

# SISTEM PAKAR UNTUK MENGIDENTIFIKASI KERUSAKAN IPHONE 5 DENGAN METODE BACKWARD CHAINING BERBASIS CLIENT-SERVER

**Ghalagghar Zakario Zalfano**

*Program Studi Teknik Informatika, Fakultas Teknologi Informasi, Universitas Budi Luhur  
Jl. Raya Ciledug, Petukangan Utara, Kebayoran Lama, Jakarta Selatan 12260  
Telp. (021) 5853753 ext.303, Fax. 5853489  
E-mail : [ghalagghar@rocketmail.com](mailto:ghalagghar@rocketmail.com)*

## ABSTRAKSI

Seiring dengan berkembangnya jaman, teknologi telepon pintar sudah menjadi hal pokok dari manusia, telepon pintar mengambil peran penting dalam kemudahan manusia beraktifitas karena semua aktifitas dapat dilakukan seperti kita menggunakan komputer, fungsi mengetik, mengirim email, membaca berita, melakukan percakapan, menonton video, mendengarkan musik dan mengakses berbagai jejaring sosial dapat dikerjakan dengan mudah. Salah satu telepon pintar yang lagi tren sekarang ini adalah iPhone. Dengan berbekal fitur keren dan kemudahan pengoperasional menjadikan iPhone menjadi favorit untuk semua kalangan. Tentunya tidak ada peralatan yang sempurna, iPhone juga kadang mengalami masalah dari perangkat lunaknya maupun dari perangkat kerasnya, itulah yang mendasari pembangunan sistem pakar diagnosa kerusakan pada iPhone ini, agar para calon teknisi mengetahui kerusakan yang terjadi secara cepat dan juga keseragaman knowledge. Dalam pengembangan sistem pakar ini dibagi menjadi 2 fase yaitu pengumpulan data dan pengembangan sistem, pengumpulan data menggunakan teknik berupa wawancara pakar sedangkan pengembangan sistem berdasarkan analisa kebutuhan sistem, representasi pengetahuan, mesin inferensi, perancangan basis data, perancangan sistem, implementasi sistem dan contoh kasus. Sedangkan untuk mesin inferensinya menggunakan metode pelacakan kebelakang (*Backward Chaining*).

**Kata kunci :** Sistem Pakar, iPhone 5, Desktop, *Best First Search*

## 1. PENDAHULUAN

Seiring dengan berkembangnya jaman, teknologi telepon genggam pun mengalami perubahan signifikan yang semula hanya digunakan untuk telepon dan mengirim pesan kini telepon genggam berubah fungsi menjadi telepon pintar.

Seperti namanya telepon pintar ini mengambil konsep dasar telepon genggam yang bisa dibawa kemana saja namun ditambahkan dengan fitur yang canggih untuk memudahkan manusia dalam menyelesaikan masalahnya yang semakin hari semakin komplek dan membutuhkan penyelesaian dalam waktu cepat dan singkat. Telepon pintar memungkinkan penggunaanya untuk melakukan pekerjaan yang biasanya hanya bisa dilakukan melalui komputer seperti mengetik, mengirim email, membaca berita, melakukan *chatting* dan mengakses berbagai jejaring sosial. Salah satu telepon pintar adalah iPhone dengan fitur iMessage di dalamnya. Tentunya seperti telepon pintar yang lain, iPhone juga kadang mengalami kerusakan dari perangkat lunaknya maupun perangkat kerasnya. Hanya saja orang yang mampu merawat atau memperbaiki iPhone masih sangat terbatas, oleh karena itu dalam penulisan pembahasan ini akan dibuat suatu sistem pakar untuk mengidentifikasi

kerusakan iPhone berbasis desktop. Sistem pakar untuk mengidentifikasi kerusakan iPhone 5 ini memuat 30 jenis kerusakan yang sering terjadi pada iPhone. Sistem pakar ini nantinya diharapkan akan mempermudah para calon teknisi untuk mengidentifikasi kerusakan iPhone. Dengan menggunakan komputer kemungkinan kinerja calon teknisi menjadi lebih cepat. Diharapkan sistem ini mampu memberikan informasi yang optimal dari timbal balik user dan sistem.

### 1.1. Batasan Masalah

Penulisan Jurnal ini dibatasi dengan ruang lingkup sebagai berikut :

- [1]. Keterbatasan teknisi untuk menangani kerusakan pada iPhone 5
- [2]. Pencarian literature iPhone 5 yang memakan waktu relatif lama

## 2. LANDASAN TEORI

### 2.1 Definisi Sistem Pakar

Menurut Siswanto (2010), sistem pakar adalah yang paling banyak aplikasinya dalam membantu menyelesaikan masalah-masalah dalam dunia nyata. Perangkat lunak ini dapat sekali dijalankan oleh

perangkat komputer pribadi, sehingga untuk aplikasi kecerdasan buatan ini dapat dilakukan dengan mudah dan dengan biaya yang relatif murah.

Sistem Pakar adalah sebuah program yang dapat :

- [1]. Menangani masalah dunia nyata, masalah yang kompleks yang sangat membutuhkan interpretasi pakar.
- [2]. Menyelesaikan masalah dengan menggunakan komputer dengan penalaran manusia dan mencapai kesimpulan yang sama dengan yang capai oleh seorang pakar jika berhadapan langsung dengan sebuah masalah yang sedang dihadapinya.

Komputer yang berbasis sistem pakar adalah program komputer yang mempunyai pengetahuan yang berasal dari manusia yang berpengetahuan luas dalam suatu domain tertentu, dimana pengetahuan disini adalah pengetahuan manusia yang sangat minim penyebarannya, mahal dan serta susah untuk didapatkannya.

Kondisi-kondisi dimana sistem pakar dapat membantu manusia dalam menyelesaikan masalahnya, antara lain :

- [1]. Kebutuhan tenaga ahli yang banyak, tetapi pakar yang tersedia jumlahnya hanya sedikit dan sangat terbatas
- [2]. Pemakaian pakar yang berlebihan dalam membuat keputusan, walaupun dalam suatu tugas yang rutin.
- [3]. Pertimbangan kritis harus dilakukan dalam waktu yang singkat untuk menghindari hal-hal yang tidak diinginkan.
- [4]. Hasil yang optimal, seperti dalam hal perencanaan atau konfigurasi.
- [5]. Sejumlah besar data yang harus diteliti oleh pakar secara berkelanjutan.

## 2.2. Pembentukan Umum Expert System

Metoda yang dipakai adalah runut balik *Backward Chaining* merupakan strategi pencarian yang arahnya kebalikan dari runut maju *Forward Chaining*. Proses pencarian dimulai dari tujuan, yaitu kesimpulan yang menjadi solusi permasalahan yang dihadapi. Mesin inferensi mencari kaidah-kaidah dalam basis pengetahuan yang kesimpulannya merupakan solusi yang ingin dicapai, kemudian dari kaidah-kaidah yang diperoleh, masing-masing kesimpulan dirunut balik ke jalur yang mengarah ke kesimpulan tersebut. Jika informasi-informasi atau nilai dari atribut-atribut yang mengarah ke kesimpulan tersebut sesuai dengan data yang diberikan maka kesimpulan tersebut merupakan solusi yang dicari, jika tidak sesuai maka kesimpulan tersebut bukan merupakan solusi yang cari. Runut balik memulai proses pencarian dengan suatu tujuan sehingga strategi ini disebut juga *goal-driven*

## 2.3. Ciri-ciri Sistem

Ciri-ciri dari sistem pakar adalah :

- [1]. Terbatasnya pada domain keahlian tertentu.
- [2]. Dapat memberikan penalaran data yang tidak pasti.
- [3]. Dapat mengemukakan rangkaian alasan-alasan yang tidak diberikannya dengan cara yang tidak dapat dipahami.
- [4]. Berdasarkan kaidah-kaidah/ketentuan/rule yang berlaku.
- [5]. Dirancang untuk dapat dikembangkan secara bertahap.
- [6]. Pengetahuan dan mekanisme penalaran jelas terpisah.
- [7]. Keluaran bersifat anjuran
- [8]. Sistem dapat mengaktifkan kaidah secara searah yang sesuai dituntun oleh dialog dengan user

## 2.4. Komponen Sistem Pakar

Menurut Siswanto (2010) komponen-komponen sistem pakar tersebut adalah sebagai berikut :

### Basis Pengetahuan (*Knowledge Base*)

- [1]. Inti program sistem pakar,
- [2]. Merupakan representasi pengetahuan dari seorang pakar,
- [3]. Tersusun atas fakta-fakta yang berupa objek dan kaidah/ketentuan yang merupakan informasi tentang cara bagaimana membangkitkan fakta baru dari fakta yang telah diketahui.

### Mesin Inferensi (*Inferensi Engine*)

- [1]. Bagian-bagian yang mengandung mekanisme fungsi berpikir dan pola-pola penalaran sistem yang digunakan seorang pakar.
- [2]. Mekanisme ini akan menganalisa masalah tertentu dan selanjutnya mencari sebuah jawaban atau kesimpulan yang terbaik.
- [3]. Memilih pengetahuan yang relevan dalam rangka mencapai suatu kesimpulan.
- [4]. Memulai pelacakannya dengan mencocokkan kaidahnya (*rule*) dalam basis pengetahuan dengan fakta-fakta yang ada dalam daftar fakta-fakta (*Fact List*) yang disimpan dalam basis pengetahuan di *harddisk*.

### User Interface

- [1]. Bagian penghubung antara sistem pakar dengan pemakai.
- [2]. Akan terjadi dialog antara program dan pemakai.
- [3]. Program akan mengajukan pertanyaan-pertanyaan dengan berbentuk panduan menu (*menu driven*), pernyataan-pernyataan bahasa alami (*natural language*), dan *graphic interface style*. Program sistem pakar akan mengambil kesimpulan berdasarkan jawaban-jawaban dari si pemakai tadi.

### Development Engine

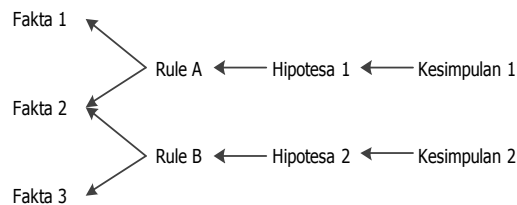
Bagian dari sistem pakar sebagai fasilitas untuk mengembangkan mesin inferensi dan penambahan basis pengetahuan yang akan dilakukan oleh

*knowledge enginner*. Knowledge enginner harus memiliki keahlian dalam mengerti bagaimana pakar menerapkan pengetahuan mereka dalam memecahkan masalah, mampu mengekstraksi penjelasan (*knowledge acquisition*) mengenai pengetahuan dari pakar jika si pakar menemukan pengetahuan dan aturan yang baru dari pengalaman dia bekerja.

### Teknik Penalaran (*Interface*)

Teknik dalam melakukan penalaran (*Interface*) menggunakan pelacakan ke belakang (*backward chaining*):

[1]. Pelacakan ke belakang (*backward chaining*) yang memulai penalarannya dari kesimpulan (*goal*), dengan mencari sekumpulan hipotesa-hipotesa yang mendukung menuju fakta-fakta yang mendukung sekumpulan hipotesa-hipotesa tersebut.



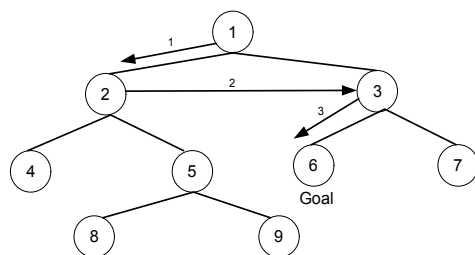
**Gambar 1 :** Teknik Pelacakan Backward Chaining

### Teknik Penelusuran (*Searching*)

Teknik penelusuran (*searching*) menggunakan teknik BFS (*Best First Search*)

[1]. Teknik Best First Search

Metode penelusuran *best first search* gabungan antara metode *depth first search* dengan metode *breadth first search*. Metode ini melakukan penelusuran pada *level* yang sama kemudian akan dilanjutkan dengan mencari ke *level* dibawahnya. Untuk melakukan penelusuran dengan jumlah *node* yang banyak, metode ini lebih efisien.



**Gambar 2 :** Teknik Best First Search

## 2.5. Tipe-tipe Sistem Pakar

Menurut Siswanto (2010), tipe-tipe sistem pakar berdasarkan struktur program, ada 3 tipe yaitu :

[1]. Program mandiri

Sistem pakar yang murni dan berdiri sendiri, artinya program utamanya tanpa mengandung *subroutine* yang memakai algoritma konvensional.

[2]. Program terkait

Sistem pakar yang dikelilingi program lainnya, artinya sebuah *subroutine* yang akan dipanggil program utama.

[3]. Program terhubung

Sistem pakar yang merupakan program yang dapat berhubungan dengan paket program lainnya.

## 2.6. Klasifikasi Sistem Pakar

[1.] Sistem Pakar Diagnosis

Digunakan untuk melakukan pencarian penyelesaian masalah atau kerusakan suatu alat atau mesin tertentu. Prinsipnya adalah menemukan masalah atau kerusakan apa yang terjadi. Sistem ini merupakan jenis sistem pakar yang paling populer saat ini, melakukan diagnosis, menggunakan deJurnal keadaan karakteristik tingkah laku, atau pengetahuan tentang pembuatan komponen sehingga dapat menentukan kemungkinan kerusakan pada sistem.

Contoh: Diagnosis medis kedokteran, diagnosis kerusakan alat elektronik dan perancangan sistem komunikasi dan radio.

[2]. Sistem Pakar Pengajaran

Digunakan untuk mengajar, mulai dari murid Sekolah Dasar sampai mahasiswa Perguruan Tinggi. Kelebihan dari sistem pakar yang digunakan untuk mengajar adalah membuat diagnosa penyebab kekurangan dari seorang siswa, kemudian memberi cara untuk memecahkannya. Contoh: Contoh sistem pakar untuk pengajaran bahasa.

[3]. Sistem Pakar Interpretasi

Digunakan untuk menganalisa data yang tidak lengkap, tidak teratur dan data yang kontradiktif, misalnya untuk interpretasi pengawasan, pengertian bahasa, analisis citra, dan lain-lain.

Contoh: sistem pakar interpretasi lingkungan pengendapan delta.

[4]. Sistem Pakar Prediksi

Digunakan untuk melakukan peramalan suatu keadaan dengan ditunjang oleh data yang diperoleh sebelumnya, sistem ini dapat memberikan kemungkinan penyelesaian tertentu.

Contoh: Prediksi Penyebab dan Solusi

Ketidaknyamanan Kerja dengan Aplikasi Sistem Pakar.

[5]. Sistem Pakar Perencanaan

Merupakan suatu sistem yang sangat luas mulai dari perencanaan mesin-mesin sampai manajemen bisnis. Penggunaan sistem pakar jenis ini menghemat biaya, waktu, dan material, sebab pembuatan model sudah tidak diperlukan lagi.

Contoh: Perencanaan Sistem Distribusi Menggunakan Sistem Pakar.

## 2.7. Kemampuan Tambahan yang Diperlukan Sistem Pakar

Untuk lebih meningkatkan kemampuan *expert system*, diperlukan komponen-komponen tambahan:

[1]. Fasilitas penjelasan

a) Untuk menjelaskan bagaimana prosesnya

sampai kesimpulan-kesimpulan tersebut di peroleh.  
b) Dengan cara memperlihatkan rule-rule yang digunakan.

[2.] Kemudahan memodifikasi

a) Di karenakan ilmu pengetahuan berkembang.

b) kemampuan seorang pakar bertambah terus.

[3.] Kompabilitas

Dapat dijalankan pada berbagai jenis komputer.

4) Kemampuan belajar

Kemampuan selama expert system untuk menambah sendiri pengolahannya, selama interaksi dengan pemakainya.

## 2.8. Struktur Sistem Pakar

Sistem pakar terdiri dari 2 bagian kelompok, yaitu: lingkungan pengembangan(*development*



**Gambar 3 : Tahapan Pembuatan Sistem Pakar**

[1]. Identifikasi Masalah

Identifikasi masalah merupakan tahapan mengidentifikasi permasalahan yang akan dikaji, dalam hal ini adalah batasan masalah yang akan dikaji, menentukan pakar yang terlibat, dan sumberdaya yang diperlukan (seperti waktu, biaya dan fasilitas komputerisasi) dan tujuan yang akan dicapai. Sebaiknya permasalahan yang akan dikaji bersifat sempit tapi kompleks (mendalam).

Pada tahapan ini pengembangan pengetahuan dapat menentukan masalah secara umum dan memilih pakar terlebih dahulu, kemudian mendiskusikannya dengan pakar tersebut untuk menentukan batasan masalah yang akan dikaji.

[2]. Tahapan Konseptualisasi

Konseptualisasi merupakan tahapan dimana pakar menentukan konsep yang akan dikembangkan menjadi sistem pakar dengan bantuan pengetahuan yang sudah ada. Dari konsep-konsep tersebut akan dirinci seluruh kegiatan-kegiatan apa saja yang akan dilakukan dan mengkaji hal-hal yang berhubungan dalam pembuatan sistem pakar itu sendiri, serta menyusun mekanisme pengendalian yang diperlukan untuk mencapai suatu solusi.

[3]. Tahapan Formalisasi

Pada tahapan formalisasi dilakukan pemilihan teknik yang tepat dan benar untuk digunakan oleh perancang sistem pakar di dalam pengembangan sistem pakar tersebut di antaranya

*environment*) dan lingkungan konsultasi (*consultation environment*). Lingkungan pengembangan digunakan sebagai pembangun sistem pakar baik dari segi pembangun komponen maupun basis pengetahuan. Lingkungan konsultasi digunakan oleh seseorang yang bukan ahli untuk berkonsultasi.

## 2.9. Tahapan Pembuatan Sistem Pakar

Proses dalam pembuatan sebuah program sistem pakar melibatkan beberapa unsur, unsur yang paling berinteraksi yaitu perekayasa pengetahuan (*Knowledge enginer*), pakar pada bidang keahlian (*domain expert*), dan pemakai akhir atau pemakai sistem pakar yang diinginkan untuk dibuat (*end user*). Tentunya dengan melalui proses dan langkah tahapan dari sistem pakar itu sendiri.

adalah struktur data, cara atau teknik inferensi dan mempresentasikan pengetahuan, strategi, kontrol serta peralatan pengembangan yang digunakan.

[4] Tahapan Implementasi

Pada tahapan ini dilakukan penterjemahan hasil formalisasi kedalam bentuk format program, yang disesuaikan dengan peralatan pengembangan sistem pakar.

[5]. Tahapan Dokumentasi

Tahapan dokumentasi adalah tahapan akhir dimana program sistem pakar yang telah dibangun harus didokumentasikan, karena dengan didokumentasikan sebuah program sistem pakar tersebut menjadi lebih baik, maka apabila suatu saat di perlukan atau terjadi permasalahan pada programnya akan mudah untuk mencarinya.

[6].Tahapan Pengujian

Pada tahapan ini sistem pakar yang telah dibuat kemudian diuji coba dengan para pakarnya untuk mencari kemungkinan kesalahan yang bisa terjadi dan melakukan perbaikan apabila diperlukan. Sehingga hasil dari sistem pakar tersebut benar-benar sesuai dengan keinginan, yaitu sebuah sistem pakar yang lengkap dan akurat serta dapat memberikan solusi terhadap masalah yang dihadapi sistem pakar tersebut.

## 3. ANALISA MASALAH DAN RANCANGAN SISTEM

### 3.1. Analisa Masalah

Sistem merupakan proses awal yang harus dilaksanakan untuk menentukan permasalahan yang sedang dihadapi. Tahap ini sangat penting karena proses analisa yang kurang akurat akan menyebabkan hasil dari suatu pengembangan perangkat lunak akan tidak sesuai dengan yang diharapkan. Jadi proses ini harus benar-benar sesuai dengan keinginan pihak pengguna agar hasil pengembangan perangkat lunak memuaskan pengguna. Tujuan sistem adalah memberikan solusi atau pemecahan dalam dalam



waktu yang cepat dan efisien. Oleh sebab itu diperlukan sebuah analisa untuk menentukan permasalahan apa yang sebenarnya terjadi. Analisa yang dilakukan harus meninjau serta mengkaji kebutuhan pengguna sehingga sesuai dengan kebutuhan dan tepat guna. Hal ini lah yang melatar belakangi perangkat lunak yang akan dikembangkan, yaitu untuk memberikan saran dan informasi yang tepat untuk mengatasi permasalahan tersebut.

Seperti yang diketahui saat ini, banyaknya informasi seputar dunia handphone terutama iphone memang sangat membantu sebagai solusi-solusi alternatif yang ditawarkan untuk pemecahan suatu masalah. Namun hal tersebut pula yang membuat kondisi menjadi kompleks yakni ketika calon teknisi dihadapkan pada situasi atau keadaan dimana calon teknisi tersebut harus memilah-milah banyaknya informasi yang didapat sedangkan waktu yang tersedia hanya sedikit. Sebagai contoh, calon teknisi yang mencoba untuk memperbaiki iphonenya sendiri namun memiliki keterbatasan waktu dan sarana. Sangat disayangkan jika calon teknisi iphone tersebut harus mencoba satu per satu dari setiap informasi yang diberikan. Calon teknisi iphone akan sangat terbantu jika terdapat sebuah sistem yang dapat memberikan saran dan informasi yang tepat juga cepat

### 3.2. Penyelesaian Masalah

Dalam menyelesaikan masalah tersebut di atas maka dibuat sistem pakar untuk mengidentifikasi kerusakan iPhone 5 dengan metode *backward Chaining* berbasis *client-server* agar dapat dipergunakan sebagai sarana informasi dan solusi. Sistem ini dirancang untuk mendeteksi gangguan yang terjadi pada bagian-bagian iphone 5 yang terdiri dari bermacam-macam komponen seperti LCD, baterai, kamera, simcard dan lain sebagainya.

Dalam sistem ini unsur-unsur yang terlibat dan saling berinteraksi adalah pembuat sistem sebagai *knowledge engineer* dan pakar. Cara kerja sistem ini dibagi menjadi 3 (tiga) bagian yaitu:

#### [1] Proses Identifikasi Kebutuhan sistem

Seorang knowledge engineer melakukan akuisisi knowledge terhadap pakar kerusakan iPhone 5 dengan cara wawancara tanya jawab dan diskusi serta mencatat hasil akuisisi kedalam matrik knowledge base. Setelah knowledge basenya jadi lalu didiskudikan kembali kepakar untuk di cek kebenaran dan bila ada yang mesti diperbaiki sampai pakarnya menyetujui matrik knowledge base yang dibuat sudah sesuai dengan knowledge kepakaran pakar kerusakan iPhone 5 seperti yang terlampir di dalam lampiran 1. Setelah matrik knowledge base disetujui oleh pakar maka knowledge engineer bisa memodelkan atau representasikan knowledge dengan metode produksi rule berupa pohon keputusan seperti yang terlampir

dalam lampiran 2, lalu dibuat rule-rule seperti yang terlampir dalam lampiran 3.

#### [2] Proses Identifikasi Kebutuhan User

Seorang knowledge engineer perlu melakukan identifikasi kondisi dan kebutuhan para pengguna jasa kepakaran pakar kerusakan iPhone 5

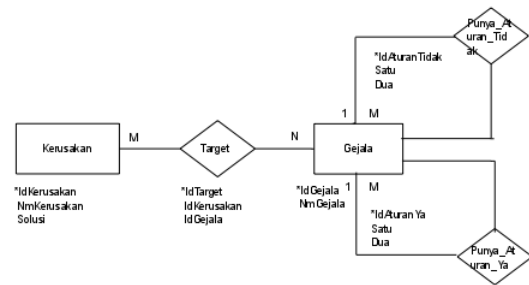
#### [3] Proses Penelusuran

Kondisi yang telah dibuat pada sistem digunakan untuk penelusuran yang nantinya akan mendapatkan solusi terbaik untuk kerusakan yang ada pada iphone.

### 3.3. Rancangan Basis Data

#### ERD (Entity Relationship Diagram)

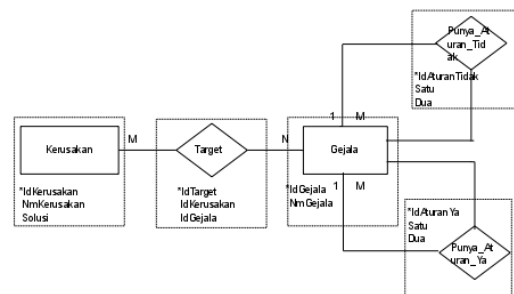
Bentuk ERD (*Entity Relationship Diagram*) dari gambar 7:



Gambar 4 : ERD (Entity Relationship Diagram)

#### Transformasi ERD (Entity Relationship Diagram) ke LRS (Logical Record Structure)

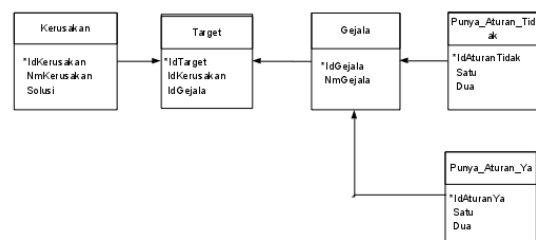
Bentuk transformasi ERD ke LRS sebagai berikut :



Gambar 5 : Transformasi ERD ke LRS

#### LRS (Logical Record Structure)

Bentuk dari ERD yang sudah menjadi LRS (*Logical Record Structure*) sebagai berikut :



Gambar 6 : LRS (Logical Record Structure)

### 3.4. Rancangan Struktur Database

#### [1]. Tabel User

Tabel *user* ini memiliki beberapa *field* antara lain : id, username, password, Tabel 1 menjelaskan tentang tabel *user*. **PRIMARY KEY = id**

*Tabel 1 : Tabel User*

No	Field	Tipe	Panjang	Keterangan
1	Id	varchar	2	Id User (PK)
2	Username	varchar	8	Nama User
3	Password	varchar	8	Kata Kunci

#### [2]. Tabel Kerusakan

Tabel kerusakan ini memiliki beberapa *field* antara lain : IdKerusakan, NmKerusakan, Solusi. Tabel 2 menjelaskan tentang tabel Kerusakan

**PRIMARY KEY = IdKerusakan**

*Tabel 2 : Tabel Kerusakan*

No	Field	Tipe	Panjang	Keterangan
1	IdKerusakan	Varchar	4	Kode Kerusakan (PK)
2	NmKerusakan	varchar	250	Nama Kerusakan
3	Solusi	varchar	1000	Solusi

#### [3].Tabel Gejala

Tabel gejala ini memiliki beberapa *field* antara lain : IdGejala, NmGejala. Tabel 3 menjelaskan tentang tabel gejala. **PRIMARY KEY = IdGejala**

*Tabel 3 : Tabel Gejala*

No	Field	Tipe	Panjang	Keterangan
1	IdGejala	Varchar	4	Id Gejala (PK)
2	NmGejala	Varchar	150	Nama Gejala

#### [4].Tabel Temp

Tabel Temp ini memiliki beberapa *field* antara lain : IdGejala dan NmGejala. Tabel 4 menjelaskan tentang tabel temporari. **PRIMARY KEY = IdGejala**

*Tabel 4 : Tabel Temp*

No	Field	Tipe	Panjang	Keterangan
1	IdGejala	Varchar	4	Id Gejala
2	NmGejala	Varchar	150	Nama Gejala

#### [5].Tabel Target

Tabel Target ini memiliki *field* antara lain : IdTarget, IdKerusakan dan IdGejala. Tabel 5 menjelaskan tentang tabel target.

**PRIMARY KEY = Id Target**

*Tabel 5 : Tabel Target*

No	Field	Tipe	Panjang	Keterangan
1	IdTarget	Varchar	5	Id target (PK)
2	IdKerusakan	Varchar	4	IdKerusakan
3	IdGejala	Varchar	4	IdGejala

#### [6].Tabel Punya\_Aturan\_Ya

Tabel Punya\_Aturan\_Ya ini memiliki *field* antara lain : IdAturanYa, Satu, Dua. Tabel 6 menjelaskan tentang tabel punya\_aturan\_ya.

**PRIMARY KEY = IdAturanYa**

*Tabel 6 : Tabel Punya\_Aturan\_Ya*

No	Field	Tipe	Panjang	Keterangan
1	IdAturanYa	Varchar	4	Id Aturan Ya(PK)
2	Satu	Varchar	4	Aturan Satu
3	Dua	varchar	4	Aturan Dua

#### [7].Tabel Punya\_Aturan\_Tidak

Tabel Punya\_Aturan\_Tidak ini memiliki *field* antara lain : IdAturanTidak, Satu, Dua. Tabel 7 menjelaskan tentang tabel punya\_aturan\_tidak.

**PRIMARY KEY = IdAturanTidak**

*Tabel 7 : Tabel Punya\_Aturan\_Tidak*

No	Field	Tipe	Panjang	Keterangan
1	IdAturanTidak	Varchar	4	Id Aturan Ya (PK)
2	Satu	Varchar	4	Aturan Satu
3	Dua	varchar	4	Aturan Dua

#### [8].Tabel Kamus

Tabel kamus ini memiliki beberapa *field* antara lain : KdKamus, Kata, dan Keterangan. Tabel 8 menjelaskan tentang tabel kamus. **PRIMARY KEY = KdKamus**

*Tabel 8 : Tabel Kamus*

No	Field	Tipe	Panjang	Keterangan
1	KdKamus	Varchar	5	Kode kamus (PK)
2	Kata	Varchar	50	Kata
3	Keterangan	varchar	150	Keterangan

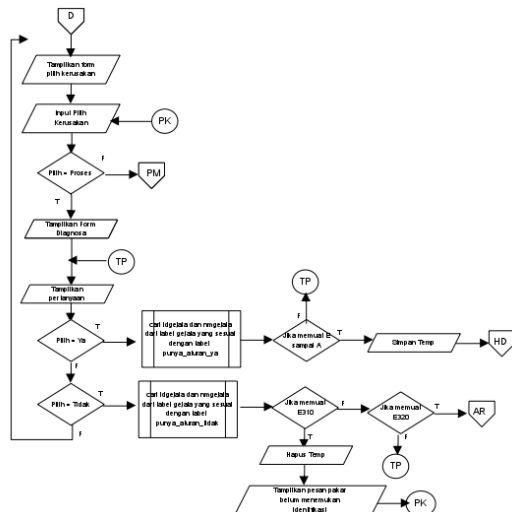
### 3.5. Mesin Inferensi

Mesin Inferensi dari sistem pakar ini menggunakan teknik pencarian *best-first search* dan menggunakan teknik pelacakan ke belakang (*backward chaining*). Cara kerja dari mesin inferensi ini adalah menggunakan pendekatan *goal-driven*, dimulai dari harapan apa yang akan terjadi (hipotesis) dan kemudian mencari bukti atau fakta-fakta yang mendukung hipotesis tersebut dengan menjawab pertanyaan untuk mendapatkan jawaban atau kesimpulan dari kerusakan tersebut. Hal ini dapat dilakukan dengan mudah oleh *user*, karena sistem yang dibuat bersifat *user friendly*.

### 3.6. Flowchart dan Algoritma

#### Flowchart Form Diagnosa

*Flowchart* dan algoritma *form* diagnosa menjelaskan tentang identifikasi kerusakan iPhone. *Flowchart form* diagnosa dapat dilihat pada gambar 10



**Gambar 7 : Flowchart Proses Menu Utama**

Berikut ini adalah algoritma *Form Diagnosa* :

1. Tampilkan *Form* Pilih Kerusakan
2. *Input* Pilih Kerusakan
3. **If** *Pilih* = Proses **Then**
4. Tampilkan *Form* Diagnosa
5. Tampilkan pertanyaan
6. *Input* Pilih
7. **If** *Pilih* = “Ya” **Then**
8. cari IdGejala dan Nmgejala dari tabel Gejala yang sesuai
9. dengan tabel punya\_aturan\_ya
10. **If** memuat E-A **Then**
11. Simpan temp
12. Tampilkan form hasil diagnosa
13. **Else**
14. Kembali ke baris 4
15. **End If**
16. **If** *Pilih* = “Tidak” **Then**
17. cari idgejala dan nmgejala dari tabel gejala yang sesuai
18. dengan tabel punya\_aturan\_tidak
19. **Else if** memuat E310 **Then**
20. Hapus temp
21. Tampilkan pesan pakar belum menemukan identifikasi
22. **Else if** memuat E320 **Then**
23. Tampilkan form Arahan
24. **End If**
25. **End If**

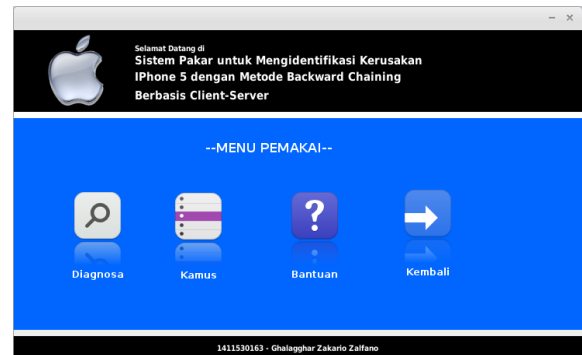
## 4. IMPLEMENTASI DAN ANALISA PROGRAM

### 4.1. Implementasi

#### [1]. Tampilan Layar *Form* Menu Pemakai

Pada tampilan layar menu pemakai seperti gambar 11 terdiri dari 4 pilihan menu yaitu menu “Diagnosa” untuk melakukan diagnosa kerusakan, menu

“Kamus” untuk mencari kata asing dan menu “Bantuan” untuk melihat petunjuk pemakaian program. Berikut ini adalah tampilan layar menu Pemakai yang ada pada file *FrmMenuPemakai..Java*



**Gambar 8 : Tampilan Layar *Form* Menu Pemakai.**

### 4.2. Spesifikasi *Hardware* dan *Software*

Spesifikasi *hardware* dan *software* pada komputer *client* sebagai berikut :

#### **Software**

- [1]. Eclipse 4.2 Juno
- [2]. JDK\_1.7.0\_51
- [3]. AppServ
- [4]. Dia Diagram 0.92.1
- [5]. Evolus Pencil
- [6.] Microsoft Office
- [7]. Microsoft Windows 7 32-bit

#### **Hardware**

- [1]. Processor Intel Core i5 CPU M480 @ 2.67GHz x 2AMD A68M
- [2]. Intel HD Graphics
- [3]. Hardisk 500 GB
- [4]. Memory 8.0 GiB

*Software* dan *hardware* yang digunakan sebagai *server* sebagai berikut :

#### **Software**

- [1]. Eclipse 4.2 Juno
- [2]. Xampp
- [3]. JDK 1.7.0
- [4]. Windows 7 32-bit

#### **Hardware**

- [1]. Processor Intel Core i3-2328M CPU @ 2.20GHz
- [2]. Memory Kingston DDR3 2GB
- [3]. Hardisk WDC 500GB
- [4]. VGA CARD AMD RADEON 1 GIGA
- [5]. Kabel Lan RJ45 1 Meter

## 5. PENUTUP

### 5.1. Kesimpulan

Berdasarkan pada pendahuluan, landasan teori, analisa dan perancangan serta implementasi dari Jurnal ini yang disusun dari bab I sampai bab IV, maka dapat diambil kesimpulan sebagai berikut :

- [1]. Sistem pakar yang dibuat ini dapat membantu pemilik atau pengguna dalam mengidentifikasi kerusakan iPhone 5.

- [2]. Sistem pakar ini juga dapat dipakai untuk membantu mempermudah teknisi junior dalam melakukan identifikasi kerusakan iPhone 5.
- [3]. Dengan menggunakan *dekstop* sebagai sarana pendukung diharapkan sistem pakar dapat menjadi *database* pengetahuan dan dokumentasi mengenai ciri-ciri kerusakan pada iPhone 5.

## 5.2 Saran

Sistem pakar untuk mengidentifikasi kerusakan iPhone ini masih belum sempurna, karena masih memerlukan perbaikan untuk meningkatkan efektifitas atau efisiensi pekerjaan. Beberapa saran yang perlu diajukan untuk perbaikan sebagai berikut:

- [1]. Dengan semakin berkembangnya ilmu pengetahuan dan teknologi, maka diharapkan sistem pakar dapat di-*update* secara berkala karena dari sisi *database* masih banyak kerusakan-kerusakan lainnya yang belum teridentifikasi.
- [2]. Data gambar diharapkan dapat ditampilkan berupa bagian-bagian yang mengalami kerusakan, sehingga akan lebih memperjelas proses identifikasi kerusakan iPhone.
- [3]. Data berupa video juga diharapkan dapat ditampilkan untuk lebih memperjelas proses identifikasi kerusakan iPhone.
- [4]. Diterapkannya perhitungan teorema Bayes agar persentase tiap-tiap kesimpulan hasil identifikasi dapat ditampilkan sehingga membantu user dalam menentukan kemungkinan kerusakan yang terjadi lewat perhitungan persentase tersebut.

## DAFTAR PUSTAKA

- [1] Siswanto, 2010, Kecerdasan Tiruan. Yogyakarta : Graha Ilmu.
- [2] Gata, Windu. 2012. Asyiknya Menenal Java. Jakarta Pusat : Elex Media Komputindo
- [3] Sitonang, Alfa dan Eva Maulina Sitonang. 2012. Java for Beginner with Eclipse 4.2 Juno, Jakarta Pusat : Elex Media Komputindo
- [4] Huda, Miftkhul dan Bunafit Komputer. 2010. Membuat Aplikasi Database dengan Java, MySQL dan Netbeans. Jakarta : Elex Media Komputindo
- [5] Josua Marinacci & Chris Adamson. 2005. Swing Hacks Tips and Tools for Killer GUI. Publisher : O'Rielly Media
- [6] Nugroho, Bunafit. 2007. Panduan Lengkap Menguasai Perintah SQL. Jakarta : Media Kita
- [7] Rochmad. 2014. Sistem Pakar Kerusakan Blackberry Berbasis Web, dilihat tanggal 1 Mei

2015. <<http://www.unisbank.ac.id/ojs/index.php/fti3/article/view/2430>>.

- [8] Purwanto, Kuwat. 2014. Sistem Pakar untuk Mendiagnosa Kerusakan Blackberry Berbasis Web dengan PHP Framework, dilihat tanggal 2 Mei 2015. <<http://eprints.unisbank.ac.id/1398/>>.

- [9] Dahria, Muhammad. 2014. Implementasi Inferensi Backward Chaining untuk Mengetahui Kerusakan Komputer, dilihat tanggal 2 Mei 2015. <[http://lppm.trigunadharma.ac.id/public/fileJurnal/hp\\_xZJurnal%2011\\_1\\_2012%20M-Dahria-4.pdf](http://lppm.trigunadharma.ac.id/public/fileJurnal/hp_xZJurnal%2011_1_2012%20M-Dahria-4.pdf)>.

- [10] Suryani, Vivin. 2014. Sistem Pakar untuk Mendiagnosa Penyakit HIV pada Manusia Menggunakan Metode Best First Search, dilihat pada 3 Mei 2015. <<http://eprints.umk.ac.id/3189/>>

- [11] Santoso, D.H & Harjono. 2012. Mendiagnosa Kerusakan Handphone Menggunakan Aplikasi Sistem Pakar, dilihat pada 4 Mei 2015. <[jurnal.ump.ac.id/index.php/juita/article/download/458/433](http://jurnal.ump.ac.id/index.php/juita/article/download/458/433)>