

# SISTEM PAKAR *TROUBLESHOOTING* KERUSAKAN *HARDWARE* KOMPUTER BERBASIS WEB DENGAN METODE *FORWARD CHAINING*

Frendy Triawan, Nurahman

Universitas Darwan Ali  
Jl. Batu Berlian No.10 Telp. (0531) 33336, Fax. (0531) 33342 Sampit

## ABSTRAK

*Banyaknya pengguna komputer yang kurang memiliki pengetahuan yang cukup terhadap penanganan awal kerusakan hardware mengakibatkan banyak sekali pengguna komputer atau suatu institusi yang mengeluarkan biaya yang tidak sedikit hanya untuk memperbaiki kerusakan yang terjadi pada hardware komputer. Oleh sebab itu aplikasi sistem pakar ini dibuat untuk membantu pengguna komputer dalam melakukan diagnosis awal terhadap suatu kerusakan hardware komputer yang dialami beserta solusi untuk mengatasi kerusakan tersebut.*

*Proses pembangunan sistem pakar ini menggunakan metode pengetahuan dengan teknik pengumpulan data yang digunakan yaitu wawancara, observasi dan studi pustaka. Sedangkan metode inferensi yang digunakan dalam pembangunan sistem pakar ini menggunakan metode forward chaining. Untuk metodologi pembangunan perangkat lunak yang digunakan menggunakan metode terstruktur yaitu Data Flow Diagram dan Entity Relationship Diagram.. Tools pemrograman yang digunakan dalam pembangunan sistem pakar ini menggunakan Adobe Dreamweaver CS3 sedangkan basis datanya menggunakan MySQL.*

*Dalam perancangan aplikasi sistem pakar ini meminta suatu input berupa jawaban dari user terhadap pertanyaan gejala-gejala, kemudian output yang dihasilkan oleh komputer berupa kemungkinan kerusakan yang dialami oleh hardware, penjelasan kerusakan serta solusi atau saran untuk mengatasi kerusakan tersebut. Adapun kesimpulan yang diperoleh yaitu aplikasi sistem pakar ini sudah dapat melakukan diagnosis awal terhadap suatu kerusakan hardware dan memberikan langkah solusi untuk mengatasinya.*

**Kata Kunci :** Sistem Pakar, *Forward Chaining*, *Troubleshooting*, *Hardware*

## 1. PENDAHULUAN

### 1.1 Latar Belakang

Perkembangan teknologi informasi telah menjadi sebuah bagian yang susah untuk dipisahkan di era seperti sekarang ini. Dengan adanya kemajuan teknologi yang semakin pesat, berpengaruh pula pada perkembangan perangkat komputer saat ini. Seiring perkembangan teknologi tersebut dikembangkan pula suatu teknologi yang mampu mengadopsi proses dan cara berpikir manusia yaitu teknologi *Artificial Intelligence* atau Kecerdasan Buatan. Kecerdasan buatan atau *Artificial Intelligence* merupakan salah satu bagian ilmu komputer yang membuat agar mesin (komputer) dapat melakukan pekerjaan seperti dan sebaik yang dilakukan manusia. Kecerdasan buatan dapat diimplementasikan menjadi sistem pakar. Sistem pakar merupakan sistem yang berusaha mengadopsi pengetahuan manusia ke komputer, agar komputer dapat memberikan alternatif solusi. (Sri Kusumadewi, 2003).

Kerusakan komputer pada saat ini masih menjadi permasalahan, hal ini dimaklumi mengingat

banyaknya *user* atau pengguna komputer yang kurang memiliki pengetahuan cukup dalam mengetahui kerusakan pada komputernya. Masalahnya, masyarakat umum atau suatu institusi mungkin tidak dapat mengidentifikasi dimana letak kerusakan dan berat tidaknya kerusakan yang terjadi pada komputernya.

Untuk itu dirasakan perlunya dibangun suatu aplikasi yang dapat membantu untuk mengidentifikasi awal kerusakan sebelum dilakukan tindakan selanjutnya. Aplikasi yang dimaksud adalah suatu aplikasi sistem pakar berbasis *web* yang mengadopsi pengetahuan yang bersumber dari seorang pakar komputer. Dengan dibangunnya aplikasi sistem pakar ini diharapkan dapat memberikan solusi alternatif agar dapat dilakukan penanganan kerusakan *hardware* pada komputer.

### 1.2 Perumusan Masalah

Berdasarkan pada latar belakang masalah yang telah dibahas, maka dirumuskan permasalahannya sebagai berikut :

1. Bagaimana merancang dan membangun sebuah perangkat lunak sistem pakar untuk *troubleshooting* kerusakan *hardware* komputer dengan akurasi diagnosis yang baik ?
2. Bagaimana memberikan solusi alternatif informasi kepada *user* komputer dalam menemukan letak permasalahannya yang terjadi pada *hardware* komputernya?

### 1.3 Tujuan Penelitian

Berdasarkan permasalahan yang diteliti, maka maksud dari penulisan Tugas Akhir ini adalah membangun sebuah aplikasi sistem pakar untuk mendiagnosis dan memberikan petunjuk penanganan kerusakan *hardware* pada komputer.

Sedangkan tujuan yang ingin dicapai dalam pembangunan perangkat lunak sistem pakar ini adalah :

1. Merancang dan membangun sebuah perangkat lunak sistem pakar untuk *troubleshooting* kerusakan pada komputer dengan akurasi diagnosis yang baik.
2. Memberikan informasi solusi alternatif kepada *user* dalam menemukan letak permasalahan yang terjadi pada komputernya.

### 1.4 Metode Penelitian

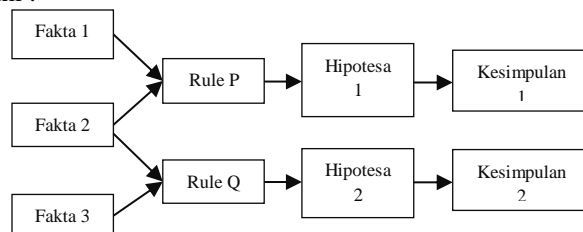
Adapun metode pengumpulan data yang dilakukan oleh penulis pada saat penelitian adalah sebagai berikut :

1. Study Pustaka (*Library Research*)  
Metode studi pustaka yaitu metode pengumpulan data diperoleh melalui buku-buku yang berhubungan dengan masalah yang akan diteliti sebagai bahan referensi bagi penulis.
2. Wawancara (*Interview*)  
Metode wawancara yaitu metode pengumpulan data dengan cara melakukan wawancara secara langsung dengan seorang pakar komputer untuk mengetahui macam kerusakan, jenis kerusakan, ciri kerusakan, serta solusi kerusakan atas yang terjadi dikomputer, yang dilakukan di TRIAD Komputer.
3. Observasi (*Observation*)  
Metode observasi yaitu suatu metode pengumpulan data dimana penulis secara langsung mengamati ke objek. Metode observasi ini digunakan untuk memperkuat data yang akan dibuat. Observasi ini digunakan untuk mengamati secara langsung cara tentang kerusakan *hardware* pada komputer serta solusi kerusakan pada TRIAD Komputer.  
Hasil observasi ini juga sekaligus untuk menginformasikan data yang telah terkumpul dari hasil wawancara dengan keadaan yang sebenarnya.

## 2. LANDASAN TEORI

### 2.1 Metode Forward Chaining

*Forward Chaining* adalah pendekatan yang dimotori data (*data-driven*). Dalam pendekatan ini pelacakan dimulai dari informasi masukan, dan selanjutnya mencoba menggambarkan kesimpulan. Pelacakan ke depan mencari fakta yang sesuai dengan bagian *IF* dari aturan *IF-THEN*. Metode pelacakan *forward chaining* dapat dilihat pada gambar dibawah ini :



**Gambar 1.** Proses Pelacakan Forward Chaining

Dalam sistem inferensi yang dibuat/ dikembangkan tentu ada pertanyaan, maka sistem yang harus dipilih, *forward* atau *backward chaining* ?

Di bawah ini terdapat panduan untuk memilih sistem yang mana yang cocok diantara keduanya untuk sistem yang akan kita kembangkan :<sup>1</sup>

1. Bagaimanakah hubungan antara rule dengan fakta – faktanya, sehingga didapatkan konklusinya.
2. Jika masalah yang dihadapi lebih dekat ke fan out ( Sekumpulan fakta yagn bisa menuju banyak konklusi) maka pilihlah *backward chaining*.
3. Jika masalah yang dihadapi lebih dekat ke fan in ( sekumpulan hipotesis yang bisa menuju banyak pertanyaan), maka pilihlah *forward chaining*.
4. Banyak cara mendapatkan sedikit konklusi → *forward chaining*.
5. Sedikit cara untuk mendapatkan banyak konklusi → *backward chaining*.

Tipe sistem yang dapat dicari dengan *forward chaining* :

1. Sistem yang dipresentasikan dengan satu atau beberapa kondisi.
2. Untuk setiap kondisi, sistem mencari rule-rule dalam knowledge base untuk rule-rule yang berkorespondensi dengan kondisi dalam bagian *IF*
3. Setiap rule dapat menghasilkan kondisi baru dari konklusi yang diminta pada bagian *THEN*. Kondisi yang ditambahkan ke sistem akan diproses. Jika ditemui suatu kondisi, sistem akan kembali ke langkah 2 dan mencari

rule-rule dalam knowledge base kembali. Jika tidak ada konklusi baru, sesi ini berakhir.

Adapun kelemahan dan kelebihan forward chaining yaitu (DUR.1994):

Kelebihan :

- Kelebihan utama dari *forward chaining* yaitu metode ini akan bekerja dengan baik ketika problem bermula dari mengumpulkan/ menyatukan informasi lalu kemudian mencari kesimpulan apa yang dapat diambil dari informasi tersebut.
- Metode ini mampu menyediakan banyak sekali informasi dari hanya jumlah kecil data.

Kelemahan :

- Kelemahan utama metode ini yaitu kemungkinan tidak adanya cara untuk mengenali dimana beberapa fakta lebih penting dari fakta lainnya.
- Sistem bisa saja menanyakan pertanyaan yang tidak berhubungan. Walaupun jawaban dari pertanyaan tersebut penting. Namun hal ini akan membingungkan *user* untuk menjawab pada subjek yang tidak berhubungan.

Sedangkan contoh Algoritma dari *forward chaining* adalah sebagai berikut :

**Tabel 1** Contoh aturan

No.	Aturan
R-1	IF A & B THEN C
R-2	IF C THEN D
R-3	IF A & E THEN F
R-4	IF A THEN G
R-5	IF F & G THEN D
R-6	IF G & E THEN H
R-7	IF C & H THEN I
R-8	IF I & A THEN J
R-9	IF G THEN J
R-10	IF J THEN K

Pada tabel 1 terlihat ada 10 aturan yang tersimpan dalam basis pengetahuan. Fakta awal yang diberikan hanya A dan F (artinya : A dan F bernilai benar). Ingin dibuktikan apakah K bernilai benar.

Langkah-langkah inferensi adalah sebagai berikut :

- Dimulai dari R.I.A merupakan fakta sehingga bernilai benar, sedangkan B belum bisa di ketahuikebenarannya.sehingga C-pun juga belum bisa diketahui kebenarannya . Oleh karena itu kita tidak dapat informasi apapun kepada R-1 ini. Sehingga kita menuju ke R-2.
- Pada R-2, kita tidak mengetahui informasi apapun tentang C, sehingga kita tidak bisa memastikan kebenaran D. Oleh karena itu kita tidak mendapatkan informasi apapun pada R-1 ini. Sehingga kita menuju ke R\_3
- Pada R-3 , baik A maupun E adalah fakta sehingga jelas benar. Demikian dengan F sebagai konsekuen juga ikut benar. Sehingga kita

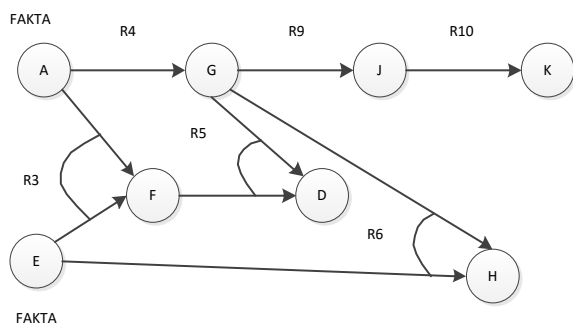
mempunyai fakta baru yaitu F. Karna F bukan hipotesis yang hendak kita buktikan (=K), maka penelusuran kita lanjutkan ke R-4

- Pada R-4, A adalah fakta sehingga jelas benar. Dengan demikian G sebagai konsekuen juga ikut benar. Sehingga sekarang kita mempunyai fakta baru yaitu G. Karena G bukan hipontesis yang hendak kita buktikan (=K), maka penerusuran kita lanjutkan ke R-5
- Pada R-5, baik F maupun G bernilai benar berdasarkan aturan R-3, dan R-4, dengan demikian D sebagai konsekuen juga ikut benar. Sehingga kita mempunyai fakta baru yaitu D. Karna D bukan hipotesis yang hendak kita buktikan (=K), maka penerusuran kita lanjutkan ke R-6
- Pada R-6, baik A maupun G adalah benar berdasarkan fakta dan R-4. Dengan demikian H sebagai konsekuen juga ikut benar. Sehingga sekarang kita mempunyai fakta baru yaitu H. Karena H bukan hipotesia yang hendak kita buktikan (=K), maka penerusuran kita lanjutkan ke R-7.
- Pada R-7. Meskipun H benar berdasarkan R-6, namun kita tidak tau kebenarannya C, sehingga I-pun juga belum bisa di ketahui kebenarannya. Oleh karena itu kita tidak mendapatkan informasi apapun pada R-7 ini. Sehingga kita menuju ke R-8
- Pada R-8, meskipun A benar karena fakta, namun kita tidak tau kebenaran I, sehingga J-pun juga belum bisa di ketahui kebenarannya. Oleh karena itu kita tidak mendapatkan informasi apapun padaR-8 ini. Sehingga kita menujuka R-9
- Pada R-9, J bernilai benar karena G benar berdasarkan R-4 karena J bukan hipotesis yang hendak kita buktikan (=K) maka penerusuran kita lanjutkan ke R-10
- Pada R-10, K bernilai benar karena J benar berdasarkan R-9. Karena H sesudah merupakan hipotesis yang hendak kita buktikan (=K), maka terbukti bahwa K adalah benar.

Tabel munculnya fakta baru pada saat inferensi terlihat pada Tabel 2. Sedangkan alur inferensi terlihat pada gambar 2.

**Tabel 2** Fakta Baru

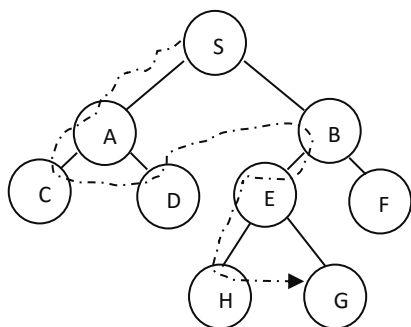
Aturan	Fakta Baru
R-3	F
R-4	G
R-5	D
R-6	H
R-9	J
R-10	K



Gambar 2. Penalaran *Forward Chaining*

## 2.2 Depth-first search

Dalam metode inferensi tersebut menggunakan penelusuran yaitu *Depth-first search*. *Depth-first search*, melakukan penelusuran kaidah secara mendalam dari simpul akar bergerak menurun ke tingkat dalam yang berurutan. Penelusuran *depth-first search* dapat dilihat pada gambar dibawah ini :



Gambar 3. Diagram Alir Teknik Penelusuran *Depth First Search*

## 3. MODEL, ANALISA, DESAIN, DAN IMPLEMENTASI

### 3.1 Model

Teknik analisis data dalam pembuatan perangkat lunak menggunakan pemodelan perangkat lunak dengan paradigma *waterfall*, yang meliputi beberapa proses diantaranya:

- Definisi Kebutuhan**  
Merupakan bagian dari sistem yang terbesar dalam pengerjaan suatu proyek, dimulai dengan menetapkan berbagai kebutuhan dari semua elemen yang diperlukan sistem.
- Analisis**  
Merupakan tahap menganalisis hal-hal yang diperlukan dalam pelaksanaan proyek pembuatan perangkat lunak.
- Design**

Tahap penerjemahan dari data yang dianalisis kedalam bentuk yang mudah dimengerti oleh user.

#### d. Coding

Tahap penerjemahan data atau pemecahan masalah yang telah dirancang kedalam bahasa pemrograman tertentu.

#### e. Pengujian

Merupakan tahap pengujian terhadap perangkat lunak yang dibangun.

## 3.2 Analisis

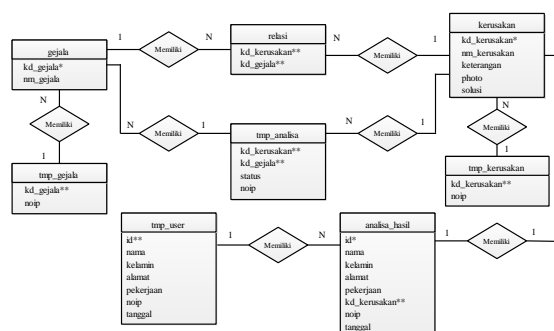
### a. Analisis Sistem

Kurangnya pengetahuan yang cukup dalam penanganan kerusakan *hardware* mengakibatkan sebagian besar masyarakat umum atau suatu institusi tidak dapat mengidentifikasi letak kerusakan yang terjadi pada *hardware* komputernya. Sehingga banyak sekali institusi yang mengeluarkan biaya yang cukup besar hanya untuk memperbaiki kerusakan yang terjadi pada *hardware* komputer kepada pakar *troubleshooting hardware*.

Berdasarkan analisis masalah diatas, maka melalui tugas akhir ini dibuat alternatif penyajian informasi dan konsultasi tentang kerusakan yang terjadi pada *hardware* beserta solusinya yang berbentuk rujukan langkah *troubleshooting* terhadap masalah kerusakan *hardware*

### b. Analisis Basis Data

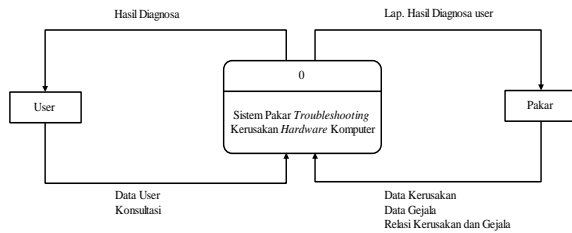
*Entity Relationship Diagram (ERD)* merupakan cara untuk mengorganisasikan data, dimana diagram ini akan memperlihatkan hubungan entitas yang terdapat didalam sistem. ERD diusulkan untuk sistem yang akan dibangun dapat dilihat pada gambar 4 dibawah ini.



Gambar 4. *Entity Relationship Diagram (ERD)*

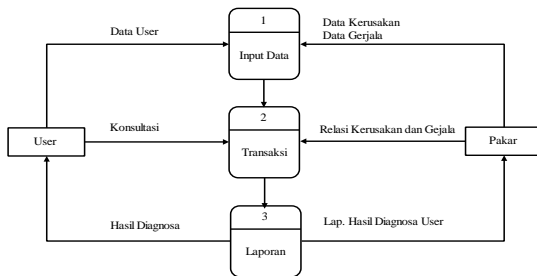
## c. Analisa Kebutuhan Fungsional

### 1. Diagram Kontek



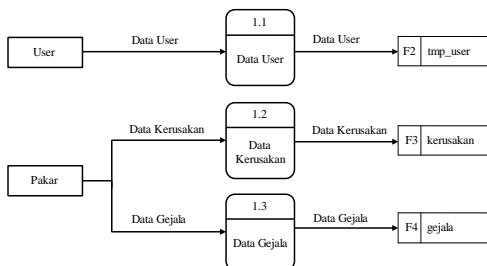
Gambar 5. Diagram Kontek

### 2. DFD Level 0

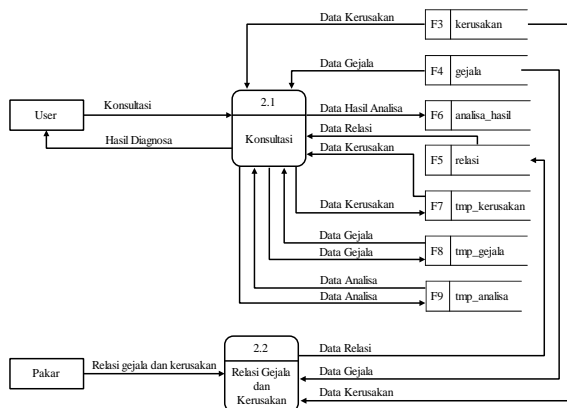


Gambar 6. DFD Level 0

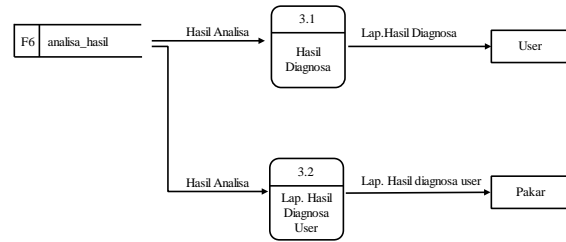
### 3. DFD Level 1



Gambar 7. DFD Level 1 Proses Master Data

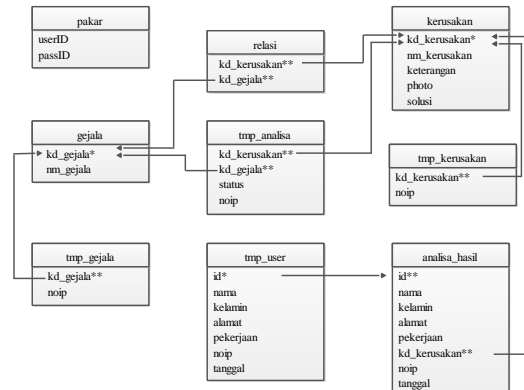


Gambar 8. DFD Level 1 Transaksi



Gambar 9. DFD Level 1 Laporan

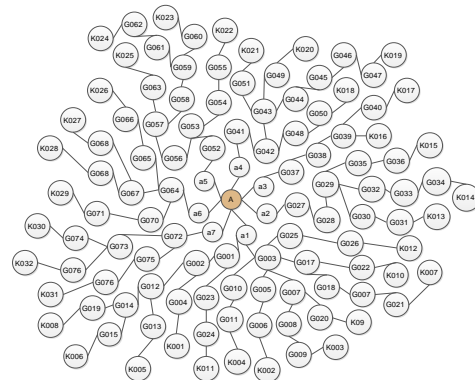
### 4. Skema Relasi



Gambar 10. Skema Relasi

### 5. Pohon Pelacakan

Pohon pelacakan adalah gambaran dari urutan proses yang terjadi dalam sistem, pohon pelacakan ini melakukan proses pelacakan dan penelusuran kerusakan yang didasarkan pada table gejala, proses pelacakannya menggunakan metode *forward chaining*.



Gambar 11. Pohon Pelacakan

### 6. Aturan Kaidah Dan Produksi

Kaidah produksi biasanya dituliskan dalam bentuk *IF-THEN*, kaidah ini dapat dikatakan sebagai hubungan implikasi dua bagian yaitu bagian premis (jika) dan bagian konklusi (maka), apabila bagian premis dipenuhi, maka bagian konklusi juga akan bernilai benar. Sebagai contoh, dapat dilihat *IF-THEN*

*rules* gejala penyakit dari area kerusakan komponen RAM.

**Rule 1 :**

IF Lampu indikator monitor hidup tetapi tidak ada tampilan di monitor

AND Pada saat komputer dihidupkan terdengar bunyi beep terus-menerus

AND Pada saat blue screen terdapat pesan “PEN\_LIST\_CORRUP”

THEN RAM Rusak

**Rule 2 :**

IF Lampu indikator monitor hidup tetapi tidak ada tampilan di monitor

AND Pada saat komputer dihidupkan terdengar bunyi beep terus-menerus

AND Ram tidak terdeteksi

AND Menambah Ram tetapi tetapi tidak terdeteksi

THEN Pemasangan RAM kurang sesuai

**Rule 3 :**

IF Lampu indikator monitor hidup tetapi tidak ada tampilan di monitor

AND Pada saat komputer dihidupkan terdengar bunyi beep terus-menerus

AND RAM tidak terdeteksi

AND Pemasangan RAM sudah benar-benar tepat

AND Komputer hang setelah memasang RAM

THEN RAM tidak Kompatibel

**Rule 4 :**

IF Lampu indikator monitor hidup tetapi tidak ada tampilan di monitor

AND Pada saat komputer dihidupkan terdengar bunyi beep terus-menerus

AND RAM tidak terdeteksi

AND Pemasangan RAM sudah benar-benar tepat

AND Slot RAM jarang dibersihkan

THEN Socket RAM/ RAM kotor

**Rule 5 :**

IF Lampu indikator monitor hidup tetapi tidak ada tampilan di monitor

AND Pada saat dihidupkan terdengar bunyi beep

AND Terdengar bunyi beep 1x panjang 3x pendek

Then VGA rusak

## 7. Struktur Tabel

Tabel 3. File – file Tabel

Kode File	Nama File Tabel
F1	pakar
F2	tmp_user
F3	kerusakan
F4	gejala
F5	relasi
F6	analisa_hasil

F7	tmp_kerusakan
F8	tmp_gejala
F9	tmp_analisa

Tabel 4. Struktur file Tabel “pakar”

No	Nama Kolom	Tipe Data	Lebar	Key	Null Able
1	userID	varchar	50		Not Null
2	passID	varchar	50		Not Null

Tabel 5. Struktur file Tabel “tmp\_user”

No	Nama Kolom	Tipe Data	Lebar	Key	Null Able
1	id	integer	4	Primary Key	Not Null
2	nama	varchar	60		Not Null
3	kelamin	enum	‘laki-laki,’ ‘wanita’		Not Null
4	alamat	varchar	100		Not Null
5	pekerjaan	varchar	60		Not Null
6	noip	varchar	60		Not Null
7	tanggal	datetime	-		Not Null

Tabel 6. Struktur file Tabel “kerusakan”

No	Nama Kolom	Tipe Data	Lebar	Key	Null Able
1	kd_kerusakan	char	6	Primary Key	Not Null
2	nama_kerusakan	varchar	150		Not Null
3	keterangan	text	250		Not Null
4	photo	text	250		Not Null
5	solusi	text	250		Not Null

Tabel 7. Struktur file Tabel “gejala”

No	Nama Kolom	Tipe Data	Lebar	Key	Null Able
1	kd_gejala	char	4	Primary Key	Not Null
2	nama_gejala	varchar	250		Not Null

Tabel 8. Struktur file Tabel “relasi”

No	Nama Kolom	Tipe Data	Lebar	Key	Null Able
1	kd_gejala	char	4	Foreign Key	Not Null
2	kd_kerusakan	char	4	Foreign Key	Not Null

Tabel 9. Struktur file Tabel “analisa\_hasil”

No	Nama Kolom	Tipe Data	Lebar	Key	Null Able
1	id	char	6	Foreign Key	Not Null
2	nama	varchar	60		Not Null
3	kelamin	enum	‘pria’, ‘wanita’		Not Null
4	alamat	varchar	100		Not Null
5	pekerjaan	varchar	60		Not Null
6	kd_kerusakan	char	4	Foreign Key	Not Null
7	noip	varchar	60		Not Null
8	tanggal	datetime	-		Not Null

Tabel 10. Struktur file Tabel “tmp\_kerusakan”

No	Nama Kolom	Tipe Data	Lebar	Key	Null Able
1	kd_kerusakan	char	4		Not Null
2	noip	varchar	60		Not Null

Tabel 11. Struktur file Tabel “tmp\_gejala”

No	Nama Kolom	Tipe Data	Lebar	Key	Null Able
1	kd_gejala	char	4	Foreign Key	Not Null
2	noip	varchar	60		Not Null

Tabel 12. Struktur file Tabel “tmp\_analisa”

No	Nama Kolom	Tipe Data	Lebar	Key	Null Able
1	noip	varchar	60		Not Null
2	kd_kerusakan	char	4	Foreign Key	Not Null
3	kd_gejala	char	4	Foreign Key	Not Null
4	status	enum	‘Ya’, ‘Tidak’		Not Null

### 3.3 Desain

#### 1. Tampilan Menu Utama

Header				
Home	Kerusakan	Konsultasi	Login Pakar	
<p>Selamat Datang Di Website Sistem Pakar Troubleshooting Kerusakan Hardware Komputer</p> <p>Sistem pakar adalah suatu program komputer yang mengandung pengetahuan dari satu atau lebih pakar manusia mengenai suatu bidang spesifik. Jenis program ini pertama kali dikembangkan oleh persei kecerdasan buatan pada dekade 1960-an dan 1970-an dan diterapkan secara komersial selama 1980-an. Bentuk umum sistem pakar adalah suatu program yang dibuat berdasarkan suatu set aturan yang menganalisis informasi (biasanya diberikan oleh pengguna suatu sistem) mengenai suatu kelas masalah spesifik serta untuk memuat dari masalah tersebut. Terkadang, dari desainnya, sistem pakar juga mampu merekomendasikan suatu rangkaian tindakan pengguna untuk dapat menerapkan koreksi. Sistem ini memanfaatkan kapabilitas penalaran untuk mencapai suatu kesimpulan.</p>				
Footer				

Gambar 12. Tampilan Menu Utama

#### 2. Tampilan Konsultasi User

Header				
Home	Kerusakan	Konsultasi	Login Pakar	
<p><b>Konsultasi</b></p> <p>JAWABLAH PERTANYAAN BERIKUT :</p> <p>Pertanyaan Gejala N2250) ?</p> <p>● Ya    ● Tidak</p> <p>Jawab :</p>				
Footer				

Gambar 13. Tampilan Konsultasi User

## 2.4 Implementasi

Tahap implementasi merupakan tahap penerapan sistem supaya dapat dioperasikan. Pada tahap ini dijelaskan mengenai implementasi perangkat lunak dan implementasi perangkat keras.

#### a. Implementasi Perangkat Lunak

Untuk mendukung sistem yang diusulkan berjalan dengan optimal, dibutuhkan *software* pengolahan data, adapun perangkat lunak yang digunakan untuk untuk menjalankan aplikasi sistem pakar sebagai berikut :

1. *Operating System* : Microsoft Windows 7
2. *Browser* : Mozilla Firefox, Google Chrome
3. *Web Server* : Xampp versi 1.7.7
4. *Tool Database* : MySQL

#### b. Implementasi Perangkat Keras

Perangkat keras (*hardware*) yaitu peralatan dalam bentuk fisik yang menjalankan komputer. *Hardware* digunakan sebagai media untuk menjalankan perangkat lunak (*software*) dan peralatan ini berfungsi untuk menjalankan instruksi – instruksi yang diberikan dan mengeluarkannya dalam bentuk informasi yang digunakan oleh manusia untuk laporan. Adapun perangkat keras berdasarkan kebutuhan minimal untuk menjalankan sistem pakar ini sebagai berikut :

1. Server
  - a. Komputer PC Pentium 4
  - b. Hardisk 80 Gb
  - c. Kabel serial dan port com
  - d. Speedy/internet
2. Client
  - a. *Processor* Minimal Pentium 4
  - b. RAM minimal 512 MB
  - c. VGA Card Minimal 128MB
  - d. Speedy/Interne
  - e. Printer

#### c. Implementasi Program

Kegiatan ini bertujuan untuk menerangkan secara singkat penggunaan program sistem pakar kerusakan *hardware* komputer berbasis web. Didalam membangun aplikasi ini, objek studi kasus yang akan dirancang adalah diagnosa *user* terhadap kerusakan dan pendokumentasian seorang pakar komputer. Setelah melalui proses analisis perancangan, proses desain dan pembuatan Aplikasi Sistem Pakar Kerusakan *Hardware* komputer.

### 1. Menu Home

Halaman *Home* digunakan untuk menampilkan halaman awal dari *User*. Adapun tampilannya adalah sebagai berikut :



Gambar 14. Menu Home

### 2. Daftar User

Halaman daftar *user* digunakan untuk mendaftarkan *user* sebelum konsultasi. Adapun tampilannya adalah seperti gambar berikut ini :



Gambar 15. Daftar User

### 3. Menu User Konsultasi

Halaman yang digunakan *user* untuk melakukan diagnosa dengan menjawab pertanyaan yang diajukan oleh sistem. Antarmuka *user* merupakan bagian yang menyediakan sarana untuk *user* agar bisa berkomunikasi dengan sistem dalam bentuk program aplikasi. Antarmuka *user* akan mengajukan beberapa pertanyaan untuk informasi awal dalam pencarian suatu solusi yang akan dilakukan. Adapun tampilannya adalah seperti berikut ini:



Gambar 16. Konsultasi User

### 4. Hasil Analisa

Setelah menjawab beberapa gejala maka akan muncul hasil analisa kerusakan. Adapun tampilannya sebagai berikut :



Gambar 17. Hasil Analisa

### 5. Laporan Hasil Konsultasi



Gambar 18. Laporan Hasil Konsultasi

## 3. PENUTUP

### 3.1 Kesimpulan

Setelah melakukan analisis, perancangan, dan pengujian, maka dapat diperoleh beberapa kesimpulan, antara lain :

1. Aplikasi sistem pakar *troubleshooting* kerusakan *hardware* komputer ini dapat melakukan diagnosis awal terhadap suatu kerusakan serta solusi yang diusulkan untuk mengatasi kerusakan tersebut.
2. Aplikasi sistem pakar *troubleshooting* kerusakan *hardware* komputer ini cukup membantu *user* dalam memperoleh informasi mengenai kerusakan-kerusakan yang sering kali terjadi pada *hardware* serta solusi untuk menanganinya.

### 3.2 Saran

Agar aplikasi sistem pakar *troubleshooting* kerusakan *hardware* komputer ini kedepannya lebih baik, maka yang dapat disarankan adalah :

1. Mengingat besarnya domain pengetahuan yang berkaitan dengan *hardware* komputer, maka ruang lingkup sistem dalam melakukan identifikasi suatu



kerusakan dapat dikembangkan menjadi lebih luas dan lebih kompleks terutama memperbanyak pertanyaan-pertanyaan gejala sehingga dapat menghasilkan kesimpulan yang lebih akurat dan dapat mengatasi permasalahan *hardware* yang cukup kompleks.

2. Memperbaiki dan memperindah tampilan antarmuka untuk menyajikan kenyamanan penggunaan oleh *user*.
3. Menyajikan solusi kerusakan *hardware* yang lebih detail dengan langkah-langkah penanganan yang rinci dan disertai dengan gambar.

#### DAFTAR PUSTAKA

- [1] Andi, *Langkah mudah Troubleshooting Komputer*. Wahana Komputer. Yogyakarta, 2011.
- [2] Arhami, Muhammad, *Konsep Dasar Sistem Pakar*, Andi, Edisi Pertama, Yogyakarta, 2005.
- [3] Jogiyanto, Hartono, *Pengenalan Komputer*, Andi, Edisi Pertama, Yogyakarta, 2002.
- [4] Kusrini, *Sistem Pakar Teori dan Aplikasi*. Andi, Edisi Pertama, Yogyakarta, 2006.
- [5] Kusumadewi, Sri, *Artificial Intelligence*. Graha Ilmu, Edisi Pertama, Yogyakarta, 2003.
- [6] Nugroho Bunafit, *Aplikasi Pemrograman Web Dinamis dengan PHP dan MySQL*, Gava Media, Edisi Pertama, Yogyakarta, 2004.
- [7] Nugroho Bunafit, *Membuat Aplikasi Sistem Pakar dengan PHP dan Editor Dreamweaver*, Gava Media, Yogyakarta, 2008.
- [8] Radiana, Sendy, *Sistem Pakar Troubleshooting Kerusakah Hardware Komputer*. Skripsi UNIKOM.
- [9] Wahyono, Teguh, *PC Troubleshooting Plus*, Gava Media, Edisi Kedua, Yogyakarta, 2008.
- [10] Wedjo, Silvester S., *Mengatasi Masalah Hardware Komputer*, Mediakita, Edisi Pertama, Jakarta, 2007.