

**APLIKASI SISTEM PAKAR UNTUK DIAGNOSA  
KERUSAKAN KOMPUTER PADA *MOBILE DEVICE*  
MENGUNAKAN TEKNOLOGI J2ME**

**TUGAS AKHIR**

**Diajukan sebagai Salah Satu Syarat  
Untuk Memperoleh Gelar Sarjana Program Studi Teknik Informatika**



*oleh :*

**Nama : Bagus Tejo Permono  
No Mahasiswa : 02 523 220**

**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS TEKNOLOGI INDUSTRI  
UNIVERSITAS ISLAM INDONESIA  
YOGYAKARTA**

**2007**

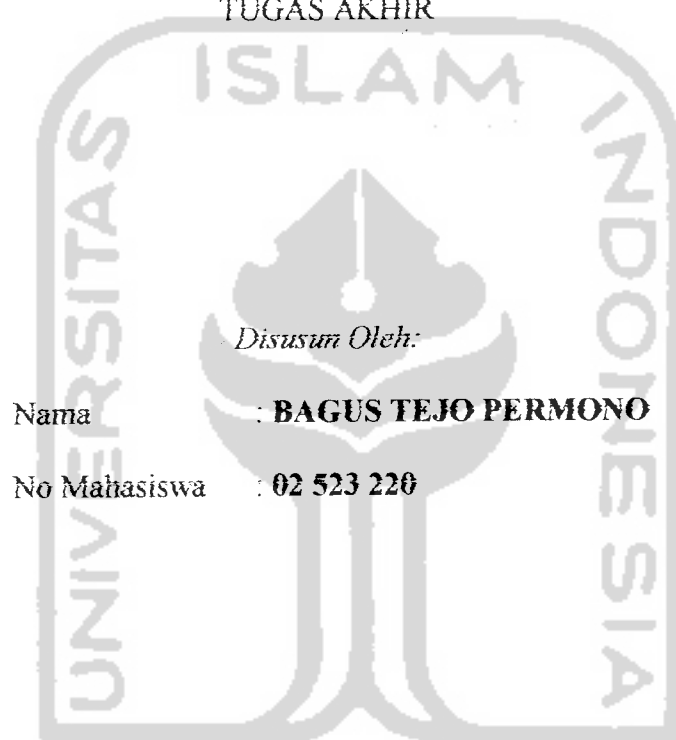
**LEMBAR PENGESAHAN PEMBIMBING**

**APLIKASI SISTEM PAKAR**

**UNTUK DIAGNOSA KERUSAKAN KOMPUTER**

**PADA MOBILE DEVICE MENGGUNAKAN TEKNOLOGI J2ME**

TUGAS AKHIR



*Disusun Oleh:*

Nama : **BAGUS TEJO PERMONO**

No Mahasiswa : **02 523 220**

Yogyakarta, 3 April 2007

Telah Diterima dan Disetujui dengan baik oleh:

Dosen Pembimbing

A handwritten signature in black ink, appearing to read 'Taufiq Hidayat'. The signature is written in a cursive style with a horizontal line underneath.

( Taufiq Hidayat, ST, MCS )

**LEMBAR PENGESAHAN PENGUJI**  
**APLIKASI SISTEM PAKAR**  
**UNTUK DIAGNOSA KERUSAKAN KOMPUTER**  
**PADA MOBILE DEVICE MENGGUNAKAN TEKNOLOGI**  
**J2ME**

**TUGAS AKHIR**

Oleh:

Nama : Bagus Tejo Permono

NIM : 02 523 220

Telah Dipertahankan di Depan Sidang Penguji Sebagai Salah Satu Syarat  
Untuk Memperoleh Gelar Sarjana Teknik Informatika  
Fakultas Teknologi Industri Universitas Islam Indonesia  
Yogyakarta, 2 Mei 2007

Tim Penguji

**Taufiq Hidayat, ST.,MCS.**

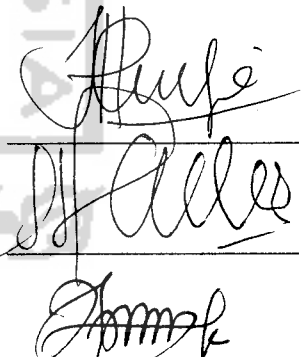
Ketua

**Sri Kusuma Dewi, S.Si, MT.**

Anggota I

**Hendrik, ST.**

Anggota II



Mengetahui,

Ketua Jurusan Teknik Informatika

Universitas Islam Indonesia



**Yudi Prayudi, S.Si.,M.Kom**

## **Alhamdulillah Rabbil 'Aalamiin**

**Puji dan syukur kami panjatkan kehadiran Allah SWT  
Yang telah memberikan rahmat dan hidayah-Nya  
serta memberikan Kemudahan sehingga dapat  
menyelesaikan Tugas akhir ini dengan baik.**

**Dengan izin-Nya pula,  
Kupersembahkan karyaku ini untuk :**

- ✦ *Bapak dan Ibu... yang dengan keikhlasan dan kesabaran tiada henti – hentinya memberikan dukungan dan semangat, serta doa Untuk keberhasilanku, Mudah-mudahan ini merupakan suatu bentuk dari sebagian kecil bakti dan patuhku..*
- ✦ *Dik Hayu dan Hendra... yang selalu memberi semangat dan memberi motivasi bagiku untuk berhasil..*
- ✦ *Buat keluarga besar di Pekalongan & Semarang... mbah saerah, eyang, pakde, tante, om.. terima kasih buat doanya..*
- ✦ *Temen & ex-temen kos ku, Chandra, Dimas, Maji, Qimin, Topik... thanks for our friendship..*
- ✦ *Pemilik kost, Bu & Pak Heri, Mbak Dita & Tika... terima kasih atas semuanya...*
- ✦ *All voip 02, khususnya Arlinto, Imam, Hawie, Shinta... thanks for support..*
- ✦ *Anak-anak KKN, thanks guys..*
- ✦ *And semua yang telah mendukung, terima kasih ya.....*

## HALAMAN MOTTO

**“Sebaik-baiknya manusia adalah yang paling bermanfaat bagi manusia”**

(HR. Tirmidzi)

**“Pelajarilah ilmu Barang siapa mempelajarinya karena Allah, itu taqwa.**

**Menuntutnya itu Ibadah, Mengulang-ulangnya itu Tasbih,**

**Membahasnya itu Jihad,**

**Mengajarkannya kepada orang yang tidak tahu itu Sedekah,**

**Memberikannya kepada ahlinya itu Mendekatkan diri kepada Tuhan”**

(Abusy Syaikh Ibnu Hibban dan Ibnu Abdil Barr, Ilya Al Ghozali)

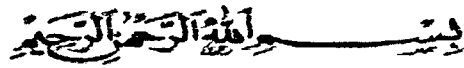
**“Sesungguhnya disamping kesukaran itu ada kemudahan”**

(Qs. Al Insyirah : 6 )

**“Allah tidak akan membebani seseorang melainkan sesuai dengan kesanggupannya”**

(Qs. Al Baqoroh : 286 )

## KATA PENGANTAR



*Assalamu'alaikum, Wr.Wb.*

*Alhamdulillah* rabbil'alam, dengan memanjatkan puji syukur kehadiran Allah SWT atas limpahan hidayah, kekuatan, petunjuk, taufik serta inayah-Nya sehingga Tugas Akhir yang berjudul "**Aplikasi Sistem Pakar Untuk Diagnosa Kerusakan Komputer Pada Mobile Device Dengan Menggunakan Teknologi J2ME**" ini dapat diselesaikan dengan baik.

Tugas Akhir ini ditulis sebagai syarat untuk mendapatkan gelar sarjana di jurusan Teknik Informatika Universitas Islam Indonesia setelah menempuh proses akademis selama kurang lebih delapan semester.

Penyusun berharap Tugas Akhir ini dapat memberikan manfaat dan memberikan kontribusi yang baik khususnya bagi Jurusan Teknik Informatika Universitas Islam Indonesia.

Dalam menyusun tugas akhir ini, penyusun telah banyak mendapat bantuan, bimbingan dari berbagai pihak, maka dari itu dalam kesempatan ini perkenankanlah penyusun menyampaikan rasa terima kasih yang sebesar-besarnya kepada:

1. Allah SWT, yang telah memberikan perlindungan, rahmat, hidayah dan semua kenikmatan-Nya hingga penyusun dapat menyelesaikan Tugas Akhir ini dengan sebaik-baiknya.

2. Kedua orang tua ayah Kaelani dan ibu Tri'ah serta adik Hayu dan Hendra tercinta yang tak henti-hentinya memberikan doa dan semua dukungan.
3. Bapak Prof. Dr. Edy Suandi Hamid, selaku rektor Universitas Islam Indonesia.
4. Bapak Fathul Wahid, ST, M.Sc, selaku Dekan Fakultas Teknologi Industri, Universitas Islam Indonesia.
5. Bapak Yudi Prayudi, S.Si., M.Kom., selaku Ketua Program Studi Teknik Informatika, Universitas Islam Indonesia
6. Bapak Taufiq Hidayat ST., MCS., selaku dosen pembimbing yang telah membimbing sehingga Tugas Akhir ini dapat diselesaikan dengan sebaik-baiknya.
7. Seluruh karyawan dan staf bagian pengajaran khususnya bagian KP/TA yang telah membantu penyusun dalam perijinan dan surat menyurat.
8. Terima kasih juga kepada semua teman-teman yang tidak dapat disebutkan satu persatu, yang telah membantu memberikan dukungan dan ilmu sehingga Tugas Akhir ini dapat diselesaikan dengan baik.

Penyusun menyadari adanya banyak kekurangan dalam Laporan Tugas Akhir ini, untuk itu penyusun sangat mengharapkan kritik dan saran dari semua pihak. Semoga Laporan Tugas Akhir ini dapat bermanfaat bagi kita semua.

*Wassalamu'alaikum , Wr.Wb.*

Jogjakarta, 3 April 2007

Penyusun

## SARI

Perkembangan teknologi telah membawa dampak positif bagi perkembangan bangsa-bangsa di seluruh dunia termasuk bangsa Indonesia. Perkembangan sebuah perangkat mobile device sangat cepat. Mobile device sangat cepat. Mobile device dirancang sedemikian hingga mudah untuk dibawa kemana saja. Sehingga dapat memudahkan pengguna untuk menggunakannya. Selain itu sekarang cenderung semakin banyak orang menggunakan mobile device dan semakin banyak fasilitas dari mobile device yang memudahkan penggunaannya. Untuk itu aplikasi sistem pakar untuk diagnosa kerusakan komputer pada mobile device ini dibuat.

Kemudahan dalam mengetahui kerusakan komputer sangat dibutuhkan oleh pengguna dan pemilik komputer. Dengan memanfaatkan teknologi mobile device dalam aplikasi ini, maka pengguna dan pemilik komputer dapat terbantu. Dengan hanya memasukkan gejala kerusakan komputer, pengguna aplikasi ini bisa dengan segera mengetahui kerusakan yang terjadi dan solusi perbaikannya. Sehingga mempermudah pengguna dalam menangani kerusakan komputer dan sebagai proses pembelajaran bagi penggunanya.

Aplikasi ini menggunakan teknologi J2ME dengan memanfaatkan pustaka RMS sebagai manajemen basis datanya karena teknologi ini bersifat open platform. Disamping itu aplikasi ini hanya memerlukan kapasitas memori kecil untuk instalasinya sehingga tidak memberatkan kinerja perangkat mobile device.

Kata kunci: J2ME, RMS, sistem pakar, expert system, mobile device



## TAKARIR

<i>Artificial intelligence</i>	Ilmu yang mengembangkan komputer supaya dapat bekerja dan berpikir serta mengambil keputusan seperti layaknya manusia
<i>Byte</i>	Karakter yang dibentuk oleh bit-bit. 1 byte = 8bit
<i>Certainty factor</i>	Faktor kepastian
<i>Class</i>	Definisi formal dari suatu objek
<i>Class diagram</i>	Diagram yang menggambarkan struktur dan deskripsi <i>class</i> dan objek beserta hubungan satu sama lain
<i>Compiler</i>	Program yang menerjemahkan serangkaian perintah ke dalam bahasa mesin. Sekali program dikompilasi, program tersebut dapat dijalankan sesering mungkin tanpa harus mengulang proses kompilasi, kecuali jika terjadi perubahan pada program sumber
<i>Database</i>	Kumpulan file/tabel arsip yang saling berhubungan yang disimpan dalam media penyimpanan elektronik
<i>Emulator</i>	berusaha menyamai. Sistem operasi yang dapat menjalankan sistem operasi lainnya
<i>File</i>	Arsip

<i>Handheld device</i>	Perangkat genggam
<i>Interface</i>	Antarmuka
<i>Library</i>	Kumpulan fungsi atau prosedur atau modul.
<i>Lifecycle</i>	Daur hidup
<i>Mobile</i>	Bergerak
<i>Platform</i>	Lingkungan struktur utama dari suatu sistem
<i>Record store</i>	Tempat penyimpanan record (kumpulan dari beberapa field)
<i>Runtime</i>	Saat suatu program dijalankan
<i>Sequence diagram</i>	Diagram yang menggambarkan interaksi antar objek di dalam dan di sekitar sistem berupa message yang digambarkan terhadap waktu
<i>Use case diagram</i>	Diagram yang menggambarkan hubungan antara user dengan proses pada aplikasi
<i>User</i>	Pemakai, pengguna
<i>User interface</i>	Tampilan yang dilihat oleh user pada sistem operasi

## DAFTAR ISI

HALAMAN JUDUL .....	i
LEMBAR PENGESAHAN DOSEN PEMBIMBING.....	ii
LEMBAR PENGESAHAN DOSEN PENGUJI.....	iii
LEMBAR PERNYATAAN KEASLIAN.....	iv
HALAMAN PERSEMBAHAN.....	v
HALAMAN MOTTO .....	vi
KATA PENGANTAR.....	vii
SARI.....	ix
TAKARIR.....	x
DAFTAR ISI.....	xii
DAFTAR TABEL.....	xv
DAFTAR GAMBAR.....	xvi
DAFTAR LAMPIRAN.....	xix
BAB I    PENDAHULUAN.....	1
1.1    Latar Belakang.....	1
1.2    Rumusan Masalah.....	3
1.3    Batasan Masalah.....	3
1.4    Tujuan Penelitian.....	3
1.5    Manfaat Penelitian.....	4
1.6    Metodologi Penelitian.....	4
1.6.1    Pengumpulan data.....	4
1.6.2    Pengembangan perangkat lunak.....	5
1.7    Sistematika Penulisan.....	5
BAB II    LANDASAN TEORI.....	8
2.1    Java.....	8
2.2    Java 2 Platform, Micro Edition (J2ME).....	11
2.3    MIDlets.....	14
2.3.1    Daur Hidup MIDlets.....	14

	2.3.2	<i>High Level API</i> .....	15
	2.3.3	<i>Low Level API</i> .....	15
	2.4	Konsep <i>Database Record Management System (RMS)</i> .....	15
	2.4.1	Paket <i>javax.microedition.rms</i> .....	16
	2.5	Sistem Pakar.....	29
	2.5.1	Ciri-ciri sistem pakar.....	18
	2.5.2	Kategori Masalah Dari Sistem Pakar.....	20
	2.5.3	Representasi Pengetahuan.....	21
	2.5.4	Faktor Kepastian.....	24
	2.5.4	Sistem Kerja Pakar.....	26
	2.6	Identifikasi Kerusakan Komputer.....	28
BAB III		ANALISIS KEBUTUHAN SISTEM PERANGKAT LUNAK.....	30
	3.1.	Metode Analisis.....	30
	3.2.	Hasil Analisis.....	30
	3.2.1	Kebutuhan Masukan.....	30
	3.2.2	Kebutuhan Keluaran.....	32
	3.2.3	Kebutuhan Proses.....	32
	3.2.4	Analisis Antarmuka Sistem.....	33
	3.2.5	Perangkat lunak yang digunakan.....	33
	3.2.6	Perangkat keras yang dibutuhkan.....	34
BAB IV		PERANCANGAN PERANGKAT LUNAK.....	35
	4.1	Metode Perancangan.....	35
	4.2	Hasil Perancangan Perangkat Lunak.....	35
	4.2.1	<i>Use Case Diagram</i> .....	35
	4.2.2	<i>Class Diagram</i> .....	37
	4.2.3	<i>Sequence Diagram</i> .....	37
	4.2.4	Perancangan Antarmuka.....	41
	4.2.5	Perancangan Basis Pengetahuan.....	50
	4.2.4	Perancangan <i>Inference Engine</i> .....	51
BAB V		IMPLEMENTASI PERANGKAT LUNAK.....	53
	5.1	Implementasi Secara Umum.....	53

5.2	Implementasi Antarmuka.....	53
5.2.1	Tampilan Halaman Muka.....	53
5.2.2	Tampilan Menu Utama.....	54
5.2.3	Tampilan Halaman Konsultasi.....	55
5.2.4	Tampilan Halaman Pakar.....	58
5.2.5	Tampilan Halaman Info Program.....	63
5.2.6	Tampilan Halaman Bantuan.....	64
5.3	Implementasi Prosedural.....	65
5.3.1	<i>Class DataGejala</i> .....	65
5.3.2	<i>Class DataKerusakan</i> .....	66
5.3.3	<i>Class DataBasisPengetahuan</i> .....	67
5.3.4	<i>Class Konsultasi</i> .....	68
5.3.5	<i>Class SistemPakar</i> .....	70
BAB VI	ANALISIS KINERJA PERANGKAT LUNAK.....	71
6.1	Pengujian Aplikasi.....	71
6.1.1	Pengujian Proses Diagnosa.....	71
6.1.1.1	Pengujian proses diagnosa.....	67
6.1.2	Proses Olah Data.....	79
BAB VII	KESIMPULAN DAN SARAN.....	86
7.1	Kesimpulan.....	86
7.2	Saran.....	87
DAFTAR	PUSTAKA.....	88
LAMPIRAN	.....	90

**DAFTAR TABEL**

Tabel 2.1 Perbandingan CDC dan CLDC.....

13



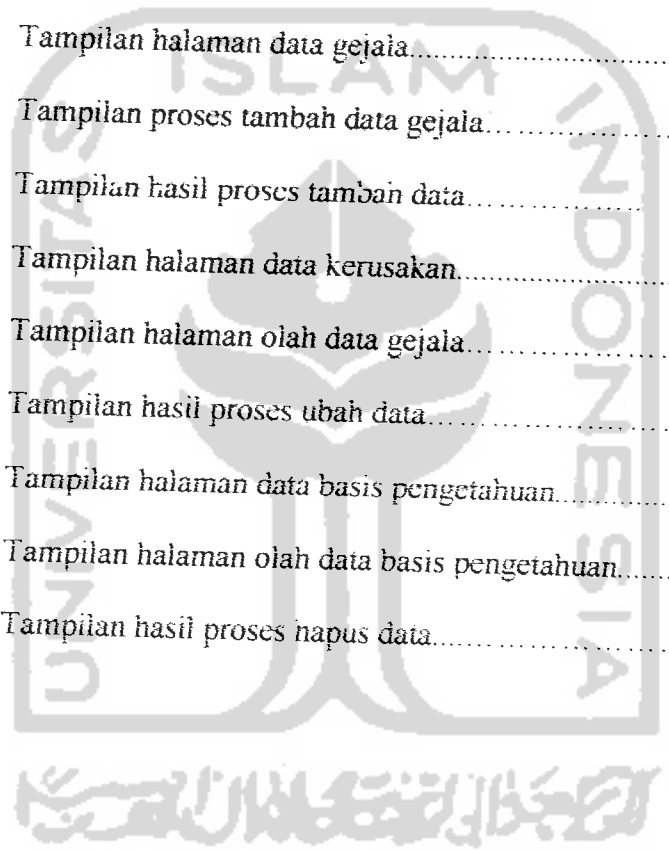
## DAFTAR GAMBAR

Gambar 2.1	Lingkungan kerja Teknologi Java.....	10
Gambar 2.2	<i>Life Cycle</i> dan perubahan status MIDiets.....	14
Gambar 2.3	Komponen utama sistem pakar.....	20
Gambar 2.4	Modul penerimaan pengetahuan.....	25
Gambar 4.1	<i>Use case diagram</i> dengan aktor <i>User</i> .....	36
Gambar 4.2	<i>Use case diagram</i> dengan aktor Pakar.....	36
Gambar 4.3	<i>Class Diagram</i> .....	37
Gambar 4.4	<i>Sequence Diagram</i> untuk olah data gejala.....	38
Gambar 4.5	<i>Sequence Diagram</i> untuk olah data kerusakan.....	38
Gambar 4.6	<i>Sequence Diagram</i> untuk olah data basis pengetahuan.....	39
Gambar 4.7	<i>Sequence Diagram</i> untuk proses konsultasi.....	40
Gambar 4.8	<i>Sequence Diagram</i> untuk lihat info program.....	39
Gambar 4.9	<i>Sequence Diagram</i> untuk lihat bantuan program.....	40
Gambar 4.10	Halaman Muka.....	41
Gambar 4.11	Halaman Menu Utama.....	42
Gambar 4.12	Halaman Konsultasi.....	43
Gambar 4.13	Halaman Pilih Gejala Kerusakan.....	43
Gambar 4.14	Halaman Hasil Diagnosa.....	44
Gambar 4.15	Halaman Pakar.....	45
Gambar 4.16	Halaman Pilih Olah Data.....	45
Gambar 4.17	Halaman View Data Gejala.....	46

Gambar 4.18	Halaman Olah Data Gejala.....	46
Gambar 4.19	Halaman View Data Kerusakan.....	47
Gambar 4.20	Halaman Olah Data Kerusakan.....	47
Gambar 4.21	Halaman View Data Basis Pengetahuan.....	48
Gambar 4.22	Halaman Olah Data Basis Pengetahuan.....	48
Gambar 4.23	Halaman Info Program.....	49
Gambar 4.24	Halaman Bantuan Program.....	49
Gambar 4.25	Struktur Tabel Basis Pengetahuan.....	50
Gambar 4.26	Inferensi <i>for,ward chaining</i> .....	52
Gambar 5.1	Tampilan Halaman Muka.....	54
Gambar 5.2	Tampilan Menu Utama.....	55
Gambar 5.3	Tampilan halaman pilih piranti konsultasi.....	56
Gambar 5.4	Tampilan halaman pilih gejala kerusakan.....	57
Gambar 5.5	Tampilan halaman hasil konsultasi.....	57
Gambar 5.6	Tampilan halaman pilih piranti pakar.....	59
Gambar 5.7	Tampilan halaman pilih jenis olah data.....	60
Gambar 5.8	Tampilan halaman data gejala.....	60
Gambar 5.9	Tampilan halaman olah data gejala.....	61
Gambar 5.10	Tampilan halaman data kerusakan.....	61
Gambar 5.11	Tampilan halaman olah data kerusakan.....	62
Gambar 5.12	Tampilan halaman data basis pengetahuan.....	62
Gambar 5.13	Tampilan halaman olah data basis pengetahuan.....	63
Gambar 5.14	Tampilan halaman info program.....	64



Gambar 5.15	Tampilan halaman bantuan.....	65
Gambar 6.1	Tampilan hasil diagnosa pengujian pertama.....	73
Gambar 6.2	Tampilan hasil diagnosa pengujian kedua.....	74
Gambar 6.3	Tampilan hasil diagnosa pengujian ketiga.....	76
Gambar 6.4	Tampilan hasil diagnosa pengujian keempat.....	78
Gambar 6.5	Tampilan hasil diagnosa pengujian kelima.....	79
Gambar 6.6	Tampilan halaman data gejala.....	80
Gambar 6.7	Tampilan proses tambah data gejala.....	81
Gambar 6.8	Tampilan hasil proses tambah data.....	81
Gambar 6.9	Tampilan halaman data kerusakan.....	82
Gambar 6.10	Tampilan halaman olah data gejala.....	83
Gambar 6.11	Tampilan hasil proses ubah data.....	83
Gambar 6.12	Tampilan halaman data basis pengetahuan.....	84
Gambar 6.13	Tampilan halaman olah data basis pengetahuan.....	85
Gambar 6.14	Tampilan hasil proses hapus data.....	85



## DAFTAR LAMPIRAN

Lampiran A Data-data analisis kinerja perangkat lunak .....	90
1. Data Gejala .....	90
2. Data Kerusakan .....	91
3. Data Basis Pengetahuan .....	94



# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Salah satu cabang ilmu komputer yang dapat membantu manusia adalah kecerdasan buatan atau *artificial intelligence*. Kecerdasan buatan adalah cabang ilmu komputer yang bertujuan untuk membuat sebuah komputer dapat berpikir dan bernalar seperti manusia. Tujuan praktis dari kecerdasan buatan ini adalah membuat komputer semakin berguna bagi manusia. Kecerdasan buatan dapat membantu manusia dalam membuat keputusan, mencari informasi secara lebih akurat, atau membuat komputer lebih mudah digunakan.

Salah satu bagian dari sistem kecerdasan buatan adalah sistem pakar dimana sistem pakar adalah bagian dari ilmu kecerdasan buatan yang secara spesifik berusaha mengadopsi kepakaran seseorang di bidang tertentu ke dalam suatu sistem atau program komputer. Implementasi sistem pakar banyak digunakan untuk kepentingan komersial karena sistem pakar dipandang sebagai cara penyimpanan pengetahuan pakar dalam bidang tertentu ke dalam program komputer sedemikian rupa sehingga dapat memberikan keputusan dan melakukan penalaran secara cerdas. Salah satu contoh implementasi dari sistem pakar ini adalah untuk mendiagnosa kerusakan pada komputer .

Pengetahuan-pengetahuan dari pakar komputer disimpan dalam suatu program komputer sehingga komputer tersebut bisa memprosesnya untuk menghasilkan suatu *output* berupa solusi seperti layaknya solusi yang diberikan

oleh pakar komputer. Untuk itu program sistem pakar ini diharapkan dapat menganalisa dan mendiagnosa kerusakan pada komputer sehingga pengguna dapat mengetahui kerusakan yang terjadi dan bagaimana langkah-langkah memperbaiki kerusakan tersebut.

Dengan semakin berkembangnya teknologi komputer, aplikasi-aplikasi komputer tidak hanya bisa diakses lewat komputer saja, akan tetapi bisa juga diakses melalui peralatan-peralatan *mobile* yang sudah memanfaatkan teknologi komputer, contohnya adalah telepon genggam atau ponsel. Karena semakin banyaknya fasilitas dari telepon genggam yang berguna untuk memudahkan para penggunanya, sekarang ini orang lebih cenderung untuk melakukan segala urusan melalui telepon genggam, terutama sekali bagi pengguna yang sibuk dan tidak memiliki banyak waktu.

Untuk mendukung mobilitas manusia, maka diperlukan sarana yang memadai. Salah satu sarana yang mendukung mobilitas manusia adalah dengan membangun suatu sistem yang menggunakan piranti bergerak. Sistem yang dibangun harus diusahakan dapat mencapai semua lapisan pengguna yang berkepentingan.

Dengan adanya Aplikasi Sistem Pakar Untuk Diagnosa Kerusakan Komputer untuk *Mobile Devices* ini dapat memudahkan pengguna dalam mendiagnosa kerusakan komputer hanya dengan mengaksesnya melalui HP tanpa harus pergi ke pakar atau teknisi komputer.

## 1.2 Rumusan Masalah

Berdasarkan latar belakang masalah yang diuraikan, perumusan masalah dalam penelitian ini adalah bagaimana membuat *Aplikasi Sistem Pakar Untuk Diagnosa Kerusakan Komputer Pada Mobile Devices dengan Menggunakan Teknologi J2ME*.

## 1.3 Batasan Masalah

Berdasarkan latar belakang masalah dan rumusan masalah maka dalam penelitian ini perlu adanya batasan masalah, yaitu:

1. Program ini hanya bisa dijalankan pada *handphone* yang mendukung teknologi Java dengan MIDP 2.0.
2. Perhitungan penalaran sistem pakar menggunakan metode faktor kepastian (*certainty factor*).
3. Sistem pakar ini dibuat menggunakan J2ME dengan basis data *Record Management System (RMS)*.

## 1.4 Tujuan Penelitian

Tujuan dari penelitian ini adalah sebagai berikut :

1. Membangun program sistem pakar untuk diagnosa kerusakan komputer untuk diterapkan pada ponsel, yaitu dengan teknologi Java (J2ME).
2. Membantu pemilik komputer, teknisi komputer, atau siapa saja dalam mengetahui kerusakan komputer dan cara memperbaikinya dengan mudah melalui ponsel.

## 1.5 Manfaat Penelitian

Manfaat dari penelitian ini adalah sebagai berikut :

1. Sebagai media pembelajaran membangun suatu aplikasi sistem pakar dalam hal ini adalah untuk diagnosa kerusakan komputer.
2. Membantu pengguna komputer atau orang umum untuk dapat mendiagnosa kerusakan komputer dari gejala-gejala yang ada dan bagaimana cara memperbaikinya.

## 1.6 Metodologi Penelitian

Metode penelitian adalah suatu cara berurutan yang dilakukan dalam penelitian. Metode yang digunakan untuk penelitian adalah metode pengumpulan data dan metode pengembangan perangkat lunak.

### 1.6.1 Metode Pengumpulan Data

Metode pengumpulan data merupakan cara pertama yang digunakan dalam melakukan penelitian. Metode pengumpulan data terdiri dari :

1. Metode Observasi

Metode observasi antara lain pengumpulan data dengan mengamati data-data apa saja yang perlu dimasukkan ke dalam *database*.

2. Metode *Library Research*

Metode *Library Research* terdiri dari pengumpulan data dan informasi dari buku-buku referensi, modul-modul, dan artikel yang sesuai dengan penelitian.

### 1.6.2 Metode Pengembangan Perangkat Lunak

Metode pengembangan perangkat lunak disusun berdasarkan hasil yang sudah diperoleh dari proses pengumpulan data. Metode ini meliputi:

#### 1. Analisis Kebutuhan Sistem

Analisis ini dilakukan untuk mengolah data yang sudah didapat dan mengelompokkan data-data tersebut sesuai kebutuhan perancangan sistem.

#### 2. Desain

Tahap ini merupakan tahap penerjemahan dari keperluan data yang telah dianalisis kedalam bentuk yang mudah dimengerti oleh pengguna ponsel.

#### 3. Implementasi

Implementasi pada perangkat lunak (*software*) menggunakan Teknologi Java 2 *Micro Edition*.

#### 4. Pengujian

Pengujian terhadap Perangkat Lunak yang telah dibangun, dengan pengujian secara normal dan tidak normal.

### 1.7 Sistematika Penulisan

Untuk memberikan gambaran menyeluruh mengenai masalah yang akan dibahas, maka digunakan sistematika penulisan tugas akhir. Sistematika penulisan tugas akhir *Aplikasi Sistem Pakar Untuk Diagnosa Komputer Pada Mobile Devices dengan Menggunakan Teknologi J2ME* adalah sebagai berikut :

## **BAB II**

### **LANDASAN TEORI**

Pada saat ini pemrograman berkembang ke teknologi *Wireless internet* sehingga memungkinkan koneksi jaringan *internet* dengan perangkat bergerak ( *Mobile devices* ) [YAN06]. Dengan teknologi “*write once run everywhere*” aplikasi-aplikasi untuk *Mobile devices* dapat dikembangkan dalam Java. Java 2 *Micro Edition* (J2ME) digunakan untuk menjalankan dan mengembangkan aplikasi-aplikasi Java pada perangkat semacam telepon genggam, PDA (*Personal Digital Assistance*), dan *pocket PC*. Dengan adanya J2ME, memungkinkan bagi para pengembang untuk bisa membuat aplikasi *wireless* yang multi *platform* dan yang dapat diimplementasikan pada berbagai merek telepon genggam yang mendukung aplikasi Java.

Aplikasi sistem pakar untuk mendiagnosa kerusakan komputer pada penelitian ini dibangun dengan menggunakan teknologi Java 2 *Micro Edition* (J2ME) dengan J2ME *wireless toolkit 2.2* sebagai *emulator compiler*-nya dan *library-library* Java.

#### **2.1 Java**

Java merupakan bahasa pemrograman berorientasi obyek yang dapat digunakan untuk pengembangan aplikasi mandiri, aplikasi berbasis *internet* maupun *intranet*, serta aplikasi perangkat-perangkat cerdas yang dapat berkomunikasi lewat jaringan komunikasi. Melalui teknologi Java, saat ini



1. dimungkinkan perangkat *audio stereo* dan perangkat lain di rumah menjadi terhubung jaringan komputer. Java tidak lagi merupakan bahasa untuk membuat applet untuk sekedar memperindah halaman *web*, tapi Java telah menjadi bahasa untuk pengembangan aplikasi skala *enterprise* jaringan luas [YAN06].
2. Sebagai bahasa pemrograman, Java dikenal sebagai bahasa pemrograman tingkat tinggi dengan fitur-fitur utama seperti :
  1. Berorientasi objek.
  2. Terdistribusi.
  3. Arsitekturnya netral. *Compiler* Java membangkitkan sebuah format *file* dengan instruksi kode *byte* yang tidak dapat dilakukan oleh arsitektur komputer tertentu.
  4. *Portabel*. Java tidak terdapat ketergantungan pada saat implementasi.
  5. *Interpreter* pada Java dapat mengeksekusi kode *byte* Java secara langsung pada komputer-komputer yang memiliki *interpreter*.
  6. *Multithreaded*. Bisa dihubungkan dengan banyak bahasa pemrograman lainnya.
  7. Dinamis. Java dirancang untuk dapat dengan mudah beradaptasi dengan lingkungan yang berkembang.

Java adalah teknologi dan bahasa pemrograman yang berjalan pada multplatform sesuai dengan semboyannya yaitu "Write Once, Run Anywhere". Pada *official site* Java dari Sun yaitu <http://java.sun.com> bisa ditemui tiga pembagian paket Java yaitu :

## 2.2 Java 2 Platform, Micro Edition (J2ME)

Komponen-komponen J2ME terdiri dari Java *Virtual Machine* (JVM) yang digunakan untuk menjalankan aplikasi Java pada *emulator* atau *handheld device*, Java API (*Application Programming Interface*) dan *tools* lain untuk pengembangan aplikasi Java semacam *emulator Java Phone*, *emulator Motorola* dari J2ME *wireless toolkit* [YAN06].

Dalam pengembangan aplikasi *wireless* dengan Java, J2ME dibagi menjadi dua buah bagian diantaranya ialah bagian *profile* dan *configuration* yang didukung oleh Java *Virtual Machine* (*C-virtual Machine* dan *K-Virtual Machine*), yaitu :

### 1. Lapisan Profil (*Profile Layer*)

J2ME menyediakan implementasi-implementasi tambahan yang sangat spesifik pada *handheld devices*.

J2ME mempunyai beberapa profil antara lain :

- a. *Mobile Information Device Profile* (MIDP)
- b. *Foundation Profile* (FP)
- c. *Personal Profile*
- d. *RMI Profile*
- e. *Personal Digital Assistance (PDA) Profile*

### 2. Lapisan Konfigurasi (*Configuration Layer*)

J2ME *Configuration* mendefinisikan lingkungan kerja J2ME *Runtime* dan menyediakan *Library* standar yang mengimplementasikan fitur standar sebuah *handheld devices* (perangkat genggam lingkungan kerja). Lingkungan

kerja yang dimaksud adalah JVM (*Java Virtual Machine*) yang digunakan untuk menjalankan program Java dan mengimplementasikan fitur-fitur penting yang berkaitan dengan perangkat *wireless* saja.

J2ME mempunyai dua konfigurasi yaitu :

a. *Connected Limited Device Configuration (CLDC)*

Kategori ini umum digunakan untuk aplikasi Java pada *handphone* seperti Motorola i85s, *organizer* atau PDA semacam Palm, *Pocket PC* dan *two way pagers*.

CLDC menyediakan *platform* Java standar yang cocok untuk perangkat kecil dengan sumber daya terdaya terbatas. Perangkat ini umumnya mempunyai pemroses 16 bit atau 32 bit dan memori 160 Kbyte sampai 512 Kbyte. Inti CLDC adalah mesin maya K (KVM).

CLDC mempunyai spesifikasi adalah sebagai berikut :

1. Mengimplementasikan subset dari J2SE
2. JVM yang digunakan dikenal dengan nama *K-virtual Machine* (KVM)
3. Digunakan pada perangkat *handheld* dengan ukuran memori terbatas (160 -512 Kbytes)
4. Prosesor : 16 bit atau 32 bit

Pada bagian ini secara detail CLDC diperlukan untuk pengembangan aplikasi *wireless* dengan MIDP implementasinya CLDC digunakan untuk program Java pada perangkat keras dengan ukuran memori yang terbatas, pada 160 sampai dengan 512 Kilobyte. Akibatnya, fitur fitur

yang kurang penting untuk diimplementasikan dalam *handheld device* yang bersangkutan dari Java 2 harus dibuang.

b. *Connected Device Configuration* (CDC).

CDC menyediakan mesin maya dan pustaka kelas dasar yang mendukung aplikasi Java untuk perangkat seperti *Smart Communicator*, *pager*, PDA (*Personal Digital Assistant*), dan *Television Set-top Box*. Perangkat-perangkat ini umumnya memiliki pemroses 32 bit dan memori lebih dari 2 MB.

Konfigurasi ini memiliki kemampuan untuk melakukan koneksi *internet*. Fungsionalitas mesin maya Java dipenuhi oleh CDV *virtual machine* yang merupakan mesin maya dengan fitur penuh.

**Tabel 2.1** Perbandingan CDC dan CLDC [WIR03]

CLDC ( <i>Connected Limited Device Configuration</i> )	CDC ( <i>Connected Device Configuration</i> )
Mengimplementasikan subset dari J2SE	Mengimplementasikan seluruh fitur dari J2SE
JVM yang digunakan adalah KVM	JVM yang digunakan adalah CVM.
Digunakan pada perangkat <i>handheld</i> ( <i>handphone</i> , PDA, <i>two way pager</i> ) dengan memori terbatas (160-512 kb)	Digunakan pada perangkat <i>handheld</i> ( <i>internet TV</i> , <i>Nokia Communicator</i> , <i>car TV</i> ) dengan memori minimal 2 Mb
Prosesor : 16/ 32 bit	Prosesor : 32 bit

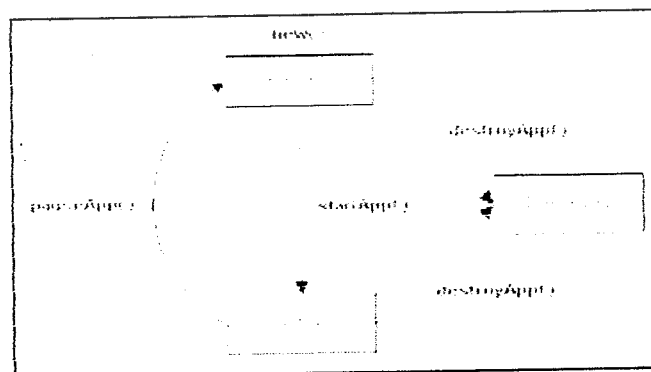
### 2.3 MIDlets

Aplikasi yang berjalan pada sebuah perangkat yang mendukung MIDP disebut dengan MIDlets, atau lebih singkatnya MIDlet merupakan aplikasi yang dibuat menggunakan Java 2 Micro Edition dengan profile *Mobile Information Device Profile* (MIDP) [YAN06].

MIDP dikhususkan untuk digunakan pada *handset* dengan kemampuan CPU, memori, *keyboard* dan *layer* yang terbatas, seperti *handphone*, *pager*, PDA dan sebagainya.

#### 2.3.1 Daur Hidup (*Life Cycle*) MIDlets

*Lifecycle* dari sebuah MIDlet ditangani oleh *Application Management Software* (AMS). AMS adalah sebuah lingkungan tempat siklus dari sebuah MIDlet, mampu untuk diciptakan, dijalankan, dihentikan maupun dihilangkan. AMS sering pula disebut dengan *Java Application Manager* (JAM). MIDlet memiliki beberapa *state*, yaitu *Pause*, *Active* dan *Destroy*. Ketika masing-masing *state* dipanggil, beberapa *method* yang bersesuaian dipanggil. *Method-method* tersebut merupakan bawaan dari J2ME. Untuk menjelaskan proses MIDlet dalam *Java Application Manager* (JAM) adalah pada Gambar 2.2.



**Gambar 2.2** *LifeCycle* dan perubahan status MIDlet

### 2.3.2 High Level API

Kelas-kelas yang menyediakan fungsionalitas untuk pembuatan GUI pada MIDP ada pada paket *javax.microedition.lcdui*. Pada paket tersebut terdapat tiga *interface* dan 21 kelas. *Interface* tersebut adalah *Display*, *Screen* dan *Form*.

### 2.3.3 Low Level API

Pada *level* pemrograman yang lebih rendah (*low level*), akan ditemukan fungsionalitas yang lebih spesifik ke jenis *handheld* yang digunakan. Kelas-kelas untuk pemrograman GUI pada *level* yang lebih rendah ini diimplementasikan oleh kelas *javax.microedition.lcdui.Canvas* dan *javax.microedition.lcdui.Graphics*. Kelas *Canvas* ini memungkinkan pengguna untuk menggambar garis, titik dan elemen-elemen dasar lain.

## 2.4 Konsep Database Record Management System (RMS)

RMS adalah suatu mekanisme pada MIDlet yang dapat mengelola database berukuran kecil dalam memori penyimpanan pada perangkat-perangkat cerdas. RMS menyediakan fungsi pada aplikasi J2ME yang sama dengan fungsi yang disediakan oleh JDBC pada aplikasi J2SE, yaitu *persistent storage facility* [GUP02].

RMS merupakan implementasi *database* yang digunakan dalam MIDlet Java yang berorientasi *Record*. Hal ini berkaitan dengan keadaan ukuran memori pada perangkat-perangkat *handphone*, *organizer* yang relatif kecil. Pada pendekatan *record-oriented*, RMS J2ME terdiri dari beberapa *RecordStore* [YAN06].

Paket yang menyediakan fungsi-fungsi untuk manajemen RMS ini berada pada *javax.microedition.rms\**. Fungsi-fungsi dalam paket *javax.microedition.rms\** menyediakan fungsi-fungsi untuk menghapus, menambah, mengubah data dan manipulasi lain pada *RecordStore*. Fungsi-fungsi dalam *javax.microedition.rms\** juga menyediakan mekanisme yang memungkinkan dua atau lebih MIDlet dalam suatu aplikasi untuk berbagi data *RecordStore*.

#### 2.4.1 Paket *javax.microedition.rms*

Paket-paket ini menyediakan fungsi-fungsi untuk mengimplementasikan *database* RMS pada MIDlet. Paket ini terdiri dari :

##### 1. *Interface*

###### a. *RecordComparator*

Mendefinisikan komparator atau pembanding ketika melakukan komparasi atau perbandingan antara dua *record* dalam RMS.

###### b. *RecordEnumerator*

Merepresentasikan *enumerator record* yang bersifat *bidirectional*.

###### c. *RecordFilter*

Mendefinisikan filter yang digunakan untuk menganalisis sebuah *record* pada *RecordStore*, apakah memenuhi suatu kriteria tertentu sesuai dengan filter yang diberikan atau tidak.

###### d. *RecordListener*

Mendeteksi *record* yang ditambahkan, diubah, atau dihapus dalam sebuah *RecordStore*.

## 2. Kelas

*RecordStore* merupakan kelas utama yang digunakan untuk membuat obyek *RecordStore*.

## 3. Exception

### a. *InvalidRecordIDException*

Jika ditemukan *recorded* yang tidak *valid*, eksepsi ini akan dilemparkan.

### b. *RecordStoreException*

Eksepsi umum yang digunakan jika terjadi kesalahan pada manajemen obyek *RecordStore*.

### c. *RecordStoreFullException*

Jika tempat penyimpanan *RecordStore* sudah penuh, eksepsi ini akan dilemparkan ke sistem.

### d. *RecordStoreNotFoundException*

Jika tempat penyimpanan *RecordStore* tidak ditemukan, eksepsi ini akan dilemparkan ke sistem.

### e. *RecordStoreNotOpenException*

Jika tempat penyimpanan *RecordStore* tidak dibuka (*opened*), eksepsi ini akan dilemparkan ke sistem.

## 2.5 Sistem Pakar

Sistem pakar adalah sistem berbasis komputer yang menggunakan pengetahuan, fakta, dan teknik penalaran dalam memecahkan masalah yang



biasanya hanya dapat dipecahkan oleh seorang pakar dalam bidang tertentu (Martin dan Oxman, 1988) [KUS06].

Sistem pakar dibuat pada wilayah pengetahuan tertentu untuk suatu kepakaran tertentu yang mendekati kemampuan manusia di salah satu bidang. Sistem pakar mencoba mencari solusi yang memuaskan sebagaimana dilakukan seorang pakar. Selain itu sistem pakar juga dapat memberikan penjelasan terhadap langkah yang diambil dan memberikan alasan atas saran dan kesimpulan yang ditemukan.

### 2.5.1 Ciri-ciri sistem pakar

Ciri-ciri sistem pakar adalah sebagai berikut [KUS06]:

1. Terbatas pada bidang yang spesifik, contoh MYCIN hanya dapat digunakan untuk melakukan diagnosis dan terapi terhadap penyakit meningitis dan infeksi bacremia.
2. Dapat memberikan penalaran untuk data data yang tidak pasti.
3. Dapat mengemukakan rangkaian alasan yang diberikannya dengan cara yang dapat dipahami.
4. Berdasarkan pada kaidah/rule tertentu.
5. Dirancang untuk dapat dikembangkan secara bertahap.
6. Keluarannya bersifat anjuran atau saran.

Komponen sistem pakar terbagi menjadi empat bagian, yaitu [FAH04]:

1. *Knowledge Base* (Basis Pengetahuan)

Knowledge Base merupakan inti dari program sistem pakar karena basis pengetahuan itu merupakan presentasi pengetahuan atau knowledge

representation basis pengetahuan adalah sebuah basis data yang menyimpan aturan-aturan tentang suatu domain *knowledge*/pengetahuan tertentu. Basis pengetahuan ini terdiri dari kumpulan objek beserta aturan dan atributnya (sifat atau cirinya). Contoh : If hewan merupakan sayap dan bertelur then hewan jenis burung.

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

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

## 3. *Inference Engine* (Mesin Penalaran)

*Inference Engine* adalah bagian yang menyediakan mekanisme fungsi berfikir dan pola-pola penalaran sistem yang digunakan oleh seorang pakar.

- Mekanisme ini akan menganalisa masalah tertentu dan selanjutnya akan mencari jawaban atau kesimpulan yang terbaik.
- Mesin ini akan dimulai pelacakannya dengan mencocokkan kaidah-kaidah dalam basis pengetahuan dengan fakta-fakta yang ada dalam basis data.

Dua teknik *Inference*, yaitu:

### a. *Backward Chaining* (Runut Balik)

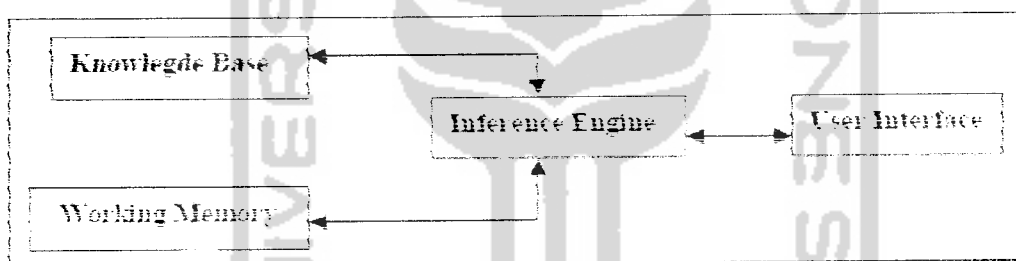
Melalui penalarannya dari sekumpulan hipotesis menuju fakta-fakta yang mendukung tersebut, jadi proses pelacakan berjalan mundur dimulai dengan menentukan kesimpulan yang akan dicari baru kemudian fakta-fakta pembangun kesimpulan atau *a Goal Driven*.

#### b. *Forward Chaining* (Runut Maju)

*Forward Chaining* merupakan kebalikan dari *Backward Chaining* yaitu mulai dari kumpulan data menuju kesimpulan. Suatu kasus kesimpulannya dibangun berdasarkan fakta-fakta yang telah diketahui atau data driven.

#### 4. *User Interface* (Antarmuka Pemakai)

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



Gambar 2.3 Komponen Utama sistem pakar

#### 2.5.2 Kategori Masalah Dari Sistem Pakar

Masalah-masalah yang dapat diselesaikan dengan sistem pakar, di antaranya [KUS06]:

1. Interpretasi – membuat kesimpulan atau deskripsi dari sekumpulan data mentah.
2. Prediksi – memproyeksikan akibat-akibat yang dimungkinkan dari situasi-situasi tertentu.

3. *Diagnosis* – menentukan sebab malfungsi dalam situasi kompleks yang didasarkan pada gejala-gejala yang teramati.
4. *Desain* – menentukan konfigurasi komponen-komponen sistem yang cocok dengan tujuan-tujuan kinerja tertentu yang memenuhi kendala-kendala tertentu.
5. *Perencanaan* – merencanakan serangkaian tindakan yang akan dapat mencapai sejumlah tujuan dengan kondisi awal tertentu.
6. *Debugging dan Repair* – menentukan dan menginterpretasikan cara-cara untuk mengatasi malfungsi.
7. *Instruksi* – mendeteksi dan mengoreksi defisiensi dalam pemahaman domain subyek.
8. *Pengendalian* – mengatur tingkah laku suatu environment yang kompleks.
9. *Selection* – mengidentifikasi pilihan terbaik dari sekumpulan (*list*) kemungkinan.
10. *Simulation* – pemodelan interaksi antara komponen-komponen system.
11. *Monitoring* – membandingkan hasil pengamatan dengan kondisi yang diharapkan.

### 2.5.3 Representasi Pengetahuan

Agar pengetahuan dapat digunakan dalam sistem, pengetahuan harus direpresentasikan dalam format tertentu yang kemudian dihimpun dalam suatu basis pengetahuan. Cara sistem pakar merepresentasikan pengetahuan akan mempengaruhi perkembangan, efisiensi, dan perbaikan sistem [KUS06].

Representasi pengetahuan merupakan metode yang digunakan untuk mengkodekan pengetahuan dalam sebuah sistem pakar yang berbasis pengetahuan. Representasi pengetahuan ini dimaksudkan untuk menangkap sifat-sifat penting problema dan membuat informasi itu dapat diakses oleh prosedur pemecahan problema.

Bahasa representasi harus dapat membuat seseorang pemrogram mampu mengekspresikan pengetahuan yang diperlukan untuk mendapat solusi problema, dapat diterjemahkan kedalam bahasa pemrograman, dan dapat disimpan. Harus dirancang agar fakta-fakta dan pengetahuan lain yang terkandung di dalamnya dapat digunakan untuk penalaran.

Beberapa model representasi pengetahuan yang penting, diantaranya adalah[KUS06]:

1. Logika
2. Jaringan Semantik
3. Object-Attribute-Value (OAV)
4. Bingkai (Frame)
5. Kaidah Produksi

Kaidah menyediakan cara formal untuk merepresentasikan rekomendasi, arahan, atau strategi. Kaidah produksi dituliskan dalam bentuk jika-maka (*if-then*). Kaidah *if-then* menghubungkan anteseden dengan konsekuensi yang diakibatkannya. Berbagai struktur kaidah *if-then* yang menghubungkan objek atau atribut yang diakibatkannya. Berbagai struktur kaidah *if-then* yang menghubungkan objek atau atribut adalah sebagai berikut:

JIKA premis MAKA konklusi

JIKA masukan MAKA keluaran

JIKA kondisi MAKA tindakan

JIKA anteseden MAKA konsekuen

JIKA data MAKA hasil

JIKA tindakan MAKA tujuan

Premis mengacu pada fakta yang harus benar sebelum konklusi tertentu dapat diperoleh. Masukan mengacu pada data yang harus tersedia sebelum keluaran dapat diperoleh. Kondisi mengacu pada keadaan yang harus berlaku sebelum tindakan dapat diambil. Anteseden mengacu pada situasi yang terjadi sebelum konsekuensi dapat diamati. Data mengacu pada kegiatan yang harus dilakukan sebelum hasil dapat diharapkan. Tindakan mengacu pada kegiatan yang harus dilakukan sebelum hasil dapat diharapkan.

Kaidah dapat diklasifikasikan menjadi dua, yaitu kaidah derajat pertama (*first order rule*) dan kaidah meta (*meta rule*). Kaidah derajat pertama adalah kaidah sederhana yang terdiri dari anteseden dan konsekuen. Misalnya:

JIKA bersin-bersin dan pusing MAKA terserang flu

Kaidah meta adalah kaidah yang anteseden atau konsekuennya mengandung tentang kaidah yang lain. Misalnya:

Aturan 1:

JIKA mengalami kehilangan kesadaran yang berlangsung sangat singkat, sehingga aktifitas yang sedang berjalan terhenti

DAN terkadang disertai dengan mata yang menatap kosong atau gerakan mioklonik dari sekelompok otot mata atau wajah, otomatisme, kehilangan tonus otot (sehingga barang yang dipegang bisa terjatuh atau bila sedang berdisri bisa terjatuh)

DAN serangan berakhir dengan diikuti dengan pulihnya kesadaran

DAN berlangsung beberapa detik sampai setengah menit, dan dapat berlangsung puluhan kali dalam sehari

MAKA mengalami tipe sawan lena

Aturan 2:

JIKA tipe sawan umum

ATAU tipe sawan mioklonik

ATAU tipe sawan lena

ATAU tipe sawan tonik-klonik

DAN EEG tidak menunjukkan kelainan fokal

DAN penyebab tidak diketahui

DAN awitan berhubungan dengan usia

MAKA terkena epilepsy idiopatik umum

Dalam contoh diatas, konsekuen dari aturan 1, yaitu mengalami sawan lena, menjadi anteseden bagi aturan 2.

#### 2.5.4 Faktor Kepastian

Faktor kepastian (*certainty factor*) menunjukkan ukuran kepastian terhadap suatu fakta atau aturan. Notasi Faktor Kepastian : [KUS06]

$$CF[h,e] = MB[h,e] - MD[h,e]$$

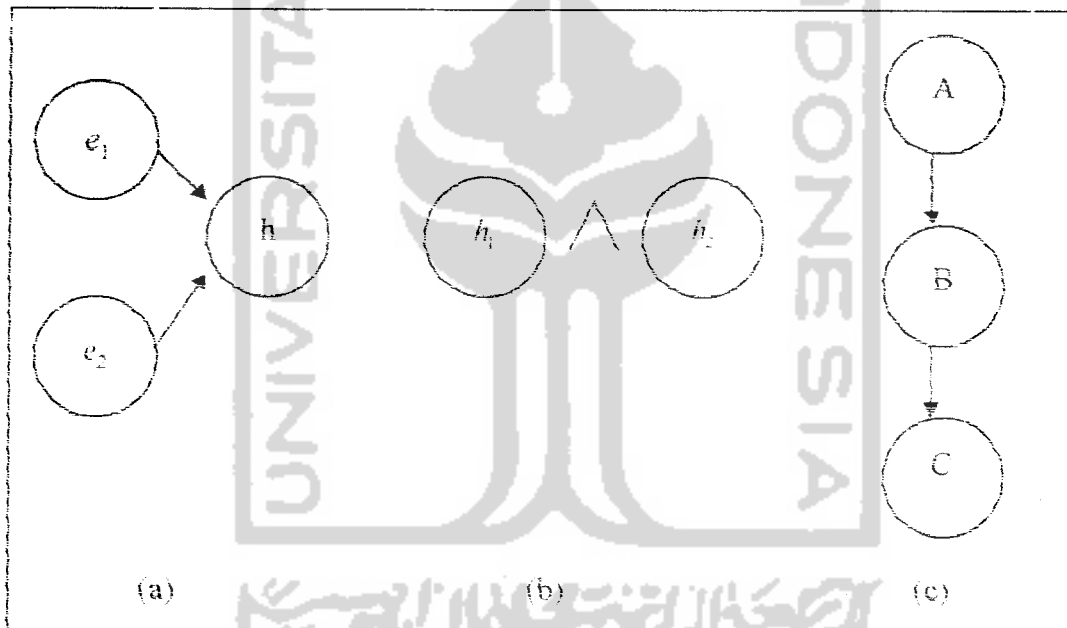
Dengan:

$CF[h,e]$  = faktor kepastian

$MB[h,e]$  = ukuran kepercayaan terhadap hipotesis  $h$ , jika diberikan *evidence*  $e$  (antara 0 dan 1)

$MD[h,e]$  = ukuran ketidakpercayaan terhadap *evidence*  $h$ , jika diberikan *evidence*  $e$  (antara 0 dan 1)

Ada 3 hal yang mungkin terjadi dengan menggunakan faktor kepastian :



Gambar 2.3 Kombinasi aturan ketidakpastian

- i. Beberapa *evidence* dikombinasikan untuk menentukan CF dari suatu hipotesis (gambar 2.3a). Jika  $e_1$  dan  $e_2$  adalah observasi, maka :

$$MB[h, e_1 \wedge e_2] = \begin{cases} 0 & \text{jika } MB[h, e_1] = 0 \text{ atau } MB[h, e_2] = 0 \\ MB[h, e_1] + MB[h, e_2] \cdot (1 - MB[h, e_1]) & \text{jika } MB[h, e_1] > 0 \text{ dan } MB[h, e_2] > 0 \end{cases}$$

lainnya



$$MD[h, e_1 \wedge e_2] = \begin{cases} 0 & \\ MD[h, e_1] + MD[h, e_2] \cdot (1 - MD[h, e_1]) & MD[h, e_1 \wedge e_2] = 1 \end{cases}$$

lainnya

2. CF dihitung dari kombinasi beberapa hipotesis (gambar 2.3b). Jika  $h_1$  dan  $h_2$  adalah hipotesis, maka :

$$MB[h_1 \wedge h_2, e] = \min(MB[h_1, e], MB[h_2, e])$$

$$MB[h_1 \vee h_2, e] = \max(MB[h_1, e], MB[h_2, e])$$

$$MD[h_1 \wedge h_2, e] = \min(MD[h_1, e], MD[h_2, e])$$

$$MD[h_1 \vee h_2, e] = \max(MD[h_1, e], MD[h_2, e])$$

3. Beberapa aturan saling bergandengan, ketidakpastian dari suatu aturan menjadi input untuk aturan yang lainnya (gambar 2.3c), maka:

$$MB[h, s] = MB^*[h, s] * \max(0, CF[s, e])$$

Dengan  $MB^*[h, s]$  adalah ukuran kepercayaan  $h$  berdasarkan keyakinan penuh terhadap validitas  $s$ .

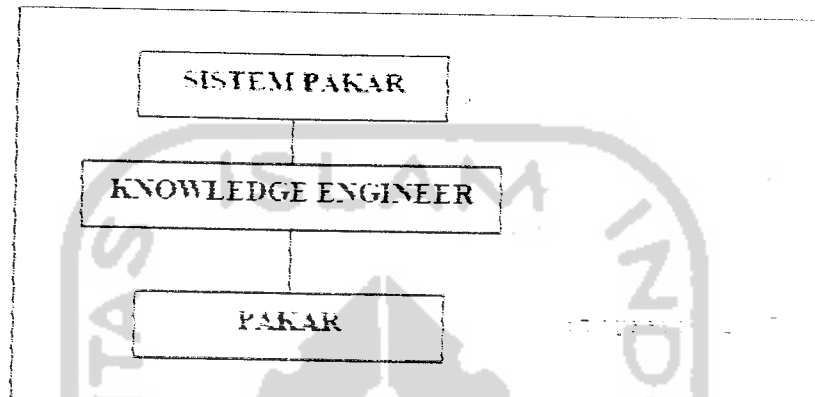
### 2.5.5 Sistem Kerja Pakar

Menurut Staugard (1987) sistem kerja pakar terbagi dalam tiga modul yaitu [FAH04]:

1. Modul Penerimaan Pengetahuan

Untuk mendapatkan pengetahuan sistem pakar dilakukan proses penerimaan pengetahuan. Proses ini dilakukan melalui interaksi dengan

pakar penerimaan pengetahuan dilakukan dengan bantuan *Knowledge Engineer* (KE), yaitu seorang spesialis sistem yang menterjemahkan pengetahuan yang dimiliki seorang pakar menjadi pengetahuan yang akan tersimpan dalam basis pengetahuan pada sebuah sistem pakar.



**Gambar 2.4** Modul Penerimaan Pengetahuan

## 2. Modul Konsultasi

Sistem pakar pada modul konsultasi apabila sistem memberikan konsultasi berupa jawaban atas permasalahan yang diajukan oleh pemakai pada modul ini pemakai yang awam berinteraksi dengan sistem dengan memasukkan data dan jawaban-jawaban pertanyaan sistem. Data yang dimasukkan oleh pemakai ditempatkan dalam database sistem dan kemudian diakses oleh pembangkit inference untuk mendapatkan kesimpulan.

## 3. Modul Penjelasan

Modul Penjelasan adalah menjelaskan proses pengambilan keputusan yang dilakukan oleh sistem.

## 2.6 Identifikasi Kerusakan Komputer

Masalah yang ditimbulkan komputer kadangkala merupakan kerusakan kecil yang untuk memperbaikinya tidak memerlukan pengetahuan yang tinggi mengenai komputer, mungkin cukup dengan seseorang yang mempunyai pengetahuan dasar tentang komputer. Tetapi terkadang kita temui kerusakan-kerusakan yang membutuhkan pengetahuan yang tinggi tentang sistem komputer dan komponen-komponennya untuk memperbaiki kerusakan tersebut. Untuk itulah kita membutuhkan teknisi komputer.

Pemeriksaan kerusakan komputer merupakan bentuk penelusuran terhadap kesalahan yang terjadi, dapat digolongkan menjadi beberapa kategori sebagai berikut[YUH05]:

### 1. Kategori Pemeriksaan Visual

Pemeriksaan visual merupakan tahap awal pemeriksaan, yaitu pemeriksaan beberapa masalah yang dapat ditangkap melalui pandangan mata. Beberapa contoh pemeriksaan visual antara lain:

- a. Pemeriksaan sambungan-sambungan kabel apakah terpasang dengan benar, seperti kabel *power* dan kabel *data*.
- b. Pemeriksaan konfigurasi "jumper", seperti jumper pada motherboard, harddisk, CD ROM.
- c. Pemeriksaan pemasangan perangkat keras, seperti RAM, kartu grafis, kartu audio, dan sebagainya.

### 2. Kategori Pemeriksaan Bunyi

Pemeriksaan bunyi memerlukan pemeriksaan komponen dengan mendengarkan bunyi-bunyi yang merupakan kode akan kondisi komputer.

Beberapa contoh pemeriksaan bunyi antara lain:

- a. Bunyi mekanikal dapat didengar dari putaran *disk*, putaran kipas, gerakan *head printer* dan lain-lain.
  - b. Pendeteksian bunyi *beep* yang muncul ketika *booting*. Bunyi *beep* yang ditimbulkan BIOS, mempunyai arti berbeda-beda.
3. Kategori Pemeriksaan Kondisi

Pemeriksaan ini memerlukan interaksi yang lebih khusus seperti misalnya menyentuh chip untuk mengetahui suhunya. CPU yang mempunyai panas berlebihan tentu akan menimbulkan masalah. Jika CPU memiliki chip yang besar seperti EPROM, seharusnya memiliki suhu yang hangat saja. Chip keramik lebih panas daripada yang terbuat dari plastic. Jika terlalu dingin mungkin menunjukkan chip tersebut mati.

4. Kategori Pemeriksaan Komponen Elektronik

Pemeriksaan komponen elektronik adalah cara terakhir jika semua langkah yang dilakukan tidak mampu mendeteksi dan menyelesaikan masalah. Pemeriksaan ini lebih sulit dan memerlukan peralatan khusus, dokumen dan gambar sistem, serta pengetahuan yang mendalam mengenai sistem elektronik.

## BAB III

### ANALISIS KEBUTUHAN PERANGKAT LUNAK

#### 3.1 Metode Analisis

Analisis kebutuhan perangkat lunak dilakukan untuk mengetahui semua permasalahan serta kebutuhan yang diperlukan dalam pengembangan aplikasi. Analisis dilakukan dengan mencari dan menentukan permasalahan yang dihadapi serta semua kebutuhan seperti kebutuhan masukan sistem, kebutuhan keluaran sistem, antarmuka sistem, dan fungsi-fungsi yang dibutuhkan.

Metode analisis yang digunakan dalam mengembangkan perangkat lunak ini adalah metode analisis berorientasi objek dengan menggunakan UML (*Unified Modelling Language*) sebagai alat bantu.

#### 3.2 Hasil Analisis

Hasil analisis yang diperoleh dalam penelitian ini adalah proses-proses yang disertai dengan masukan dan keluaran sistem. Hasil analisis ini merupakan pemilihan kebutuhan sistem yang harus diwujudkan dalam perangkat lunak, yang meliputi fungsi-fungsi yang dibutuhkan dan antarmuka yang *user friendly*.

##### 3.2.1 Kebutuhan Masukan

Masukan dari sistem pakar untuk mendiagnosa kerusakan komputer pada *mobile device* ini adalah sebagai berikut:

### 1. *Input User*

*Input user* disini berupa masukan yang diberikan oleh pengguna aplikasi yang berupa:

- a. Pilihan piranti komputer yang mengalami kerusakan (piranti pemrosesan, piranti *input*, piranti *output*).
- b. Pilihan gejala kerusakan pada masing-masing piranti komputer yang telah disediakan dalam bentuk *ChoiceGroup*.

### 2. *Input Pakar*

*Input pakar* pada aplikasi ini berupa masukan yang diberikan oleh seorang pakar sebagai basis pengetahuan yang terdiri dari:

- a. Data gejala kerusakan

*Input* ini berupa gejala-gejala kerusakan yang sering terjadi pada piranti komputer. Faktor-faktor penyebab yang terjadi dalam kasus kerusakan komputer diubah menjadi gejala-gejala sehingga memudahkan mengetahui jenis kerusakan yang lebih spesifik.

- b. Data kemungkinan kerusakan

*Input* ini berupa kemungkinan kerusakan yang terjadi pada piranti komputer beserta dengan solusi perbaikannya.

- c. Data basis pengetahuan

*Input* ini berupa basis pengetahuan atau aturan-aturan untuk proses diagnosa.

Data yang dimasukkan oleh pakar tersebut akan digunakan oleh sistem pakar untuk penelusuran diagnosa kerusakan komputer.

### 3.2.2 Kebutuhan Keluaran

Keluaran dari sistem ini menampilkan kemungkinan kerusakan komputer, nilai kepastian berdasarkan metode *Certainty Factor*, dan solusi perbaikannya.

Kebutuhan data keluaran yang digunakan dalam penelitian ini diperoleh dari:

1. Pengumpulan data dengan cara mempelajari buku-buku tentang kerusakan pada komputer.
2. Wawancara atau konsultasi dengan pakar teknisi komputer yang dapat menentukan nilai kemungkinan pada kerusakan komputer.

Hasil keluaran diagnosa kerusakan komputer ini diberikan setelah *user* melakukan konsultasi dengan memilih piranti komputer yang rusak dan kemudian mengisi gejala-gejala kerusakan komputer untuk masing-masing piranti komputer yang telah dipilih.

### 3.2.3 Kebutuhan Proses

Kebutuhan proses dalam sistem pakar ini meliputi :

- a. Proses untuk memetakan data (*Data Mapping*) pada *database* RMS agar data bisa dikelola dengan baik (simpan data, ubah data, hapus data, dan ambil data) dan bisa digunakan untuk penelusuran data untuk proses diagnosa.
- b. Proses diagnosa, proses ini dilakukan dengan menggunakan penalaran *backward chaining* dan untuk penghitungannya menggunakan metode *certainty factor*. Data untuk proses ini berasal dari gejala-gejala yang telah

diisi pada formGejala dan menggunakan basis pengetahuan yang telah disimpan dalam *RecordStore* bernama "BP".

### 3.2.4 Analisis Antarmuka Sistem

Antarmuka pemakai atau yang lebih dikenal dengan *user interface* adalah bagian penghubung antara program sistem pakar dengan pemakai sistem atau *user*. Pada bagian ini akan terjadi interaksi atau komunikasi antara sistem dengan *user*. Interaksi yang diberikan sistem saat *user* melakukan konsultasi kerusakan komputer adalah berbentuk pilihan piranti komputer yang mengalami kerusakan dan pilihan gejala kerusakan komputer yang ditampilkan dalam menu pilihan *ChoiceGroup*. Setelah proses diagnosaselesai maka sistem akan menampilkan hasilnya kepada *user* berupa informasi kemungkinan kerusakan, nilai *CF*-nya dan disertai dengan solusi perbaikannya.

Antarmuka pemakai juga menyediakan fasilitas untuk akuisisi pengetahuan, yaitu fasilitas untuk dialog antara pakar dengan sistem. Fasilitas ini berupa *input* untuk olah data (simpan, ubah, hapus), yaitu data kerusakan, data gejala, dan data basis pengetahuan.

### 3.2.5 Perangkat lunak yang dibutuhkan

Untuk mengembangkan aplikasi ini digunakan perangkat lunak, antara lain:

1. Microsoft Windows XP Professional, merupakan sistem operasi yang digunakan dalam pembuatan aplikasi.



2. Java 2 SDK 1.5 (JDK 1.5), merupakan bahasa pemrograman yang digunakan untuk mengembangkan aplikasi.
3. NetBeans IDE (Integrated Development Environment) 5.0, untuk mendesain program dan penulisan kode program.
4. NetBeans Mobility 5.0 for Windows sebagai paket untuk pengembangan aplikasi *mobile*.

### 3.2.6 Perangkat keras yang dibutuhkan

Perangkat keras yang digunakan adalah sebagai berikut :

1. Untuk pembuatan aplikasi ini, penulis menggunakan Laptop dengan spesifikasi:
  - a. *Processor* Intel Pentium M 1.5 GHz.
  - b. *Monitor* XGA TFT LCD
  - c. *Memory* DDR2 256 MB
  - d. *Harddisk* 60 GB
2. Untuk uji coba langsung, aplikasi sistem pakar ini memerlukan *mobile device* yang mendukung teknologi java. Untuk pengujian ini, penulis menggunakan Nokia 6600 dengan spesifikasi teknologi java sebagai berikut :
  - a. CLDC 1.0
  - b. MIDP 2.0

## **BAB IV**

### **PERANCANGAN PERANGKAT LUNAK**

#### **4.1 Metode Perancangan**

Metode perancangan yang digunakan dalam membangun *Aplikasi Sistem Pakar Untuk Diagnosa Kerusakan Komputer pada Mobile Devices dengan Menggunakan Teknologi J2ME* ini adalah metode perancangan berorientasi objek dengan menggunakan *tool* UML (*Unified Modelling Language*). UML digunakan karena merupakan bahasa yang *process-independent*, yaitu bahasa yang saling bebas terhadap proses yang ada di dalamnya. Tahapan perancangan yang dibahas merupakan perancangan yang akan menghasilkan kebutuhan sistem aplikasi, hal ini mencakup perancangan desain.

#### **4.2 Hasil Perancangan Perangkat Lunak**

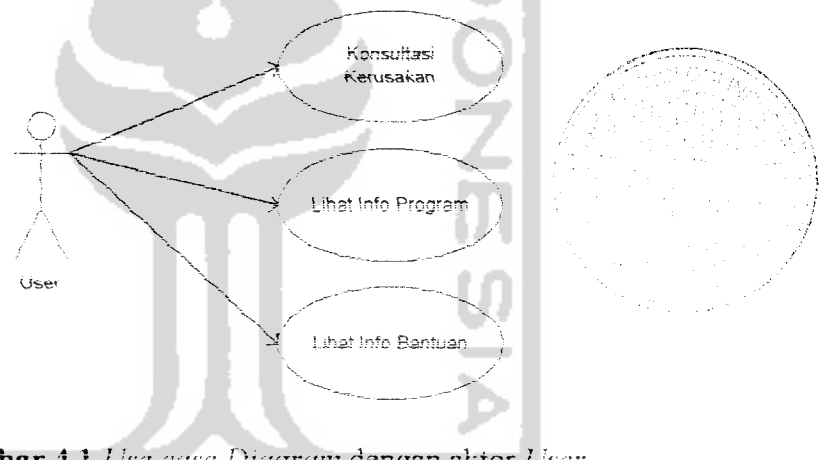
Hasil perancangan perangkat lunak ini adalah berupa gambaran umum aplikasi secara keseluruhan yang akan menjelaskan proses implementasi sistem. Pada penelitian ini, rancangan proses implementasi sistem berupa *Use Case Diagram*, *Class Diagram*, *Sequence Diagram*, dan perancangan antarmuka.

##### **4.2.1 Use Case Diagram**

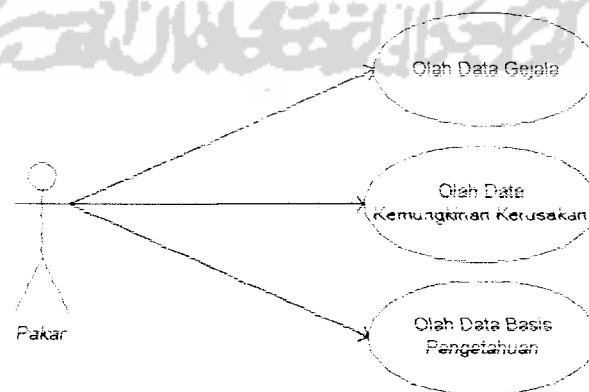
*Use Case Diagram* digunakan untuk mendeskripsikan apa yang seharusnya dilakukan oleh sistem dari sudut pandang pengguna. *Use Case Diagram* menyediakan cara untuk mendeskripsikan pandangan eksternal terhadap

sistem dan interaksi-interaksinya dengan dunia luar. Dalam *use case diagram* ada dua pihak yang saling berhubungan, yaitu aktor dan *use case*. Dengan diagram ini dapat diketahui cakupan dari sistem, siapa saja aktor yang berperan dalam sistem dan interaksi antara aktor dengan elemen-elemen *use case* dalam sistem.

Pada rancangan aplikasi ini terdapat dua aktor, yaitu aktor *User* dan aktor *Pakar*. Adapun kedua aktor tersebut berinteraksi dengan sistem melalui fungsi-fungsi yang dimiliki oleh sistem. Hubungan antara aktor dengan *use case* dapat dilihat lebih jelas pada gambar 4.1 dan 4.2.



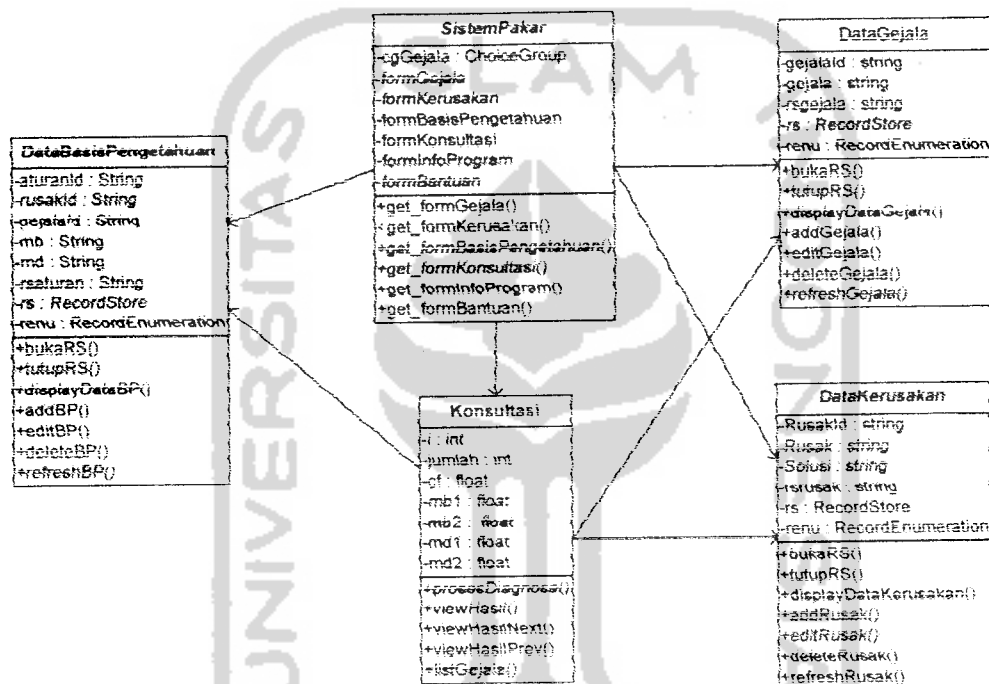
**Gambar 4.1** *Use case Diagram* dengan aktor *User*



**Gambar 4.2** *Use case Diagram* dengan aktor *Pakar*

#### 4.2.2 Class Diagram

Aplikasi yang menggunakan perancangan berorientasi obyek dapat diilustrasikan dalam struktur kelas-kelas dan hubungan antar kelas yang ada. Dalam UML digunakan notasi class diagram untuk menggambarkan hal tersebut. Lebih jelasnya dapat dilihat pada gambar 4.3.



Gambar 4.3 Class Diagram

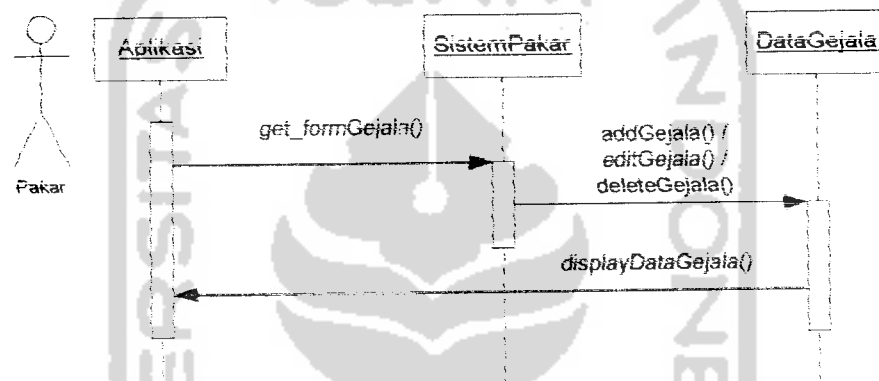
#### 4.2.3 Sequence Diagram

Sequence Diagram menggambarkan perilaku sistem secara dinamis dan memperlihatkan interaksi dari obyek-obyek. Interaksi antar obyek dapat disusun berdasarkan urutan waktu yang menunjukkan skenario dan urutan-urutan pertukaran data.

Dari tahapan analisis kebutuhan yang dilakukan sebelumnya maka dapat dibentuk beberapa *sequence diagram* untuk menunjukkan urutan-urutan proses dari masing-masing *use case*. Pada aplikasi sistem pakar untuk diagnosa kerusakan komputer ini terdapat beberapa *Sequence Diagram*, yaitu:

### 1. Olah Data Gejala

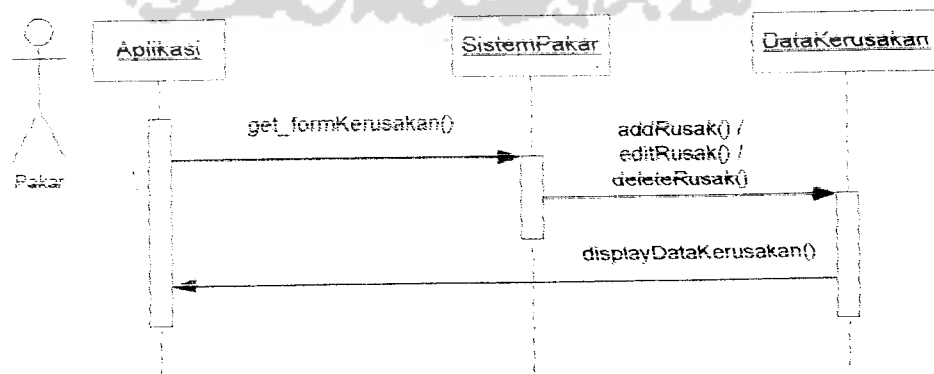
*Sequence Diagram* ini menunjukkan proses untuk olah data gejala.



**Gambar 4.4** *Sequence Diagram* untuk Olah Data Gejala

### 2. Olah Data Kerusakan

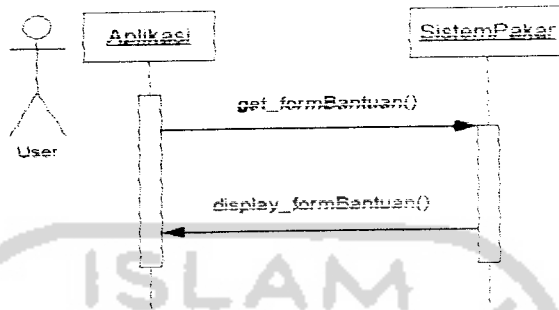
*Sequence Diagram* ini menunjukkan proses untuk olah data kerusakan.



**Gambar 4.5** *Sequence Diagram* untuk Olah Data Kerusakan

## 5. Lihat Bantuan

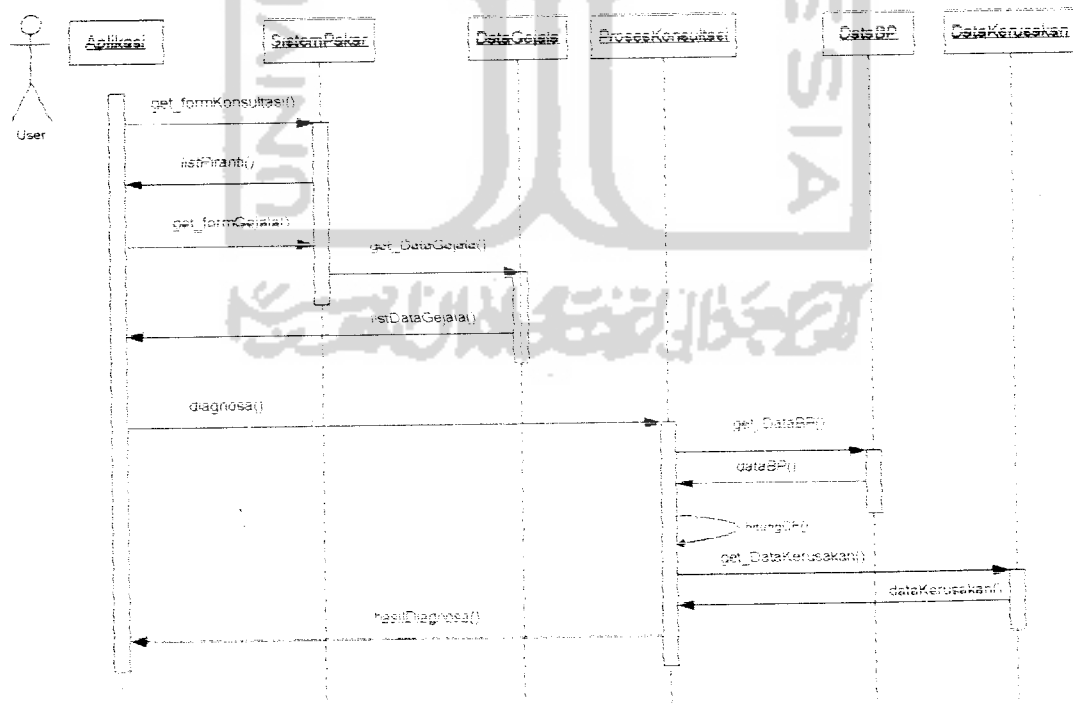
*Sequence Diagram* ini menunjukkan proses untuk melihat bantuan.



**Gambar 4.9** *Sequence Diagram* untuk Lihat Bantuan Program

## 6. Konsultasi

*Sequence Diagram* ini menunjukkan proses untuk konsultasi kerusakan komputer.



**Gambar 4.7** *Sequence Diagram* untuk Proses Konsultasi

#### 4.2.4 Perancangan Antarmuka

Perancangan antarmuka menggambarkan desain tampilan dari sistem, yaitu ilustrasi dari rancangan antarmuka terhadap sistem yang akan diaplikasikan. Desain antarmuka ditujukan bagi *user*, dimana antarmuka didesain sedemikian rupa untuk memudahkan penggunaan sistem aplikasi ini. Berikut ini desain antarmukanya :

##### 1. Halaman Muka

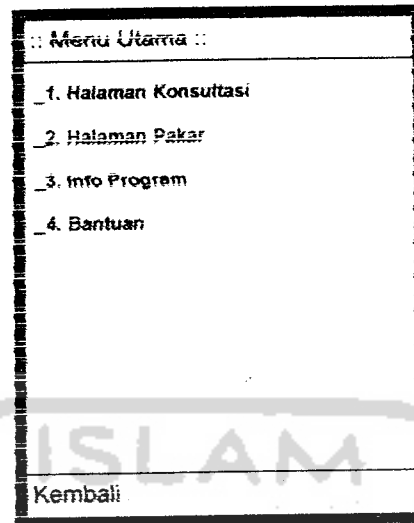
Begitu aplikasi dibuka maka halaman pertama yang muncul adalah halaman muka. Tombol “Keluar” untuk keluar dari aplikasi, sedangkan tombol “Masuk” untuk masuk ke Menu Utama.



Gambar 4.10 Halaman Muka

##### 2. Menu Utama

Pada halaman Menu Utama ini terdapat 4 macam pilihan menu, yaitu Halaman Konsultasi, Halaman Pakar, Info Program dan Bantuan. Tombol “Kembali” untuk kembali ke Halaman Muka.



**Gambar 4.11** Halaman Menu Utama

### 3. Halaman Konsultasi

Halaman Konsultasi ini berfungsi untuk proses konsultasi. Untuk melakukan proses konsultasi, *user* harus melakukan beberapa tindakan berikut:

- a. Membuka Halaman Konsultasi (lihat gambar 4.12), terdapat tiga pilihan piranti komputer yang harus dipilih.
- b. Setelah memilih piranti, maka akan ditampilkan Halaman Pilih Gejala Kerusakan (lihat gambar 4.13). Kemudian *user* harus memilih gejala-gejala kerusakan yang dialami dan tekan tombol "Diagnosa".
- c. Setelah itu akan muncul Halaman Hasil Diagnosa (lihat gambar 4.14) yang memberikan hasil berupa kemungkinan kerusakan, nilai CF-nya dan solusi perbaikannya. Untuk kembali ke Halaman Konsultasi, tekan tombol "Kembali".



**:: Halaman Konsultasi ::**

**Pilih Piranti Komputer Yang Rusak :**

- 1. Piranti Pemrosesan
- 2. Piranti Input
- 3. Piranti Output

Kembali

Gambar 4.12 Halama Konsultasi

**:: Pilih Gejala Kerusakan ::**

- Gejala 1
- Gejala 2
- Gejala 3

Batal      Diagnosa

Gambar 4.13 Halama Pilih Gejala Kerusakan

**Gambar 4.14** Halaman Hasil Diagnosa

#### 4. Halaman Pakar

Halaman Pakar ini berfungsi untuk menyimpan data pengetahuan seorang pakar. Untuk menyimpan pengetahuan pakar dalam basis data, maka disediakan antarmuka sebagai berikut:

- a. Halaman Pakar (lihat gambar 4.15), disini terdapat tiga pilihan piranti perangkat komputer untuk proses olah data. Setelah memilih piranti, akan ditampilkan Halaman Pilihan Olah Data.
- b. Halama Pilihan Olah Data (lihat gambar 4.16), terdapat tiga macam pilihan olah data, yaitu data gejala, data kerusakan, dan data basis pengetahuan. Pilih salah satu untuk menampilkan Halaman View Data.
- c. Halaman View Data ini ada tiga macam, yaitu Halaman View Data Gejala (lihat gambar 4.17), Halaman View Data Kerusakan (lihat gambar 4.19), dan Halaman View Data Basis Pengetahuan (lihat gambar 4.21). Halaman View Data ini untuk melihat data yang ada di basis data pakar dan terdapat

**:: Data Gejala ::**

Id Record : \_\_\_\_

Id Gejala : \_\_\_\_

Gejala : \_\_\_\_

Jumlah Record : \_\_\_\_

Menu
1 Data Selanjutnya
2 Data Sebelumnya
3 Tambah Data
4 Ubah Data
5 Hapus Data

Kembali Menu

Gambar 4.17 Halaman View Data Gejala

**:: Olah Data Gejala ::**

Id Gejala :

Gejala :

Batal Ok

Gambar 4.18 Halaman Olah Data Gejala

**:: Data Kerusakan ::**

**Id Record :** \_\_\_\_\_

**Id Kerusakan :** \_\_\_\_\_

**Kerusakan :** \_\_\_\_\_

**Solusi :** \_\_\_\_\_

**Jumlah Record :** \_\_\_\_\_

Menu
1 Data Selanjutnya
2 Data Sebelumnya
3 Tambah Data
4 Ubah Data
5 Hapus Data

**Kembali** **Menu**

**Gambar 4.19** Halaman View Data Kerusakan

**:: Olah Data Gejala ::**

**Id Kerusakan :**

**Kerusakan :**

**Solusi :**

**Batal** **Ok**

**Gambar 4.20** Halaman Olah Data Kerusakan

**:: Data Basis Pengetahuan ::**

Id Record : \_\_\_\_\_

Id Aturan : \_\_\_\_\_

Id Kerusakan : \_\_\_\_\_

Id Gejala : \_\_\_\_\_

Nilai MB : \_\_\_\_\_

Nilai MD : \_\_\_\_\_

Jumlah Record : \_\_\_\_\_

Menu
1 Data Selanjutnya
2 Data Sebelumnya
3 Tambah Data
4 Ubah Data
5 Hapus Data

Kembali Menu

Gambar 4.21 Halaman View Data Basis Pengetahuan

**:: Olah Data Gejala ::**

Id Aturan : \_\_\_\_\_

Id Kerusakan : \_\_\_\_\_

Id Gejala : \_\_\_\_\_

Nilai MB : \_\_\_\_\_

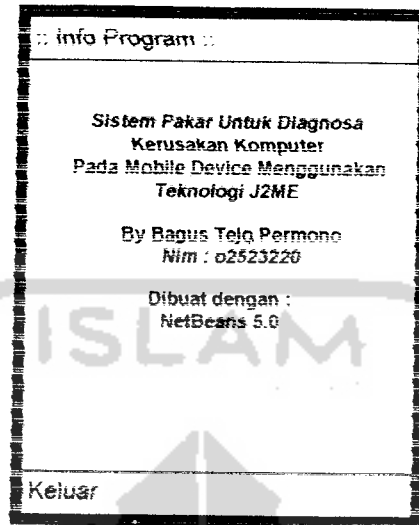
Nilai MD : \_\_\_\_\_

Batal OK

Gambar 4.22 Halaman Olah Data Basis Pengetahuan

### 5. Halaman Info Program

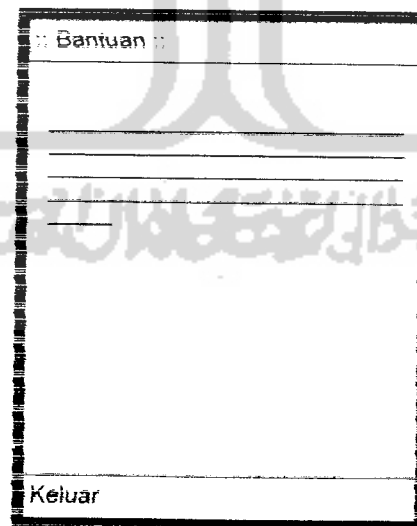
Halaman ini menunjukkan informasi program.



Gambar 4.23 Halaman Info Program

### 6. Halaman Bantuan Program

Halaman ini memberikan bantuan petunjuk penggunaan aplikasi sistem pakar ini.



Gambar 4.24 Halaman Bantuan Program

### 4.3 Perancangan Basis Pengetahuan

Representasi basis pengetahuan pada aplikasi ini adalah menggunakan kaidah produksi (*production rule*). Pada kaidah produksi ini pengetahuan direpresentasikan dengan menggunakan aturan berbentuk: IF-THEN. Bentuk ini digunakan apabila kita memiliki sejumlah pengetahuan pakar pada suatu permasalahan tertentu, dan si pakar dapat menyelesaikan masalah tersebut secara berurutan.

Adapun basis pengetahuan pada aplikasi ini diimplementasikan dalam bentuk tabel basis pengetahuan dengan struktur tabel sebagai berikut:

- |                 |
|-----------------|
| 1. Id aturan    |
| 2. Id gejala    |
| 3. Id kerusakan |
| 4. MB           |
| 5. MD           |

**Gambar 4.25** Struktur tabel basis pengetahuan

Pada struktur tabel basis pengetahuan tersebut diatas, memungkinkan sebuah kerusakan bisa memiliki beberapa gejala yang mempengaruhinya dengan nilai MB dan MD-nya masing-masing untuk setiap gejala yang mempengaruhinya tersebut atau bisa juga kebalikannya yaitu sebuah gejala memiliki beberapa kerusakan yang dipengaruhinya. Untuk lebih jelasnya dapat dilihat beberapa contoh implementasi basis pengetahuan dibawah ini ( data selengkapnya dapat dilihat pada data basis pengetahuan di halaman Lampiran).

- IF Indikator power on tidak menyala THEN Masalah kelistrikan dengan MB=0.8 MD=0.1
- IF Indikator power on tidak menyala THEN Masalah power supply dengan MB=0.9 MD=0.02
- IF Kabel power tidak terpasang benar THEN Masalah kelistrikan dengan MB=0.9 MD=0.1
- IF Kabel power terpasang benar THEN Masalah power supply dengan MB=0.9 MD=0.01
- IF Layar monitor gelap dan bunyi beep berulang THEN Hardisk bermasalah dengan MB=0.7 MD=0.05

**Tabel 4.1** Tabel identifikasi kerusakan dari contoh basis pengetahuan

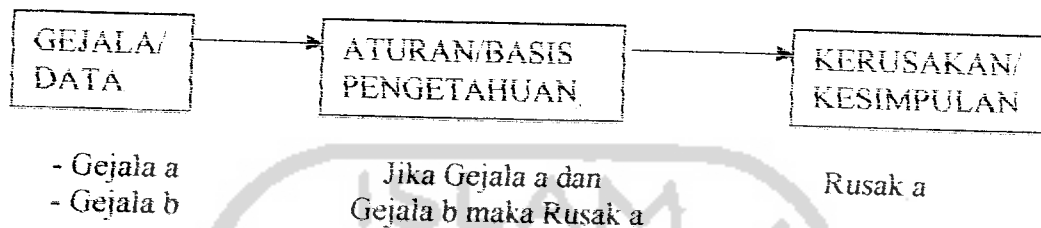
Gejala	Kerusakan		
	Masalah kelistrikan	Masalah power supply	Hardisk bermasalah
Indikator power on tidak menyala	√	√	
Kabel power tidak terpasang benar	√		
Kabel power terpasang benar		√	
Layar monitor gelap dan bunyi beep berulang			√
Tegangan listrik sering turun	√		

#### 4.4 Perancangan *Inference Engine*

*Inference Engine* adalah bagian yang menyediakan mekanisme fungsi berfikir dan pola-pola penalaran sistem yang digunakan oleh seorang pakar. Terdapat dua teknik penelusuran, yaitu *forward chaining* dan *backward chaining*. Dalam perancangan aplikasi ini, penulis memilih teknik *forward chaining* untuk mesin inferensinya.



*Forward Chaining* merupakan teknik penelusuran yang dimulai dari kumpulan data atau dalam aplikasi ini berupa gejala-gejala menuju kesimpulan yang berupa jenis kerusakan. Gambar 4.26 berikut ini menunjukkan bagaimana cara kerja teknik inferensi *forward chaining* pada aplikasi ini.



Gambar 4.26 Inferensi *forward chaining*

Berikut ini merupakan contoh inferensi dengan menggunakan metode inferensi *forward chaining* :

JIKA Indikator power on tidak menyala dengan  $MB=0.8$  dan  $MD=0.1$

DAN Kabel power tidak terpasang benar dengan  $MB=0.9$   $MD=0.1$

DAN Tegangan listrik sering turun dengan  $MB=0.7$   $MD=0.05$

MAKA Masalah kelistrikan

## **BAB V**

### **IMPLEMENTASI PERANGKAT LUNAK**

#### **5.1 Implementasi Secara Umum**

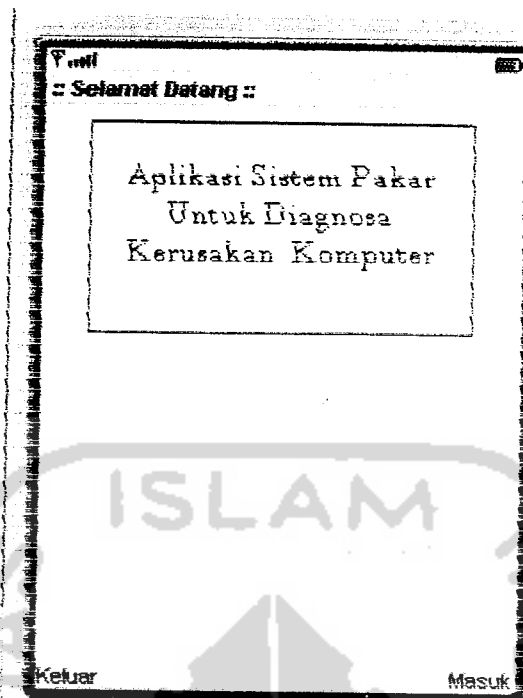
Implementasi sistem merupakan tahap dimana sistem mampu diaplikasikan dalam keadaan yang sesungguhnya. Dari implementasi ini akan diketahui apakah sistem yang dibuat dapat berjalan dengan baik atau tidak dan menghasilkan *output* yang sesuai dengan perancangan yang ada.

#### **5.2 Implementasi Antarmuka**

Tampilan antarmuka ini menggunakan *emulator platform* J2ME Wireless Toolkit 2.2 dengan *device* DefaultColorPhone pada NetBeans Mobility Pack. Tampilan pada *emulator* dapat berbeda-beda tergantung jenis dan versi *emulator* yang digunakan.

##### **5.2.1 Tampilan Halaman Muka**

Pada halaman muka terdapat dua buah tombol yaitu tombol "Masuk" untuk masuk ke halaman Menu Utama dan tombol "Keluar" yang digunakan untuk menutup aplikasi. Gambar 5.1 menunjukkan tampilan Halaman Muka aplikasi sistem pakar untuk diagnosa kerusakan komputer.

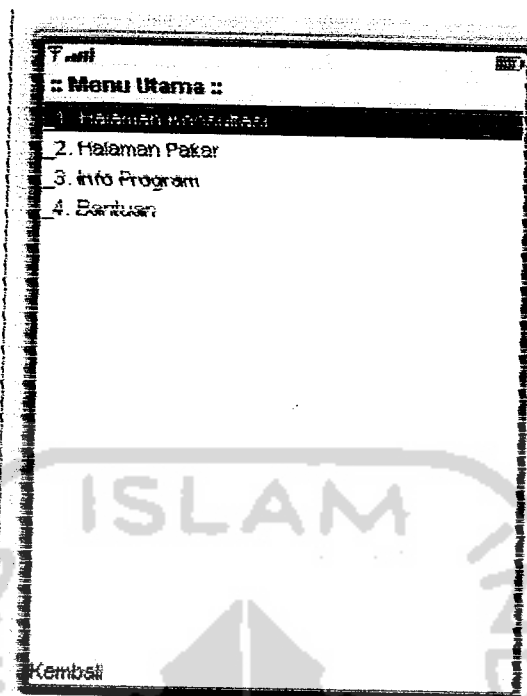


Gambar 5.1 Tampilan Halaman Muka

### 5.2.2 Tampilan Menu Utama

Pada tampilan Menu Utama terdapat empat pilihan menu, yaitu :

1. Halaman Konsultasi, merupakan antarmuka untuk melakukan proses konsultasi kerusakan komputer.
2. Halaman Pakar, merupakan antarmuka untuk menyimpan data pakar dalam sistem pakar.
3. Halaman Info Program, merupakan antarmuka untuk mengetahui informasi tentang aplikasi sistem pakar ini.
4. Halaman Bantuan, merupakan antarmuka untuk melihat petunjuk penggunaan program.



Gambar 5.2 Tampilan Menu Utama

### 5.2.3 Tampilan Halaman Konsultasi

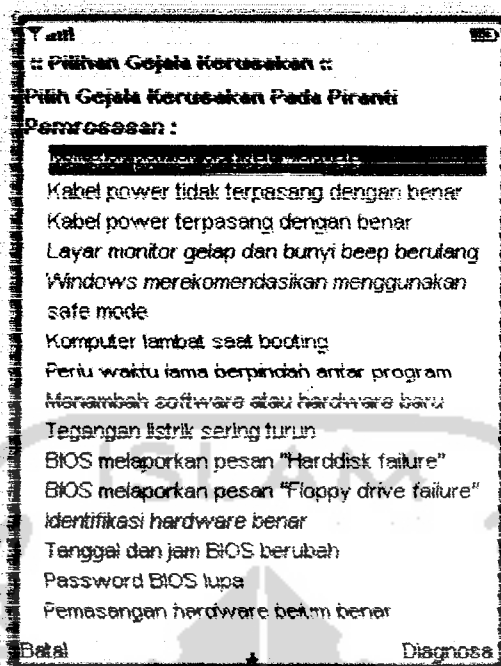
Halaman untuk proses konsultasi terdiri dari tiga buah halaman, yaitu:

1. Halaman pilih piranti komputer. Terdapat tiga buah pilihan piranti konsultasi. Pilih salah satu untuk menuju halaman pilih gejala. Lebih jelasnya dapat dilihat pada gambar 5.3.
2. Halaman pilih gejala kerusakan. Data gejala pada halaman ini berdasarkan data gejala pada masing-masing piranti yang dipilih pada halaman pilih piranti. Tombol "Diagnosa" digunakan untuk mendiagnosa kerusakan komputer berdasarkan gejala - gejala yang telah dipilih. Untuk lebih jelasnya dapat dilihat pada gambar 5.4.

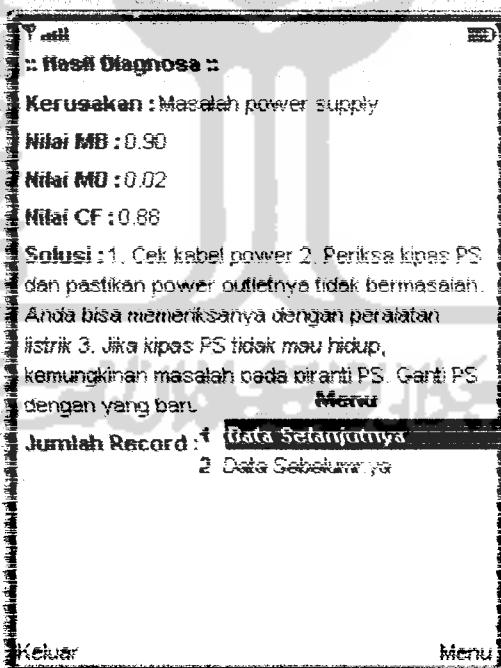
3. Halaman hasil konsultasi. Halaman ini menampilkan hasil dari proses diagnosa pada halaman pilih gejala. Tampilan halaman hasil konsultasi dapat dilihat pada gambar 5.5.



Gambar 5.3 Tampilan halaman pilih piranti konsultasi



Gambar 5.4 Tampilan halaman pilih gejala kerusakan



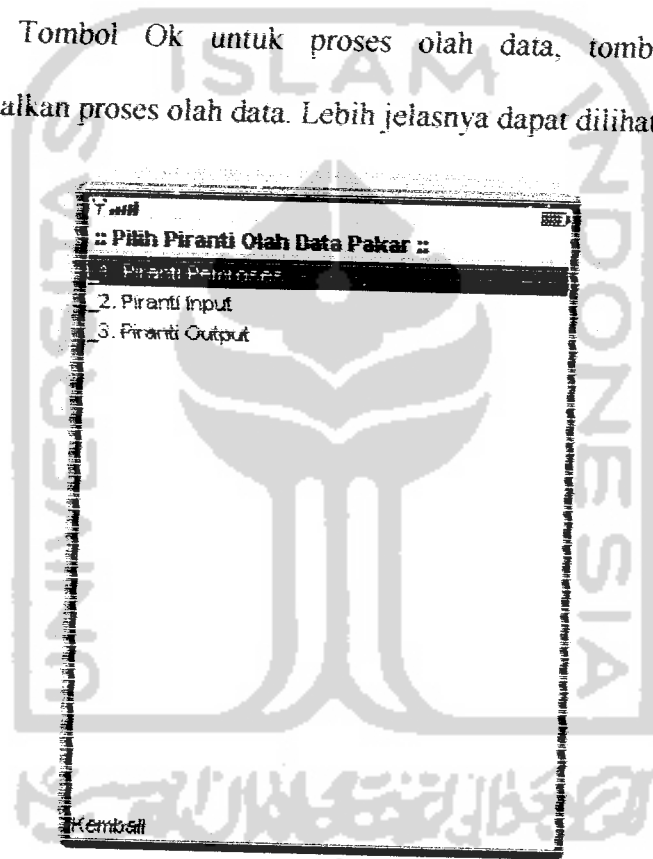
Gambar 5.5 Tampilan halaman hasil konsultasi

#### 5.2.4 Tampilan Halaman Pakar

Halaman pakar terdiri dari delapan bagian yaitu:

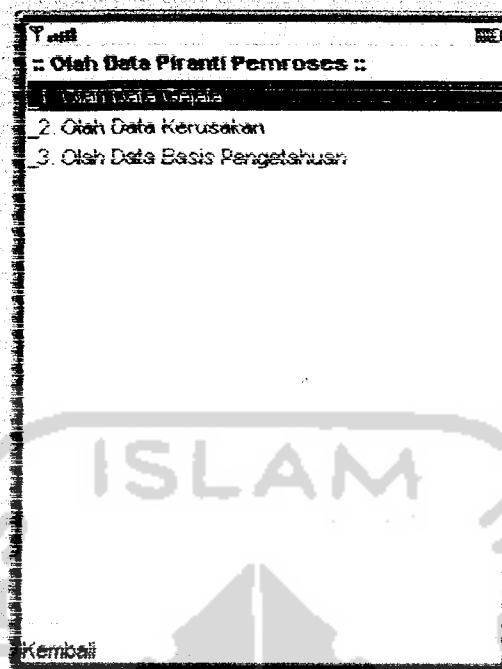
1. Halaman pilih piranti pakar. Terdapat tiga buah pilihan piranti, begitu salah satu dipilih akan ditampilkan halaman pilih jenis olah data. Lebih jelasnya dapat dilihat gambar 5.6.
2. Halaman pilih jenis olah data. Terdapat tiga buah pilihan olah data, yaitu gejala, kerusakan, dan basis pengetahuan. Lebih jelasnya dapat dilihat gambar 5.7.
3. Halaman data gejala. Halaman ini menampilkan data gejala yang telah di-input-kan ke basis data pakar. Terdapat lima buah menu pilihan untuk proses olah data. Lebih jelasnya dapat dilihat gambar 5.8.
4. Halaman olah data gejala. Halaman ini muncul ketika menu pilihan tambah, ubah, atau hapus pada halaman data gejala dipilih. Tombol Ok untuk proses olah data, tombol batal untuk membatalkan proses olah data. Lebih jelasnya dapat dilihat gambar 5.9.
5. Halaman data kerusakan. Halaman ini menampilkan data kerusakan yang telah di-input-kan ke basis data pakar. Terdapat lima buah menu pilihan untuk proses olah data. Lebih jelasnya dapat dilihat gambar 5.10.
6. Halaman olah data kerusakan. Halaman ini muncul ketika menu pilihan tambah, ubah, atau hapus pada halaman data kerusakan dipilih. Tombol Ok untuk proses olah data, tombol batal untuk membatalkan proses olah data. Lebih jelasnya dapat dilihat gambar 5.11.

7. Halaman data basis pengetahuan. Halaman ini menampilkan data basis pengetahuan yang telah di-input-kan ke basis data pakar. Terdapat lima buah menu pilihan untuk proses olah data. Lebih jelasnya dapat dilihat gambar 5.12.
8. Halaman olah data basis pengetahuan. Halaman ini muncul ketika menu pilihan tambah, ubah, atau hapus pada halaman data basis pengetahuan dipilih. Tombol Ok untuk proses olah data, tombol batal untuk membatalkan proses olah data. Lebih jelasnya dapat dilihat gambar 5.13.

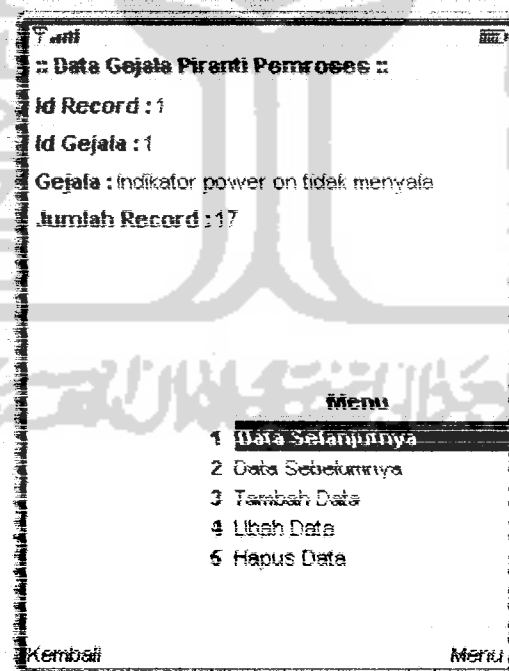


Gambar 5.6 Tampilan halaman pilih piranti pakar





Gambar 5.7 Tampilan halaman pilih jenis olah data



Gambar 5.8 Tampilan halaman data gejala

The screenshot shows a mobile application interface with a title bar containing 'T' and 'ABC'. The main content area is titled 'Olah Data Gejala ::' and contains the following text: 'Id Gejala :' followed by an empty input field, and 'Gejala :' followed by a larger empty text area. At the bottom, there are two buttons: 'Batal' on the left and 'OK' on the right.

Gambar 5.9 Tampilan halaman olah data gejala

The screenshot shows a mobile application interface with a title bar containing 'T' and 'ABC'. The main content area is titled 'Data Kerusakan Piranti Pemroses ::' and contains the following text: 'Id Record : 1', 'Id Kerusakan : 1', 'Kerusakan : Masalah kelistrikan', 'Solusi : 1. Cek kabel power dan stabilizer, apakah berfungsi dengan baik 2. Jika PC tetap tidak mau hidup, ganti kabel power', and 'Jumlah Record : 4'. Below this text is a 'Menu' section with a list of options: '1 Data Selanjutnya', '2 Data Sebelumnya', '3 Tambah Data', '4 Ubah Data', and '5 Hapus Data'. At the bottom, there are two buttons: 'Kembali' on the left and 'Menu' on the right.

Gambar 5.10 Tampilan halaman data kerusakan

KAMI REC

== Olah Data Kerusakan ==

Id Kerusakan :

Kerusakan :

Solusi :

Batalan OK

Gambar 5.11 Tampilan halaman olah data kerusakan

KAMI REC

== Data Aturan Piranti Pemroses ==

Id Record : 1

Id Aturan : 1

Id Kerusakan : 1

Id Gejala : 1

Nilai MB : 0.8

Nilai MD : 0.03

Jumlah Record : 6

Menu

- 1 Data Selanjutnya
- 2 Data Sebelumnya
- 3 Tambah Data
- 4 Ubah Data
- 5 Hapus Data

Kembali Menu

Gambar 5.12 Tampilan halaman data basis pengetahuan

Yout REC

== Olah Data Basis Pengetahuan ==

Id Aturan :  
1

Id Gejala :  
1

Id Kerusakan :  
1

Nilai MB :  
0.8

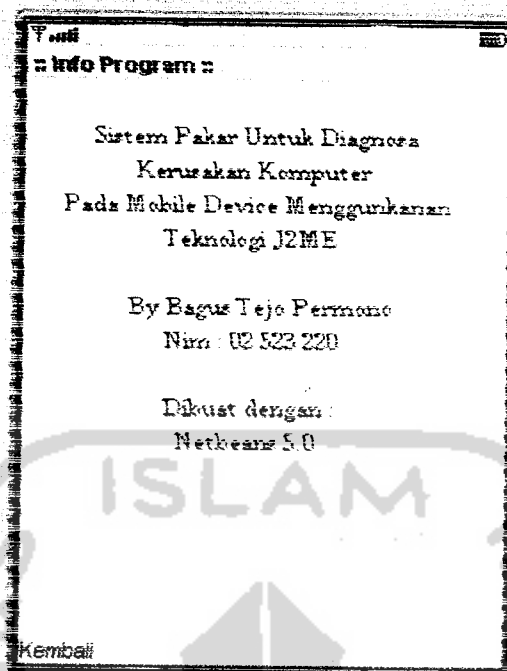
Nilai MD :  
0.03

Batal OK

Gambar 5.13 Tampilan halaman olah data basis pengetahuan

### 5.2.5 Tampilan Halaman Info Program

Halaman info program digunakan untuk menampilkan informasi tentang program. Lebih jelasnya dapat dilihat pada gambar 5.14.



**Gambar 5.14** Tampilan halaman info program

#### **5.2.6 Tampilan Halaman Bantuan**

Halaman bantuan berisi petunjuk penggunaan program. Tampilan antarmuka halaman bantuan dapat dilihat pada gambar 5.15.

Jenis : public

3. Prosedur untuk melihat data gejala

Nama prosedur : viewDataGejala

Jenis : public

4. Prosedure untuk menambah data gejala

Nama prosedur : addRecordGejala

Jenis : public

5. Prosedur untuk mengubah data gejala

Nama prosedur : editRecordGejala

Jenis : public

6. Prosedur untuk menghapus data gejala

Nama prosedur : delRecordGejala

Jenis : public

### 5.3.2 Class DataKerusakan

Prosedur-prosedur yang terdapat pada *class* DataKerusakan adalah sebagai berikut :

1. Prosedur untuk membuat/membuka RecordStore RSRusak

Nama prosedur : bukaRS

Jenis : public

2. Prosedur untuk menutup RecordStore RSRusak

Nama prosedur : tutupRS

Jenis : public

3. Prosedur untuk melihat data kerusakan
  - Nama prosedur : viewDataRusak
  - Jenis : public
4. Prosedure untuk menambah data kerusakan
  - Nama prosedur : addRecordRusak
  - Jenis : public
5. Prosedur untuk mengubah data kerusakan
  - Nama prosedur : editRecordRusak
  - Jenis : public
6. Prosedur untuk menghapus data kerusakan
  - Nama prosedur : delRecordRusak
  - Jenis : public

### 5.3.3 Class DataBasisPengetahuan

Prosedur-prosedur yang terdapat pada class DataBasisPengetahuan adalah sebagai berikut :

1. Prosedur untuk membuat/membuka RecordStore RSAaturan
  - Nama prosedur : bukaRS
  - Jenis : public
2. Prosedur untuk menutup RecordStore RSAaturan
  - Nama prosedur : tutupRS
  - Jenis : public

3. *Prosedur untuk melihat data basis pengetahuan*

Nama prosedur : *viewDataAturan*

Jenis : *public*

4. *Prosedure untuk menambah data basis pengetahuan*

Nama prosedur : *addRecordAturan*

Jenis : *public*

5. *Prosedur untuk mengubah data basis pengetahuan*

Nama prosedur : *editRecordAturan*

Jenis : *public*

6. *Prosedur untuk menghapus data basis pengetahuan*

Nama prosedur : *delRecordAturan*

Jenis : *public*

#### 5.3.4 *Class Konsultasi*

Prosedur-prosedur yang terdapat pada *class* Konsultasi adalah sebagai berikut :

1. *Prosedur untuk membuat/membuka RecordStore Fakta*

Nama prosedur : *bukaRS*

Jenis : *public*

2. *Prosedur untuk membuat/membuka RecordStore Alur*

Nama prosedur : *bukaRS1*

Jenis : *public*



3. Prosedur untuk membuat/membuka RecordStore Hasil
- Nama prosedur : bukaRS2
- Jenis : public
4. Prosedur untuk menghapus RecordStore
- Nama prosedur : hapusRS
- Jenis : public
5. Prosedur untuk menampilkan gejala-gejala pada saat *user* melakukan konsultasi sesuai piranti yang dipilih *user*
- Nama prosedur : listRecord
- Jenis : public
- Keterangan : prosedur ini menggunakan data gejala pada *class* DataGejala
6. Prosedur untuk melakukan proses diagnosa, yaitu melakukan penghitungan nilai *certainty factor* berdasarkan gejala yang dipilih *user*
- Nama prosedur : diagnosa
- Jenis : public
- Keterangan : prosedur ini menggunakan data gejala pada *class* DataGejala, data kerusakan pada *class* DataKerusakan, dan data basis pengetahuan pada *class* DataBasisPengetahuan
7. Prosedur untuk menampilkan hasil perhitungan nilai *certainty factor*
- Nama prosedur : viewHasil
- Jenis : public

### 5.3.5 Class SistemPakar

Prosedur-prosedur yang terdapat pada *class* SistemPakar merupakan prosedur-prosedur untuk menampilkan halaman, misalnya:

1. Prosedur untuk menampilkan halaman muka

Nama prosedur : initialize

Jenis : private

2. Prosedur untuk mengatur penekanan tombol (*command action*)

Nama prosedur : commandAction

Jenis : public



## BAB VI

### ANALISIS KINERJA PERANGKAT LUNAK

Analisis kinerja perangkat lunak merupakan proses pengujian terhadap aplikasi sehingga dapat dilihat seberapa besar kemampuan aplikasi dalam pencapaian tujuan pembuatan aplikasi. Pengujian ini dilakukan untuk menemukan kesalahan-kesalahan yang mungkin terjadi pada aplikasi sistem pakar ini.

#### 6.1 Pengujian Aplikasi

Pengujian kinerja aplikasi ini dilakukan untuk mengetahui kesalahan-kesalahan yang ada dan juga untuk mengetahui upaya penanganan kesalahannya. Beberapa pengujian memiliki dua kemungkinan, yaitu prosedur normal (benar) dan prosedur tidak normal (salah). Pengujian pada prosedur tidak normal digunakan untuk melihat apakah aplikasi mampu menangani kesalahan dengan baik serta mampu mengkomunikasikannya kepada pemakai aplikasi untuk kemudian memberikan saran atau solusi kepada pemakai aplikasi.

##### 6.1.1 Pengujian Proses Diagnosa

Tujuan pengujian pada proses diagnosa ini adalah untuk menguji apakah aplikasi dapat melakukan proses diagnosa dengan benar, yaitu menghitung nilai *certainty factor* dari setiap kerusakan berdasarkan nilai MB dan MD gejala-gejala yang mempengaruhinya.

Cara pengujian proses diagnosa ini adalah dengan langkah-langkah sebagai berikut

1. Aplikasi diisi dengan data gejala, data kerusakan, data basis pengetahuan seperti pada pengujian proses olah data
2. Selanjutnya aplikasi akan diuji dengan mengakses halaman konsultasi. Pilih gejala-gejala yang telah diberikan pada halaman konsultasi, kemudian lakukan proses diagnosa dengan menekan tombol Diagnosa
3. Bandingkan perhitungan nilai *certainty factor* yang dihasilkan oleh aplikasi dengan perhitungan yang dilakukan secara manual

Data gejala dan kerusakan pada proses pengujian ini dapat dilihat pada Lampiran. Pengujian proses diagnosa pada aplikasi ini akan dilakukan dengan cara sebagai berikut :

1. Pengujian untuk kerusakan yang dipengaruhi oleh satu gejala saja

Misalkan dalam halaman konsultasi dipilih satu gejala berikut : Layar monitor gelap dan bunyi beep berulang.

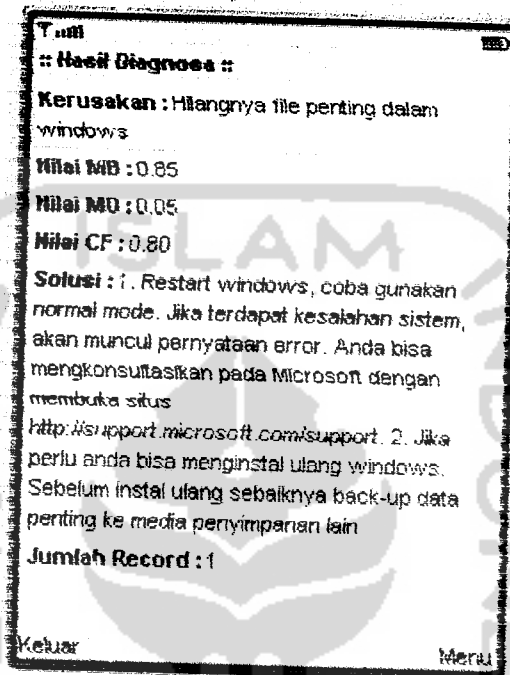
Kerusakan komputer yang dipengaruhi oleh gejala tersebut adalah : Hilangnya file penting dalam windows, dengan nilai  $MB = 0,85$  dan  $MD = 0,05$ .

**Hasil pengujian :**

Pada pengujian ini karena kerusakan Hilangnya file penting dalam windows cuma dipengaruhi oleh satu gejala maka penghitungannya adalah :

$$CF = MB - MD = 0,85 - 0,05 = 0,80$$

Setelah dilakukan pengujian sistem, didapatkan hasil yang sama untuk penghitungan manual maupun penghitungan oleh sistem (dapat dilihat pada gambar 6.1), yaitu Hilangnya file penting dalam windows dengan  $CF = 0,80$ .



Gambar 6.1 Tampilan hasil diagnosa pengujian pertama

2. Pengujian untuk kerusakan yang dipengaruhi oleh dua gejala

Misalkan dalam halaman konsultasi dipilih dua buah gejala kerusakan komputer, yaitu :

Gejala a : Indikator power on tidak menyala

Gejala b : Kabel power tidak terpasang dengan benar

Kerusakan komputer dalam basis pengetahuan yang dipengaruhi oleh kedua gejala kerusakan tersebut adalah :

Gejala a : - Masalah kelistrikan, dengan  $MB = 0,8$  dan  $MD = 0,1$

Gejala b : - Masalah kelistrikan, dengan MB = 0,9 dan MD = 0,1

### Hasil pengujian :

Pada pengujian ini karena Masalah kelistrikan dipengaruhi oleh dua gejala, maka penghitungannya adalah sebagai berikut :

$$MB \text{ Gejala a (MBa)} = 0,80 \quad MD \text{ Gejala a (MDa)} = 0,10$$

$$MB \text{ Gejala b (MBb)} = 0,90 \quad MD \text{ Gejala b (MDb)} = 0,10$$

$$MB = MBa + MBb * (1 - MBa) = 0,80 + 0,90 * (1 - 0,80) = 0,98$$

$$MD = MDa + MDb * (1 - MDa) = 0,10 + 0,10 * (1 - 0,10) = 0,19$$

$$CF = MB - MD = 0,98 - 0,19 = 0,79$$

Setelah dilakukan pengujian pada sistem, didapatkan hasil yang sama untuk penghitungan manual maupun penghitungan oleh sistem (dapat dilihat pada gambar 6.2), yaitu Masalah kelistrikan dengan CF = 0,79.



Gambar 6.2 Tampilan hasil diagnosa pengujian kedua

### 3. Pengujian pada kerusakan yang dipengaruhi oleh tiga gejala

Gejala yang dipilih :

Gejala a : Indikator power on tidak menyala

Gejala b : Kabel power tidak terpasang benar

Gejala c : Tegangan listrik sering turun

Kerusakan yang dipengaruhi oleh gejala tersebut adalah :

Masalah kelistrikan : - gejala a mempunyai MB = 0,8 dan MD = 0,1

- gejala b mempunyai MB = 0,9 dan MD = 0,1

- gejala c mempunyai MB = 0,8 dan MD = 0,1

**Hasil pengujian :**

Pada pengujian ini karena Masalah kelistrikan dipengaruhi oleh tiga gejala, maka penghitungannya adalah sebagai berikut :

$$MB \text{ Gejala a (MBa)} = 0,80 \quad MD \text{ Gejala a (MDa)} = 0,10$$

$$MB \text{ Gejala b (MBb)} = 0,90 \quad MD \text{ Gejala b (MDb)} = 0,10$$

$$MB \text{ Gejala c (MBc)} = 0,80 \quad MD \text{ Gejala c (MDc)} = 0,10$$

$$MBab = MBa + MBb * (1 - MBa) = 0,80 + 0,90 * (1 - 0,80) = 0,98$$

$$MDab = MDa + MDb * (1 - MDa) = 0,10 + 0,10 * (1 - 0,10) = 0,19$$

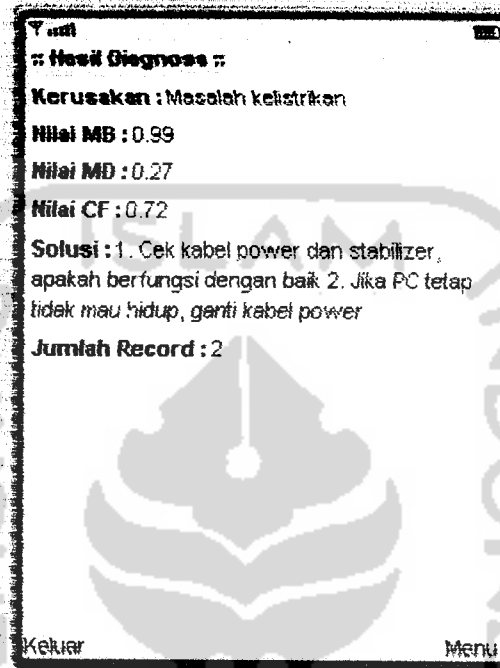
$$MBabc = MBab + MBc * (1 - MBab) = 0,98 + 0,80 * (1 - 0,98) = 0,996$$

$$MDabc = MDab + MDc * (1 - MDab) = 0,19 + 0,10 * (1 - 0,19) = 0,279$$

$$CFabc = MBabc - MDabc = 0,996 - 0,279 = 0,715$$

Setelah dilakukan pengujian pada sistem, didapatkan hasil yang sedikit berbeda untuk penghitungan manual dan penghitungan oleh sistem (dapat

dilihat pada gambar 6.3), hal ini dikarenakan sistem melakukan pembulatan untuk angka desimal lebih dari dua digit menjadi angka desimal dua digit.



Gambar 6.3 Tampilan hasil diagnosa pengujian ketiga

4. Pengujian untuk kerusakan yang dipengaruhi dua gejala dimana salah satu gejala mempunyai nilai  $MB = 1$

Dipilih dua gejala sebagai berikut :

Gejala a : Layar monitor gelap dan bunyi beep berulang

Gejala b : Hardware tidak terdeteksi BIOS

Kerusakan yang dipengaruhi oleh gejala tersebut adalah :

Masalah koneksi hardware : - gejala a dengan  $MB = 0,9$  dan  $MD = 0,05$

- gejala b dengan  $MB = 1$  dan  $MD = 0$



**Hasil pengujian :**

Pada pengujian ini karena Masalah koneksi hardware dipengaruhi oleh dua gejala, maka seharusnya penghitungan MD-nya menggunakan rumus sebagai berikut :  $MDab = MDa + MDb * (1 - MDa)$

akan tetapi karena ada salah satu gejala dengan nilai  $MB = 1$ , maka  $MDab = 0$ .

Jadi penghitungan CF-nya adalah sebagai berikut :

$$MB \text{ Gejala a (MBa)} = 0,90 \quad MD \text{ Gejala a (MDa)} = 0,05$$

$$MB \text{ Gejala b (MBb)} = 1 \quad MD \text{ Gejala b (MDb)} = 0$$

$$MBab = MBa + MBb * (1 - MBa) = 0,90 + 1 * (1 - 0,90) = 1$$

$$MDab = 0$$

$$CF = MBab - MDab = 1 - 0 = 1$$

Setelah dilakukan pengujian pada sistem, didapatkan hasil yang sama untuk penghitungan manual maupun penghitungan oleh sistem (dapat dilihat pada gambar 6.4), yaitu Masalah koneksi hardware dengan  $CF = 1$ .



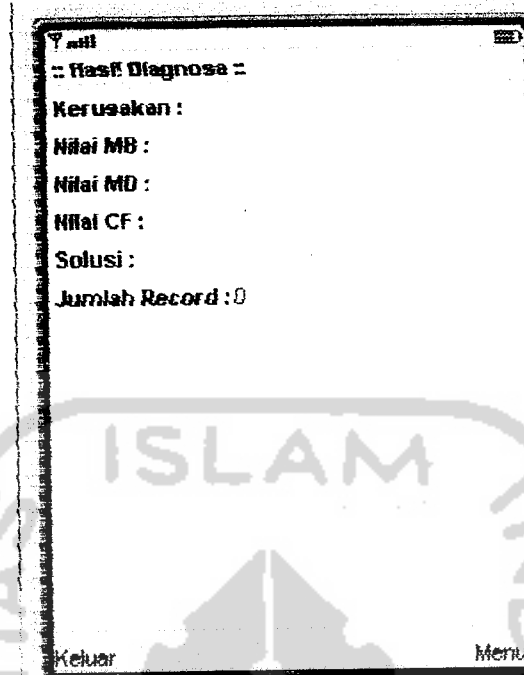
Gambar 6.4 Tampilan hasil pengujian keempat

#### 5. Pengujian tanpa memilih satupun gejala

Dalam pengujian ini tidak dipilih satupun gejala kerusakan jadi tidak ada satupun kerusakan komputer dalam basis pengetahuan yang dipengaruhi.

#### Hasil pengujian :

Karena tidak ada gejala yang dipilih, maka kerusakan yang dipengaruhi juga tidak ada. Sehingga hasilnya kosong, untuk lebih jelasnya lihat pada gambar 6.5.



Gambar 6.5 Tampilan hasil pengujian kelima

### 6.1.2 Proses Olah Data

Tujuan dari pengujian pada proses olah data adalah untuk menguji apakah proses olah data ini bisa melakukan proses tambah data, ubah data, dan hapus data dengan benar. Cara pengujiannya adalah dengan melakukan proses tambah data, ubah data, dan hapus data untuk data gejala, data kerusakan, dan data basis pengetahuan.

Untuk pengujian proses olah data ini terdapat tiga kategori, yaitu proses tambah data, proses ubah data, dan proses hapus data:

#### 1. Tambah data

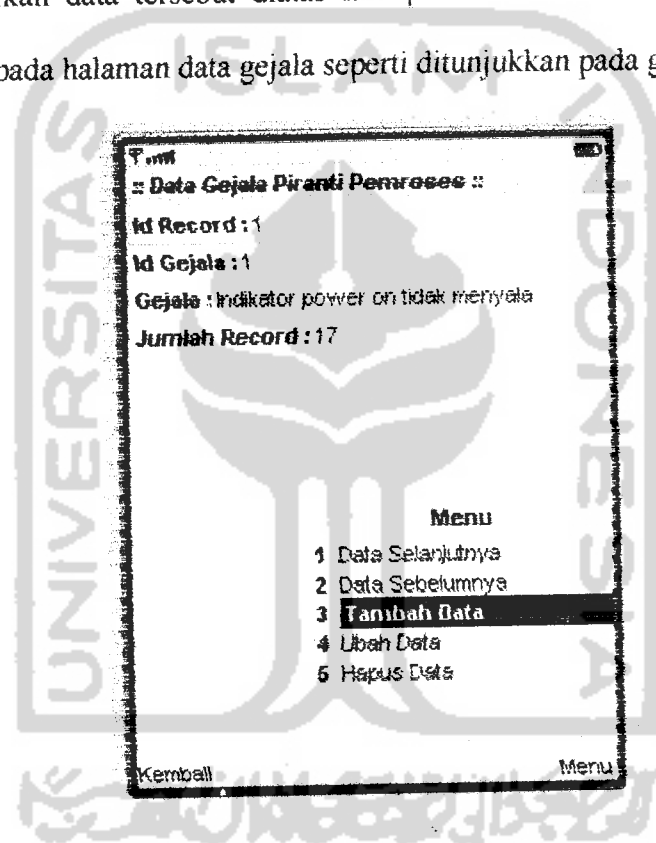
Proses tambah data baik data gejala, data kerusakan, maupun data basis pengetahuan mempunyai prosedur yang hampir sama.

Misalnya untuk menambah data gejala :

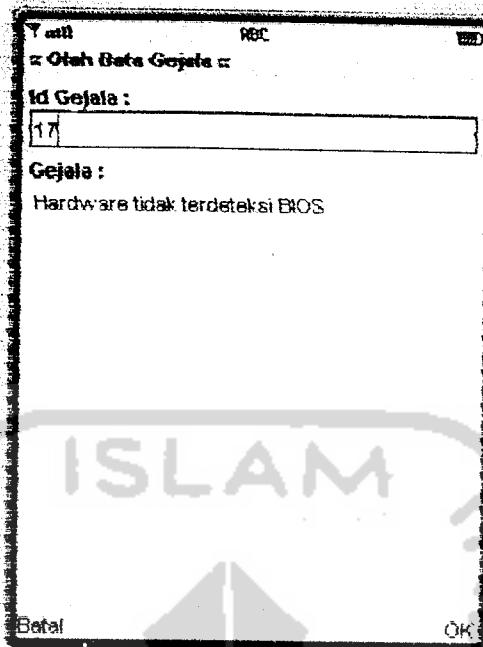
Id Gejala : 17

Gejala : Hardware tidak terdeteksi BIOS

maka pada halaman data gejala pilih menu tambah data (lihat gambar 6.6) untuk menampilkan halaman oiah data gejala seperti ditunjukkan pada gambar 6.7, kemudian isikan data tersebut diatas lalu pilih tombol Ok dan hasilnya akan ditampilkan pada halaman data gejala seperti ditunjukkan pada gambar 6.8.

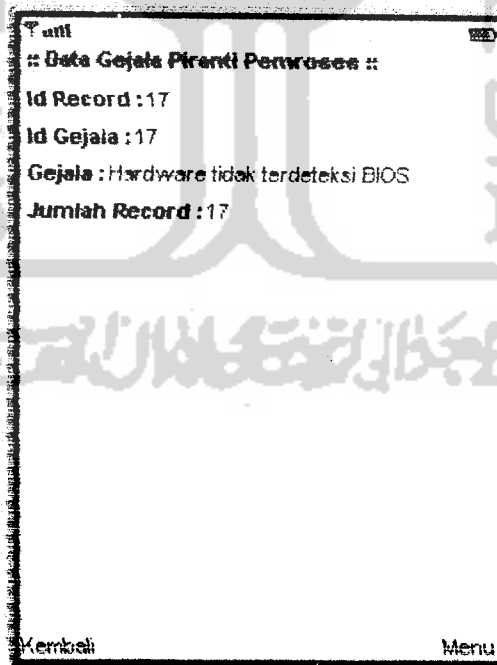


Gambar 6.6 Tampilan halaman data gejala



The screenshot shows a mobile application interface with a title bar containing 'Tutill' and 'REC'. The main content area is titled '=: Otah Data Gejala =:'. It features a text input field labeled 'Id Gejala:' with the value '17' entered. Below this is a text label 'Gejala:' followed by the description 'Hardware tidak terdeteksi BIOS'. At the bottom of the form, there are two buttons: 'Batal' on the left and 'OK' on the right.

Gambar 6.7 Tampilan proses tambah data gejala



The screenshot shows a mobile application interface with a title bar containing 'Tutill' and 'REC'. The main content area is titled '=: Data Gejala Piranti Perwosee =:'. It displays the following information: 'Id Record : 17', 'Id Gejala : 17', 'Gejala : Hardware tidak terdeteksi BIOS', and 'Jumlah Record : 17'. At the bottom of the form, there are two buttons: 'Kembali' on the left and 'Menu' on the right.

Gambar 6.8 Tampilan hasil proses tambah data

## 2. Ubah Data

Proses ubah data baik data gejala, data kerusakan, maupun data basis pengetahuan memiliki prosedur yang hampir sama.

Misalkan akan dilakukan proses ubah data kerusakan dengan Id Kerusakan = 1 dan isi dari Kerusakan diubah dari Masalah kelistrikan menjadi Masalah tegangan listrik, maka pada halaman data kerusakan pilih menu Ubah data (lihat gambar 6.9) lalu akan ditampilkan halaman olah data kerusakan (lihat gambar 6.10). Isi bagian yang ingin diubah lalu pilih tombol Ok untuk melihat hasilnya (lihat gambar 6.11).



Gambar 6.9 Tampilan halaman data kerusakan

Y. wati ABC

**:: Olah Data Kerusakan ::**

**Id Kerusakan :**  
1

**Kerusakan :**  
Masalah tegangan listrik

**Solusi :**  
1. Cek kabel power dan stabilizer, apakah berfungsi dengan baik 2. Jika PC tetap tidak mau hidup, ganti kabel power

Kembali OK

Gambar 6.10 Tampilan halaman olah data kerusakan

Y. wati ABC

**:: Data Kerusakan Piranti Pemroses ::**

**Id Record : 1**

**Id Kerusakan : 1**

**Kerusakan : Masalah tegangan listrik**

**Solusi : 1. Cek kabel power dan stabilizer, apakah berfungsi dengan baik 2. Jika PC tetap tidak mau hidup, ganti kabel power**

**Jumlah Record : 4**

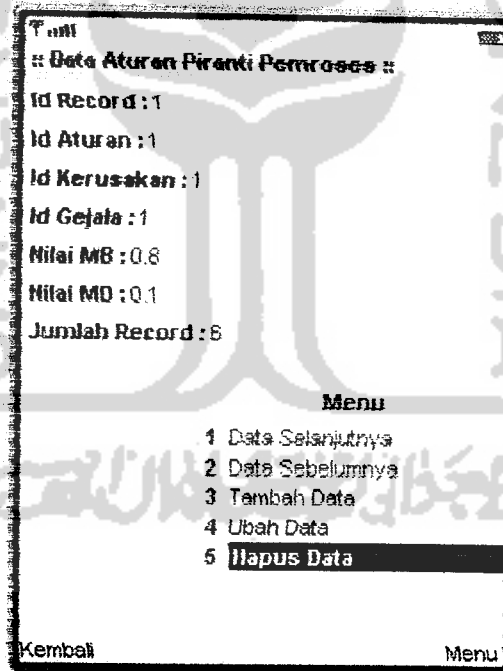
Kembali Menu

Gambar 6.11 Tampilan hasil proses ubah data

### 3. Hapus data

Proses ubah data baik data gejala, data kerusakan, maupun data basis pengetahuan memiliki prosedur yang hampir sama.

Misalkan akan dilakukan pengujian untuk menghapus data basis pengetahuan dengan Id Aturan = 1, pada halaman data basis pengetahuan pilih menu Hapus Data (lihat gambar 6.12) lalu pada halaman olah data basis pengetahuan akan ditampilkan data basis pengetahuan yang akan dihapus (lihat gambar 6.13). Pilih tombol Ok untuk menghapus data, tampilan halaman data basis pengetahuan akan menunjuk ke data lain (lihat gambar 6.14).



Gambar 6.12 Tampilan halaman data basis pengetahuan



## BAB VII

### KESIMPULAN DAN SARAN

#### 7.1 Kesimpulan

Setelah menyelesaikan aplikasi sistem pakar untuk mendiagnosa kerusakan computer pada *mobile device* dengan menggunakan teknologi J2ME ini, dapat diambil beberapa kesimpulan sebagai berikut:

1. Aplikasi ini hanya dapat diakses dengan ponsel berbasis java
2. Dengan menggunakan metode berorientasi objek untuk perancangan sistem, memudahkan dalam pengembangan sistem secara keseluruhan, sebagai contoh ketika ingin menambahkan suatu fungsi tertentu, cukup dengan menambahkan fungsi tersebut ke dalam *class*. Demikian pula ketika diimplementasikan dalam kode program, dengan menggunakan pemrograman berorientasi objek, maka fungsi tambahan tersebut cukup ditambahkan ke dalam *class* yang sesuai.
3. Aplikasi sistem pakar untuk dignosa kerusakan komputer pada *mobile device* ini dapat membantu *user* dalam memperoleh informasi tentang kerusakan komputer berdasarkan gejala-gejala kerusakan yang terjadi dan bagaimana langkah-langkah perbaikannya, dengan fasilitas *mobile*-nya menjadikan aplikasi ini semakin mudah digunakan.
4. Dengan menggunakan bahasa Java sebagai bahasa pemrograman maka aplikasi ini dapat dijalankan di berbagai *platform* sistem operasi seperti Linux, Windows maupun Unix.

## DAFTAR PUSTAKA

- [FAH04] Fahrial, Jaka. 2004. Sistem Pakar Mengidentifikasi Kerusakan Gangguan Sambungan Telepon PT. TELKOM. Ilkom
- [HAR03] Hartanto, Antonius Aditya. 2003. Java 2 Micro Edition Mobile Interface device programming. Jakarta: Elex Media Komputindo
- [HAR04] Hartanto, Antonius Aditya. 2004. Pemrograman Mobile Java dengan MIDP 2.0. Yogyakarta: ANDI Yogyakarta
- [HAR06] Haryanti, Arfiani. 2006. Aplikasi AlQuran Untuk Mobile Devices Dengan Menggunakan Teknologi J2ME. Skripsi, tidak diterbitkan. Yogyakarta: Fakultas Teknologi Industri Universitas Islam Indonesia.
- [KUS06] Kusriani, S.Kom. 2006. Sistem Pakar : Teori & Aplikasi. Yogyakarta: ANDI Yogyakarta.
- [RAH05] Rahman, A.S. 2005. Aplikasi Teleakses Perangkat Bergerak. Yogyakarta: ANDI Yogyakarta.
- [RIC03] Rickyanto, Isak, ST. 2003. Dasar Pemrograman Berorientasi Objek Dengan Java 2 (JDK 1.4). Yogyakarta: ANDI Yogyakarta.
- [SUY05] Suyoto, Dr. 2005. Membuat Sendiri Aplikasi Ponsel. Yogyakarta: Gava Media.
- [WIR03] Wirasantika, Faisal. 2003. Membangun Wireless Application Menggunakan Teknologi J2ME. Available at [ilmukomputer.com](http://ilmukomputer.com)

- [YUH05] Yuhana, Andri. 2005. Sistem Pakar Untuk Mendiagnosa Kerusakan pada Komputer dengan Metode Certanty Factor. Skripsi, tidak diterbitkan. Yogyakarta: Fakultas Teknologi Industri Universitas Islam Indonesia.



# LAMPIRAN

## 1. DATA GEJALA

### a. Gejala Piranti Pemroses

1	Indikator power on tidak menyala
2	Kabel power tidak terpasang benar
3	Kabel power terpasang benar
4	Layar monitor gelap dan bunyi beep berulang
5	Windows merekomendasikan menggunakan safe mode
6	Komputer lambat saat booting
7	Perlu waktu lama berpindah antar program
8	Menambah software atau hardware baru
9	Tegangan listrik sering turun
10	BIOS melaporkan pesan 'Harddisk failure'
11	BIOS melaporkan pesan 'Floppy drive failure'
12	Identifikasi hardware benar
13	Tanggal dan jam BIOS berubah
14	Password BIOS lupa
15	Pemasangan hardware belum benar
16	Pemasangan hardware sudah benar
17	Hardware tidak terdeteksi BIOS

### b. Gejala Piranti Input

No	Gejala
1	Saat booting muncul pesan kesalahan
2	Keyboard tidak terdeteksi pada control panel
3	Tombol keyboard tidak berfungsi
4	Mouse tidak terdeteksi pada control panel
5	Tombol mouse tidak berfungsi
6	CD drive tidak terdeteksi pada window explorer
7	CD drive lambat
8	CD drive mengeluarkan bunyi berisik

### c. Gejala Piranti Output

No	Gejala
1	Monitor tidak mau hidup
2	Monitor mati saat digunakan

3	Muncul bercak kebiruan pada sudut monitor
4	Monitor tampak buram dan sering terjadi kedipan
5	Printer tidak mau mencetak
6	Lampu indikator monitor mati
7	Printer lambat mencetak
8	Hasil cetak tidak sempurna
9	Pada control panel printer terdeteksi
10	Speaker tidak mengeluarkan suara
11	Lampu indikator speaker mati
12	Sound card tidak terdeteksi
13	Pemasangan kabel speaker tidak benar

## 2. DATA KERUSAKAN

### a. Kerusakan Piranti Pemroses

No	Kerusakan	Solusi
1	Masalah kelistrikan	<ol style="list-style-type: none"> <li>1. Cek koneksi kabel, periksa apakah kabel power tersambung dengan baik</li> <li>2. Jika menggunakan stabilizer, periksa apakah stabilizer berfungsi baik saat dihidupkan</li> <li>3. Pastikan power outletnya tidak bermasalah, Anda dapat memeriksa dengan peralatan listrik</li> <li>4. Apabila PC tetap tidak mau hidup, kemungkinan masalah pada power. Gantilah kabel power dengan yang baru</li> </ol>
2	Masalah power supply	<ol style="list-style-type: none"> <li>1. Cek koneksi kabel, periksa apakah kabel power tersambung dengan baik</li> <li>2. Periksa apakah kipas power supplya(PS) berfungsi dan power outletnya tidak bermasalah. Anda bisa memeriksanya dengan peralatan listrik</li> <li>3. Apabila kipas PS tidak mau hidup, kemungkinan masalah pada PS. Ganti PS dengan yang baru</li> </ol>
3	Hilangnya file penting windows	<ol style="list-style-type: none"> <li>1. Restart windows dan paksaan menggunakan normal mode. Jika ada kesalahan, windows akan mengeluarkan pernyataan error. Anda dapat konsultasi dengan Microsoft dengan mengunjungi situs <a href="http://support.microsoft.com/support">http://support.microsoft.com/support</a>.</li> <li>2. Jika perlu anda dapat menginstal ulang windows. Sebelum instal ulang sebaiknya back-up data penting ke media penyimpanan lain</li> </ol>
4	Masalah koneksi hardware	<ol style="list-style-type: none"> <li>1. Tiap merek BIOS punya kode bunyi beep tersendiri, untuk mengetahui kerusakan komputer dengarkan bagaimana bunyi beep terdengar pada speaker.</li> <li>2. Cari informasi bunyi beep pada situs penyedia BIOS</li> </ol>

5	Masalah koneksi floppy	<ol style="list-style-type: none"> <li>1. Cek BIOS, apakah setting floppy drive sudah benar. Lihat kapasitas floppy yang terpasang dan nama drive yaitu A</li> <li>2. Jika floppy tak dikenali, cek terlebih dulu sambungan kabel case dan motherboard. Mungkin belum terpasang dengan benar</li> <li>3. Jika sambungan benar tapi floppy tidak terdeteksi, kemungkinan floppy rusak</li> </ol>
6	Hardisk bermasalah	<ol style="list-style-type: none"> <li>1. Cek setup BIOS, apakah masih bisa mendeteksi harddisk</li> <li>2. Jika harddisk tak dikenali, pertama cek sambungan kabel harddisk pada case dan motherboard. Mungkin tidak tertancap dengan benar</li> <li>3. Jika sambungan sudah benar dan tidak terdeteksi, kemungkinan harddisk anda memang rusak</li> </ol>
7	Kebanyakan program yg di up-load saat start-up	<ol style="list-style-type: none"> <li>1. Tutup semua aplikasi yang sedang aktif pada komputer</li> <li>2. Munculkan kotak dialog System configuration utilities dengan cara klik <b>Start - Run</b></li> <li>3. Ketik "msconfig" pada kotak open dan tekan enter. Akan muncul kotak dialog System configuration utilities</li> <li>4. Klik <b>Startup</b> pada kotak dialog tsb. Hilangkan tanda centang pada beberapa item yang tidak anda anggap perlu.</li> <li>5. Klik tombol <b>Apply</b> dan <b>Ok</b> untuk mengaktifkannya. Klik <b>Yes</b> pada konfirmasi untuk merestart sistem</li> </ol>
8	Sumber daya memori kurang memadai	<p>Anda bisa menambah kapasitas memori dengan cara:</p> <ol style="list-style-type: none"> <li>1. Buka kunci soket</li> <li>2. Masukkan modul memori pada slot memori</li> <li>3. Kunci atau tekan kembali kunci soket</li> </ol>
9	Terjadi konflik antara hardware dan software	<ol style="list-style-type: none"> <li>1. Lakukan reinstal windows, tp sebelum itu back-up dulu data penting ke media penyimpanan lain</li> <li>2. Jika setelah langkah tsb. komputer masih mengalami gangguan, bawalah ke teknisi komputer</li> </ol>
10	Baterai CMOS habis	Cabut baterai CMOS pada motherboard dan ganti dengan yang baru
11	Update password	
12	Fasilitas plug and play gagal	<p>Ada 2 alternatif :</p> <ol style="list-style-type: none"> <li>1. Gunakan fasilitas Add/Remove Hardware pada Control Panel. Lakukan langkah berikut : - Klik Start - Settings - Control Panel. Akan muncul jendela Control Panel, klik Add/Remove Hardware - Sebuah wizard akan muncul untuk mendeteksi hardware, jika berhasil akan muncul pesan hardware ditemukan</li> <li>2. Jika alternatif 1 belum berhasil, setting BIOS perlu diatur agar fitur Plug N Play dapat berjalan.</li> </ol>

### c. Kerusakan Piranti Output

No	Kerusakan	Solusi
1	Masalah power monitor	<ol style="list-style-type: none"> <li>1. Jika lampu indikator monitor tak menyala. Pastikan kabel power monitor terpasang dengan benar</li> <li>2. Jika lampu indikator pada monitor hidup dan berwarna orange dan berkedip-kedip. Periksa kabel video yang menghubungkan monitor dengan CPU terpasang dengan benar</li> <li>3. Jika dengan pengecekan diatas masalah tak teratasi, berarti ada problem dengan video board video adapter CRT. Coba ganti kartu grafis dengan yang baru</li> </ol>
2	Ada medan magnet disekitar monitor	Cobalah memakai fitur Degaussing yang ada pada kontrol monitor
3	Tabung katoda bermasalah	Permasalahan ini sering terjadi pada monitor berusia lebih dari 3 tahun. Kemungkinan disebabkan oleh fosfor pada tabung katoda. Untuk mengatasinya lebih baik berkomunikasi dengan teknisi komputer karena berhubungan dengan komponen elektronik
4	Masalah koneksi printer	<ol style="list-style-type: none"> <li>1. Periksa kabel power dan kabel koneksi terpasang dengan benar</li> <li>2. Periksa driver printer pada Control Panel, apakah terinstal dengan benar</li> <li>3. Jika anda yakin koneksi sudah tepat tapi tidak dapat mencetak berarti printer rusak</li> </ol>
5	Kerusakan head printer	Bersihkan head printer, jika sudah bersih tapi tetap tidak dapat mencetak kemungkinan printer rusak
6	Isi tinta habis	<ol style="list-style-type: none"> <li>1. Isi tangki tinta atau ganti tangki tinta dengan yang baru</li> <li>2. Jika tetap tidak mau mencetak, kemungkinan printer rusak</li> </ol>
7	Masalah kelistrikan speaker	<ol style="list-style-type: none"> <li>1. Cek koneksi kabel apakah kabel power dan kabel koneksi ke komputer terpasang dengan benar</li> <li>2. Jika speaker tetap tidak mau hidup, kemungkinan masalah pada piranti speaker</li> </ol>

## 3. DATA BASIS PENGETAHUAN

### a. Basis Pengetahuan Piranti Pemroses

Idr	Kerusakan	Idg	Gejala	MB	MD
1	Masalah kelistrikan	1	Indikator power on tidak menyala	0,8	0,1
		2	Kabel power tidak terpasang benar	0,9	0,1
		7	Perlu waktu lama berpindah antar program	0,6	0,1

		9	Tegangan listrik sering turun	0,8	0,1
2	Masalah power supply	1	Indikator power on tidak menyala	0,9	0,02
		3	Kabel power terpasang benar	0,9	0,01
3	Hilangnya file penting dalam windows	4	Layar monitor gelap dan bunyi beep berulang	0,85	0,05
		5	Windows merekomendasikan menggunakan safe mode	0,7	0,05
4	Masalah koneksi hardware	4	Layar monitor gelap dan bunyi beep berulang	0,9	0,05
		15	Pemasangan hardware belum benar	1	0
		17	Hardware tidak terdeteksi BIOS?	1	0
5	Masalah koneksi floppy	4	Layar monitor gelap dan bunyi beep berulang	0,7	0,1
		11	BIOS melaporkan pesan 'Floppy drive failure'	1	0
6	Hardisk bermasalah	4	Layar monitor gelap dan bunyi beep berulang	0,7	0,05
		10	BIOS melaporkan pesan 'Harddisk failure'	1	0
7	Kebanyakan program yg di up-load saat start-up	6	Komputer lambat saat booting	0,92	0,02
8	Sumber daya memori kurang memadai	6	Komputer lambat saat booting	0,9	0,01
		7	Perlu waktu lama berpindah antar program	0,9	0,01
9	Terjadi konflik antar hardware	7	Perlu waktu lama berpindah antar program	0,6	0,1
		8	Menambah software atau hardware baru	0,85	0,05
		12	Identifikasi hardware benar	0,85	0,05
10	Baterai CMOS habis	13	Tanggal dan jam BIOS berubah	0,92	0,02
11	Update password	14	Password BIOS lupa	0,85	0,01
12	Fasilitas plug and play gagal	8	Menambah software atau hardware baru	0,9	0,01
		16	Pemasangan hardware sudah benar	0,9	0,01
13	Versi BIOS harus di up-date	16	Pemasangan hardware sudah benar	0,9	0,05
		17	Hardware tidak terdeteksi BIOS	0,9	0,02

#### b. Basis Pengetahuan Piranti Input

Idr	Kerusakan	Idg	Gejala	MB	MD
1	Keyboard tidak terdeteksi komputer	1	Saat booting muncul pesan kesalahan	0,9	0,01
		2	Keyboard tidak terdeteksi pada control panel	1	0
2	Keyboard kotor	3	Tombol keyboard tidak berfungsi	1	0
3	Mouse tidak terdeteksi komputer	1	Saat booting muncul pesan kesalahan	0,9	0,05



		4	Mouse tidak terdeteksi pada control panel	1	0
4	Mouse kotor	5	Tombol mouse tidak berfungsi	1	0
5	Masalah koneksi CD drive	6	CD drive tidak terdeteksi pada window explorer	1	0
6	Optik pada CD drive kotor	7	CD drive lambat	0,8	0,1
		8	CD drive mengeluarkan bunyi berisik	0,75	0,1

**c. Basis Pengetahuan Piranti Output**

Idr	Kerusakan	Idg	Gejala	MB	MD
1	Masalah power monitor	1	Monitor tidak mau hidup	1	0
		5	Lampu indikator monitor mati	1	0
2	Ada medan magnet disekitar monitor	2	Muncul bercak kebiruan pada sudut monitor	0,9	0,1
		3	Monitor tampak buram dan sering terjadi kedipan	0,9	0,1
3	Tabung katoda bermasalah	3	Monitor tampak buram dan sering terjadi kedipan	0,9	0,1
4	Masalah koneksi printer	4	Printer tidak mau mencetak	0,9	0,05
5	Kerusakan head printer	6	Printer lambat mencetak	0,8	0,2
		7	Hasil cetak tidak sempurna	0,9	0,1
6	Isi tinta habis	8	Pada control panel printer terdeteksi	0,7	0,1
		4	Printer tidak mau mencetak	0,9	0,05
7	Masalah kelistrikan speaker	9	Speaker tidak mengeluarkan suara	0,8	0,1
		10	Lampu indikator speaker mati	1	0