

# SISTEM PAKAR UNTUK MENDIAGNOSA KERUSAKAN HARDWARE KOMPUTER DENGAN METODE FORWARD CHAINING DAN BACKWARD CHAINING BERBASIS WEB

Sri Kurniasih, ST.,M.Kom.<sup>1</sup>, Rizki Hardian<sup>2</sup>

<sup>1</sup>Program Studi Manajemen Informatika PKN &STMIK LPKIA

<sup>2</sup>Konsentrasi Teknik Informatika PKN LPKIA

Jln. Soekarno Hatta No. 456 Bandung 40266, Telp. +62 22 75642823, Fax. +62 22 7564282

<sup>1</sup>[sri.kurniasih@yahoo.co.id](mailto:sri.kurniasih@yahoo.co.id), <sup>2</sup>[rizkyhardian@gmail.com](mailto:rizkyhardian@gmail.com)

---

## Abstrak

*Hardware* komputer merupakan alat yang sangat penting dalam suatu sistem komputer, terdapat banyak kasus komputer rusak dikarenakan sesuatu hal yang tidak kita ketahui gejala kerusakan itu tergolong kendala, jenis kerusakan hardware yang mana, sehingga kita harus pergi ke teknisi/pakarnya untuk menemukan solusi dari kasus kerusakan yang ada. Untuk permasalahan ini di rancang sebuah sistem pakar yang dapat memandu masyarakat awam tanpa harus ke teknisi/pakarnya. Sistem pakar ini merupakan penerapan dari hasil analisis kerusakan-kerusakan hardware komputer yang sering terjadi.

Sistem pakar ini dikembangkan dengan menggunakan bahasa pemrograman *PHP*, *Css Framework* sebagai web editor dan *MYSQL* sebagai tempat untuk membuat dan menyimpan database servernya sedangkan metode yang di gunakan adalah Runut maju (*Forward Chaining*) dan Runut balik (*Backward Chaining*), dengan menggunakan metode ini pengguna dapat memberitahukan keluhan kerusakan komputernya ke sistem dengan metode *Forward Chaining* kemudian sistem akan memastikan gejala kerusakan mana yang di alami pengguna sistem di antara beberapa kemungkinan penyebab terjadinya kerusakan komputer yang dipilih oleh user berdasarkan gejala yang terjadi sampai sistem menemukan solusinya, sedangkan metode *Backward Chaining* user dapat memilih kerusakan yang kemungkinan dialami kemudian sistem akan menampilkan daftar gejala dan solusi perbaikan yang ada pada kasus kerusakan tersebut dengan demikian kedua metode tersebut dapat saling melengkapi.

**Kata kunci** : Sistem Pakar, *Forward Chaining*, *Backward Chaining*, *Hardware*

---

## 1. Pendahuluan

Saat ini komputer merupakan sebuah alat bantu manusia yang digunakan dalam menyelesaikan permasalahan, dengan perkembangan teknologi saat ini maka dibutuhkanlah suatu sistem komputer yang terkomputerisasi sehingga dapat membantu user dalam pekerjaannya. Orang-orang dari berbagai kalangan mulai beralih menggunakan teknologi komputer dalam aktivitas sehari-hari nya.

*Hardware* komputer merupakan alat yang sangat penting dalam suatu sistem komputer, terdapat banyak kasus komputer rusak dikarenakan sesuatu hal yang tidak kita ketahui gejala kerusakan itu tergolong kendala atau jenis kerusakan hardware yang mana, sehingga kita harus pergi ke teknisi/pakarnya untuk menemukan solusi dari kasus kerusakan yang ada. Oleh karena itu dibutuhkan sebuah media/alat bantu

untuk menyelesaikan masalah itu secara cepat sehingga user tidak perlu pergi ke teknisi/pakarnya. dalam kasus ini peneliti akan mengambil tema tentang sistem pakar yang bermaksud untuk membantu/mendiagnosa masalah yang ada di dalam komputer anda sehingga kerusakan pada komputer dapat segera diketahui solusi terbaiknya tanpa perlu datang ke teknisi/pakarnya sehingga lebih menghemat waktu.

Perancangan Sistem pakar ini merupakan penerapan dari hasil analisis terhadap berbagai jenis kerusakan-kerusakan yang terjadi pada hardware komputer. Adapun pembangunan sistem pakar ini menggunakan bahasa pemrograman *PHP*, dengan *Css framework* sebagai web editor dan *MY SQL* sebagai database servernya. Sistem pakar yang di rancang tidak terlepas dari metode, untuk itu metode yang digunakan yaitu dengan menggunakan Runut maju

(*Forward Chaining*) dan Runut balik (*Backward Chaining*). Dengan penggabungan dua metode ini dapat memandu pengguna untuk dapat memberitahukan keluhan kerusakan komputernya ke system. Metode *Forward Chaining* digunakan untuk memastikan gejala kerusakan mana yang dialami pengguna sistem diantara beberapa kemungkinan penyebab terjadinya kerusakan komputer yang dipilih oleh user berdasarkan gejala yang terjadi sampai sistem menemukan solusinya. Sedangkan metode *Backward Chaining* user dapat memilih kerusakan yang kemungkinan dialaminya kemudian sistem akan menampilkan daftar gejala dan solusi perbaikan yang ada pada kasus kerusakan tersebut dengan demikian kedua metode tersebut dapat saling melengkapi.

### 1.1. Rumusan Masalah

Berdasarkan uraian di atas, maka dapat di rumuskan permasalahan yang muncul sebagai berikut :

1. Terbatasnya pengetahuan user (orang awam) dalam menangani kerusakan.
2. Kurang efektifnya sistem konvensional karena mengharuskan user untuk pergi ke seorang ahli sehingga membuang waktu dan biaya lebih.
3. Kurangnya media komunikasi dan visualisasi antara ahli dan user mengenai konsultasi kerusakan komputer.

### 1.2. Tujuan Penelitian

Adapun tujuan penelitian adalah untuk :

1. Memberikan arahan dan informasi kerusakan awal dan solusi awal tentang bagaimana cara mengatasi suatu permasalahan kerusakan pada komputer
2. Memberikan kemudahan dan kecepatan dalam memproses kendala kendala yang user alami sehingga lebih menghemat waktu dan biaya
3. Memberikan fasilitas kepada user untuk menyampaikan keluhan - keluhan yang berkaitan dengan masalah *hardware* komputer.

### 1.3. Ruang Lingkup

Agar penelitian lebih fokus pada permasalahan, maka sejumlah batasan masalah dan asumsi yang ada dapat di uraikan sebagai berikut :

1. Isi yang di sajikan dalam websitenya hanya yang berkaitan dengan hardware komputer dan masalah masalah yang ada di basis pengetahuan hanya meliputi *power supply, hardisk, vga card, Motherboard, Ram, sound card.*
2. Metode yang digunakan yaitu *Forward chaining* dan *Backward chaining* kedua metode tersebut bukan merupakan gabungan melainkan saling melengkapi yang diaplikasikan di halaman yang berbeda

3. Implementasi metode *Backward chaining* yang ada di halaman kerusakan bersifat statis yaitu pengguna hanya bisa memastikan dan memilih kemudian menampilkan kerusakan yang ada.

## 2. Dasar Teori

Sistem pakar (*expert system*) adalah sistem yang berusaha mengadopsi pengetahuan manusia ke komputer, agar komputer dapat menyelesaikan masalah seperti yang biasa dilakukan oleh para ahli. (Kusumadewi, 2003) Sistem Pakar disusun oleh dua bagian utama, yaitu lingkungan pengembangan (*development environment*) dan lingkungan konsultasi (*consultation environment*) (Minarni, Rahmad hidayat, 2013).

Komponen-komponen yang terdapat dalam sistem pakar terdiri dari yaitu *User Interface* (antarmuka pengguna), basis pengetahuan, akuisisi pengetahuan, mesin inferensi, *workplace*, fasilitas penjelasan, perbaikan pengetahuan.

### 1. Antarmuka Pengguna (*User Interface*)

*User Interface* merupakan mekanisme yang digunakan oleh pengguna dan sistem pakar untuk berkomunikasi. Antarmuka menerima informasi dari pemakai dan mengubahnya ke dalam bentuk yang dapat diterima oleh sistem. Selain itu antarmuka menerima informasi dari sistem dan menyajikan ke dalam bentuk yang dapat dimengerti oleh pemakai. Pada bagian ini terjadi dialog antara program dan pemakai, yang memungkinkan sistem pakar menerima instruksi dan informasi (*input*) dari pemakai, juga memberikan informasi (*output*) kepada pemakai (McLeod, 1995).

### 2. Basis Pengetahuan

Basis pengetahuan mengandung pengetahuan untuk pemahaman, formulasi, dan penyelesaian masalah. Komponen sistem pakar ini disusun atas dua elemen dasar, yaitu fakta dan aturan. Fakta merupakan informasi tentang obyek dalam area permasalahan tertentu, sedangkan aturan merupakan informasi tentang cara bagaimana memperoleh fakta baru dari fakta yang telah diketahui

### 3. Akuisisi pengetahuan (*Knowledge Acquisition*)

Akuisisi pengetahuan adalah akumulasi, transfer dan transformasi keahlian dalam menyelesaikan masalah dari sumber pengetahuan kedalam program komputer. Turban (2001) dalam pendapatnya mengungkapkan bahwa terdapat tiga metode utama dalam akuisisi pengetahuan, yaitu wawancara, analisis protocol, observasi pada pekerjaan pakar dan induksi aturan dari contoh.

#### 4. Mesin Inferensi

Mekanisme *inferensi* yang utama pada sistem pakar dapat dibedakan menjadi inferensi dengan mekanisme pelacak mundur (*Backward chaining*) dan pelacak maju (*forward chaining*). Penalaran dengan *Backward chaining* dimulai dari sekumpulan hipotesis menuju fakta-fakta yang mendukung hipotesis tersebut. *Forward chaining* merupakan kebalikan dari *Backward chaining*, yaitu penalaran dimulai sekumpulan data menuju suatu kesimpulan atau goal.

#### 5. Workplace

*Workplace* merupakan area dari sekumpulan memori kerja (*working memory*). *Workplace* digunakan untuk merekam hasil-hasil antara dan kesimpulan yang dicapai. Ada 3 tipe keputusan yang dapat direkam yaitu rencana, agenda dan solusi.

Dalam melakukan inferensi diperlukan adanya proses pengujian kaidah-kaidah dalam urutan tertentu untuk mencari yang sesuai dengan kondisi awal atau kondisi yang berjalan yang sudah dimasukkan pada basis data. Perunutan adalah proses pencocokan fakta, pernyataan atau kondisi berjalan yang tersimpan pada basis pengetahuan maupun pada memori kerja dengan kondisi yang dinyatakan pada premis atau bagian kondisi pada kaidah.

Runut maju (*Forward Chaining*) merupakan proses perunutan yang dimulai dengan menampilkan kumpulan data atau fakta yang meyakinkan menuju konklusi akhir. Runut maju bisa juga disebut sebagai penalaran *forward* (*forward reasoning*) atau pencarian yang dimotori data (*data driven search*). Jadi dimulai dari premis-premis atau informasi masukan (*if*) dahulu kemudian menuju konklusi atau *derived information* (*then*). Informasi masukan dapat berupa data, bukti, temuan, atau pengamatan. Sedangkan konklusi dapat berupa tujuan, hipotesa, penjelasan, atau diagnosis. Sehingga jalannya penalaran runut maju dapat dimulai dari data menuju tujuan, dari bukti menuju hipotesa, dari temua menuju penjelasan, atau dari pengamatan menuju diagnosa.

Runut balik (*Backward*) merupakan proses perunutan yang arahnya kebalikan dari runut maju. Proses penalaran runut balik dimulai dengan tujuan/goal kemudian merunut balik ke jalur yang akan mengarahkan ke goal tersebut, mencari bukti-bukti bahwa bagian kondisi terpenuhi. Jadi secara umum runut balik itu diaplikasikan ketika tujuan atau hipotesis yang dipilih itu sebagai titik awal penyelesaian masalah dan bisa juga disebut dengan *goal-driven search*.

*Unified Modeling Language* (UML) adalah “*Salah satu standar bahasa yang digunakan dalam*

*dunia industri untuk mendefinisikan requirement, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemograman berorientasi objek.*” (Rosa dan M. shalahuddin, 2013:133)

Dengan menggunakan UML kita dapat membuat model untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta dapat digunakan dalam bahasa pemograman apapun. Tetapi karena UML juga menggunakan *class* dan *operation* dalam konsep dasarnya, maka UML lebih cocok untuk penulisan piranti perangkat lunak dalam bahasa-bahasa berorientasi objek seperti C++, Java, C# atau Basic. Sedangkan Perangkat Lunak merupakan sebuah program yang didalamnya terdapat perintah-perintah yang terdokumentasi.

### 3. Hasil Penelitian

Hasil analisis dan perancangan dimodelkan menggunakan diagram UML (*Unified Modeling Language*) yang merupakan bahasa standar pemodelan visual (*visual modelling*) dalam rekayasa perangkat lunak yang berorientasi objek. Pembangunan perangkat lunak ini berbasis web, dibuat dengan menggunakan PHP. Perangkat lunak yang dirancang menggunakan konsep *client server* dan dapat dijalankan dengan menggunakan komputer yang telah terinstal (aplikasi) *web browser*. Terdapat dua kategori pengguna dalam aplikasi ini yakni *user* dan *administrator*, pembagian kategori dimaksudkan untuk membatasi hak akses. Pengguna dengan hak akses *user* dapat mengakses halaman publik yang merupakan halaman yang dapat diakses oleh *user* (pengunjung website).

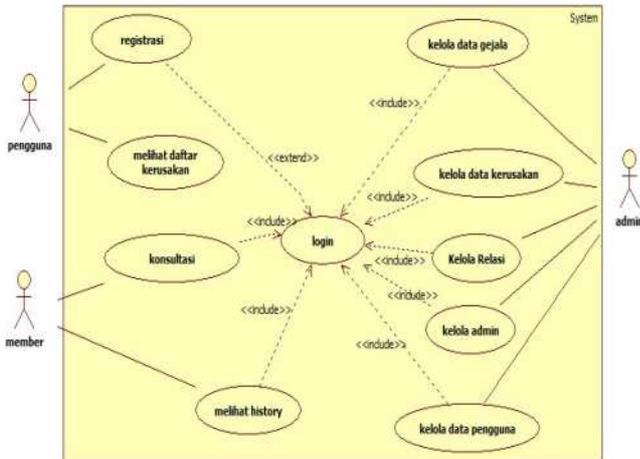
Pada halaman ini tersedia berbagai fitur untuk melakukan registrasi *user*, *loginuser*, mengubah akun *loginuser*, mengisi dan mengubah data profil *user*, melihat daftar kerusakan, konsultasi, Sedangkan pengguna dengan hak akses administrator memiliki hak akses penuh untuk melakukan berbagai proses transaksi seperti melihat history, kelola data gejala, kelola data kerusakan, kelola relasi dan kelola pengguna.

#### 3.1 Use Case Diagram

*Use cases* diagram digunakan untuk memodelkan bisnis proses berdasarkan *prespektif* pengguna sistem. *Use case* diagram terdiri atas diagram *use case* dan *actor*. *Use case* diagram dapat mengilustrasikan kebutuhan sistem, sedangkan *actor* meliputi semua yang ada diluar sistem yang berinteraksi dengan sistem perangkat lunak.

Setiap *use case* disertai dengan penjelasan yang diuraikan dalam *use case scenario*, meliputi nama *use case*, aksi *actor* dan *response* dari perangkat lunak.

Use case diagram dalam Sistem Pakar untuk mendiagnosa kerusakan pada hardware komputer dapat digambarkan sebagai berikut.



**Gambar 3.1 Use case diagram Sistem Pakar untuk mendiagnosa kerusakan pada hardware komputer**

### 3.2 UseCaseScenario

Use case scenario merupakan bagian dari use case diagram yang berisi penjelasan tekstual dari setiap use case, mulai dari nama use case, keterangan, kondisi awal, kondisi akhir, actor yang terlibat, serta use case pengecualian jika terdapat hubungan extend dalam diagram use case.

**Tabel 3.1 Use case scenario Registrasi**

Aksi aktor	Reaksi Sistem
1. aktor memilih menu Registrasi	2. menampilkan halaman Registrasi
3. aktor mengisi semua data isian yang telah disediakan	4. menyimpan dan memverifikasi data user di database
Kondisi Akhir	Aktor telah berhasil melakukan registrasi
Alternatif Skenario	
1. aktor memilih menu Registrasi	
2. aktor langsung mengakses menu aplikasi konsultasi	3. sistem tidak merespon action karena belum Registrasi / login
4. aktor tidak mengisikan salah satu data Registrasi	5. sistem akan memunculkan informasi bahwa data belum lengkap

**Tabel 3.2 Use case scenario Konsultasi**

Aktor	Sistem
1. aktor memilih menu konsultasi	2. Menampilkan halaman konsultasi
3. memulai proses konsultasi	4. menampilkan pertanyaan mengenai gejala-gejala yang dialami
5. menjawab Ya atau Tidak pada pertanyaan yang diajukan oleh sistem	6. menampilkan hasil dan solusi
Kondisi akhir	Pengguna telah selesai melakukan proses konsultasi
Alternatif Skenario	
1. Aktor memilih menu konsultasi	2. Menampilkan halaman konsultasi
3. Aktor tidak memilih pertanyaan	4. Sistem tidak akan menampilkan solusi
5. aktor memilih pertanyaan yang tidak sesuai dengan rule basis pengetahuan	6. sistem akan memberikan pemberitahuan bahwa gejala yang di pilih tidak terdiagnosa

**Tabel 3.3 Use case scenario kelola data gejala**

Aktor	Reaksi Sistem
1. aktor melakukan login admin	2. sistem menampilkan form login
3. aktor masuk ke menu gejala	4. menampilkan menu gejala
5. aktor menyimpan gejala kerusakan baru	6. menyimpan gejala baru di database
7. merubah gejala kerusakan lama	8. merubah data gejala kerusakan lama menjadi baru ke database
9. menghapus gejala kerusakan	10. menghapus data gejala kerusakan yang telah di pilih 11. logout
Kondisi akhir	Admin berhasil mengelola data gejala kerusakan
Alternatif Skenario	
1. aktor tidak melakukan login admin	2. sistem akan memberi peringatan bahwa hanya admin yang dapat mengakses menu tersebut
3. aktor menginputkan data yang sama aktor	4. sistem akan memberikan peringatan bahwa data telah ada
5. menyimpan gejala baru tapi tidak	6. sistem akan memberikan peringatan bahwa data

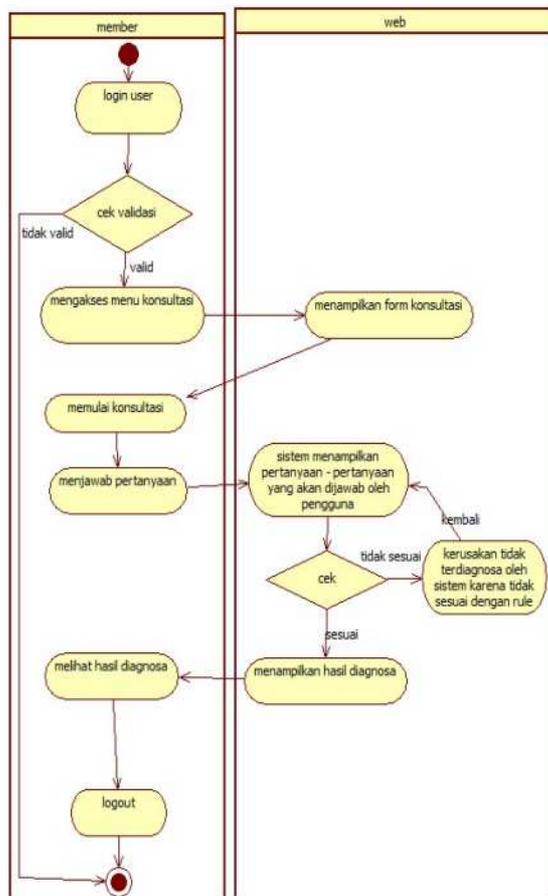
mengisi semua datanya	harus di isi semua
7. aktor merubah gejala baru tapi tidak mengisi semua datanya	8. sistem akan memberikan peringatan bahwa data harus di isi semua

### 3.3. Aliran Kerja

Dalam pembuatan aliran kerja dimodelkan dengan menggunakan *activity diagram*, yang dimaksudkan untuk memberikan gambaran alur aktivitas perangkat lunak dari satu aktivitas ke aktivitas lainnya. Diagram aktivitas atau *activity diagram* adalah sebuah cara untuk memodelkan aliran kerja atau *workflow* dari *use case* dalam bentuk *grafik*, diagram ini menunjukkan langkah-langkah didalam aliran kerja, titik-titik keputusan didalam aliran kerja, siapa yang bertanggung jawab menyelesaikan masing-masing aktivitas, dan objek-objek yang digunakan dalam aliran kerja.

#### 3.3.1 ActivityDiagram

Gambar di bawah ini menjelaskan sebuah aktifitas mengenai perangkat lunak sistem Pakar.



**Gambar 3.2** *Activity diagram* konsultasi

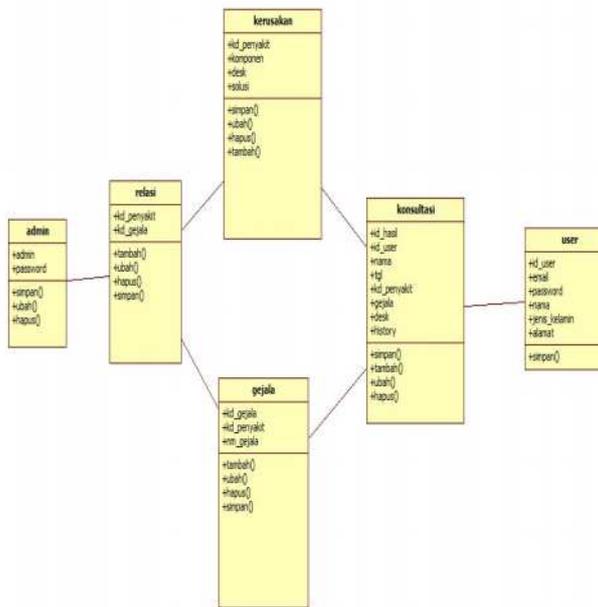
#### 3.3.2. Uraian Workflow

Pada sub bab ini akan menjelaskan diagram aktifitas perangkat lunak Sistem Pakar secara tekstual yang disajikan poin-poin dari pemodelan perangkat lunak yang dikembangkan. Dibawah ini uraian activity diagram Konsultasi:

1. Member melakukan login untuk dapat melakukan diagnosa
2. Sistem melakukan validasi pada login Member
3. Apabila login gagal maka Member tidak dapat mengakses menu diagnosa dan secara otomatis akan menampilkan form login kembali
4. Jika Member telah sukses melakukan login maka Member dapat masuk ke halaman konsultasi
5. Member mengakses menu konsultasi
6. Sistem menampilkan halaman konsultasi
7. Member memulai konsultasi
8. Member menjawab pertanyaan yang diajukan oleh sistem berupa gejala-gejala yang dialami dengan format "YA" atau "TIDAK"
9. Sistem mengecek apakah jawaban yang telah dipilih oleh Member sesuai dengan rule atau tidak, apabila tidak maka Member akan kembali diberi pertanyaan kembali
10. Apabila hasil sudah sesuai rule maka sistem akan menampilkan hasil proses konsultasi
11. Member melakukan logout

### 3.4. Pemodelan Data

Dalam pembuatan pemodelan data ini dimodelkan dengan menggunakan *class diagram* yang menggambarkan *class object*, dan *class object description* untuk menjelaskan fungsi, setiap atribut yang digunakan dan *method* atau operasi yang dimilikinya.

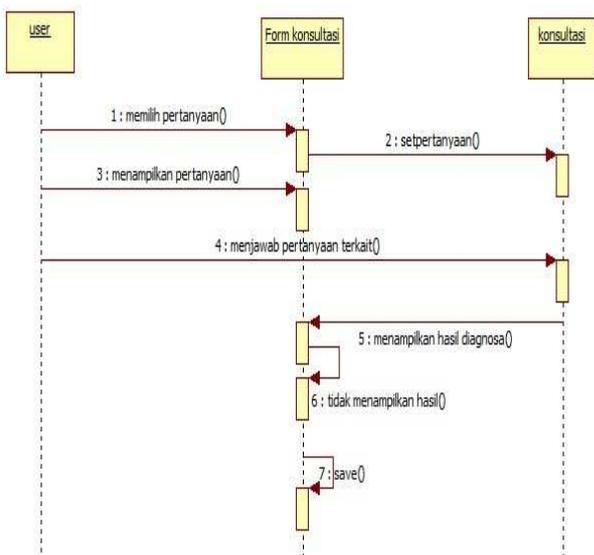


**Gambar 3.3 Class diagram sistem pakar untuk mendiagnosa kerusakan hardware**

### 3.5. Struktur Organisasi Obyek dan Pesan

Dalam sub bab ini dibahas mengenai bagaimana memodelkan interaksi antara objek-objek dalam sistem, ada dua tipe diagram interaksi yakni *sequence diagram* diagram dan *collaboration diagram*. Pada pemodelan perangkat lunak ini akan digambarkan *sequence diagram* diagram dan *collaboration diagram* selain itu juga ada narasi sebagai penjelasan dari model yang telah dibuat.

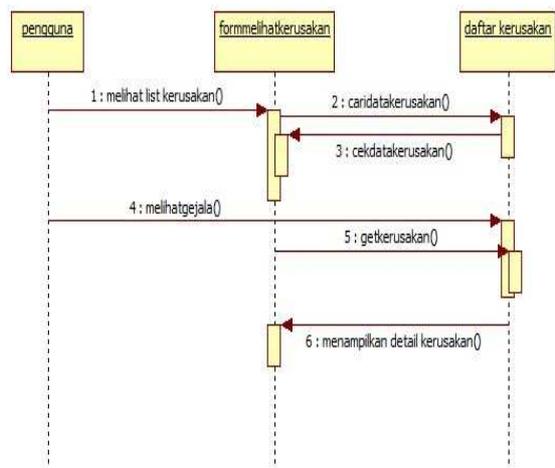
#### 3.5.1. Sequence Diagram



**Gambar 3.4 Sequence diagram Konsultasi kerusakan**

#### Narasi *sequence diagram* untuk *use case* diagram konsultasi kerusakan :

Uraian tekstual dari gambar 3.4 menggambarkan sebuah aktor mengirim pesan kepada objek satu ke objek lainnya, yakni aktor User memilih pertanyaan sesuai dengan kerusakan, dengan memanggil method memilih pertanyaan() kemudian mengirim ke object form konsultasi dengan memanggil method set pertanyaan(). object ini menampilkan beberapa pertanyaan yang dipanggil melalui object konsultasi. Setelah menampilkan pertanyaan kemudian user menjawab semua daftar pertanyaan yang dikirim melalui method menjawab pertanyaan terkait() sehingga object konsultasi menampilkan hasil diagnosa() sebagai hasil untuk diperiksa dan Menjawab pertanyaan yang diajukan oleh sistem dan tidak menampilkan hasil() jika daftar jawaban kurang memenuhi persyaratan dari daftar pertanyaan.



**Gambar 3.5 Sequence diagram untuk melihat daftar kerusakan**

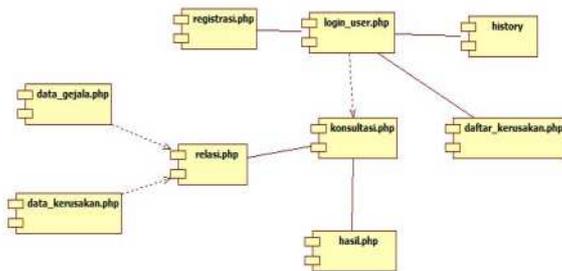
Narasi *sequence diagram* diatas adalah Aktor Pengguna mencari data kerusakan melalui method melihat list kerusakan() untuk memanggil formmelihatkerusakan yang kemudian akan akan mengirim method caridatakerusakan() untuk memanggil object daftar\_kerusakan untuk mengecek data kerusakan yang di pilih dan mengirim method tsb ke formmelihatkerusakan agar dapat di tampilan kembali untuk user. Setelah user melihat daftar gejala kerusakan method memanggil getkerusakan() untuk menampilkan kembali daftar kerusakan

### 3.6. Pemodelan Pemrosesan Sistem

Sub bab ini akan menjelaskan secara terperinci mengenai struktur perangkat lunak dan kebutuhan spesifikasi *hardware* dan *software* untuk implementasi pengembangan perangkat lunak yang digambarkan dalam *deployment* diagram dan *component* diagram

### 3.6.1 Component Diagram

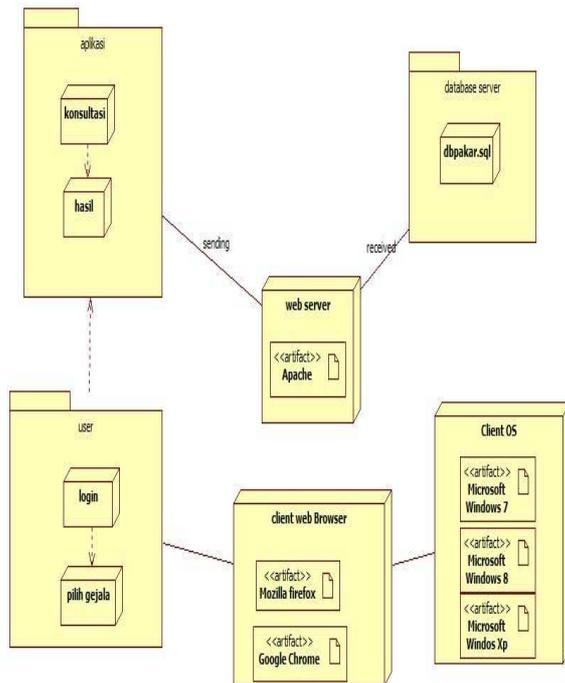
Menggambarkan tentang *component* apa saja yang saling berhubungan didalam suatu sistem.



**Gambar 3.6** *Component diagrams* sistem pakar untuk mendiagnosa kerusakan hardware komputer

Pada gambar 3.6 dapat dijelaskan bahwa setiap modul pada perangkat lunak sistem pakar ini saling berhubungan, dimulai data\_gejala.php dan data\_kerusakan.php terhadap modul relasi.php. Kemudian login\_user.php berhubungan dengan history.php, daftar\_kerusakan.php dan konsultasi.php, sedangkan hasil sangat bergantung pada module konsultasi.php

### 3.6.2. DeploymentDiagram



**Gambar 3.7** Deployment diagram sistem pakar mendiagnosa kerusakan hardware komputer

Gambar 3.7 menggambarkan deployment diagram yang menjelaskan tentang pemrosesan yang dilakukan oleh user dilakukan dengan memanggil sistem melalui client OS dan Client Web Browser mengakses dan meminta layanan melalui aplikasi Browser kemudian dari dua node tersebut user dapat mengakses aplikasi dan data yang di tampung oleh aplikasi langsung berhubungan dengan web server kemudian database server memberikan sebuah basis pengetahuan menggunakan akses internet

### 3.7. Perancangan Antar Muka

Nama Antarmuka : Halaman utama  
 Fugsi : Menu utama dari aplikasi sistem pakar untuk mendiagnosa kerusakan hardware komputer, yang meliputi menu home, registrasi,

about, help, kerusakan dan sign-in

Bukti :



**Gambar 3.8** Halaman utama perangkat lunak

Nama Antarmuka : Halaman Menu Registrasi  
 Fugsi : Setelah memilih menu registrasi maka akan muncul tampilan seperti ini pengguna harus mengisi form-form yang ada seperti nama, password, email, jenis kelamin dan alamat setelah di isi semua klik simpan

Bukti :



**Gambar 3.9** Halaman Menu Registrasi Pengguna

Nama Antarmuka : Halaman Menu Konsultasi  
 Fugsi : Setelah login, pengguna dapat melakukan proses konsultasi dengan cara memilih menu konsultasi, pengguna dapat memulai proses konsultasi dengan memilih tombol mulai konsultasi

Bukti :



**Gambar 3.10**Halaman Menu Konsultasi

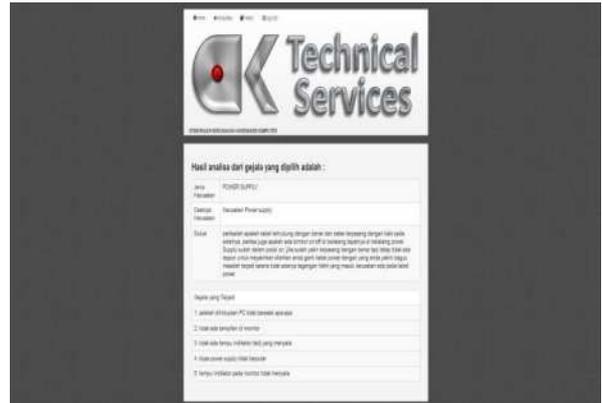
Nama Antarmuka : Halaman Proses Konsultasi  
 Fugsi : Proses konsultasi pengguna dengan sistem pakar . pada menu ini pengguna dapat menjawab beberapa pertanyaan seputar kerusakan hardware.

Bukti :



**Gambar 3.11**Halaman Proses Konsultasi

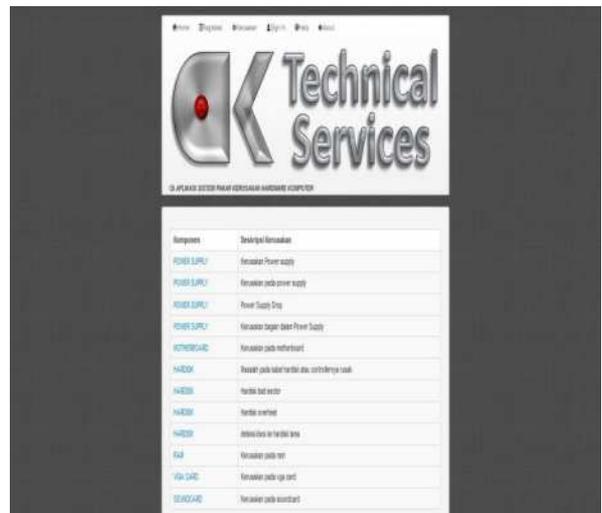
Nama Antarmuka : Halaman Hasil Konsultasi  
 Fugsi : Menampilkan hasil konsultasi pengguna dengan sistem pakar  
 Bentuk :



**Gambar 3.11** Halaman Hasil Konsultasi

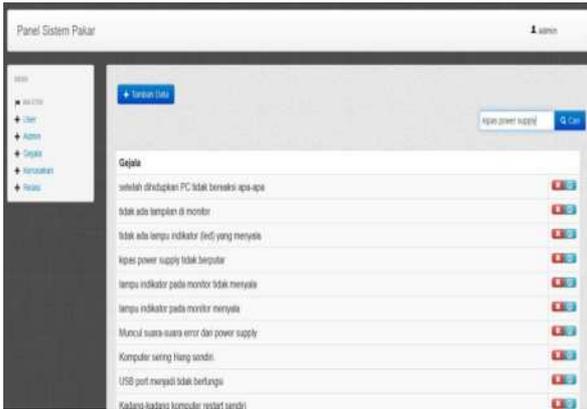
Nama Antarmuka : Lihat List Daftar kerusakan Fugsi : Memberikan pelayanan terhadap pengguna dimana Pengguna dapat melihat list daftar kerusakan yang ada pada daftar, klik salah satu list kerusakannya maka akan muncul detail dari kerusakan tersebut

Bentuk :



**Gambar 3.12**Lihat List Daftar kerusakan

Nama Antarmuka : Halaman Gejala Kerusakan Fugsi : Menampilkan beberapa gejala kerusakan yang ada sebagai list untuk konsultasi pengguna dengan sistem pakar  
 Bentuk :



**Gambar 3.12 Halaman Daftar Gejala**

Nama Antarmuka : Halaman Menu Kerusakan  
 Fungsi : form ini digunakan untuk Admin pada saat menambah kerusakan baru, merubah, mencari dan menghapus kerusakan yang ada untuk proses menyimpan klik icon tambah data untuk menuju ke proses selanjutnya  
 Bentuk :



**Gambar 3.12 Halaman Menu untuk menambah, merubah dan menyimpan Gejala kerusakan**

#### 4. Implementasi Dan Pengujian

##### 4.1. Implementasi

Rencana implementasi merupakan tahapan awal dari penerapan perangkat lunak yang akan dibuat dengan tujuan agar perangkat yang dibuat dapat berjalan sesuai dengan rencana yang diharapkan. Langkah-langkah kegiatan dalam mengimplementasikan perangkat lunak sistem pakar ini sebagai berikut :

1. Perencanaan  
 Merencanakan setiap kegiatan dan tahapan yang akan dilakukan untuk setiap kebutuhan perangkat lunak sistem pakar.
2. Analisis Kebutuhan Perangkat Lunak

Tahap ini peneliti melakukan analisa terhadap user mengenai data kerusakan hardware, kebutuhan aplikasi, metode yang tepat untuk menentukan solusi masalah hardware, menyusun dan mengumpulkan data mengenai gejala dan kemungkinan kerusakan yang sering terjadi pada hardware komputer, mengumpulkan data dari para ahli/pakar sebagai solusi untuk permasalahan yang ada.

3. Desain Perangkat Lunak  
 Membuat design perangkat lunak meliputi desain antar muka, desain database, interface input dan output, dll.
4. Pengkodean  
 Membangun atau membuat aplikasi sistem pakar sesuai dengan apa yang telah direncanakan sebelumnya.
5. Pengujian  
 Menguji aplikasi yang telah dibangun. Uji coba dilakukan terhadap banyak user untuk menguji setiap fungsi yang ada apakah berjalan dengan baik atau tidak dan menyesuaikan kembali dengan setiap kemungkinan jawaban yang di berikan user sebagai data tambahan untuk pengembangan aplikasi selanjutnya.

#### 4.2. Lingkup dan Batasan Implementasi

Pembatasan implementasi dimaksudkan agar ruang lingkup implementasi menjadi lebih jelas. Lingkup dan batasan implementasi dari perangkat lunak sistem pakar ini sebagai berikut :

1. Aplikasi sistem pakar ini meliputi pembahasan tentang hardware komputer dan permasalahan – permasalahan yang sering terjadi .
2. Pada saat user melakukan konsultasi, user harus memilih beberapa gejala yang dialami sehingga sistem dapat menuntun user untuk menemukan kerusakan atau masalah yang dihadapi
3. Sistem pakar ini hanya memberikan informasi gejala dan kerusakan hardware, sehingga menu aplikasi masih terbatas dan perlu tambahan data serta uji kelayakan kembali untuk pengembangan bagi peneliti.

#### 4.2 Kebutuhan Sumberdaya

Didalam penggunaan perangkat lunak sistem pakar ini dibutuhkan spesifikasi yang harus menunjang dengan perangkat lunak ini, diantaranya diperlukan beberapa komponen sebagai berikut :

##### Hardware

1. Processor Pentium IV 2,0 Ghz
2. Harddisk 20 GB
3. Ram 512 GB
4. VGA Card Onboard 128 Mb
5. Mouse dan Keyboard Standard

##### Software

1. Microsoft Window XP, 7 , 8. 32 bit dan 64 bit
2. Browser : Mozilla Firefox, Internet Explorer, Google Chrome
3. Adobe reader
4. xampp 1.7.4

#### Brainware

Kebutuhan sumber daya lain selain kebutuhan akan perangkat keras dan perangkat lunak agar sistem yang telah dibuat dapat berjalan dengan semestinya yaitu SDM. Kebutuhan SDM disini adalah *ahli hardware, user* yang berperan untuk menggunakan perangkat lunak sistem pakar berbasis web yang telah dibuat.

#### 4.4. Pengujian

Pengujian Sistem Pakar berbasis web ini menggunakan pengujian *black-box* yaitu menguji perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program, dimaksudkan untuk mengetahui apakah fungsi-fungsi, masukan, dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan seperti di uraikan di bawah ini :

- a. Pengujian terhadap fungsi – fungsi pada setiap form
- b. Pengujian terhadap kesalahan antar muka
- c. Pengujian untuk memeriksa kesalahan pada kinerja sistem
- d. Pengujian terhadap validasi form form yang ada
- e. Pengujian terhadap proses konsultasi user dengan system
- f. Pengujian untuk menampilkan hasil diagnosa sesuai gejala apa saja yang dipilih user sesuai rule yang telah di susun
- g. Menguji *Hyperlink* setiap menu
- h. Fungsionalitas menu admin dan user

**Tabel 3.4** Pengujian login admin & fungsionalitas mode administrator

No	Fungsi Yang diuji	Cara Pengujian	Hasil Yang Diharapkan	Hasil Pengujian
1	Login ke mode administrator	Memasukan username & password	admin dapat masuk dan mengakses aplikasi	ok
2	Tombol cari data pengguna	Mencari nama pengguna	admin dapat mencari nama pengguna	ok
3	Tombol rubah data admin	Merubah data admin	admin dapat merubah data admin	ok
4	Tombol cari data admin	Mencari data admin	admin dapat mencari data admin	ok
5	Tombol tambah data gejala	Menambah data gejala	admin dapat menambah data gejala	ok

6	Tombol rubah data gejala	Merubah data gejala	admin dapat merubah data gejala	ok
7	Tombol hapusdata gejala	Menghapus data gejala	admin dapat menghapus data gejala	ok
8	Tombol cari data gejala	Mencari data gejala	admin dapat mencari data gejala	ok
9	listbox data gejala	Memilih gejala sesuai kerusakan	admin dapat memilih data gejala sesuai kerusakan	ok
10	Tombol tambah data kerusakan	Menambah kerusakan komponen	admin dapat menambah data kerusakan komponen	ok
11	Tombol rubah data kerusakan	Merubah kerusakan komponen	admin dapat merubah data kerusakan komponen	ok
12	Tombol cari data kerusakan	Mencari kerusakan komponen	admin dapat mencari data kerusakan komponen	ok
13	Tombol tambah solusi	Menambah solusi sesuai kerusakan	admin dapat mnambah solusi sesuai kerusakan	ok

#### 5. Kesimpulan dan Saran

##### 5.1. Kesimpulan

Dari semua uraian yang telah peneliti kemukakan dapat peneliti simpulkan sebagai berikut :

1. Aplikasi sistem pakar ini dapat membantu user khususnya orang awam untuk menyelesaikan masalah kerusakan yang dialaminya
2. Dengan Implementasi berbasis web aplikasi ini dapat di akses dimana saja, cepat dan mudah untuk digunakan.
3. Aplikasi sistem pakar ini dapat memberikan fasilitas konsultasi untuk user layaknya seorang teknisi komputer sesuai ruang lingkup sistem.

##### 5.2. Saran

Berdasarkan kesimpulan yang telah peneliti kemukakan, maka peneliti merekomendasikan beberapa saran untuk pengembangan perangkat lunak sebagai berikut :

1. Diharapkan penelitian berikutnya dapat memperluas isi database yang lebih kompleks sehingga aplikasi sistem pakar ini lebih lengkap dalam hal data-data tentang hardware komputer.
2. Penyempurnaan terhadap implementasi metode *forward chaining* dan *backward chaining*, agar user dapat berkonsultasi lebih baik lagi.

Penggabungan kedua metode ini akan semakin dinamis karena keduanya memakai sistem pertanyaan.

3. Jika Aplikasi ini sudah diimplementasikan di dunia nyata maka kedepannya diharapkan agar bisa mendukung multiplatform sehingga pengguna dapat leluasa menggunakan aplikasi ini.

#### **Daftar Pustaka**

1. Rosa A.S M Shalahudin 2013. “*Rekayasa Perangkat Lunak*”, Informatika, Bandung.
2. Sri Hartati, Sari Iswanti. 2012, “ Sistem Pakar & Pengembangannya”, Graha Ilmu.
3. Arya Dipanegara. 2011, “ Komputer Klinik solusi perbaikan dan perawatan Komputer”, Agobos.
4. Wahana Komputer. 2011, Langkah Mudah Troubleshooting Komputer, Andi Yogyakarta.
5. Minarni, Rahmad Hidayat. 2013, Jurnal Penelitian Institut Teknologi Padang, Rancang bangun Aplikasi Sistem Pakar Untuk Kerusakan Komputer Dengan Metode Backward Chaining.