antarmuka Hasil Identifikasi

Pada Gambar 5.4 adalah halaman identifikasi untuk pengguna. Di halaman tersebut dapat dilihat gejala-gejala yang dapat dipilih oleh pengguna sesuai dengan fakta yang dialami oleh pengguna. Kemudian di Gambar 5.5 adalah halaman hasil identifikasi, pengguna bisa melihat hasil dari identifikasi sesuai dengan gejala yang dimasukkan.

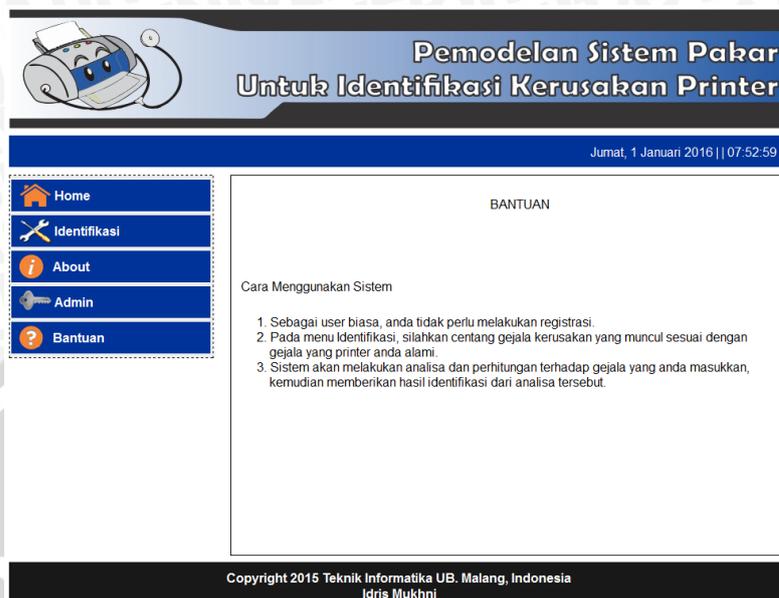
5.4.3 Implementasi Halaman About



Gambar 5.6 Antarmuka Halaman about

Pada Gambar 5.6 adalah halaman yang menerangkan tentang biodata pembuat aplikasi identifikasi kerusakan menggunakan metode *Dempster-Shafer*.

5.4.4 Implementasi Halaman Bantuan



Gambar 5.7 Antarmuka Halaman Bantuan

Pada Gambar 5.7 adalah antarmuka halaman bantuan. Halaman ini berisi informasi cara penggunaan sistem.

5.4.5 Implementasi Halaman Login



Gambar 5.8 Antarmuka Halaman Login Admin

Pada Gambar 5.8 adalah antarmuka halaman *login* admin. Admin memasukkan *username* dan *password*, kemudian sistem akan mencocokkan dengan *database*.

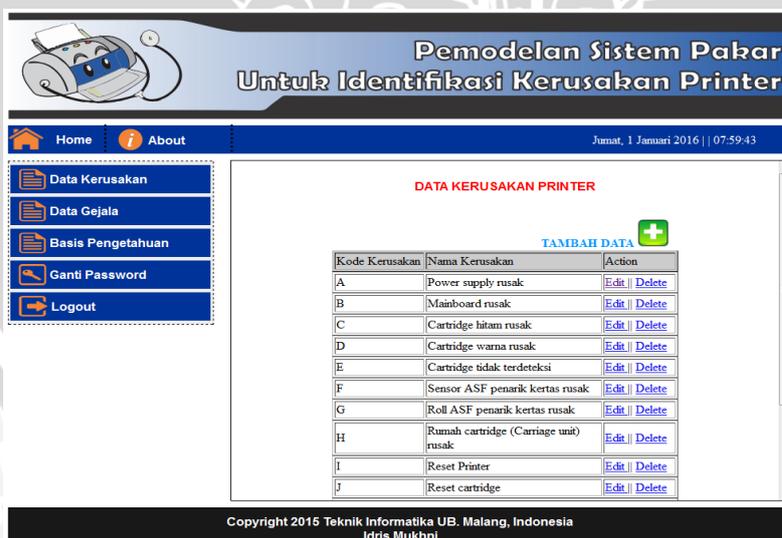
5.4.6 Implementasi Halaman Admin



Gambar 5.9 Antarmuka Halaman Admin

Pada Gambar 5.9 adalah antarmuka halaman Admin. Pada halaman ini terdapat menu data kerusakan, data gejala, basis pengetahuan, ganti *password* dan *logout*.

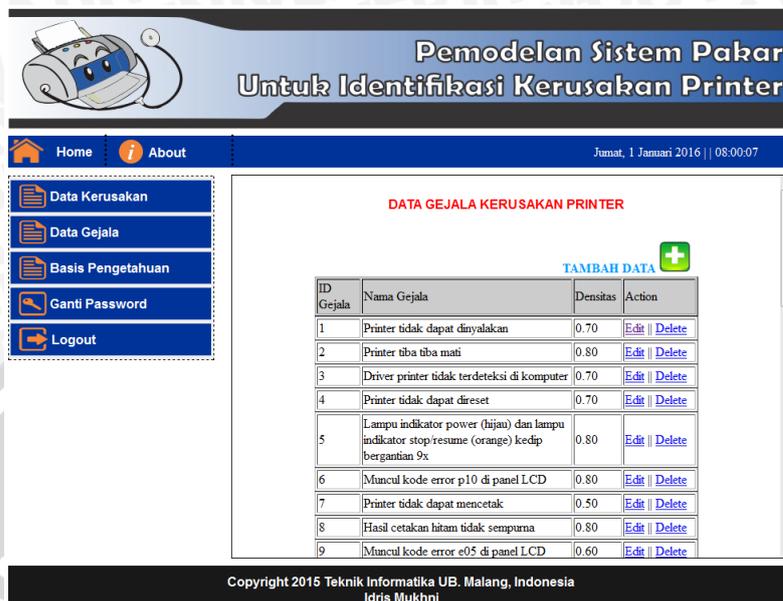
5.4.7 Implementasi Halaman Data Kerusakan



Gambar 5.10 Antarmuka Halaman Data Kerusakan

Pada Gambar 5.10 adalah antarmuka halaman data kerusakan. Pada halaman ini terdapat kode kerusakan, nama kerusakan dan *action* yang dapat dilakukan oleh admin yaitu *edit* dan *delete* kerusakan.

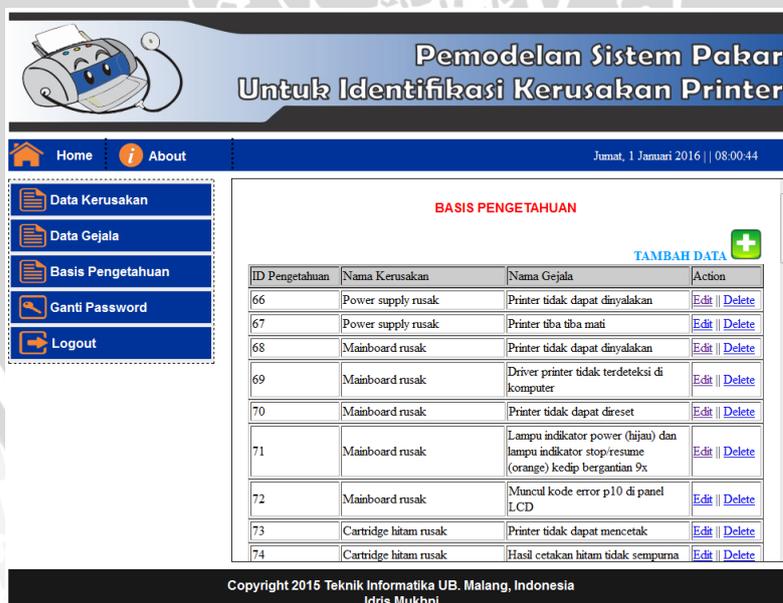
5.4.7 Implementasi Halaman Data Gejala



Gambar 5.11 Antarmuka Halaman Data Kerusakan

Pada Gambar 5.11 adalah antarmuka halaman data kerusakan. Di halaman ini admin menambah, mengedit dan menghapus data gejala.

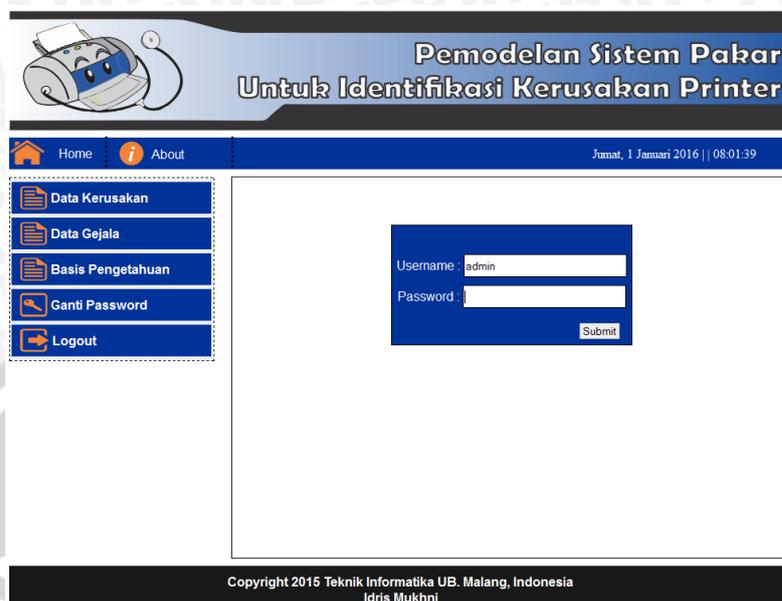
5.4.8 Implementasi Halaman Basis Pengetahuan



Gambar 5.12 Antarmuka Halaman Basis Pengetahuan

Pada Gambar 5.12 adalah halaman basis pengetahuan, dimana dalam halaman tersebut admin dapat melihat data-data gejala yang sudah dimasukkan sebelumnya, menambah, mengedit dan menghapus data.

5.4.9 Implementasi Halaman Ganti Password Admin



Gambar 5.13 Antarmuka Halaman Ganti *Password* Admin

Pada Gambar 5.13 adalah halaman untuk admin. Dimana fungsi dari halaman ini adalah pergantian *password* baru hanya untuk akun admin.

5.4.10 Implementasi Halaman Tambah/Ubah Data Kerusakan



Gambar 5.14 Antarmuka Halaman Tambah/Ubah Data Kerusakan

Pada Gambar 5.14 adalah halaman yang muncul apabila admin menambah atau mengubah data kerusakan yang ada pada menu data kerusakan.

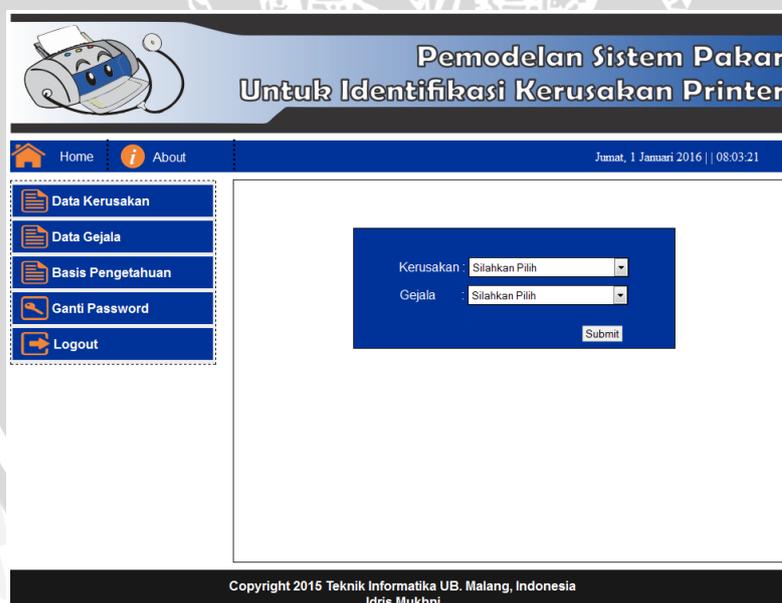
5.4.11 Implementasi Halaman Tambah/Ubah Data Gejala



Gambar 5.15 Antarmuka Halaman Tambah/Ubah Data Gejala

Pada Gambar 5.15 adalah halaman yang muncul apabila admin menambah atau mengubah data gejala yang ada pada menu data gejala.

5.4.12 Implementasi Halaman Tambah/Ubah Basis Pengetahuan

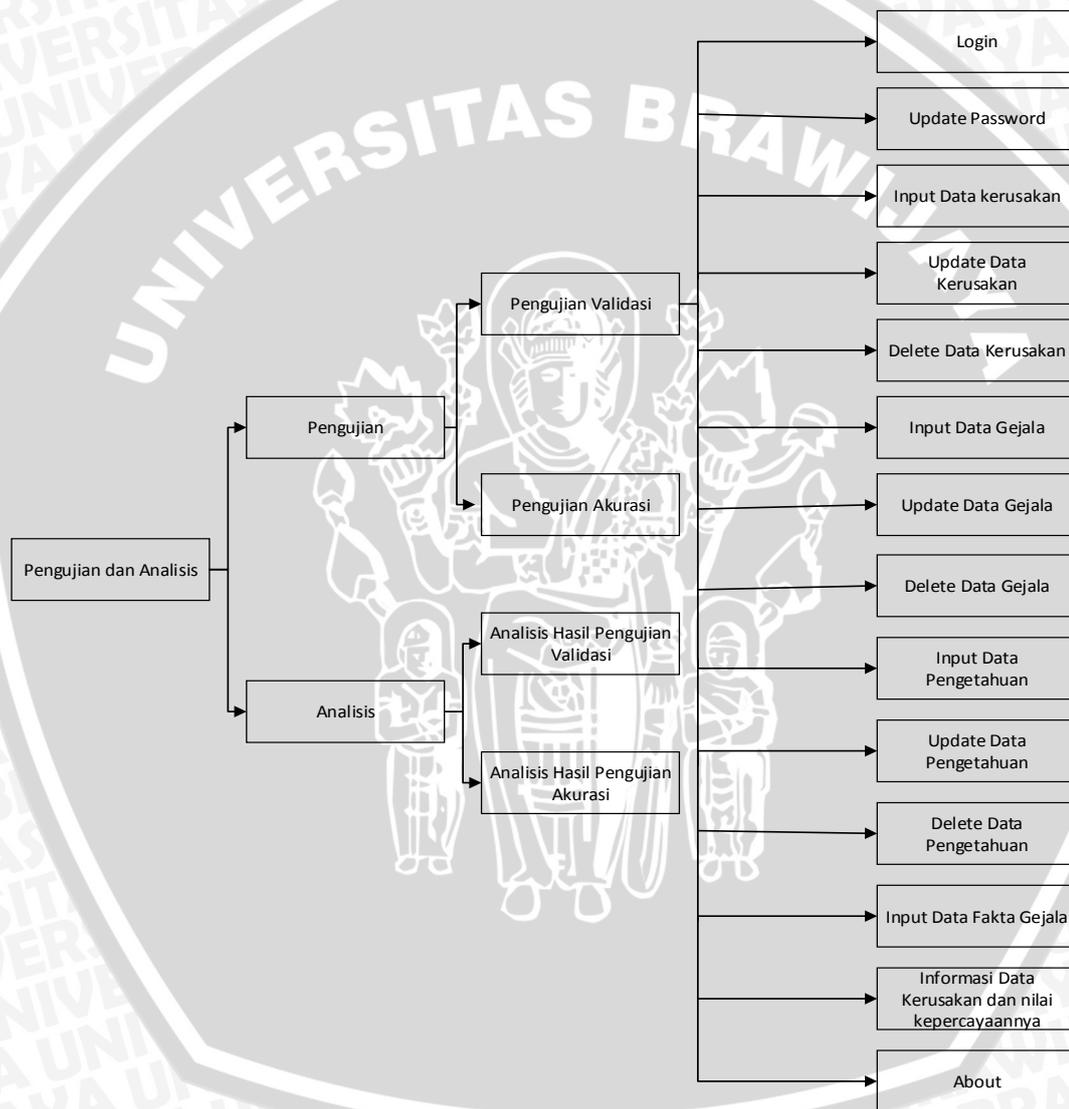


Gambar 5.16 Antarmuka Halaman Tambah/Ubah Data Kerusakan

Pada Gambar 5.16 adalah halaman yang muncul apabila admin menambah atau mengubah data basis pengetahuan yang ada pada menu basis pengetahuan.

BAB VI PENGUJIAN DAN ANALISIS

Pada bab ini membahas mengenai pengujian dan analisis sistem terhadap aplikasi identifikasi kerusakan dengan metode *Dempster-Shafer*. Proses pengujian dilakukan dengan 2 cara, yaitu pengujian validasi system dan pengujian akurasi. Pada pengujian validasi sistem akan digunakan teknik pengujian *black box testing*. Pengujian akurasi digunakan untuk menguji tingkat akurasi antar perhitungan kasus dari pakar dengan perhitungan kasus yang telah diimplementasikan menjadi sistem. Analisa hasil pengujian dilakukan untuk menganalisa hasil pengujian yang telah dilakukan. Diagram pengujian dan analisis dapat dilihat pada Gambar 6.1.



Gambar 6.1 Diagram Pengujian dan Analisis

6.1 Pengujian

Proses pengujian dilakukan dengan pengujian validasi dan pengujian tingkat akurasi. Pengujian validasi dilakukan untuk mengetahui fungsional sistem yang dibangun sudah sesuai dengan daftar kebutuhan yang ada. Pengujian akurasi dilakukan untuk mengetahui seberapa besar akurasi yang didapat dari penggunaan metode *Dempster-Shafer* dalam menyelesaikan masalah aplikasi identifikasi kerusakan .

6.1.1 Pengujian Validasi

Pengujian validasi digunakan untuk mengetahui apakah sistem yang dibangun sudah sesuai dengan yang dibutuhkan. Item-item yang telah dirumuskan dalam daftar kebutuhan akan menjadi acuan untuk melakukan pengujian validasi. Pengujian validasi menggunakan metode pengujian *black box*, Karena tidak difokuskan terhadap alurnya jalan algoritma program namun lebih ditekankan untuk menemukan kesesuaian antara kinerja sistem dengan daftar kebutuhan. Pada setiap kebutuhan dilakukan proses pengujian dengan kasus uji masing-masing untuk mengetahui kesesuaian antar kebutuhan dengan kinerja sistem.

6.1.1.1 Kasus Uji Login

Pada kasus uji *login* akan dilakukan pengujian untuk validasi *login* yang sah dan validasi *login* yang tidak sah dimana akan ditunjukkan pada Tabel 6.1 dan Tabel 6.2.

Tabel 6.1 Kasus Uji untuk Pengujian Validasi Login Sah

Nama Kasus Uji	Login Sah
Tujuan pengujian	Untuk menguji validasi terhadap kinerja sistem dalam menyediakan fasilitas <i>login</i> bagi admin untuk bisa masuk ke halaman admin.
Prosedur uji	<ol style="list-style-type: none">1. Sistem akan berjalan ketika program mulai dieksekusi2. Akan muncul halaman <i>login</i>3. Admin akan mengisi username dan password pada form <i>login</i>4. Admin menekan tombol <i>login</i> untuk masuk ke halaman admin
Hasil yang diharapkan	<ol style="list-style-type: none">1. Sistem akan melakukan pemeriksaan terhadap <i>username</i> dan <i>password</i> yang dimasukkan dengan data yang ada dalam <i>database</i>, jika benar admin akan masuk ke halaman admin
Hasil yang diharapkan	<ol style="list-style-type: none">2. Sistem akan melakukan pemeriksaan terhadap form <i>login</i> apakah sudah terisi semua atau belum, jika sudah terisi semua admin akan masuk ke halaman admin

Tabel 6.2 Kasus Uji untuk Pengujian Validasi Login Tidak Sah

Nama Kasus Uji	Login Tidak Sah
Tujuan pengujian	Untuk menguji validasi terhadap kinerja sistem dalam menyediakan fasilitas <i>login</i> bagi admin untuk bisa masuk ke halaman admin
Prosedur uji	<ol style="list-style-type: none"> 1. Sistem akan berjalan ketika program mulai dieksekusi 2. Akan muncul halaman <i>login</i> 3. Admin akan mengisi <i>username</i> dan <i>password</i> pada form <i>login</i> 4. Admin menekan tombol <i>login</i> untuk masuk ke halaman admin
Hasil yang diharapkan	<ol style="list-style-type: none"> 1. Sistem akan melakukan pemeriksaan terhadap <i>username</i> dan <i>password</i> yang dimasukkan dengan data yang ada dalam <i>database</i>, jika salah admin tidak dapat masuk ke halaman admin dan sistem akan memberikan pesan peringatan 2. Sistem akan melakukan pemeriksaan terhadap form <i>login</i> apakah sudah terisi semua atau belum, jika belum terisi semua sistem akan memberikan pesan peringatan

6.1.1.2 Kasus Uji Data Kerusakan

Pada kasus uji pengolahan data kerusakan akan dilakukan pengujian untuk validasi tambah data kerusakan yang sah, validasi tambah data kerusakan tidak sah, validasi edit dan validasi hapus dimana akan ditunjukkan pada Tabel 6.3, Tabel 6.4, Tabel 6.5 dan Tabel 6.6.

Tabel 6.3 Kasus Uji untuk Pengujian Validasi Tambah Data Kerusakan Sah

Nama Kasus Uji	Tambah Data Kerusakan
Tujuan pengujian	Untuk menguji validasi terhadap kinerja sistem dalam menyediakan fasilitas tambah data kerusakan yang akan dilakukan oleh admin

Nama Kasus Uji	Tambah Data Kerusakan
Prosedur uji	<ol style="list-style-type: none"> 1. Admin masuk ke halaman admin 2. Admin memilih menu input data kerusakan 3. Admin mengisi data pada form data kerusakan 4. Admin menekan tombol simpan untuk menambahkan data kerusakan
Hasil yang diharapkan	<ol style="list-style-type: none"> 1. Sistem akan melakukan pemeriksaan terhadap data pada form data kerusakan apakah sudah terisi semua 2. Sistem akan melakukan pemeriksaan terhadap id_kerusakan yang diinputkan sudah tersedia atau belum di database, jika id_kerusakan belum terdapat dalam database maka data kerusakan bisa ditambahkan dan akan ditampilkan pada tabel data kerusakan

Tabel 6.4 Kasus Uji untuk Pengujian Validasi Tambah Data Kerusakan Tidak Sah

Nama Kasus Uji	Tambah Data Kerusakan Tidak Sah
Tujuan pengujian	Untuk menguji validasi terhadap kinerja sistem dalam menyediakan fasilitas tambah data kerusakan yang akan dilakukan oleh admin
Prosedur uji	<ol style="list-style-type: none"> 1. Admin masuk ke halaman admin 2. Admin memilih menu <i>input</i> data kerusakan 3. Admin mengisi data pada form data kerusakan 4. Admin menekan tombol simpan untuk menambahkan data kerusakan
Hasil yang diharapkan	<ol style="list-style-type: none"> 1. Sistem akan melakukan pemeriksaan terhadap data pada form data kerusakan apakah sudah terisi semua atau belum 2. Sistem akan melakukan pemeriksaan terhadap id_kerusakan yang diinputkan sudah tersedia atau belum dalam database, jika id_kerusakan sudah terdapat dalam <i>database</i> maka data kerusakan tidak bisa ditambahkan dan akan muncul pesan peringatan

Tabel 6.5 Kasus Uji untuk Pengujian Validasi Edit Data Kerusakan

Nama Kasus Uji	Edit Data Kerusakan
Tujuan pengujian	Untuk menguji validasi terhadap kinerja sistem dalam menyediakan fasilitas edit data kerusakan oleh admin
Prosedur uji	<ol style="list-style-type: none"> 1. Admin masuk ke halaman admin 2. Admin memilih menu data kerusakan 3. Sistem akan menampilkan daftar data kerusakan 4. Admin menekan <i>icon</i> edit pada baris yang diinginkan 5. Sistem akan menampilkan form edit sesuai dengan data yang dipilih 6. Admin melakukan perubahan pada data dan menekan tombol simpan
Hasil yang diharapkan	Sistem akan melakukan perubahan data dan menyimpan ke dalam <i>database</i> serta menampilkan kembali pada tabel data kerusakan

Tabel 6.6 Kasus Uji untuk Pengujian Validasi Hapus Data Kerusakan

Nama Kasus Uji	Hapus Data Kerusakan
Tujuan pengujian	Untuk menguji validasi terhadap kinerja sistem dalam menyediakan fasilitas hapus data kerusakan yang dilakukan oleh admin
Prosedur uji	<ol style="list-style-type: none"> 1. Admin masuk ke halaman admin 2. Admin memilih menu data kerusakan 3. Sistem akan menampilkan daftar data kerusakan 4. Admin menekan <i>icon</i> hapus pada baris yang diinginkan
Hasil yang diharapkan	Sistem akan melakukan hapus data kerusakan dari database dan secara otomatis data juga akan terhapus dalam tabel data kerusakan

2.3.4.1 Kasus Uji Pengolahan Data Gejala

Pada kasus uji pengolahan data gejala akan dilakukan pengujian untuk validasi tambah data gejala yang sah, tambah data gejala yang tidak sah, validasi edit dan validasi hapus dimana akan ditunjukkan pada Tabel 6.7, Tabel 6.8, Tabel 6.9 dan Tabel 6.10.

Tabel 6.7 Kasus Uji untuk Pengujian Validasi Tambah Data Gejala Sah

Nama Kasus Uji	Tambah Data Gejala Sah
Tujuan pengujian	Untuk menguji validasi kinerja sistem dalam menyediakan fasilitas tambah data gejala yang akan dilakukan oleh admin
Prosedur uji	<ol style="list-style-type: none"> 1. Admin masuk ke menu utama 2. Admin memilih menu <i>input</i> data gejala 3. Admin mengisi data pada form data gejala 4. Admin menekan tombol simpan untuk menambahkan data gejala
Hasil yang diharapkan	<ol style="list-style-type: none"> 1. Sistem akan melakukan pemeriksaan terhadap data pada form data gejala apakah sudah terisi semua 2. Sistem akan melakukan pemeriksaan terhadap <i>id_gejala</i> yang diinputkan sudah tersedia atau belum dalam database, jika <i>id_gejala</i> belum terdapat dalam database maka data gejala bisa ditambahkan dan akan ditampilkan pada tabel data gejala

Tabel 6.8 Kasus Uji untuk Pengujian Validasi Tambah Data Gejala Tidak Sah

Nama Kasus Uji	Tambah Data Gejala Tidak Sah
Tujuan Pengujian	Untuk menguji validasi kinerja sistem dalam menyediakan fasilitas tambah data gejala yang akan dilakukan oleh admin
Prosedur uji	<ol style="list-style-type: none"> 1. Admin masuk ke menu utama 2. Admin memilih menu input data gejala 3. Admin mengisi data pada form data gejala 4. Admin menekan tombol simpan untuk menambahkan data gejala
Hasil yang diharapkan	<ol style="list-style-type: none"> 1. Sistem akan melakukan pemeriksaan terhadap data pada form data gejala apakah sudah terisi semua

Nama Kasus Uji	Tambah Data Gejala Tidak Sah
Hasil yang diharapkan	2. Sistem akan melakukan pemeriksaan terhadap id_gejala yang diinputkan sudah terisi atau belum dalam database, jika sudah terdapat dalam database maka data gejala tidak bisa ditambahkan dan akan muncul pesan peringatan

Tabel 6.9 Kasus Uji untuk Pengujian Validasi Edit Data Gejala

Nama Kasus Uji	Edit Data Gejala
Tujuan Pengujian	Untuk menguji validasi terhadap kinerja sistem dalam menyediakan fasilitas edit data gejala oleh admin
Prosedur uji	<ol style="list-style-type: none"> 1. Admin masuk ke halaman admin 2. Admin memilih menu data gejala 3. Sistem akan menampilkan daftar data gejala 4. Admin menekan icon edit pada baris yang diinginkan 5. Sistem akan menampilkan form edit sesuai dengan data yang dipilih 6. Admin melakukan perubahan pada data dan menekan tombol simpan
Hasil yang diharapkan	Sistem akan melakukan perubahan data dan menyimpan ke dalam database serta menampilkan kembali pada tabel data gejala

Tabel 6.10 Kasus Uji untuk Pengujian Validasi Hapus Data Gejala

Nama Kasus Uji	Hapus Data Gejala
Tujuan pengujian	Untuk menguji validasi terhadap kinerja sistem dalam menyediakan fasilitas hapus data gejala yang dilakukan oleh admin
Prosedur uji	<ol style="list-style-type: none"> 1. Admin masuk ke halaman admin 2. Admin memilih menu data gejala 3. Sistem akan menampilkan daftar data gejala 4. Admin menekan icon hapus pada baris yang diinginkan

Nama Kasus Uji	Hapus Data Gejala
Hasil yang diharapkan	Sistem akan melakukan hapus data gejala dari <i>database</i> dan secara otomatis data juga akan terhapus dalam tabel data gejala

2.3.4.2 Kasus Uji Pengolahan Basis Pengetahuan

Pada kasus uji pengolahan data basis pengetahuan akan dilakukan pengujian untuk validasi tambah data basis pengetahuan yang sah, tambah data basis pengetahuan yang tidak sah, validasi edit dan validasi hapus dimana akan ditunjukkan pada Tabel 6.11, Tabel 6.12, Tabel 6.13 dan Tabel 6.14.

Tabel 6.11 Kasus Uji untuk Pengujian Validasi Tambah Data Basis Pengetahuan Sah

Nama Kasus Uji	Tambah Data Basis Pengetahuan Sah
Tujuan pengujian	Untuk menguji validasi kinerja sistem dalam menyediakan fasilitas tambah data basis pengetahuan yang akan dilakukan oleh admin
Prosedur uji	<ol style="list-style-type: none"> 1. Admin masuk ke menu utama 2. Admin memilih menu input data basis pengetahuan 3. Admin mengisi data pada form data basis pengetahuan 4. Admin menekan tombol simpan untuk menambahkan data basis pengetahuan
Hasil yang diharapkan	Sistem akan melakukan pemeriksaan terhadap data pada form data gejala apakah sudah terisi semua, jika benar maka otomatis akan ditambahkan dalam <i>database</i> tabel daftar basis pengetahuan

Tabel 6.12 Kasus Uji untuk Pengujian Validasi Tambah Data Basis Pengetahuan Tidak Sah

Nama Kasus Uji	Tambah Data Basis Pengetahuan Tidak Sah
Tujuan Pengujian	Untuk menguji validasi kinerja sistem dalam menyediakan fasilitas tambah data basis pengetahuan yang akan dilakukan oleh admin
Prosedur uji	<ol style="list-style-type: none"> 1. Admin masuk ke menu utama 2. Admin memilih menu <i>input</i> data basis pengetahuan

Nama Kasus Uji	Tambah Data Basis Pengetahuan Tidak Sah
Prosedur uji	<ol style="list-style-type: none"> 3. Admin mengisi data pada form data basis pengetahuan 4. Admin menekan tombol simpan untuk menambahkan data basis pengetahuan
Hasil yang diharapkan	Sistem akan melakukan pemeriksaan terhadap data pada form data basis pengetahuan apakah sudah terisi semua, jika belum maka data tidak dapat ditambahkan ke dalam <i>database</i> daftar basis pengetahuan dan akan mengeluarkan pesan peringatan

Tabel 6.13 Kasus Uji untuk Pengujian Validasi Edit Basis Pengetahuan

Nama Kasus Uji	Edit Data Basis Pengetahuan
Tujuan Pengujian	Untuk menguji validasi terhadap kinerja sistem dalam menyediakan fasilitas edit data basis pengetahuan oleh admin
Prosedur uji	<ol style="list-style-type: none"> 1. Admin masuk ke halaman admin 2. Admin memilih menu data basis pengetahuan 3. Sistem akan menampilkan daftar data basis pengetahuan 4. Admin menekan <i>icon</i> edit pada baris yang diinginkan 5. Sistem akan menampilkan form edit sesuai dengan data yang dipilih 6. Admin melakukan perubahan pada data dan menekan tombol simpan
Hasil yang diharapkan	Sistem akan melakukan perubahan data dan menyimpan ke dalam <i>database</i> serta menampilkan kembali pada tabel data basis pengetahuan

Tabel 6.14 Kasus Uji untuk Pengujian Validasi Hapus Data Basis Pengetahuan

Nama Kasus Uji	Hapus Data Basis Pengetahuan
Tujuan pengujian	Untuk menguji validasi terhadap kinerja sistem dalam menyediakan fasilitas hapus data basis pengetahuan yang dilakukan oleh admin
Prosedur uji	<ol style="list-style-type: none"> 1. Admin masuk ke halaman admin 2. Admin memilih menu data basis pengetahuan 3. Sistem akan menampilkan daftar data basis pengetahuan 4. Admin menekan icon hapus pada baris yang diinginkan
Hasil yang diharapkan	Sistem akan melakukan hapus data basis pengetahuan dari <i>database</i> dan secara otomatis data juga akan terhapus dalam tabel data basis pengetahuan

6.1.1.3 Kasus Uji Ganti Password

Pada kasus uji ganti password akan dilakukan pengujian untuk validasi dalam perubahan data password admin dimana akan ditunjukkan pada Tabel 6.15.

Tabel 6.15 Kasus Uji untuk Pengujian Ganti *Password*

Nama Kasus Uji	Ganti <i>Password</i>
Tujuan Pengujian	Untuk menguji validasi terhadap kinerja sistem dalam menyediakan fasilitas ganti <i>password</i> admin
Prosedur uji	<ol style="list-style-type: none"> 1. Admin masuk ke halaman admin 2. Admin memilih menu ganti <i>password</i> 3. Sistem akan menampilkan form ganti <i>password</i> 4. Admin menekan tombol simpan
Hasil yang diharapkan	Sistem akan melakukan perubahan data dan menyimpan ke dalam <i>database</i>

6.1.1.4 Kasus Uji Proses *Logout*

Pada kasus uji ganti password akan dilakukan pengujian untuk validasi dalam proses keluar dari sistem dimana akan ditunjukkan pada Tabel 6.16.

Tabel 6.16 Kasus Uji untuk Proses Logout

Nama Kasus Uji	Proses Logout
Tujuan Pengujian	Untuk menguji validasi terhadap kinerja sistem dalam menyediakan fasilitas <i>logout</i> bagi admin
Prosedur uji	1. Admin masuk ke halaman admin 2. Admin menekan tombol <i>logout</i>
Hasil yang diharapkan	Sistem akan keluar dari halaman admin dan kembali ke halaman utama

6.1.1.5 Kasus Uji Profil Admin

Pada kasus uji profil admin akan dilakukan pengujian untuk validasi sistem dalam menampilkan data profil admin dimana akan ditunjukkan pada Tabel 6.17.

Tabel 6.17 Kasus Uji untuk Menampilkan Profil Admin

Nama Kasus Uji	Menampilkan Profil Admin
Tujuan pengujian	Untuk menguji validasi terhadap kinerja sistem dalam menampilkan profil admin
Prosedur uji	Pengguna menekan tombol about
Hasil yang diharapkan	Sistem akan menampilkan profil admin

6.1.1.6 Kasus Uji untuk Input Data Fakta Gejala**Tabel 6.18 Kasus Uji untuk Input Data Fakta Gejala**

Nama Kasus Uji	Input Data Fakta Gejala
Tujuan Pengujian	Untuk menguji validasi terhadap kinerja sistem dalam menyediakan fasilitas input data fakta gejala yang dilakukan oleh pengguna
Prosedur uji	1. Pengguna memilih menu halaman Identifikasi 2. Pengguna memilih data fakta gejala yang ada pada tabel gejala dengan cara memberi tanda checklist 3. Pengguna menekan tombol <i>submit</i>
Hasil yang diharapkan	1. Sistem akan menampilkan data gejala dan nilai densitasnya 2. Sistem dapat menyimpan data fakta gejala sesuai yang telah diinputkan sebelumnya oleh pengguna

Tabel 6.19 Kasus Uji untuk Informasi Hasil Identifikasi

Nama Kasus Uji	Informasi Hasil Identifikasi
Tujuan Pengujian	Untuk menguji validasi terhadap kinerja sistem dalam menampilkan informasi hasil Identifikasi
Prosedur uji	1. Pengguna memilih data fakta gejala yang ada pada tabel gejala dengan cara memberi tanda <i>checklist</i> 2. Pengguna menekan tombol <i>submit</i>
Hasil yang diharapkan	Sistem dapat menampilkan informasi data hasil Identifikasi berupa nama kerusakan dan nilai kepercayaannya

Berdasarkan kasus uji yang telah dilaksanakan sesuai dengan prosedur pengujian, maka didapatkan hasil pengujian seperti yang ditunjukkan pada Tabel 6.20.

Tabel 6.20 Hasil Pengujian Validasi

No	Nama Kasus	Hasil yang diharapkan	Hasil yang didapatkan	Status Validasi
1	Login Sah	1. Sistem akan melakukan pemeriksaan terhadap username dan password yang dimasukkan dengan data yang ada dalam database, jika benar admin akan masuk ke halaman admin 2. Sistem akan melakukan pemeriksaan terhadap form login apakah sudah terisi semua atau belum, jika sudah terisi semua admin akan masuk ke halaman admin	1. Sistem akan melakukan pemeriksaan terhadap username dan password yang dimasukkan dengan data yang ada dalam database, jika benar admin akan masuk ke halaman admin 2. Sistem akan melakukan pemeriksaan terhadap form login apakah sudah terisi semua atau belum, jika sudah terisi semua admin akan masuk ke halaman admin	Valid

No	Nama Kasus	Hasil yang diharapkan	Hasil yang didapatkan	Status Validasi
2	Login Tidak Sah	<p>1. Sistem akan melakukan pemeriksaan terhadap <i>username</i> dan <i>password</i> yang dimasukkan dengan data yang ada dalam <i>database</i>, jika salah admin tidak dapat masuk ke halaman admin dan sistem akan memberikan pesan peringatan</p> <p>2. Sistem akan melakukan pemeriksaan terhadap form <i>login</i> apakah sudah terisi semua atau belum, jika belum terisi semua sistem akan memberikan pesan peringatan</p>	<p>1.Sistem akan melakukan pemeriksaan terhadap <i>username</i> dan <i>password</i> yang dimasukkan dengan data yang ada dalam <i>database</i>, jika salah admin tidak dapat masuk ke halaman admin dan sistem akan memberikan pesan peringatan</p> <p>2. Sistem akan melakukan pemeriksaan terhadap form <i>login</i> apakah sudah terisi semua atau belum, jika belum terisi semua sistem akan memberikan pesan peringatan</p>	Valid
3	Tambah Data Kerusakan yang Sah	<p>1. Sistem akan melakukan pemeriksaan terhadap data pada form data kerusakan apakah sudah terisi semua</p> <p>2. Sistem akan melakukan pemeriksaan terhadap <i>id_kerusakan</i> yang diinputkan sudah tersedia atau belum di <i>database</i>, jika <i>id_kerusakan</i> belum terdapat</p>	<p>1. Sistem akan melakukan pemeriksaan terhadap data pada form data kerusakan apakah sudah terisi semua</p> <p>2. Sistem akan melakukan pemeriksaan terhadap <i>id_kerusakan</i> yang diinputkan sudah tersedia atau belum di <i>database</i>, jika <i>id_kerusakan</i> belum terdapat dalam</p>	Valid

No	Nama Kasus	Hasil yang diharapkan	Hasil yang didapatkan	Status Validasi
		dalam database maka data kerusakan bisa ditambahkan dan akan ditampilkan pada tabel data kerusakan	database maka data kerusakan bisa ditambahkan dan akan ditampilkan pada tabel data kerusakan	
4	Tambah Data Kerusakan Tidak Sah	<ol style="list-style-type: none"> 1. Sistem akan melakukan pemeriksaan terhadap data pada form data kerusakan apakah sudah terisi semua atau belum 2. Sistem akan melakukan pemeriksaan terhadap id_kerusakan yang diinputkan sudah tersedia atau belum dalam <i>database</i>, jika id_kerusakan sudah terdapat dalam database maka data kerusakan tidak bisa ditambahkan dan akan muncul pesan peringatan 	<ol style="list-style-type: none"> 1. Sistem akan melakukan pemeriksaan terhadap data pada form data kerusakan apakah sudah terisi semua atau belum 2. Sistem akan melakukan pemeriksaan terhadap id_kerusakan yang diinputkan sudah tersedia atau belum dalam <i>database</i>, jika id_kerusakan sudah terdapat dalam database maka data kerusakan tidak bisa ditambahkan dan akan muncul pesan peringatan 	Valid
5	Edit Data Kerusakan	Sistem akan melakukan perubahan data dan menyimpan ke dalam database serta menampilkan kembali pada tabel data kerusakan	Sistem akan melakukan perubahan data dan menyimpan ke dalam <i>database</i> serta menampilkan kembali pada tabel data kerusakan	Valid
6	Hapus Data Kerusakan	Sistem akan melakukan hapus data	Sistem akan melakukan hapus data kerusakan	Valid

No	Nama Kasus	Hasil yang diharapkan	Hasil yang didapatkan	Status Validasi
		kerusakan dari <i>database</i> dan secara otomatis data juga akan terhapus dalam tabel data kerusakan	dari <i>database</i> dan secara otomatis data juga akan terhapus dalam tabel data kerusakan	
7	Tambah Data Gejala Sah	<p>1. Sistem akan melakukan pemeriksaan terhadap data pada form data gejala apakah sudah terisi semua</p> <p>2. Sistem akan melakukan pemeriksaan terhadap <i>id_gejala</i> yang diinputkan sudah tersedia atau belum dalam <i>database</i>, jika <i>id_gejala</i> belum terdapat dalam <i>database</i> maka data gejala bisa ditambahkan dan akan ditampilkan pada tabel data gejala</p>	<p>1. Sistem akan melakukan pemeriksaan terhadap data pada form data gejala apakah sudah terisi semua</p> <p>2. Sistem akan melakukan pemeriksaan terhadap <i>id_gejala</i> yang diinputkan sudah tersedia atau belum dalam <i>database</i>, jika <i>id_gejala</i> belum terdapat dalam <i>database</i> maka data gejala bisa ditambahkan dan akan ditampilkan pada tabel data gejala</p>	Valid
8	Tambah Data Gejala Tidak Sah	<p>1. Sistem akan melakukan pemeriksaan terhadap data pada form data gejala apakah sudah terisi semua</p> <p>2. Sistem akan melakukan pemeriksaan terhadap <i>id_gejala</i> yang diinputkan</p>	<p>1. Sistem akan melakukan pemeriksaan terhadap data pada form data gejala apakah sudah terisi semua</p> <p>2. Sistem akan melakukan pemeriksaan terhadap <i>id_gejala</i> yang diinputkan</p>	Valid

No	Nama Kasus	Hasil yang diharapkan	Hasil yang didapatkan	Status Validasi
		sudah terisi atau belum dalam database, jika sudah terdapat dalam database maka data gejala tidak bisa ditambahkan dan akan muncul pesan peringatan	sudah terisi atau belum dalam database, jika sudah terdapat dalam database maka data gejala tidak bisa ditambahkan dan akan muncul pesan peringatan	
9	Edit Data Gejala	Sistem akan melakukan perubahan data dan menyimpan ke dalam database serta menampilkan kembali pada tabel data gejala	Sistem akan melakukan perubahan data dan menyimpan ke dalam database serta menampilkan kembali pada tabel data gejala	Valid
10	Hapus Data Gejala	Sistem akan melakukan hapus data gejala dari database dan secara otomatis data juga akan terhapus dalam tabel data gejala	Sistem akan melakukan hapus data gejala dari database dan secara otomatis data juga akan terhapus dalam tabel data gejala	Valid
11	Tambah Data Basis Pengetahuan Sah	Sistem akan melakukan pemeriksaan terhadap data pada form data gejala apakah sudah terisi semua, jika benar maka otomatis akan ditambahkan dalam database tabel daftar basis pengetahuan	Sistem akan melakukan pemeriksaan terhadap data pada form data gejala apakah sudah terisi semua, jika benar maka otomatis akan ditambahkan dalam database tabel daftar basis pengetahuan	Valid
12	Tambah Data Basis Pengetahuan Tidak Sah	Sistem akan melakukan pemeriksaan terhadap data pada form data basis pengetahuan apakah sudah terisi semua, jika belum	Sistem akan melakukan pemeriksaan terhadap data pada form data basis pengetahuan apakah sudah terisi semua, jika belum maka data tidak dapat	Valid

No	Nama Kasus	Hasil yang diharapkan	Hasil yang didapatkan	Status Validasi
		maka data tidak dapat ditambahkan ke dalam database daftar basis pengetahuan dan akan mengeluarkan pesan peringatan	ditambahkan ke dalam database daftar basis pengetahuan dan akan mengeluarkan pesan peringatan	
13	Edit Data Basis Pengetahuan	Sistem akan melakukan perubahan data dan menyimpan ke dalam database serta menampilkan kembali pada tabel data basis pengetahuan	Sistem akan melakukan perubahan data dan menyimpan ke dalam database serta menampilkan kembali pada tabel data basis pengetahuan	Valid
14	Hapus Data Basis Pengetahuan	Sistem akan melakukan hapus data basis pengetahuan dari database dan secara otomatis data juga akan terhapus dalam tabel data basis pengetahuan	Sistem akan melakukan hapus data basis pengetahuan dari database dan secara otomatis data juga akan terhapus dalam tabel data basis pengetahuan	Valid
15	Ganti Password	Sistem akan melakukan perubahan data dan menyimpan ke dalam database	Sistem akan melakukan perubahan data dan menyimpan ke dalam database	Valid
16	Proses Logout	Sistem akan keluar dari halaman admin dan kembali ke halaman utama	Sistem akan keluar dari halaman admin dan kembali ke halaman utama	Valid
17	Profil Admin	Sistem akan menampilkan profil admin	Sistem akan menampilkan profil admin	Valid
18	Input Data Fakta Gejala	<ol style="list-style-type: none"> 1. Sistem akan menampilkan data gejala dan nilai densitasnya 2. Sistem dapat menyimpan data 	<ol style="list-style-type: none"> 1. Sistem akan menampilkan data gejala dan nilai densitasnya 2. Sistem dapat menyimpan data 	Valid

No	Nama Kasus	Hasil yang diharapkan	Hasil yang didapatkan	Status Validasi
		fakta gejala sesuai yang telah diinputkan sebelumnya oleh pengguna	fakta gejala sesuai yang telah diinputkan sebelumnya oleh pengguna	
19	Hasil Identifikasi	Sistem dapat menampilkan informasi data hasil Identifikasi berupa nama kerusakan dan nilai kepercayaannya	Sistem dapat menampilkan informasi data hasil Identifikasi berupa nama kerusakan dan nilai kepercayaannya	Valid

Berdasarkan pengujian validasi terhadap 19 tindakan dalam daftar kebutuhan dengan metode *black box testing* menunjukkan bahwa sistem memiliki fungsionalitas yang dihitung menggunakan persamaan 2.9.

$$\begin{aligned}
 \text{Fungsionalitas} &= \frac{\text{jumlah tindakan yang dilakukan}}{\text{jumlah tindakan dalam daftar kebutuhan}} \times 100\% \\
 &= \frac{19}{19} \times 100\% \\
 &= 100\%
 \end{aligned}$$

Dari 19 kasus uji yang telah dilakukan pengujian *black box* menunjukkan nilai valid sebesar 100% yang menandakan bahwa fungsionalitas sistem dapat berjalan dengan baik sesuai dengan daftar kebutuhan.

Berdasarkan hasil perbandingan fungsional sistem dengan daftar kebutuhan fungsional dihasilkan nilai 100% sesuai pada Tabel 6.20. Nilai persentase 100% diperoleh dari pembagian data yang valid sebanyak 19 dari 19 daftar kebutuhan. Sehingga fungsional sistem sudah memenuhi kebutuhan yang diperlukan pada sistem ini.

6.1.2 Pengujian Akurasi

Pengujian akurasi dilakukan untuk mengetahui kesesuaian sistem dalam memberikan keluaran yang berupa nama kerusakan dan nilai kepercayaannya. Data yang diuji berjumlah 25 sampel data analisa pakar. Hasil rekomendasi yang diperoleh dari perhitungan di sistem, dicocokkan dengan hasil analisa pakar.

Prosedur pengujian akurasi dilakukan dengan memasukkan data fakta gejala ke dalam sistem yang dibuat, kemudian sistem akan menghitung dengan metode *Dempster-Shafer* untuk mendapatkan hasil kerusakannya. Setelah didapatkan hasil inferensi dari sistem kemudian hasil tersebut dibandingkan dengan hasil keputusan pakar. Hasil pengujian akurasi sistem dari 25 sampel yang telah diuji ditunjukkan pada Tabel 6.21.



Tabel 6.21 Hasil Pengujian Akurasi hasil identifikasi sistem dengan pakar

Printer ke-	Gejala yang dialami	Hasil Identifikasi Sistem	Hasil Identifikasi Pakar	Akurasi
1	Printer tidak dapat dinyalakan, printer tiba – tiba mati.	Power supply rusak	Power supply rusak	1
2	Printer tidak dapat dinyalakan, driver printer tidak terdeteksi di komputer, printer tidak dapat direset	Mainboard rusak	Mainboard rusak	1
3	Printer tidak dapat mencetak, hasil cetakan hitam tidak sempurna,	Cartridge hitam rusak	Cartridge hitam rusak	1
4	Printer tidak dapat mencetak, kertas macet, lampu indikator stop (orange) kedip 3x,	Sensor ASF penarik kertas rusak	Sensor penarik ASF kertas rusak	1
5	Printer tidak dapat mencetak, muncul kode error p07/e08 pada panel LCD	Rset printer	Reset printer	1
6	Printer tidak dapat mencetak, muncul kode error p22 pada panel LCD,	Scanner rusak	Scanner rusak	1
7	Hasil cetakan bergaris putih vertikal, hasil cetakan patah – patah.	Sensor timing disk rusak	Sensor timing disk rusak	1
8	Hasil cetakan bergaris putih	Sensor pita encoder rusak	Sensor pita encoder rusak	1

Printer ke-	Gejala yang dialami	Hasil Identifikasi Sistem	Hasil Identifikasi Pakar	Akurasi
	horizontal, hasil cetakan patah – patah.			
9	Printer tidak dapat mencetak, muncul kode error e15/e16 pada pnel LCD	Reset cartridge	Reset cartridge	1
10	Printer menarik kertas double, lampu indikator stop (orange) kedip 2x,	Roll ASF penarik kertas rusak	Roll ASF penarik kertas rusak	1
11	Printer tidak dapat dinyalakan, driver printer tidak terdeteksi di komputer, printer tidak dapat direset,	Mainboard rusak	Mainboard rusak	1
12	Printer tidak dapat mencetak, hasil cetakan tidak bagus, muncul kode error p02 pada panel LCD,	Rumah cartridge rusak	Rumah cartridge rusak	1
13	Printer tidak dapat mencetak, hasil cetakan hitam tidak sempurna, hasil cetakan hanya muncul warna tertentu saja	Cartridge hitam rusak	Cartridge hitam rusak	1
14	Printer tidak dapat mencetak, lampu indikator	Reset printer	Reset printer	1

Printer ke-	Gejala yang dialami	Hasil Identifikasi Sistem	Hasil Identifikasi Pakar	Akurasi
	stop (orange) kedip 7x			
15	Printer tiba-tiba mati, printer tidak dapat direset, muncul kode error p10 di panel LCD	Mainboard rusak	Mainboard rusak	1
16	Printer tidak dapat dinyalakan, printer tiba-tiba mati, printer tidak dapat mencetak	Mainboard rusak	Mainboard rusak	1
17	Printer tidak dapat mencetak, hasil cetakan hitam tidak sempurna, hasil cetakan hanya muncul warna tertentu saja	Cartridge hitam rusak	Cartridge hitam rusak, cartridge warna rusak	0
18	Printer tidak dapat mencetak, lampu indikator stop (orange) kedip 15x	Reset cartridge	Reset cartridge	1
19	Printer tidak dapat mencetak, muncul kode error p22 pada panel LCD	Scanner rusak	Scanner rusak	1
20	Hasil cetakan tidak bagus, Hasil cetakan hitam tidak sempurna, hasil cetakan hanya muncul warna tertentu saja	Cartridge hitam rusak	Cartridge hitam rusak, cartridge warna rusak	1

Printer ke-	Gejala yang dialami	Hasil Identifikasi Sistem	Hasil Identifikasi Pakar	Akurasi
21	Printer tidak dapat mencetak, hasil cetakan patah-patah	Sensor pita encoder rusak, sensor timing disk rusak	Sensor pita encoder rusak, sensor timing disk rusak	1
22	Driver printer tidak terdeteksi di komputer, lampu indikator stop kedip 3x	Mainboard rusak	Sensor penarik ASF rusak	0
23	Printer tidak dapat mencetak, lampu indikator stop (orange) kedip 15x	<i>Reset printer</i>	<i>Reset printer</i>	1
24	Printer tiba-tiba mati, muncul kode error p10 di panel lcd, muncul kode error e05 di panel LCD	Power supply rusak	Power supply rusak, mainboard rusak.	0
25	Muncul peringatan "ink absorber is full" di layar monitor, printer tidak dapat direset,	<i>Reset printer</i>	<i>Mainboard rusak</i>	0

Hasil akurasi bernilai 1 artinya keluaran dari perhitungan sistem sama dengan keluaran dari data pakar. Sebaliknya, hasil akurasi bernilai 0 artinya keluaran dari perhitungan sistem tidak sama dengan keluaran dari data pakar. Berdasarkan tabel 6.21 telah dilakukan pengujian akurasi dengan 25 sampel data kerusakan printer dan menghasilkan nilai akurasi sesuai dengan persamaan 2.9.

$$\text{Nilai akurasi} = \frac{\text{jumlah data akurat}}{\text{jumlah seluruh data}} \times 100\%$$

$$\text{Nilai akurasi} = \frac{21}{25} \times 100\% = 84\%$$

Dapat disimpulkan bahwa akurasi sistem berdasarkan 25 data yang diuji adalah 84% yang menunjukkan bahwa sistem ini dapat berfungsi dengan baik sesuai dengan identifikasi pakar.

Berdasarkan data observasi yang diberikan oleh pakar mengenai kasus-kasus kerusakan printer yang pernah terjadi maka dihasilkan nilai akurasi sebesar 84% dari penggunaan perhitungan metode *Dempster-Shafer* yang terdapat pada tabel 6.21. Nilai persentase 84% diperoleh dari pembagian data benar sebanyak 21 dari 25 data kasus sebenarnya.

6.2 Analisis

Proses analisis bertujuan untuk mendapatkan kesimpulan dari hasil pengujian pemodelan sistem pakar identifikasi kerusakan menggunakan metode *Dempster-Shafer* yang telah dilakukan. Analisis dilakukan terhadap hasil pengujian di setiap tahap pengujian. Proses analisis yang dilakukan meliputi analisis hasil pengujian validasi dan analisis hasil pengujian akurasi.

6.2.1 Analisis Hasil Pengujian Validasi

Proses analisis terhadap hasil pengujian validasi dilakukan dengan melihat kesesuaian antara hasil kinerja sistem dengan daftar kebutuhan. Hasil pengujian validasi dengan metode *black box testing* adalah 100%, sehingga dapat disimpulkan bahwa pemodelan sistem pakar identifikasi kerusakan printer menggunakan metode *Dempster-Shafer* dapat berjalan sesuai kebutuhan yang telah dijabarkan pada tahap analisis kebutuhan.

6.2.2 Analisis Hasil Pengujian Akurasi

Proses analisis terhadap hasil pengujian akurasi pemodelan sistem pakar identifikasi kerusakan printer menggunakan metode *Dempster-Shafer* dengan melihat persentase keakurasian dan ketidakakurasian sistem dalam menghasilkan identifikasi kerusakan. Pada hasil pengujian sistem menunjukkan adanya ketidaksesuaian hasil identifikasi kerusakan yang muncul. Ketidaksesuaian disebabkan karena beberapa kemungkinan perbedaan nilai densitas yang disesuaikan dengan pengalaman yang didapatkan oleh pakar selama ini dan ketidaksesuaian juga terjadi karena data yang diproses oleh sistem bernilai angka yang hasilnya bernilai pasti.



BAB 7 PENUTUP

7.1 Kesimpulan

Berdasarkan hasil perancangan dan pengujian yang dilakukan pada pemodelan sistem pakar identifikasi kerusakan printer menggunakan metode *Dempster-Shafer*, maka dapat diambil kesimpulan sebagai berikut :

1. Pemodelan sistem pakar identifikasi kerusakan printer mampu mengidentifikasi kerusakan printer serta pengambilan kesimpulan kerusakan dihitung menggunakan metode *Dempster-Shafer* dengan memasukkan fakta gejala dari pengguna.
2. Berdasarkan hasil pengujian didapatkan hasil sebagai berikut :
 - a. Hasil pengujian validasi fungsionalitas pemodelan sistem pakar untuk identifikasi kerusakan printer menggunakan metode *Dempster-Shafer* memiliki tingkat kesesuaian persentase sebesar 100%. Nilai persentase 100% diperoleh dari pembagian data yang valid sebanyak 19 dari 19 daftar kebutuhan. Sehingga fungsional pemodelan sistem pakar ini sudah memenuhi kebutuhan yang diperlukan.
 - b. Hasil pengujian akurasi pemodelan sistem pakar untuk identifikasi kerusakan printer menggunakan metode *Dempster-Shafer* dengan menggunakan 25 sampel data, memiliki tingkat kesesuaian persentase sebesar 84%.

7.2 Saran

Pemodelan sistem pakar untuk identifikasi kerusakan printer menggunakan metode *Dempster-Shafer* ini masih memiliki beberapa kekurangan. Saran yang dapat diberikan untuk pengembangan penelitian dimasa yang akan datang adalah sebagai berikut :

1. Dapat dilakukan penelitian lebih lanjut dalam menentukan nilai densitas dari tiap gejala sehingga akurasi sistem menjadi lebih optimum.
2. Pemodelan sistem pakar ini dapat dikembangkan atau diimplementasikan kedalam sistem pakar.
3. Sistem dapat dikembangkan dengan menambahkan gejala baru dan kerusakan printer jika ditemukannya gejala dan kerusakan baru pada printer.



DAFTAR PUSTAKA

- Aston.2013.<http://astonprinter.malang.com/info/mengenal-istilah-istilah-dalam-dunia-printer>. Diakses pada tanggal 5 November 2015
- Desiani, Anita, 2006, *Konsep Kecerdasan Buatan*, Yogyakarta.
- Dewi, Mustika. 2014. *Aplikasi Sistem Pakar Untuk Pendeteksian Dan Penanganan Dini Pada Penyakit Sapi Dengan Metode Dempster-Shafer Berbasis Web*. Malang: Universitas Brawijaya.
- Fahraini, Bacharuddin. 2014. *Pemodelan Dan Simulasi*. Jakarta: Universitas Mercu Buana
- Fitrianti, Rakhma Indah, 2012. *Sistem Pakar Pada Bidang Teknologi Informasi Untuk Rekomendasi Profesi Pekerjaan Berdasarkan Kepribadian Menggunakan Pendekatan Personality Factor*. Malang: Universitas Brawijaya.
- Jogiyanto, Hartono. 2004. *Pengenalan Komputer*. Yogyakarta: C.V.ANDI OFFSET.
- Wijaya, Hendra. 2010. *Definisi, Karakteristik dan Prinsip-Prinsip Pemodelan Sistem*. Tersedia di: <[http:// www.slideserve.com/molimo/analisis-dan-perancangan-perangkat-lunak](http://www.slideserve.com/molimo/analisis-dan-perancangan-perangkat-lunak)> [Diakses 30 Oktober 2015]
- Hidayati, Iswari Nur, 2010. *Pemanfaatan Teori Bukti Dempster-Shafer Untuk Optimalisasi Penggunaan Lahan Berdasarkan Data Spasial dan Citra Multisumber*. Yogyakarta: Universitas Gajah Mada.
- Kurniawati, I., 2013, *Pengembangan Model Matematika Untuk Penjadwalan Ruang Operasi (Studi Kasus Di Bagian Instalasi Bedah Sentral Rspu Dr.Sardjito)*. Yogyakarta.
- Kusumadewi Sri, 2003, *Artificial Intelligence (Teknik dan Aplikasinya)*. Graha Ilmu.Yogyakarta.
- Lufitasari, Innes Yunia, 2015, *Sistem Pakar Pendeteksi Hama Dan Penyakit Tanaman Cabai Dengan Metode Dempster-Shafer* Malang: Universitas Brawijaya
- Muis, 2015. *Wawancara Gejala dan Kerusakan Printer Inkjet Merk Canon*. Diwawancara oleh Idris Mukhni[tatap muka], 5 November 2015, 19:30.
- Prihatini, Putu Manik, 2011. *Metode Ketidakpastian Dan Kesamaran Dalam Sistem Pakar*. Bali: Politeknik Negeri Bali.
- Sulistyohati, Aprilia, 2008. *Aplikasi Sistem Pakar Diagnosis Penyakit Ginjal dengan Metode Dempster-Shafer*. Yogyakarta: Universitas Islam Indonesia.
- Suriyanti, 2013. *Aplikasi Sistem Pakar Pendeteksi Kerusakan Printer Dengan Case Based Reasoning*. Pelita Informatika Budi Darma.
- Wahyuni, Gustri Ellyza & Prijodiprojo, Widodo, 2013. *Sistem Pakar untuk Mendeteksi Tingkat Resiko Penyakit Jantung Koroner dengan Metode*

Dempster-shafer. Jurusan Ilmu Komputer dan Elektronika FMIPA Universitas Gajah Mada

