

**PENGEMBANGAN SISTEM PAKAR UNTUK DIAGNOSA  
PENYAKIT GIGI DAN MULUT DENGAN MENGGUNAKAN  
METODE BACKWARD CHAINING**

**SKRIPSI**



**Disusun Oleh :**  
**NASHRUL HAQ**  
**05.12.583**

**JURUSAN TEKNIK ELEKTRO S-1  
KONSENTRASI TEKNIK KOMPUTER DAN INFORMATIKA  
FAKULTAS TEKNOLOGI INDUSTRI  
INSTITUT TEKNOLOGI NASIONAL MALANG  
2011**

---

**LEMBAR PERSETUJUAN**

**PENGEMBANGAN SISTEM PAKAR UNTUK DIAGNOSA  
PENYAKIT GIGI DAN MULUT DENGAN MENGGUNAKAN  
METODE BACKWARD CHAINING**

**SKRIPSI**

*Disusun dan Diajukan Untuk Melengkapi dan Memenuhi Syarat  
Guna Mencapai Gelar Sarjana Teknik*

Disusun oleh :

**NASHRUL HAQ**

**NIM: 05.12.583**



Mengetahui,  
Ketua Jurusan T. Elektro S-1

  
**Ir. Yusuf Ismail Nakhoda, MT**  
**NIP.Y.1018800189**




Diperiksa dan Disetujui,

Dosen Pembimbing I

Dosen Pembimbing II

  
**Ir. Yusuf Ismail Nakhoda, MT**  
**NIP.Y.1018800189**

  
**Dr. Eng. Aryanto Soetedjo, ST. MT**  
**NIP.P. 1030800417**

**JURUSAN TEKNIK ELEKTRO S-1  
KONSENTRASI TEKNIK KOMPUTER DAN INFORMATIKA  
FAKULTAS TEKNOLOGI INDUSTRI  
INSTITUT TEKNOLOGI NASIONAL MALANG  
2011**

## ABSTRAKSI

### **PENGEMBANGAN SISTEM PAKAR UNTUK DIAGNOSA PENYAKIT GIGI DAN MULUT DENGAN MENGGUNAKAN METHODE BACKWARD CHAINNING**

**Nashrul Haq, 05.12.583**

**Email :** [Nashrul\\_haq24@yahoo.co.id](mailto:Nashrul_haq24@yahoo.co.id)

**Jurusan Teknik Elektro**

**Konsentrasi Teknik Komputer & Informatika S-1**

**Fakultas Teknologi Industri**

**Institut Teknologi Nasional Malang**

**Dosen Pembimbing I: Ir.Yusuf Ismail Nakhoda,MT**

**Dosen Pembimbing II: Dr. Eng. Aryuanto Soetedjo, ST. MT**

Perkembangan dunia kedokteran saat ini sudah mulai canggih. Banyak obat-obat baru yang ditemukan untuk mengatasi berbagai macam penyakit. Seperti halnya dalam bidang kedokteran gigi dan mulut. Penyakit gigi dan mulut saat ini merupakan masalah yang dihadapi oleh masyarakat banyak, semakin berkembang dari waktu ke waktu.

Mulut adalah salah satu bagian tubuh manusia yang berfungsi untuk berbicara, bernyanyi, tersenyum dan tertawa. Mulut dibentuk oleh dua rahang yaitu rahang atas dan rahang bawah, gigi merupakan salah satu bagian dari mulut. Yang kelihatan dalam mulut hanya sebagian dari seluruh gigi. Bagian yang kelihatan ini disebut mahkota gigi.

Aplikasi ini masih jauh dari sempurna dan masih banyak kekurangan. Oleh karena itu perlu dilakukan pengembangan dan penyempurnaan lebih lanjut. Adapun saran yang dapat dikemukakan agar aplikasi ini bisa berfungsi dengan lebih optimal adalah:

1. User diharapkan mematuhi semua aturan yang sudah ada dalam penggunaan aplikasi, untuk mendapatkan hasil kesimpulan yang optimal.
2. Untuk daftar jenis-jenis penyakit, diharapkan bisa dikembangkan ke yang lebih banyak lagi dengan syarat sudah diketahui oleh penulis dan pakar bagaimana mengatasi penyakit dan cara solusinya dalam bidang gigi dan mulut.

**Kata Kunci** :Sistem pakar, gigi dan mulut, *backward chaining*.

## KATA PENGANTAR

Dengan mengucapkan syukur kepada Tuhan Yang Maha Esa yang dengan segala rahmat dan anugerah – Nya, telah memberikan kekuatan, kesabaran, bimbingan dan perlindungan sehingga penulis dapat menyelesaikan laporan skripsi dengan judul :  
**PENGEMBANGAN SISTEM PAKAR UNTUK DIAGNOSA PENYAKIT GIGI  
DAN MULUT DENGAN MENGGUNAKAN METODE BACKWARD  
CHAINING**

Pembuatan skripsi ini disusun guna memenuhi syarat akhir kelulusan pendidikan jenjang Strata I di Institut Teknologi Nasional Malang. Dalam penyusunan skripsi ini penulis banyak mendapat bantuan baik moril maupun materiil, saran dan dorongan semangat dari berbagai pihak, untuk itu penulis mengucapkan terima kasih kepada :

1. Bapak Prof. Dr. Ir. Abraham Lomi, MSEE selaku Rektor ITN Malang
2. Bapak Ir. Sidik Noertjahjono, MT selaku Dekan Fakultas Teknologi Industri.
3. Bapak Ir. Yusuf Ismail Nakhoda, MT selaku Ketua Jurusan Teknik Elektro S-1 ITN Malang.
4. Bapak Ir. Yusuf Ismail Nakhoda, MT & Bapak Dr. Eng. Aryuanto Soetedjo, ST, MT selaku Dosen Pembimbing.
5. Kedua orangtua yang telah memberikan dukungan yang tiada hentinya.
6. Semua teman-teman dan semua pihak yang telah membantu penulis dalam menyelesaikan skripsi ini yang tidak bisa penulis sebutkan satu persatu.

Penulis menyadari bahwa laporan ini masih banyak yang perlu disempurnakan. Oleh sebab itu kritik dan saran yang membangun sangat diharapkan.

Akhir kata, penulis mohon maaf kepada semua pihak bilamana selama penyusunan skripsi ini penyusun membuat kesalahan secara tidak sengaja atau menyinggung pihak lain. Semoga skripsi ini dapat bermanfaat bagi kita semua.

Malang, Maret 2011

## DAFTAR ISI

<b>LEMBAR PERSETUJUAN.....</b>	<b>i</b>
<b>ABSTRAK.....</b>	<b>ii</b>
<b>KATA PENGANTAR.....</b>	<b>iv</b>
<b>DAFTAR ISI.....</b>	<b>v</b>
<b>DAFTAR TABEL.....</b>	<b>ix</b>
<b>DAFTAR GAMBAR.....</b>	<b>x</b>
<b>BAB I PENDAHULUAN.....</b>	<b>1</b>
1.1.Latar Belakang.....	1
1.2.Rumusan Masalah.....	2
1.3.Identifikasi Masalah.....	2
1.4.Maksud dan Tujuan.....	2
1.5.Batasan Masalah.....	2
1.6.Metodelogi Penelitian.....	3
1.6.1 Tahap Pengumpulan Data.....	3
1.7Sistematika Penulisan.....	4
<b>BAB II LANDASAN TEORI.....</b>	<b>5</b>
2.1 Kecerdasan Buatan.....	5
2.2 Sistem Pakar.....	5
2.2.1 Manfaat Dan Keterbatasan Sistem Pakar.....	6
2.2.1.1 Manfaat Sistem Pakar.....	6
2.2.1.2 Keterbatasan Sistem Pakar.....	7
2.2.2 Komponen Sistem Pakar.....	8
2.2.3 Rekayasa Pengetahuan.....	10
2.2.4 Partisipan Dalam Proses Pengembangan.....	10
2.2.5 Akuisisi Pengetahuan.....	11
2.2.6 Decision Tree.....	12
2.3 Metode Inferensi Backward Chaining.....	13
2.3.1 Keuntungan Methode Backward Chaining.....	15
2.4 Gigi Dan Mulut.....	16
2.4.1 Trench Mouth.....	17

2.4.2	Abses Periapikal.....	18
2.4.3	Bau Mulut.....	18
2.4.4	Gingivitis.....	20
2.4.5	Infeksi Herpes Pada Mulut.....	23
2.4.6	Pulpitis.....	24
2.4.7	Periodontitis.....	26
2.5	Borland Delphi.....	27
2.5.1	Kegunaan Delphi.....	28
2.5.2	Keunggulan Delphi.....	28
2.5.3	Menu Aplikasi Delphi.....	28
2.5.4	Komponen Delphi.....	31
2.5.5	Alite.....	31
2.6	MySQL.....	31
2.7	Xampp.....	32
<b>BAB III PERANCANGAN DAN DESAIN SISTEM.....</b>		<b>34</b>
3.1	Desain Sistem.....	34
3.2	Algoritma Metode Backward Chaining.....	36
3.2.1	Inisialisasi.....	36
3.2.2	Mulai Inferensi.....	36
3.2.3	Pengecekan Kaidah.....	37
3.2.4	Query.....	37
3.2.5	Pembaharuan Status Kaidah Premis.....	37
3.2.6	Evaluasi Kaidah.....	37
3.3	Diagram Blok Sistem.....	39
3.4	Metode Perancangan.....	39
3.4.1	Perancangan Sistem Aplikasi.....	39
3.4.2	Teknik Perancangan.....	39
3.4.3	Proses sistem Pakar.....	40
3.5	Proses Penelusuran Sistem Pakar.....	44
3.6	Perancangan Struktur Basis Data.....	45
3.6.1	Perancangan Tabel.....	45
3.7	Perancangan Form Aplikasi.....	48

3.7.1	Form Menu Utama.....	48
3.7.2	Form Registrasi Pasien.....	49
3.7.3	Form Diagnosa Awal.....	49
3.7.4	Form Hasil Diagnosa Awal.....	49
3.7.5	Form Diagnosa Penyakit.....	50
3.7.6	Form Hasil Diagnosa Penyakit.....	50
3.7.7	Form Login user.....	51
3.7.8	Form Pakar.....	51
3.7.9	Form Penyakit.....	52
3.7.10	Form Rule Base.....	52
3.7.11	Form Gejala.....	53
3.7.12	Form Pertanyaan.....	53
3.8	Struktur Basis Pengetahuan.....	54
 <b>BAB IV IMPLEMENTASI DAN PENGUJIAN SISTEM.....</b>		<b>62</b>
4.1	Implementasi Sistem.....	62
4.1.1	Form Menu Utama.....	62
4.1.2	Form Registrasi Pasien.....	62
4.1.3	Form Diagnosa Awal.....	63
4.1.4	Form Hasil Diagnosa Awal.....	63
4.1.5	Form Diagnosa Penyakit.....	64
4.1.6	Form Hasil Diagnosa Penyakit.....	65
4.1.7	Form Login User.....	65
4.1.8	Form Pakar.....	66
4.1.9	Form Penyakit.....	66
4.1.10	Form Rule Base.....	67
4.1.11	Form Gejala.....	67
4.1.12	Form Pertanyaan.....	68
4.2	Pengujian Program.....	68
4.2.1	Alpha Test.....	68
 <b>BAB V KESIMPULAN DAN SARAN .....</b>		<b>70</b>
5.1	Kesimpulan.....	70
5.2	Saran.....	70



<b>DAFTAR PUSTAKA.....</b>	<b>72</b>
<b>LAMPIRAN.....</b>	<b>73</b>

## DAFTAR TABEL

### BAB III PERANCANGAN DAN DESAIN SISTEM

Tabel 3.1 Tabel DIAGNOSA.....	46
Tabel 3.2 Tabel GEJALA .....	46
Tabel 3.3 Tabel GEJALA PENYAKIT .....	46
Tabel 3.4 Tabel PASIEN.....	47
Tabel 3.5 Tabel PENYAKIT.....	47
Tabel 3.6 Tabel RULE .....	47
Tabel 3.7 Tabel USER .....	48
Tabel 3.8 Tabel PENGOLAHAN DATA PENYAKIT GIGI DAN MULUT.....	54
Tabel 3.9 Tabel PENGURUTAN DATA GEJALA .....	60

### BAB IV IMPLEMENTASI DAN PENGUJIAN SISTEM

Tabel 4.1 Tabel PERTANYAAN ALPHA TEST.....	68
--	----

## DAFTAR GAMBAR

### BAB II LANDASAN TEORI

Gambar 2.1	Struktur Skematis Sebuah Sistem Pakar .....	9
Gambar 2.2	Proses dalam rekayasa pengetahuan .....	10
Gambar 2.3	Metode Akuisisi Pengetahuan.....	12
Gambar 2.4	Binery Tree.....	12
Gambar 2.5	Prosess Backward Chaining.....	13
Gambar 2.6	Konsep Backward Chaining.....	15
Gambar 2.7	Bagan Dari Backward Chaining .....	15
Gambar 2.8	Solusi Backward Chaining.....	16
Gambar 2.9	Menu Bar.....	28
Gambar 2.10	Tool Bar .....	28
Gambar 2.11	Component Palette .....	29
Gambar 2.12	Form .....	29
Gambar 2.13	Object Inspector .....	29
Gambar 2.14	Object Tree View .....	30
Gambar 2.15	Code Editor .....	30

### BAB III PERANCANGAN DAN DESAIN SISTEM

Gambar 3.1	Konsep Dasar Sistem Pakar .....	34
Gambar 3.2	Struktur Penelusuran Backward Chaining Gigi dan Mulut.....	35
Gambar 3.3	Diagram Algoritma Backward Chaining .....	38
Gambar 3.4	Diagram Blok Sistem Pakar.....	39
Gambar 3.5	Model Proses PL Waterfall .....	41
Gambar 3.6	Konsep Dasar Sistem Pakar Gigi dan Mulut.....	41
Gambar 3.7	Diagram Konteks Perancangan Sistem Pakar Gigi dan Mulut .....	42
Gambar 3.8	Konseptual Sistem Pakar Gigi dan Mulut.....	44
Gambar 3.9	Flowchart Sistem Pakar Gigi dan Mulut.....	45
Gambar 3.10	Form Menu Utama .....	48
Gambar 3.11	Form Registrasi Pasien.....	49
Gambar 3.12	Form Diagnosa Awal .....	49
Gambar 3.13	Form Hasil Diagnosa Awal.....	50

Gambar 3.14 Form Diagnosa Penyakit.....	50
Gambar 3.15 Form Hasil Diagnosa Penyakit.....	51
Gambar 3.16 Form Login User .....	51
Gambar 3.17 Form Pakar .....	52
Gambar 3.18 Form Penyakit .....	52
Gambar 3.19 Form Rule Base.....	53
Gambar 3.20 Form Gejala.....	53
Gambar 3.21 Form Pertanyaan .....	54

#### **BAB IV IMPLEMENTASI DAN PENGUJIAN SISTEM**

Gambar 4.1 Form Menu Utama .....	62
Gambar 4.2 Form Registrasi Pasien.....	63
Gambar 4.3 Form Diagnosa Awal .....	63
Gambar 4.4 Form Hasil Diagnosa Awal.....	64
Gambar 4.5 Form Diagnosa Penyakit.....	64
Gambar 4.6 Form Hasil Diagnosa Penyakit.....	65
Gambar 4.7 Form Login User .....	65
Gambar 4.8 Form Pakar .....	66
Gambar 4.9 Form Penyakit .....	66
Gambar 4.10 Form Rule base .....	67
Gambar 4.11 Form Gejala.....	67
Gambar 4.12 Form Pertanyaan .....	68

## BAB I

### PENDAHULUAN

#### 1.1 LATAR BELAKANG

Perkembangan dunia kedokteran saat ini sudah mulai canggih, banyak obat-obat baru yang ditemukan untuk mengatasi berbagai macam penyakit. Seperti halnya dalam bidang kedokteran gigi dan mulut. Penyakit gigi dan mulut saat ini merupakan masalah yang dihadapi oleh masyarakat banyak, semakin berkembang dari waktu ke waktu. Hal ini menuntut para dokter, baik itu yang sudah lama berkecimpung dalam dunia kedokteran maupun mahasiswa -mahasiswa yang akan dilantik sebagai dokter harus mengikuti perkembangan dan *update* terbaru tentang masalah-masalah yang ada pada gigi dan mulut beserta cara penanganannya. Hal ini tidaklah mudah mengingat bahwa sampai saat ini sudah sangat banyak sekali permasalahan penyakit gigi dan mulut yang bervariasi, apalagi jika ditambah gejala-gejala baru yang akan timbul.

Selain itu, penderita penyakit gigi juga bisa terkena efek yang lain, dari penyakit gigi yang dideritanya, karena berdasarkan ilmu kedokteran, penyakit gigi bisa merambat dan dapat menyebabkan organ-organ tubuh yang lain mengalami sakit. Hal ini tentu juga memiliki cara penanganan tersendiri yang tentunya semakin menambah beban para ahli, yaitu untuk mendiagnosa apakah penyakit gigi yang diderita oleh pasien ini bisa merambat atau tidak berdasarkan dengan ciri-ciri yang ada sesuai dengan teori dalam ilmu kedokteran.

Dengan begitu banyaknya informasi yang harus dikuasai, membuat para ahli kesulitan, karena kapasitas memori dan kemampuan yang terbatas. Untuk mengatasi hal tersebut dibutuhkan suatu hardware yaitu komputer agar para ahli mudah mengingat kembali suatu informasi. Dari komputer tersebut digunakanlah Jaringan syaraf tiruan gigi dan mulut sumber referensi dari siti nursalam itn malang yang dapat menciptakan suatu pola pengetahuan melalui pengaturan atau kemampuan belajar dan kemampuan perhitungan secara paralel sehingga proses lebih singkat. Akan tetapi jaringan syaraf tiruan memiliki kelemahan kurang mampu melakukan operasi algoritma aritmatik, operasi logika dan simbolis, lamanya proses training yang mungkin terjadi dalam waktu yang sangat lama untuk jumlah data yang besar.

Melihat dari kelemahan Jaringan syaraf tiruan disini dikembangkanlah sistem pakar dengan kelebihan mampu menjelaskan permasalahan secara detail dan

merespon lebih cepat, sehingga menghemat waktu dalam pengambilan keputusan dan juga menghemat biaya.

## 1.2 RUMUSAN MASALAH

Berdasarkan latar belakang di atas, maka permasalahan yang diangkat pada penelitian ini adalah apakah sistem pakar dengan metode *backward chaining* dapat mendeteksi jenis penyakit gigi dan mulut yang diderita pasien berdasarkan faktor / gejala klinis yang ada.

## 1.3 IDENTIFIKASI MASALAH

Secara garis besar permasalahan yang harus dihadapi adalah sebagai berikut:

- a. Kebutuhan masyarakat akan sarana pembelajaran yang efektif mengenai dunia gigi dan mulut.
- b. Kebutuhan masyarakat akan sarana konsultasi mandiri, yang dapat membantu diagnosa penyakit gigi dan mulut.
- c. Kebutuhan masyarakat akan sarana informasi kesehatan yang terjangkau lebih-lebih pada teknik penyembuhan yang dapat dilakukan sendiri.

## 1.4 MAKSUD DAN TUJUAN

Tujuan dari skripsi adalah membuat aplikasi sistem pakar untuk mendeteksi jenis penyakit gigi dan mulut yang diderita pasien berdasarkan faktor-faktor/gejala-gejala klinis yang ada dengan menggunakan *Backward Chaining*.

Adapun tujuan dibangunnya aplikasi ini adalah sebagai berikut :

- a. Aplikasi yang akan dibuat dapat memudahkan dokter dan calon dokter (mahasiswa) dalam menentukan diagnosa penyakit sesuai dengan keluhan yang dialami pasien.
- b. Aplikasi yang akan dibuat diharapkan dapat mengurangi waktu dan biaya yang akan dipakai dan dikeluarkan oleh user.

## 1.5 BATASAN MASALAH

Agar pembahasan yang dilakukan lebih terarah, maka perlu ditentukan batas permasalahan. Adapun batasan masalahnya adalah:

- a. Aplikasi yang dibuat dapat memudahkan mengidentifikasi diagnosa penyakit berdasarkan *backward chaining*.
-

- b. Sistem Pakar ini secara bertahap dirancang agar dapat bersifat fleksibel, sehingga dapat memberikan kemudahan pada pengembangan selanjutnya.
- c. Jenis pertanyaan keluhan dibatasi 7 macam pertanyaan selebihnya tidak dibahas.
- d. Aplikasi ini berdasarkan *rule-rule* yang telah ditanamkan pada program aplikasi yang dibuat.
- e. Database yang dibuat ada 7 macam yaitu :  
Tb\_diagnosa, tb\_gejala, tb\_gejala\_penyakit, tb\_pasien, tb\_penyakit, tb\_rule, tb\_user
- f. Membahas penyakit secara detail untuk mendapatkan goal driven.
- g. Tidak membahas *forward chaining*
- h. Hasil kesimpulan berupa, nama penyakit beserta gambar dan penyebab serta solusi penyakit.
- i. Perangkat lunak yang digunakan untuk membangun aplikasi ini adalah:
  - *Borland Delphi 7*
  - *Mysql-connector-odbc*
  - *Xampp-win32-1.7.3*

## 1.6 METODELOGI PENELITIAN

Metodologi yang digunakan dalam penulisan skripsi ini adalah sebagai berikut :

### 1.6.1. Tahap Pengumpulan Data

Metode pengumpulan data yang digunakan dalam penelitian ini adalah sebagai berikut :

#### a. Studi literature

Mempelajari masalah-masalah dan penyakit-penyakit pada gigi beserta cara penanganannya berdasarkan ilmu kedokteran yang ada saat ini.

#### b. Perencanaan dan Pembuatan Perangkat Lunak

Merancang *knowledge base* dan *inference engine* yang digunakan, juga mengimplementasikan rancangan sistem ke dalam *coding software* dengan menggunakan Borland Delphi 7 dan database *Xampp-win32-1.7.3*

#### c. Pengujian dan Analisa Perangkat Lunak

Menguji program yang telah dibuat, menganalisa dan memperbaiki kesalahan-kesalahan yang ditemukan.

---

#### **d. Pengambilan Kesimpulan**

Pengambilan kesimpulan dengan membandingkan hasil *output* program dengan analisa secara manual yang dilakukan berdasarkan teori kedokteran yang ada saat ini.

### **1.7. SISTEMATIKA PENULISAN**

Pada penulisan skripsi ini penulis menggunakan sistematika penulisan sebagai berikut :

#### **BAB I : PENDAHULUAN**

Menguraikan tentang latar belakang permasalahan, mencoba merumuskan inti permasalahan yang dihadapi, menentukan tujuan dan kegunaan penelitian, yang kemudian diikuti dengan pembatasan masalah, metodologi penelitian serta sistematika penulisan.

#### **BAB II : LANDASAN TEORI**

Membahas berbagai konsep dasar dan teori-teori yang berkaitan dengan topik penelitian yang dilakukan dan hal-hal yang berguna dalam proses analisis permasalahan serta tinjauan terhadap penelitian-penelitian serupa yang telah pernah dilakukan sebelumnya termasuk sintesisnya.

#### **BAB III : PERANCANGAN DAN DESAIN SISTEM**

Menganalisis masalah yang akan dibahas serta merancang aplikasi yang akan dibangun.

#### **BAB IV : IMPLEMENTASI DAN PENGUJIAN**

Berisi tentang implementasi software terhadap sistem yang telah dirancang dan berikut pengujian serta analisa dari perancangan sistem yang telah dibuat. Pengujian aplikasi dilakukan untuk mengetahui permasalahan-permasalahan yang mungkin muncul. Analisa dilakukan untuk menyempurnakan aplikasi yang dibuat jika terdapat kekurangan atau kesalahan pada jalannya program maupun dari hasil yang diperoleh.

#### **BAB V : KESIMPULAN DAN SARAN**

Berisi tentang kesimpulan yang dapat diambil berdasarkan hasil uraian pada bab-bab sebelumnya dan saran serta petunjuk pengembangan dan penyempurnaan hasil skripsi ini.

---



## BAB II LANDASAN TEORI

### 2.1 KECERDASAN BUATAN

Kecerdasan buatan merupakan salah satu bagian ilmu komputer yang mempelajari bagaimana cara membuat mesin (komputer) agar dapat melakukan pekerjaan seperti yang dilakukan oleh manusia.

Kecerdasan Buatan (*Artificial Intelligence* atau *AI*) didefinisikan sebagai kecerdasan yang ditunjukkan oleh suatu entitas buatan. Sistem seperti ini umumnya dianggap komputer. Kecerdasan diciptakan dan dimasukkan ke dalam suatu mesin (komputer) agar dapat melakukan pekerjaan seperti yang dapat dilakukan manusia. Beberapa macam bidang yang menggunakan kecerdasan buatan antara lain sistem pakar, permainan komputer, logika fuzzy, jaringan syaraf tiruan dan robotika.

Banyak hal yang kelihatannya sulit untuk kecerdasan manusia, tetapi untuk Informatika relatif tidak bermasalah. Seperti contoh: mentransformasikan persamaan, menyelesaikan persamaan integral, membuat permainan catur atau Backgammon. Di sisi lain, hal yang bagi manusia kelihatannya menuntut sedikit kecerdasan, sampai sekarang masih sulit untuk direalisasikan dalam informatika. Seperti contoh: Pengenalan Obyek/Muka, bermain Sepakbola.

Walaupun *AI* memiliki konotasi fiksi ilmiah yang kuat, *AI* membentuk cabang yang sangat penting pada ilmu komputer, berhubungan dengan perilaku, pembelajaran dan adaptasi yang cerdas dalam sebuah mesin. Penelitian dalam *AI* menyangkut pembuatan mesin untuk mengotomatisasikan tugas-tugas yang membutuhkan perilaku cerdas. Termasuk contohnya adalah pengendalian, perencanaan dan penjadwalan, kemampuan untuk menjawab diagnosa dan pertanyaan pelanggan, serta pengenalan tulisan tangan, suara dan wajah. Hal-hal seperti itu telah menjadi disiplin ilmu tersendiri, yang memusatkan perhatian pada penyediaan solusi masalah kehidupan yang nyata. Sistem *AI* sekarang ini sering digunakan dalam bidang ekonomi, obat-obatan, teknik dan militer, seperti yang telah dibangun dalam beberapa aplikasi perangkat lunak komputer rumah dan video game.

### 2.2 Sistem Pakar

Ketika hendak membuat suatu keputusan yang komplek atau memecahkan masalah, seringkali kita meminta nasehat atau berkonsultasi dengan seorang pakar atau ahli. Seorang pakar adalah seseorang yang mempunyai pengetahuan dan pengalaman

spesifik dalam suatu bidang; misalnya pakar komputer, pakar uji tak merusak, pakar politik dan lain-lain. Semakin tidak terstruktur situasinya, semakin mengkhusus (dan mahal) konsultasi yang dibutuhkan.

- **Sistem Pakar** (*Expert System*) adalah usaha untuk menirukan seorang pakar. Biasanya Sistem Pakar berupa perangkat lunak pengambil keputusan yang mampu mencapai tingkat performa yang sebanding seorang pakar dalam bidang problem yang khusus dan sempit. Ide dasarnya adalah: kepakaran ditransfer dari seorang pakar (atau sumber kepakaran yang lain) ke komputer, pengetahuan yang ada disimpan dalam komputer, dan pengguna dapat berkonsultasi pada komputer itu untuk suatu nasehat, lalu komputer dapat mengambil inferensi (menyimpulkan, mendeduksi, dll.) seperti layaknya seorang pakar, kemudian menjelaskannya ke pengguna tersebut, bila perlu dengan alasan-alasannya. Sistem Pakar malahan terkadang lebih baik unjuk kerjanya daripada seorang pakar manusia.
- **Kepakaran** (*expertise*) adalah pengetahuan yang ekstensif (meluas) dan spesifik yang diperoleh melalui rangkaian pelatihan, membaca, dan pengalaman. Pengetahuan membuat pakar dapat mengambil keputusan secara lebih baik dan lebih cepat daripada non-pakar dalam memecahkan problem yang kompleks. Kepakaran mempunyai sifat berjenjang, pakar top memiliki pengetahuan lebih banyak daripada pakar junior.

Tujuan Sistem Pakar adalah untuk mentransfer kepakaran dari seorang pakar ke komputer, kemudian ke orang lain (yang bukan pakar). Proses ini tercakup dalam rekayasa pengetahuan (*knowledge engineering*) yang akan dibahas kemudian

## 2.2.1 Manfaat dan Keterbatasan Sistem Pakar

### 2.2.1.1. Manfaat Sistem Pakar

Beberapa kemampuan dan manfaat yang diberikan oleh Sistem Pakar, diantaranya:

1. Meningkatkan output dan produktivitas, karena Sistem Pakar dapat bekerja lebih cepat dari manusia.
  2. Meningkatkan kualitas, dengan memberi nasehat yang konsisten dan mengurangi kesalahan.
  3. Mampu menangkap kepakaran yang sangat terbatas.
  4. Memudahkan akses ke pengetahuan.
-

5. **Handal.** Sistem Pakar tidak pernah menjadi bosan dan kelelahan atau sakit. Sistem Pakar juga secara konsisten melihat semua detail dan tidak akan melewatkan informasi yang relevan dan solusi yang potensial.
6. Meningkatkan kapabilitas sistem terkomputerisasi yang lain, integrasi Sistem Pakar dengan sistem komputer lain membuat lebih efektif, dan mencakup lebih banyak aplikasi.
7. Mampu bekerja dengan informasi yang tidak lengkap atau tidak pasti. Berbeda dengan sistem komputer konvensional, Sistem Pakar dapat bekerja dengan informasi yang tidak lengkap. Pengguna dapat merespon dengan: “tidak tahu” atau “tidak yakin” pada satu atau lebih pertanyaan selama konsultasi, dan Sistem Pakar tetap akan memberikan jawabannya.
8. Mampu menyediakan pelatihan. Pengguna pemula yang bekerja dengan Sistem Pakar akan menjadi lebih berpengalaman. Fasilitas penjelas dapat berfungsi sebagai guru.
9. Meningkatkan kemampuan problem solving, karena mengambil sumber pengetahuan dari banyak pakar.
10. Meniadakan kebutuhan perangkat yang mahal.
11. Fleksibel.

#### **2.2.1.2 Keterbatasan Sistem Pakar**

Metodologi Sistem Pakar yang ada tidak selalu mudah, sederhana dan efektif. Berikut adalah keterbatasan yang menghambat perkembangan Sistem Pakar:

1. Pengetahuan yang hendak diambil tidak selalu tersedia.
  2. Kepakaran sangat sulit diekstrak dari manusia.
  3. Pendekatan oleh setiap pakar untuk suatu situasi atau problem bisa berbeda-beda, meskipun sama-sama benar.
  4. Adalah sangat sulit bagi seorang pakar untuk mengabstraksi atau menjelaskan langkah mereka dalam menangani masalah
  5. Pengguna Sistem Pakar mempunyai batas kognitif alami, sehingga mungkin tidak bisa memanfaatkan sistem secara maksimal.
  6. Sistem Pakar bekerja baik untuk suatu bidang yang sempit.
  7. Banyak pakar yang tidak mempunyai jalan untuk mengecek apakah kesimpulan mereka benar dan masuk akal.
-

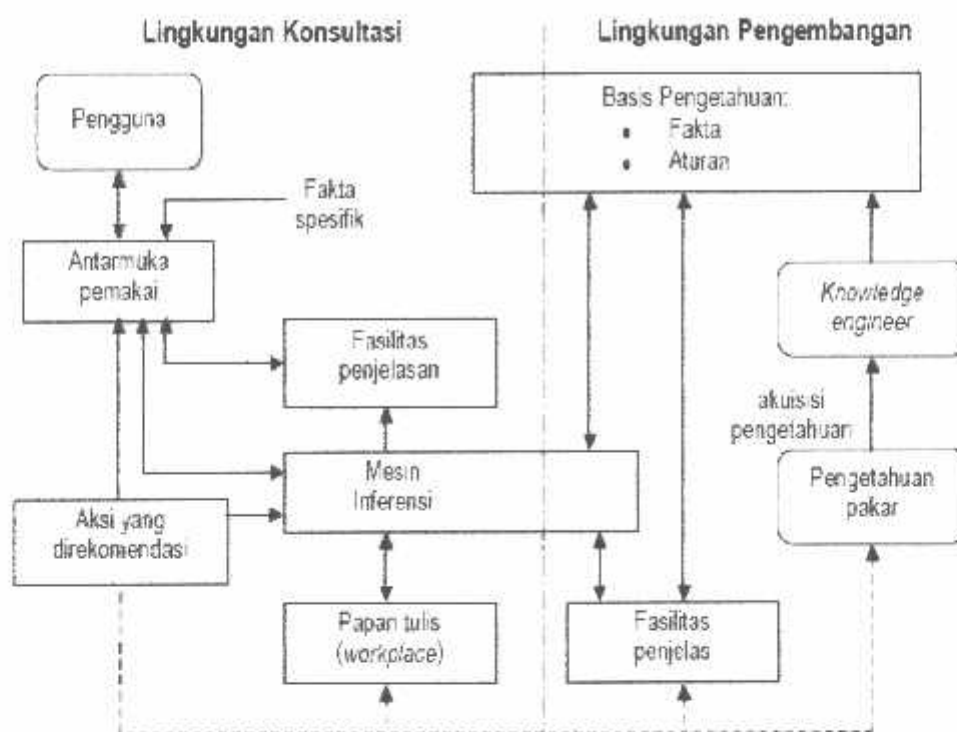
8. Istilah dan yang dipakai oleh pakar dalam mengekspresikan fakta seringkali terbatas dan tidak mudah dimengerti oleh orang lain.
9. Pengembangan Sistem Pakar seringkali membutuhkan perencana pengetahuan (*knowledge engineer*) yang langka dan mahal.
10. Kurangnya rasa percaya pengguna menghalangi pemakaian Sistem Pakar.
11. Transfer pengetahuan dapat bersifat subyektif dan bisa.

### 2.2.2 Komponen Sistem Pakar

Secara umum, Sistem Pakar biasanya terdiri atas beberapa komponen yang masing-masing berhubungan seperti terlihat pada Gambar 2.3.

Basis Pengetahuan, berisi pengetahuan yang dibutuhkan untuk memahami, memformulasi, dan memecahkan masalah. Basis pengetahuan tersusun atas 3 elemen dasar:

1. Fakta, misalnya: situasi, kondisi, dan kenyataan dari permasalahan yang ada, serta teori dalam bidang itu
  2. Aturan, yang mengarahkan penggunaan pengetahuan untuk memecahkan masalah yang spesifik dalam bidang yang khusus
  3. Mesin Inferensi (*Inference Engine*), merupakan otak dari Sistem Pakar. Juga dikenal sebagai penerjemah aturan (*rule interpreter*). Komponen ini berupa program komputer yang menyediakan suatu metodologi untuk memikirkan (*reasoning*) dan memformulasi kesimpulan. Kerja mesin inferensi meliputi:
    - a. Menentukan aturan mana akan dipakai
    - b. Menyajikan pertanyaan kepada pemakai, ketika diperlukan.
    - c. Menambahkan jawaban ke dalam memori Sistem Pakar.
    - d. Menyimpulkan fakta baru dari sebuah aturan
    - e. Menambahkan fakta tadi ke dalam memori.
-



Gambar 2.1 Struktur skematis sebuah Sistem Pakar

*Papan Tulis (Blackboard/Workplace)*, adalah memori/lokasi untuk bekerja dan menyimpan hasil sementara. Biasanya berupa sebuah basis data.

*Antarmuka Pemakai (User Interface)*. Sistem Pakar mengatur komunikasi antara pengguna dan komputer. Komunikasi ini paling baik berupa bahasa alami, biasanya disajikan dalam bentuk tanya-jawab dan kadang ditampilkan dalam bentuk gambar/grafik. Antarmuka yang lebih canggih dilengkapi dengan percakapan (voice communication).

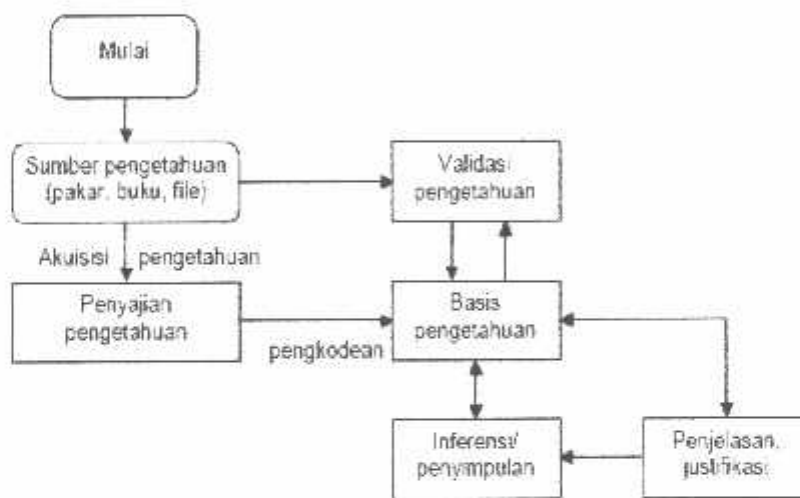
*Subsistem Penjelasan (Explanation Facility)*. Kemampuan untuk menjejak (tracing) bagaimana suatu kesimpulan dapat diambil merupakan hal yang sangat penting untuk transfer pengetahuan dan pemecahan masalah. Komponen subsistem penjelasan harus dapat menyediakannya yang secara interaktif menjawab pertanyaan pengguna.

*Sistem Penghalusan Pengetahuan (Knowledge Refining System)*. Seorang pakar mempunyai sistem penghalusan pengetahuan, artinya, mereka bisa menganalisa sendiri performa mereka, belajar dari pengalaman, serta meningkatkan pengetahuannya untuk konsultasi berikutnya. Pada Sistem Pakar, swa-evaluasi ini penting sehingga dapat menganalisa alasan keberhasilan atau kegagalan pengambilan kesimpulan, serta memperbaiki basis pengetahuannya.

### 2.2.3 Rekayasa Pengetahuan (Knowledge Engineering)

Proses dalam rekayasa pengetahuan meliputi (Gambar 2.2):

1. Akuisisi pengetahuan, yaitu bagaimana memperoleh pengetahuan dari pakar atau sumber lain (sumber terdokumentasi, buku, sensor, file komputer, dll.).
2. Validasi pengetahuan, untuk menjaga kualitasnya misalnya dengan uji kasus.
3. Representasi pengetahuan, yaitu bagaimana mengorganisasi pengetahuan yang diperoleh, mengkodekan dan menyimpannya dalam suatu basis pengetahuan.
4. Penyimpulan pengetahuan, menggunakan mesin inferensi yang mengakses basis pengetahuan dan kemudian melakukan penyimpulan.
5. Transfer pengetahuan (penjelasan). Hasil inferensi berupa nasihat, rekomendasi, atau jawaban, kemudian dijelaskan ke pengguna oleh subsistem penjelas.



Gambar 2.2 Proses dalam rekayasa pengetahuan

### 2.2.4 Partisipan Dalam Proses Pengembangan

Pakar, yaitu seseorang yang mempunyai pengetahuan, pengalaman, dan metode khusus, serta mampu menerapkannya untuk memecahkan masalah atau memberi nasihat. Pakar menyediakan pengetahuan tentang bagaimana nantinya Sistem Pakar bekerja.

Perekayasa pengetahuan (*knowledge engineer*), yang membantu pakar untuk menyusun area permasalahan dengan menerjemahkan dan mengintegrasikan jawaban pakar terhadap pertanyaan-pertanyaan dari klien, menarik analogi, serta memberikan contoh-contoh yang berlawanan, kemudian menyusun basis pengetahuan.

Pengguna, yang mungkin meliputi: seorang klien non-pakar yang sedang membutuhkan nasihat (Sistem Pakar sebagai konsultan atau *advisor*), seorang siswa

yang sedang belajar (Sistem Pakar sebagai instruktur), seorang pembuat Sistem Pakar yang hendak meningkatkan basis pengetahuan (Sistem Pakar sebagai *partner*), seorang pakar (Sistem Pakar sebagai kolega atau asisten, yang dapat memberikan opini kedua).

Partisipan lain, dapat meliputi: pembangun sistem (*system builder*), *tool builder*, staf administrasi dsb.

### 2.2.5 Akuisisi Pengetahuan

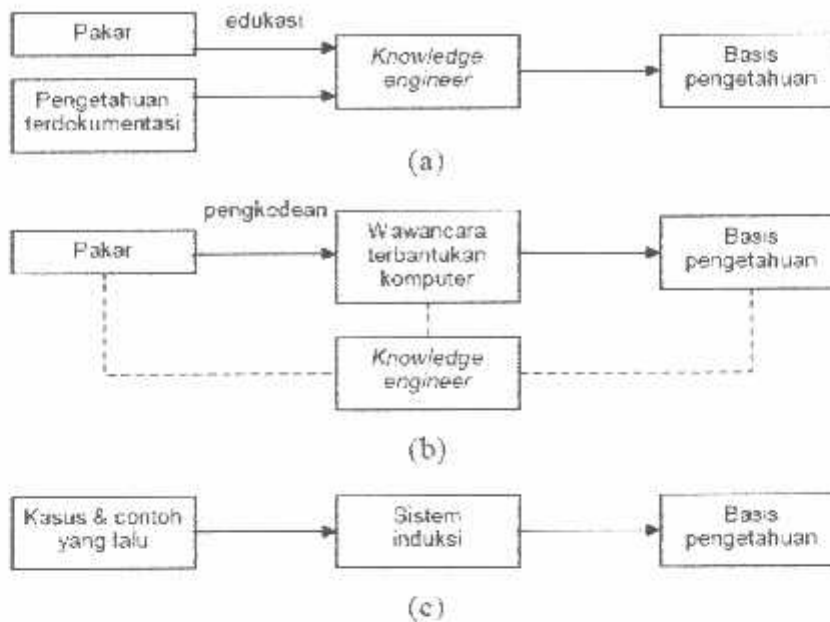
Dalam proses akuisisi pengetahuan, seorang perekayasa pengetahuan menjembatani antara pakar dengan basis pengetahuan. Perekayasa pengetahuan mendapatkan pengetahuan dari pakar, mengolahnya bersama pakar tersebut, dan menaruhnya dalam basis pengetahuan, dengan format tertentu. Pengambilan pengetahuan dari pakar dapat dilakukan secara *manual*, di mana perekayasa pengetahuan mendapatkan pengetahuan dari pakar (melalui wawancara) dan/atau sumber lain, kemudian mengkodekannya dalam basis pengetahuan. Proses ini biasanya berlangsung lambat, mahal, serta kadangkala tidak akurat.

*Semi-otomatik*, di mana terdapat peran komputer untuk:

1. mendukung pakar dengan mengijinkannya membangun basis pengetahuan tanpa (atau dengan sedikit) bantuan dari perekayasa pengetahuan.
2. membantu perekayasa pengetahuan sehingga kerjanya menjadi lebih efisien dan efektif.

*Otomatik*, di mana peran pakar, perekayasa pengetahuan, dan pembangun basis pengetahuan (*system builder*) digabung. Misalnya dapat dilakukan oleh seorang *system analyst* seperti pada metode induksi.

---

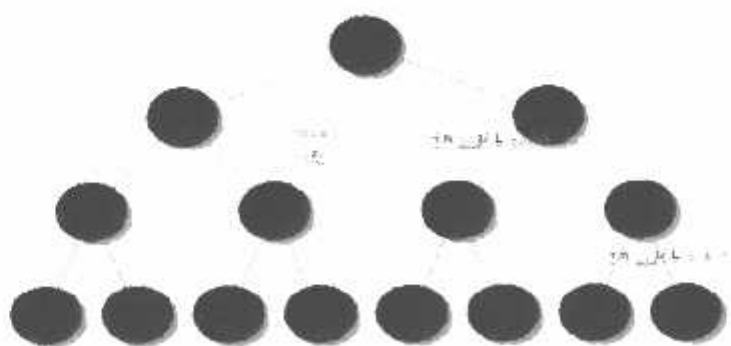


Gambar 2.3 Metode akuisisi pengetahuan (a) manual (b) akuisisi terkendali-pakar (c) induksi

### 2.2.6 Decision Tree

Suatu *Tree* adalah suatu hierarki struktur yang terdiri dari simpul (node) yang menyimpan informasi atau pengetahuan dan cabang yang menghubungkan simpul. Cabang disebut juga link atau edge dan simpul disebut juga dengan vertex. Akar simpul adalah simpul yang tertinggi dalam hierarki dan daun adalah simpul yang paling bawah. Tree dapat dianggap sebagai tipe khusus dari jaringan semantik yang setiap simpulnya, kecuali akar pasti mempunyai suatu simpul orang tua dan mempunyai nol atau lebih simpul anak.

Untuk tipe biasa dari *Binery Tree*, maksimum mempunyai 2 anak untuk setiap simpulnya, sisi kiri dan kanan dari simpul anak dibedakan. Jika simpul mempunyai lebih dari satu orang tua maka disebut jaringan. *Binery Tree* ditunjukkan pada gambar 2.4.



Gambar 2.4 Binery Tree



Aplikasi dari *tree* adalah pembuatan keputusan dan biasa disebut dengan istilah *Decision Tree* (pohon keputusan). Contoh *Decision Tree* mendiagnosa awal gangguan kesehatan. Simpul berisikan pertanyaan, cabangnya berisi jawaban **Ya** atau **Tidak** untuk menjawab pertanyaan dan daunnya berisikan penyebab dan solusi dari gangguan kesehatan.

### 2.3 Metode Inferensi *Backward Chaining*

Metode *Backward chaining* adalah suatu metode untuk menemukan suatu fakta dengan cara menelusuri subgoals yang ada secara rekursif. Cara kerjanya inference engine memulai dari goal yang telah ditentukan kemudian berjalan mundur untuk membuktikan kebenaran goal tersebut berdasarkan rule-rule apa saja yang dapat membentuk goal tersebut.

Pelacakan ke belakang (*Backward Chaining*) adalah pendekatan yang dimotori oleh tujuan (*goal driven*).

Dalam pendekatan ini pelacakan dimulai dari tujuan, selanjutnya dicari aturan yang memiliki tujuan tersebut untuk kesimpulannya. Selanjutnya proses pelacakan menggunakan premis untuk aturan tersebut sebagai tujuan baru dan mencari aturan lain dengan tujuan baru sebagai kesimpulannya. Proses berlanjut sampai semua kemungkinan ditemukan (Kusumadewi, 2003). Gambar 4 menunjukkan proses *backward chaining*.



Gambar 2.5 Proses *backward chaining*

*Backward chaining* memulai interpreter kaidah dari fakta yang ada dalam pangkalan data, yaitu hipotesa. Interpreter kaidah selanjutnya menguji kaidah sebelah kanan, yaitu bagian **THEN**, yang bertujuan untuk mencari yang sesuai. Motor inferensi melacak bukti-bukti yang mendukung hipotesa yang diperbarui yang mendukung kesimpulan yang sesuai. Proses ini dilakukan berulang-ulang untuk mencocokkan kaidah sebelah kanan terhadap status sistem yang berlaku. Semua sisi kaidah IF

yang cocok digunakan untuk menghasilkan hipotesa-hipotesa antara yang baru dan keadaan tujuan yang kemudian direkam dalam pangkalan data. Seterusnya, backward chaining meneruskan pekerjaannya sampai hipotesa terbukti benar.

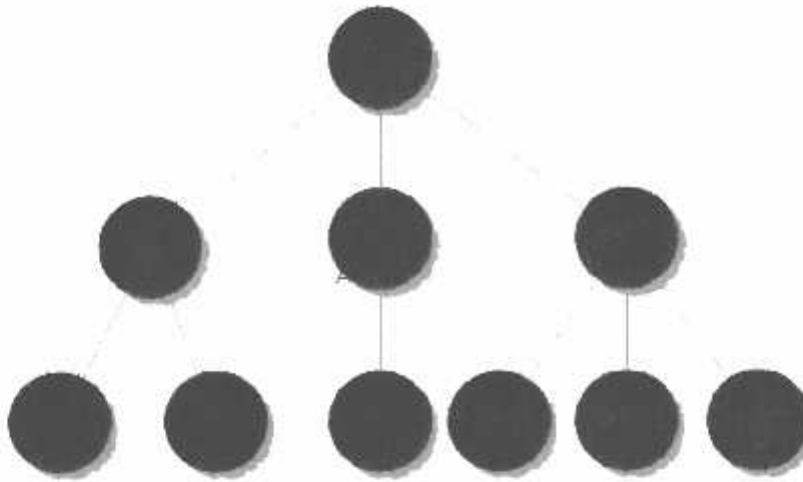
Dalam sistem pakar backward chaining, semua hipotesa harus diketahui dan semua kemungkinan terakhir terpenuhi. Pemilihan sudah diketahui sejak awal. Kesemuanya itu mempunyai makna sendiri, karena para pakar menciptakan sistem agar mengetahui jawaban. Selama proses backward chaining, interpreter kaidah akan melihat berbagai premis yang ada dalam pangkalan data. Bila ia tidak menemukan, maka ia meneruskan pelacakannya sampai pada apa yang harus ditemukannya.

Sebagai contoh misalkan Anda sedang mengemudi dan tiba-tiba Anda melihat mobil polisi dengan cahaya kelap-kelip dan bunyi sirine. Jika mobil polisi membuntuti di belakang Anda atau polisi melambaikan tangan memberhentikan Anda. Dengan mengadopsi ini sebagai suatu kerja hipotesis, maka Anda dapat menggunakan metode Backward Chaining untuk alasan mengapa?

Beberapa kemungkinan hipotesis lanjut dari contoh tersebut adalah membuang sampah sembarangan dari mobil, melampau batas kecepatan, kesalahan pemakaian peralatan dan mengemudi kendaraan curian. Selanjutnya Anda menguji fakta yang mendukung hipotesis lanjut. Apakah botol minuman yang Anda buang keluar melalui jendela, kecepatan 100 dalam 30 mil perjam dalam zona cepat, lampu belakang yang rusak atau plat mobil yang Anda gunakan mengidentifikasi mobil yang Anda kemudi adalah mobil curian? Dalam kasus ini setiap bagian dari fakta mendukung hipotesis lebih lanjut ini merupakan alasan yang mungkin untuk membuktikan hipotesis kerja yang diinginkan polisi. dalam zona cepat, lampu belakang yang rusak atau plat mobil yang Anda gunakan mengidentifikasi mobil yang Anda kemudi adalah mobil curian? Dalam kasus ini setiap bagian dari fakta mendukung hipotesis lebih lanjut ini merupakan alasan yang mungkin untuk membuktikan hipotesis kerja yang diinginkan polisi.

Dalam *backward chaining*, sistem bekerja dari tujuan/hipotesis kemudian menghubungkan fakta yang diperoleh dengan aturan yang ada untuk mencocokkan dengan tujuan yang diharapkan

---

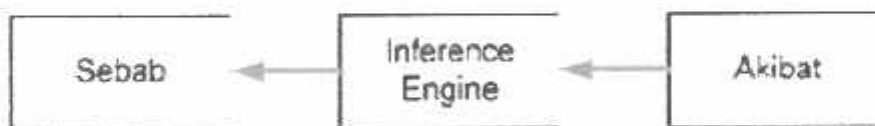


Gambar 2.6 Konsep *backward chaining*

### 2.3.1 Keuntungan metode *Backward Chaining*

Keuntungan dengan menggunakan metode *backward chaining* ialah:

1. *Backward chaining* terfokus pada *goal* yang diberikan. Prosedur ini akan menanyakan hal-hal yang perlu saja dan ini merupakan kenyamanan bagi *user*.
2. Bila *forward chaining* mencoba semua kemungkinan dari informasi yang ada, *backward chaining* mencoba menyelesaikan masalah dengan mencari basis pengetahuan yang relevan dengan masalah sekarang.
3. *Backward chaining* merupakan pendekatan yang baik untuk menyelesaikan suatu diagnostik, preskripsi, dan *debugging*.



Gambar 2.7 bagan dari *Backward Chaining*.

Dalam menganalisis masalah, maka komputer berusaha memenuhi syarat dari posisi “JIKA” pada *rule* yang konklusinya merupakan goal atau premis dari *rule* lain.

Contoh Inference Engine Backward chaining :

Diketahui sistem pakar dengan aturan-aturan sebagai berikut :

R1 : IF suku bunga turun THEN harga obligasi naik

R2 : IF suku bunga naik THEN harga obligasi turun

R3 : IF suku bunga tidak berubah THEN harga obligasi tidak berubah

R4 : IF dollar naik THEN suku bungah turun

R5 : IF dollar turun THEN suku bungah naik

R6 : IF harga obligasi turun TIEN beli obligasi

Pertanyaan : apabila diketahui dollar turun, maka buatlah keputusan apakah akan membeli obligasi atau tidak?



Gambar 2.8 Solusi *Backward Chaining*

Ket :

- Dari solusi yaitu membeli obligasi, dengan menggunakan rule 6 diperoleh anteseden harga obligasi turun.
- Dari rule 2 dibuktikan harga obligasi turun bernilai benar jika suku bungah naik bernilai benar.

## 2.4 Gigi dan Mulut

Mulut adalah salah satu bagian tubuh manusia yang berfungsi untuk berbicara, bernyanyi, tersenyum dan tertawa. Mulut dibentuk oleh dua rahang yaitu rahang atas dan rahang bawah. Sedangkan bagian muka dari bawah mata sampai ke dagu dibentuk oleh kedua rahang tersebut diatas. Dengan lain kata, mulut ikut memberikan bentuk kepada muka kita. Kalau pertumbuhan rahang kurang baik, maka bentuk muka kita menjadi kurang serasi. Bagian terpenting dari mulut adalah bibir, lidah, gigi gusi, dan ludah.

Gigi merupakan salah satu bagian dari mulut. Yang kelihatan dalam mulut hanya sebagian dari seluruh gigi. Bagian yang kelihatan ini disebut mahkota gigi. Sebagian dari gigi tertanam di dalam rahang. Karena itu bagian ini tidak terlihat kalau kita membuka mulut. Bagian yang tidak kelihatan ini di sebut akar gigi. Akar gigi ini diikat pada tulang rahang maka gigi tidak mudah copot. Gigi gunanya untuk :

- Memotong, mencabit dan menghaluskan makanan.
- Mengucapkan kata-kata dengan jelas.
- Mendorong pertumbuhan rahang sehingga bentuk wajah

menjadi harmonis.

d. Kecantikan.

Jenis penyakit gigi dan mulut pada aplikasi ini adalah:

**2.4.1 Trench Mouth**

*Trench Mouth* (Infeksi *Vincent*, *Gingivitis ulserativa nekrotikan akut*) adalah suatu infeksi gusi yang tidak menular dan terasa nyeri, menyebabkan nyeri, demam, dan kelelahan.

Beberapa hal yang mendukung terjadinya penyakit ini :

- a) Kebersihan mulut yang jelek
- b) Stres fisik maupun stres emosional
- c) Diet yang kurang
- d) Kurang istirahat.

Infeksi paling sering terjadi pada penderita *gingivitis simplek* yang mengalami saat-saat yang menegangkan (misalnya ujian di sekolah atau ganti pekerjaan). Lebih sering terjadi pada perokok.

Gejala-gejala dari *Trench Mouth* adalah:

Biasanya, *trench mouth* dimulai secara tiba-tiba berupa nyeri gusi, gelisah dan kelelahan. Dapat juga timbul bau mulut yang busuk. Ujung-ujung gusi yang terletak diantara dua gigi mengalami pengikisan dan tertutup oleh jaringan mati yang berupa lapisan yang berwarna abu-abu. Gusi mudah berdarah. Mengunyah dan menelan menyebabkan nyeri. Kelenjar getah bening dibawah rahang seringkali membengkak dan timbul demam ringan.

Diagnosa :

Pemeriksaan mulut menunjukkan adanya peradangan gusi disertai kerusakan jaringan gusi diantara gigi geligi. Mungkin ditemukan selaput berwarna abu-abu yang berasal dari jaringan gusi yang mati. Bisa terjadi pembengkakan kelenjar getah bening dikepala dan leher. Rontgen gigi atau rontgen wajah dilakukan untuk menentukan luasnya infeksi dan kerusakan jaringan.

Pengobatan :

Pengobatan dilakukan dengan pembersihan, dimana semua jaringan gusi yang mati dan karang gigi dibuang. Karena pembersihan ini menimbulkan nyeri, maka digunakan obat bius lokal. Beberapa hari pertama setelah pembersihan, penderita diharuskan berkumur-kumur dengan larutan hidrogen peroksida (setengah bagian hidrogen peroksida 3% dicampur dengan setengah bagian air), beberapa kali dalam sehari. Selama 2 minggu,

penderita mengunjungi dokter gigi setiap 1-2 hari. Pembersihan berlanjut sampai terjadinya penyembuhan. Jika bentuk dan posisi gusi tidak kembali normal, dokter gigi akan melakukan pembedahan untuk kembali membentuk gusi sebagai pencegahan terhadap kekembuhan dan pencegahan terhadap *periodontitis*. Antibiotik diberikan jika trench mouth sangat berat atau tidak dapat dilakukan perawatan gigi.

Pencegahan :

Menjaga kebersihan mulut sangat penting dalam mencegah terjadinya trench mouth. Makanan dan keadaan kesehatan yang baik juga membantu mencegah terjadinya penyakit ini. Tindakan pencegahan lainnya adalah berhenti merokok dan mencoba mengatasi stres.

#### 2.4.2 Abses Periapikal

Abses Periapikal adalah pengumpulan nanah yang telah menyebar dari sebuah gigi ke jaringan di sekitarnya, biasanya bersal dari suatu infeksi.

Penyebab :

Tubuh menyerang infeksi dengan sejumlah besar sel darah putih, nanah adalah sekumpulan sel darah putih dan jaringan yang mati. Biasanya nanah dari infeksi gigi pada awalnya dialirkan ke gusi, sehingga gusi yang berada di dekat akar gigi tersebut membengkak. Nanah bisa dialirkan ke kulit, mulut, tenggorokan, atau tengkorak, tergantung kepada lokasi gigi yang terkena.

Gejala :

Gigi terasa sakit, bila mengunyah juga timbul nyeri. Kemungkinan ada demam disertai pembengkakan kelenjar getah bening di leher. Jika sangat berat, di daerah rahang terjadi pembengkakan.

Diagnosa :

Diagnosis ditegakkan berdasarkan gejala dan hasil pemeriksaan fisik.

Pengobatan :

Abses atau *selulitis* diatasi dengan menghilangkan infeksi dan membuang nanah melalui bedah mulut atau pengobatan saluran akar. Untuk membantu menghilangkan infeksi seringkali diberikan antibiotik. Tindakan yang terpenting adalah mencabut *pulpa* yang terkena dan mengeluarkan nanahnya.

#### 2.4.3 Bau Mulut

Bau Mulut (*Halitosis*) adalah bau nafas yang tidak enak, tidak menyenangkan dan menusuk hidung.

Penyebab :

Jika bau napas yang sebelumnya normal berubah menjadi halitosis, maka penyebabnya adalah :

- a) Makanan (misalnya bawang mentah, bawang putih, kol)
- b) Vitamin (terutama dalam dosis tinggi)
- c) Kebersihan gigi yang jelek
- d) Gigi *karies*
- e) Merokok
- f) Alkohol
- g) Infeksi tenggorokan
- h) Sinusitis
- i) Infeksi paru-paru
- j) *Sindroma Sjogren*
- k) Penyakit gusi (*gingivitis, gingivostomatitis*)
- l) Abses gigi
- m) Impaksi gigi
- n) Benda asing di hidung (pada anak-anak)
- o) Obat-obatan (paraldehid, triamteren, dan obat bius yang dihirup, suntikan insulin).

Penyakit-penyakit yang bisa menyebabkan bau mulut :

- a. *Gingivitis ulseratif nekrotisasi akut*
  - b. *Mukositis ulseratif nekrotisasi akut*
  - c. Gagal ginjal akut
  - d. Penyumbatan usus
  - e. *Bronkiektasis*
  - f. Gagal ginjal kronis
  - g. *Diabetes Melitus*
  - h. Kanker kerongkongan
  - i. Karsinoma lambung
  - j. *Fistula gastrojejunokolik*
  - k. *Ensefalopati hepatikum*
  - l. *Ketoasidosis diabetikum*
  - m. Abses paru
  - n. *Ozena*
-

- o. Penyakit periodontal
- p. Faringitis
- q. *Divertikulum Zenker*

Gejala :

Bau nafas tercium tidak enak, tidak menyenangkan atau menusuk hidung.

Diagnosa :

Diagnosis ditegakkan berdasarkan gejala dan hasil pemeriksaan fisik yang menyeluruh pada mulut dan hitung. Biakan tenggorokan dilakukan jika terdapat luka di tenggorokan atau dimulut.

Pemeriksaan lainnya yang mungkin harus dilakukan adalah:

- a. *Endoskopi*
- b. *Rontgen perut*
- c. *Rontgen dada*

Pengobatan :

Daun *parsley* segar atau permen *mint* bisa menghasilkan bau mulut yang bersifat sementara.

Pencegahan :

Hal-hal yang bisa dilakukan untuk mencegah terjadinya bau mulut

- a. Menjaga kebersihan gigi
- b. Obat kumur tidak efektif dalam mengatasi penyebab bau mulut dan penggunaan obat kumur yang mengandung alkohol konsentrasi tinggi bisa meningkatkan resiko terjadinya kanker mulut.
- c. Berhenti merokok.

#### 2.4.4 Gingivitis

Gingivitis adalah peradangan pada gusi (gingiva). Gingivitis sering terjadi dan bisa timbul kapan saja setelah tumbuhnya gigi.

Penyebab :

Gingivitis hampir selalu terjadi akibat penggosokan dan *flosing* (membersihkan gigi dengan menggunakan benang gigi) yang tidak benar, sehingga *plak* tetap ada di sepanjang garis gusi. Plak merupakan suatu lapisan yang terutama terdiri dari bakteri. Plak lebih sering menempel pada tambalan yang salah atau di sekitar gigi yang terletak berschelah dengan gigi palsu yang jarang dibersihkan. Jika plak tetap melekat pada gigi selama lebih dari 72 jam, maka akan mengeras dan membentuk karang gigi (*kalkulusflosing* (benang gigi)). Plak merupakan penyebab utama dari *gingivitis*.

---



Faktor lainnya yang akan semakin memperburuk peradangan adalah :

- a. Kehamilan
- b. Pubertas
- c. Pil KB

Obat-obat tertentu bisa menyebabkan pertumbuhan gusi yang berlebihan sehingga plak sulit dibersihkan dan terjadilah gingivitis.

Obat-obat tersebut adalah :

- a. Fenitoin (obat anti kejang)
- b. Siklosporin (diminum oleh penderita yang menjalani pencangkokan organ).
- c. Calcium channel blockers (misalnya nifedipin, obat untuk mengendalikan tekanan darah dan kelainan irama jantung)
- d. Pil atau suntikan KB

**Kekurangan vitamin C** bisa menyebabkan gingivitis, dimana gusi meradang dan mudah berdarah. Kekurangan niasin (*pellagra*) juga bisa menyebabkan peradangan dan perdarahan gusi, serta mempermudah terjadinya infeksi mulut.

Pada kehamilan, gingivitis bisa semakin memburuk. Hal ini terutama disebabkan oleh perubahan hormonal. Keadaan ini didukung oleh kurangnya menjaga kebersihan mulut karena wanita hamil sering mengalami mual di pagi hari. Selama kehamilan, iritasi ringan (yang paling sering adalah pembentukan karang gigi) bisa menyebabkan pertumbuhan berlebih dari jaringan gusi yang menyerupai benjolan. Keadaan ini disebut tumor kehamilan. Jika terluka atau pada saat makan, jaringan gusi yang membengkak ini mudah mengalami perdarahan.

Gingivitis *deskuamativa* merupakan suatu keadaan yang paling sering ditemukan pada wanita pasca menopause. Lapisan gusi yang paling luar terpisah dari jaringan dibawahnya. Gusi menjadi sangat longgar sehingga lapisan terluarnya bisa digerakkan dengan kapas lidi.

Pada perikoronitis, yang membengkak adalah gusi pada sebuah gigi yang belum keluar seluruhnya. Cairan, potongan makanan dan bakteri bisa terperangkap di dalam bagian gusi yang menutupi gigi ini. Bisa terjadi infeksi, yang selanjutnya bisa menyebar ke tenggorokan atau pipi.

Gejala :

Pada **gingivitis simplek**, gusi tampak merah, bukan pink. Gusi membengkak dan mudah digerakkan. Jika penderita menggosok gigi atau makan, gusi seringkali berdarah.

---

Jika gingivitisnya berat, maka pada saat bangun pagi bantal akan dipenuhi oleh bercak darah, terutama jika pada saat tidur penderita bernafas melalui mulutnya.

Gingivostomatitis **herpetik akut** merupakan infeksi virus pada gusi dan bagian mulut lainnya, yang menimbulkan nyeri. Gusi tampak berwarna merah terang dan terdapat banyak luka terbuka yang berwarna putih atau kuning di dalam mulut.

Gingivitis **pada leukemia** merupakan tanda awal dari leukemia pada sekitar 25 % penderita anak-anak. Penyusupan (*infiltrasi*) sel-sel leukemia ke dalam gusi menyebabkan gingivitis dan berkurangnya kemampuan untuk melawan infeksi akan semakin memperburuk keadaan ini. Gusi tampak merah dan mudah berdarah. Perdarahan seringkali berlanjut sampai beberapa menit atau lebih karena pada penderita leukemia, darah tidak membeku secara normal.

Diagnosa :

Diagnosis ditegakkan berdasarkan gejala dan hasil pemeriksaan fisik. Gusi yang meradang tampak merah, membengkak dan mudah berdarah.

Pengobatan :

Kondisi media yang menyebabkan atau memperburuk gingivitis harus diatasi. Jika penyebabnya adalah obat-obatan, maka pertumbuhan gusi yang berlebihan harus diangkat melalui pembedahan.

Jika terjadi kekurangan vitamin C dan niasin, maka diberikan tambahan vitamin. Gingivostomatitis herpetik akut biasanya membaik tanpa pengobatan dalam waktu 2 minggu. Bisa diberikan obat kumur *anestetik* untuk mengurangi rasa tidak nyaman ketika penderita makan dan minum. Tumor kehamilan diangkat melalui pembedahan, tetapi tumor ini cenderung tumbuh kembali selama kehamilan masih berlangsung.

Pada gingivitis deskuamativa diberikan terapi *sulih hormon*. Pilihan pengobatan lainnya adalah tablet kortikosteroid atau salep kortikosteroid yang langsung dioleskan ke gusi. Untuk mencegah terjadinya perdarahan pada leukemia, sebaiknya penderita membersihkan giginya tidak dengan sikat gigi, tetapi menggunakan bantalan atau busa. Obat kumur klorheksidin bisa diberikan untuk mengendalikan plak dan mencegah infeksi mulut. Pada perikoronitis, sisa makanan dan bakteri dibawah lipatan gusi dibersihkan oleh dokter gigi. Jika rontgen menunjukkan bahwa gigi geraham bawah tidak mungkin tumbuh secara sempurna, maka gigi geraham atas dicabut dan diberikan antibiotik selama beberapa hari sebelum gigi geraham bawah juga dicabut.

---

#### 2.4.5 Infeksi Herpes pada Mulut

Infeksi Herpes Mulut Primer (*Gingivostomatitis Herpetik Primer, Herpes Labialis*) adalah suatu infeksi awal oleh *virus herpes simpleks* yang dengan segera bisa menyebabkan terbentuknya luka yang terasa nyeri digusi dan bagian mulut lainnya. *Herpes sekunder (Herpes Labialis Berulang)* adalah suatu *reaktivasi* (pengaktifan kembali) virus lokal yang menyebabkan terbentuknya *cold sore* (luka didekat mulut akibat demam).

Penyebab :

Virus herpes simpleks.

Gejala :

Secara khusus, seorang bayi mendapatkan virus simpleks dari orang dewasa yang memiliki *cold sore*. Infeksi awal pada bayi ini (herpes primer) menyebabkan peradangan gusi biasa dan sakit mulut yang luar biasa. Bisa terjadi demam, pembengkakan kelenjar getah bening di leher dan tidak enak badan, sehingga anak menjadi rewel. Sebagian besar kasus bersifat ringan dan menghilang dengan sendirinya. Orang tua seringkali menduganya sebagai akibat dari pertumbuhan gigi atau penyakit lainnya. Dalam 2-3 hari, timbul lepuhan yang sangat kecil (*vesikel*) di mulut. Vesikel ini mungkin tidak disadari karena mereka segera pecah dan meninggalkan luka terbuka di mulut. Rasa sakit dirasakan diseluruh mulut, terutama gusi. Seminggu kemudian anak akan membaik, tetapi virus herpes simpleks tetap berada dalam tubuhnya, dan infeksi sering berulang di kemudian hari (herpes sekunder).

Infeksi awal menyebabkan sakit yang menyebar di mulut, tetapi infeksi ulangan biasanya menyebabkan timbulnya *cold sore (fever blister, lepuhan yang timbul karena demam)*. Infeksi ulangan biasa dipacu oleh :

- a. Sengatan matahari pada bibir
- b. Demam
- c. Cuzca dingin
- d. Alergi makanan
- e. Cedera di mulut
- f. Pengobatan gigi
- g. Kecemasan.

1-2 hari sebelum timbulnya lepuhan, penderita merasakan kesemutan atau rasa tidak nyaman (gejala *prodroma*) pada daerah dimana lepuhan akan muncul. Perasaan ini sulit

---

untuk diungkapkan, tetapi mudah dikenali pada seseorang yang sebelumnya menderita herpes.

Luka terbuka bisa timbul di bibir bagian luar dan kemudian terbentuk keropeng. Di dalam mulut, luka ini paling sering ditemukan di langit-langit (*palatum*). Luka di mulut berawal sebagai lepuhan-lepuhan kecil yang dengan segera akan bergabung dan membentuk luka merah yang menimbulkan nyeri.

Diagnosa :

Diagnosis ditegakkan berdasarkan hasil pemeriksaan fisik dan hasil biakan dari luka. Pemeriksaan fisik juga bisa menunjukkan adanya pembesaran kelenjar getah bening di leher atau selangkangan. Tes *Tzanck* atau biakan virus dari luka dikulit bisa menunjukkan adanya virus herpes.

Pengobatan :

Tujuan pengobatan pada herpes primer adalah untuk mengurangi rasa sakit, sehingga penderita bisa tidur, makan dan minum secara normal.

Rasa nyeri bisa menyebabkan anak tidak mau makan dan tidak mau minum, bila disertai demam, hal ini bisa dengan segera menyebabkan *dehidrasi* (kekurangan cairan tubuh). Karena itu anak yang sakit harus minum cairan sebanyak mungkin. Untuk mengurangi nyeri pada penderita dewasa atau anak yang lebih besar, bisa digunakan obat kumur *anestetik* (misalnya lidokain) atau bisa juga digunakan obat kumur yang mengandung baking soda.

Pencegahan :

Tindakan berikut bisa dilakukan untuk mencegah terjadinya herpes labialis

- a. Menghindari kontak langsung dengan *cold sore* atau luka herpes lainnya.
- b. Memperkecil kemungkinan terjadinya penularan secara tidak langsung dengan cara mencuci benda-benda yang telah digunakan oleh penderita dengan air panas (lebih baik direbus).
- c. Tidak memakai benda bersama-sama dengan penderita herpes, terutama ketika lukanya sedang aktif.
- d. Menghindari faktor pencetus (misalnya sinar matahari).

#### 2.4.6 *Pulpitis*

*Pulpitis* adalah peradangan pada *pulpa* gigi yang menimbulkan rasa nyeri.

*Pulpa* adalah bagian gigi yang paling dalam, yang mengandung saraf dan pembuluh darah.

Penyebab:

---

Penyebab *pulpitis* yang paling sering ditemukan adalah pembusukan gigi, penyebab kedua adalah cedera. Pulpa terbungkus dalam dinding yang keras sehingga tidak memiliki ruang yang cukup untuk membengkak ketika terjadi peradangan. Yang terjadi hanyalah peningkatan tekanan di dalam gigi. Peradangan yang ringan, jika berhasil diatasi, tidak akan menimbulkan kerusakan gigi yang permanen. Peradangan yang berat akan mematikan pulpa. Meningkatnya tekanan di dalam gigi bisa mendorong pulpa melalui ujung akar sehingga bisa melukai tulang rahang dan jaringan disekitarnya.

Gejala :

*Pulpitis* menyebabkan sakit gigi yang luar biasa.

Diagnosa :

Diagnosis ditegakkan berdasarkan gejala dan hasil pemeriksaan fisik. Untuk menentukan apakah pulpa masih bisa diselamatkan, bisa dilakukan beberapa pengujian :

a. Diberikan rangsangan dingin.

Jika setelah rangsangan dihentikan nyerinya hilang, berarti pulpa masih sehat. Pulpa bisa dipertahankan dengan cara mencabut bagian gigi yang membusuk dan menambalnya. Jika nyeri tetapa ada meskipun rangsangan dingin telah dihilangkan atau jika nyeri timbul secara spontan, maka pulpa tidak dapat dipertahankan.

b. Penguji pulpa elektrik.

Alat ini digunakan untuk menunjukkan apakah pulpa masih hidup, bukan untuk menentukan apakah pulpa masih sehat. Jika penderita merasakan aliran listrik pada giginya, berarti pulpa masih hidup.

c. Menepuk gigi dengan sebuah alat.

Jika dengan pengetukan gigi timbul nyeri, berarti peradangan telah menyebar ke jaringan dan tulang dan sekitarnya.

d. Rontgen gigi.

Dilakukan untuk memperkuat adanya pembusukan gigi dan menunjukkan apakah penyebaran peradangan telah menyebabkan pengeroposan tulang disekitar akar gigi.

Pengobatan :

Peradangan mereda jika penyebabnya diobati. Jika *pulpitis* diketahui pada stadium dini, maka penambalan sementara yang mengandung obat penenang saraf bisa menghilangkan nyeri. Tambalan ini bisa dibiarkan sampai 6-8 minggu dan kemudian diganti dengan tambalan permanen. Jika terjadi kerusakan pulpa yang luas dan tidak dapat diperbaiki, satu-satunya cara untuk menghilangkan nyeri adalah dengan mencabut pulpa, baik melalui pengobatan saluran akar maupun dengan pencabutan gigi.

### 2.4.7 Periodontitis

Periodontitis (*pioire*) terjadi jika gingivitis menyebar ke struktur penyangga gigi. Periodontitis merupakan salah satu penyebab utama lepasnya gigi pada dewasa dan merupakan penyebab utama lepasnya gigi pada lanjut usia.

Penyebab :

Sebagian besar periodontitis merupakan akibat dari penumpukan plak dan karang gigi (*tartar*) diantara gigi dan gusi.

Akan terbentuk kantong diantara gigi dan gusi dan meluas ke bawah diantara akar gigi dan tulang dibawahnya. Kantong ini mengumpulkan plak dalam suatu lingkungan bebas oksigen, yang mempermudah pertumbuhan bakteri. Jika keadaan ini terus berlanjut, pada akhirnya banyak tulang rahang di dekat kantong yang dirusak sehingga gigi lepas.

Kecepatan tumbuhnya periodontitis berbeda pada orang-orang yang memiliki jumlah tartar yang sama. Hal ini mungkin karena plak dari masing-masing orang tersebut mengandung jenis dan jumlah bakteri yang berbeda, dan karena respon yang berbeda terhadap bakteri.

Beberapa keadaan medis yang bisa mempermudah terjadinya periodontitis adalah :

- a. Diabetes melitus
- b. Sindroma *Down*
- c. Penyakit *Crohn*
- d. Kekurangan sel darah putih
- e. AIDS.

Gejala-gejala dari periodontitis adalah :

- a. Perdarahan gusi
- b. Perubahan warna gusi
- c. Bau mulut (*halitosis*).

Diagnosa :

Pada pemeriksaan mulut dan gigi, gusi tampak bengkak dan berwarna merah keunguan. Akan tampak endapan plak atau karang di dasar gigi disertai kantong yang melebar digusi. Dokter gigi akan mengukur kedalaman kantong dalam gusi dengan suatu alat tipis dan dilakukan rontgen gigi untuk mengetahui jumlah tulang yang keropos. Semakin banyak tulang yang keropos, maka gigi akan lepas dan berubah posisinya. Gigi depan seringkali menjadi miring ke luar. Periodontitis biasanya tidak menimbulkan

---

nyeri kecuali jika gigi sangat longgar sehingga ikut bergerak ketika mengunyah atau jika terbentuk *abses* (pengumpulan nanah).

Pengobatan :

Seorang dokter gigi bisa membersihkan kantong sampai kedalaman 0,5 cm dengan alat khusus, yang dapat membuang seluruh karang gigi dan permukaan akar gigi yang sakit. Untuk kantong yang dalamnya mencapai 0,6 cm atau lebih, seringkali diperlukan pembedahan. Seorang dokter gigi juga dapat mengangkat sebagian gusi yang terpisah sehingga gusi yang tertinggal bisa direkatkan lagi dengan lebih erat ke gigi dan penderita bisa membersihkan plaknya di rumah. Jika terbentuk abses, diberikan antibiotik. Ke dalam kantong yang dalam bisa dimasukkan *filamen* yang mengandung antibiotik sehingga obat bisa mencapai daerah yang sakit dalam konsentrasi yang tinggi. *Abses Periodontal* menyebabkan serangan pengrusakan tulang, tetapi pengobatan segera dengan pembedahan dan antibiotik memungkinkan tulang yang rusak untuk tumbuh kembali. Jika setelah pembedahan timbul luka terbuka di mulut, diberikan obat kumur klorheksidin selama 1 menit, 2 kali/hari untuk sementara waktu, menggantikan gosok gigi dan pemakaian benang gigi.

Pencegahan :

Pencegahan terbaik adalah menjaga kebersihan mulut dan gigi. Pengobatan dan pencegahan *gingivitis* dapat mengurangi resiko terjadinya periodontitis.

## 2.5 Borland Delphi

Delphi adalah suatu bahasa pemrograman (*development language*) yang digunakan untuk merancang suatu aplikasi program. Delphi adalah compiler/penterjemah bahasa Pascal (awalnya dari Pascal) yang merupakan bahasa tingkat tinggi sekelas dengan basic, C. Bahasa pemrograman di delphi disebut bahasa *procedural* artinya bahasa/sintaknya mengikuti urutan tertentu/prosedur. Delphi termasuk keluarga visual sekelas Visual Basic, Visual C, artinya perintah-perintah untuk membuat objek dapat dilakukan secara visual. Delphi juga merupakan bahasa berorientasi objek. Saat anda pertama kali masuk ke Visual Basic maupun delphi, anda akan diperhadapkan pada sebuah form kosong yang akan dibuat secara otomatis. Form tersebut diberi nama Form1. Form ini merupakan tempat bekerja untuk membuat antarmuka pengguna (*user interface*).

---

### 2.5.1 Kegunaan Delphi

1. Untuk membuat aplikasi windows.
2. Untuk merancang aplikasi program berbasis grafis.
3. Untuk membuat program berbasis jaringan (*client / server*).
4. Untuk merancang program .net (berbasis internet).

### 2.5.2 Keunggulan Delphi

1. IDE (*Integrated Development Environment*) atau lingkungan pengembangan aplikasi sendiri adalah satu dari beberapa keunggulan delphi, didalamnya terdapat menu – menu yang memudahkan kita untuk membuat suatu proyek program.
2. Proses kompilasi cepat, pada saat aplikasi yang kita buat dijalankan pada delphi, maka secara otomatis akan dibaca sebagai sebuah program, tanpa dijalankan terpisah.
3. Mudah digunakan, source code delphi yang merupakan turunan dari pascal, sehingga tidak perlu suatu penyesuaian lagi.
4. Bersifat multi purphase, artinya bahasa pemrograman delphi dapat digunakan untuk pengembangan aplikasi.

### 2.5.3 Menu Aplikasi Delphi

#### ▪ Menu Bar

Berfungsi untuk memilih tugas-tugas tertentu, seperti memulai, membuka, dan menyimpan

*project*, mengompilasi *project* menjadi *file executable (EXE)*, dan lain-lain



Gambar 2.9 Menu Bar

#### ▪ Tool Bar/Speed Bar

Memiliki fungsi yang sama seperti menu bar, tetapi berfungsi seperti jalan pintas karena lebih

praktis dalam penggunaannya



Gambar 2.10 Tool Bar

#### ▪ Component Palette

*Component Palette* berisi kumpulan ikon yang melambangkan komponen-komponen yang terdapat pada VCL (*Visual Component Library*). Pada *Component Palette*, akan



ditemukan beberapa *page control*, seperti Standard, Additional, Win32, System, Data Access dan lain-lain. Ikon tombol *pointer* terdapat di setiap *page control*



Gambar 2.11 Component Palette

- Form

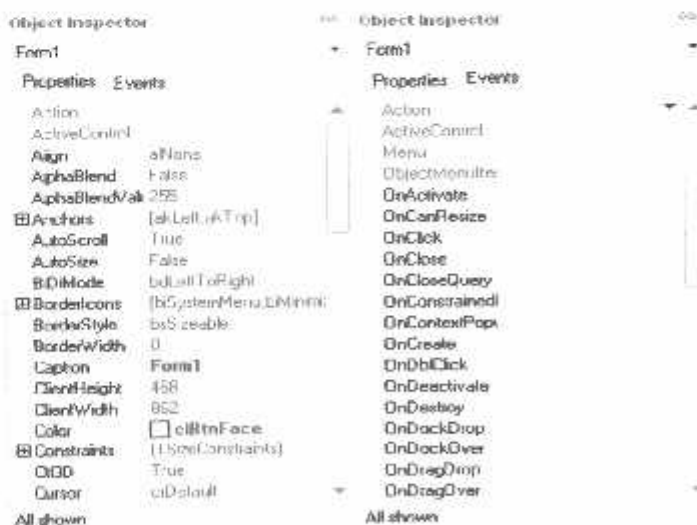
*Form Designer* merupakan suatu objek yang dapat dipakai sebagai tempat untuk merancang program aplikasi. *Form* berbentuk sebuah meja kerja yang dapat diisi dengan komponen-komponen yang diambil dari *Component Palette*



Gambar 2.12 Form

- Object Inspector

*Object Inspector* digunakan untuk mengubah properti dan karakteristik dari sebuah komponen. *Object Inspector* terdiri dari dua tab, yaitu *Properties* dan *Events*. Tab *Properties* digunakan untuk mengubah properti komponen. Properti dengan tanda + menunjukkan bahwa properti tersebut mempunyai subproperti. Tab *Events*, bagian yang dapat diisi dengan kode program tertentu yang berfungsi untuk menangani *event-event* (kejadiankejadian yang berupa sebuah *procedure*) yang dapat direspon oleh sebuah komponen



Gambar 2.13 Object Inspector

- Object Tree View

*Object Tree View* menampilkan diagram pohon dari komponen-komponen yang bersifat visual maupun nonvisual yang telah terdapat dalam *form*, data *module*, atau *frame*. *Object Tree View* juga menampilkan hubungan logika antarkomponen



Gambar 2.14 Object Tree View

- Code Editor

*Code Editor* merupakan tempat menuliskan kode program atau pernyataan-pernyataan dalam *Object Pascal*. *Code Editor* dilengkapi dengan fasilitas *highlight* yang memudahkan pemakai menemukan kesalahan.

*Title bar* yang terletak pada bagian atas jendela *code editor* menunjukkan nama *file* yang sedang disunting, serta pada bagian informasi yang perlu untuk diperhatikan, yaitu : Nomor baris/kolom yang terletak pada bagian paling kiri. Bagian ini berfungsi untuk menunjukkan posisi kursor di dalam jendela *Code Editor*. *Modified* menunjukkan bahwa *file* yang sedang disunting telah mengalami perubahan tersebut belum disimpan. Teks ini akan hilang jika telah menyimpan perubahan. *Insert/Overwrite* yang terletak pada bagian paling kanan menunjukkan bahwa modus pengetikan teks dalam jendela *Code Editor*. *Insert* menunjukkan bahwa modus penyisipan teks dalam keadaan aktif, sedangkan *Overwrite* menunjukkan bahwa modus penimpaan teks dalam keadaan aktif.



Gambar 2.15 Code Editor

#### 2.5.4 Komponen Delphi

Dalam pemrograman *delphi*, komponen digunakan untuk membantu seorang pembuat program untuk membuat suatu program. Kelebihan komponen itu sendiri adalah membuat program itu bisa tampil lebih bagus dari pada komponen bawaan asli *delphi* sendiri. Adapun komponen yang digunakan dalam pembuatan program aplikasi ini adalah :

#### 2.5.5 Alite

Komponen ini digunakan untuk memperindah tampilan dari aplikasi.

### 2.6 MySQL

MySQL adalah Relational Database Management System (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL (General Public License). Dimana setiap orang bebas untuk menggunakan MySQL, namun tidak boleh dijadikan produk turunan yang bersifat komersial. MySQL sebenarnya merupakan turunan salah satu konsep utama dalam database sejak lama, yaitu SQL (Structured Query Language).

SQL adalah sebuah konsep pengoperasian database, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis. Keandalan suatu sistem database (DBMS) dapat diketahui dari cara kerja optimizer-nya dalam melakukan proses perintah-perintah SQL, yang dibuat oleh user maupun program-program aplikasinya.

Sebagai database server, MySQL dapat dikatakan lebih unggul dibandingkan database server lainnya dalam query data. Hal ini terbukti untuk query yang dilakukan oleh single user, kecepatan query MySQL bisa sepuluh kali lebih cepat dari PostgreSQL dan lima kali lebih cepat dibandingkan Interbase<sup>[6]</sup>.

MySQL memiliki beberapa keistimewaan, antara lain :

1. **Portabilitas.** MySQL dapat berjalan stabil pada berbagai sistem operasi seperti Windows, Linux, FreeBSD, Mac Os X Server, Solaris, Amiga, dan masih banyak lagi.
  2. **Open Source.** MySQL didistribusikan secara *open source*, dibawah lisensi GPL sehingga dapat digunakan secara cuma-cuma.
  3. **'Multiuser'.** MySQL dapat digunakan oleh beberapa user dalam waktu yang bersamaan tanpa mengalami masalah atau konflik.
-

4. **'Performance tuning'**. MySQL memiliki kecepatan yang menakjubkan dalam menangani query sederhana, dengan kata lain dapat memproses lebih banyak SQL per satuan waktu.
5. **Jenis Kolom**. MySQL memiliki tipe kolom yang sangat kompleks, seperti signed / unsigned integer, float, double, char, text, date, timestamp, dan lain-lain.
6. **Perintah dan Fungsi**. MySQL memiliki operator dan fungsi secara penuh yang mendukung perintah Select dan Where dalam perintah (*query*).
7. **Keamanan**. MySQL memiliki beberapa lapisan sekuritas seperti level subnetmask, nama host, dan izin akses *user* dengan sistem perizinan yang mendetail serta sandi terenkripsi.
8. **Skalabilitas dan Pembatasan**. MySQL mampu menangani basis data dalam skala besar, dengan jumlah rekaman (records) lebih dari 50 juta dan 60 ribu tabel serta 5 milyar baris. Selain itu batas indeks yang dapat ditampung mencapai 32 indeks pada tiap tabelnya.
9. **Konektivitas**. MySQL dapat melakukan koneksi dengan klien menggunakan protokol TCP/IP, Unix socket (UNIX), atau Named Pipes (NT).
10. **Lokalisasi**. MySQL dapat mendeteksi pesan kesalahan pada klien dengan menggunakan lebih dari dua puluh bahasa. Meski pun demikian, bahasa Indonesia belum termasuk di dalamnya.
11. **Antar Muka**. MySQL memiliki interface (antar muka) terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API (Application Programming Interface).
12. **Klien dan Peralatan**. MySQL dilengkapi dengan berbagai peralatan (tool) yang dapat digunakan untuk administrasi basis data, dan pada setiap peralatan yang ada disertakan petunjuk online.
13. **Struktur tabel**. MySQL memiliki struktur tabel yang lebih fleksibel dalam menangani ALTER TABLE, dibandingkan basis data lainnya semacam PostgreSQL ataupun Oracle.

## 2.7 Xampp

Xampp adalah perangkat lunak gratis, yang mendukung banyak system operasi, merupakan kompilasi dari beberapa program untuk menjalankan fungsinya sebagai server yang berdiri sendiri, yang terdiri atas program apache Http server, mysql databasedan penterjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan

---

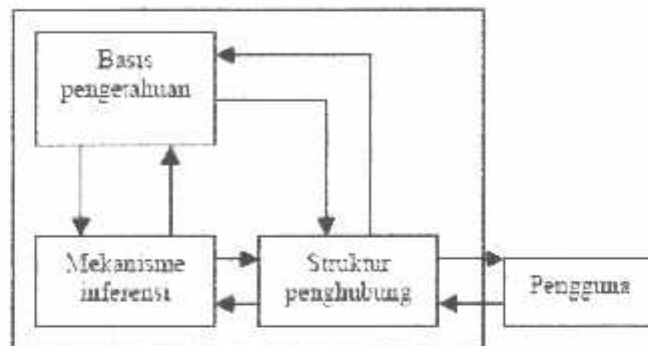
Perl. Xampp adalah nama yang merupakan singkatan dari X (empat system operasi apapun),apache,Mysql,PHP dan Perl. Program ini tersedia dalam GNU (General Public License)dan bebas. Xampp juga merupakan web server yang mudah digunakan yang dapat melayani tampilan halaman web yang dinamis.

---

## BAB III PERANCANGAN DAN DESAIN SISTEM

### 3.1 Desain Sistem

Desain Sistem disajikan untuk mengetahui bagaimana aplikasi ini berjalan berdasarkan rule-rule yang telah ditentukan sebelumnya adapun konsep-konsep dari pada desain system pakar untuk gigi dan mulut adalah sebagai berikut :



Gambar 3.1 konsep dasar Sistem Pakar

Dari gambar tersebut dapat kita ketahui bahwa struktur dari system pakar tersusun atas tiga komponen utama yaitu system berbasis pengetahuan, mekanisme inferensi dan struktur penghubung antara pengguna dan system.

Pengguna dan sistem dihubungkan dengan sebuah aplikasi, aplikasi inilah yang dinamakan sebagai struktur penghubung antara pengguna dan sistem pakar yang dibuat. Basis pengetahuan merupakan inti dari suatu sistem pakar, yaitu berupa representasi pengetahuan dari pakar. Basis pengetahuan tersusun atas fakta dan kaidah. Fakta adalah informasi tentang objek, peristiwa, atau situasi.

Mesin inferensi berperan sebagai otak dari sistem pakar. Mesin inferensi berfungsi untuk memandu proses penalaran terhadap suatu kondisi, berdasarkan pada basis pengetahuan yang tersedia. Di dalam mesin inferensi terjadi proses untuk memanipulasi dan mengarahkan kaidah, model, dan fakta yang disimpan dalam basis pengetahuan dalam rangka mencapai solusi atau kesimpulan. Dalam prosesnya, mesin inferensi menggunakan strategi penalaran dan strategi pengendalian. Strategi penalaran terdiri dari strategi penalaran pasti (*Exact Reasoning*) dan strategi penalaran tak pasti (*Inexact Reasoning*). *Exact reasoning* akan dilakukan jika semua data yang dibutuhkan untuk menarik suatu kesimpulan tersedia, sedangkan *inexact reasoning* dilakukan pada keadaan sebaliknya. Strategi pengendalian berfungsi sebagai panduan arah dalam melakukan proses penalaran.

- Ada dua metode inferensi (penalaran) yang biasa dipergunakan dalam sistem pakar, yaitu metode *Forward Chaining* dan *Backward Chaining* :

### 3.1. *Backward Chaining*

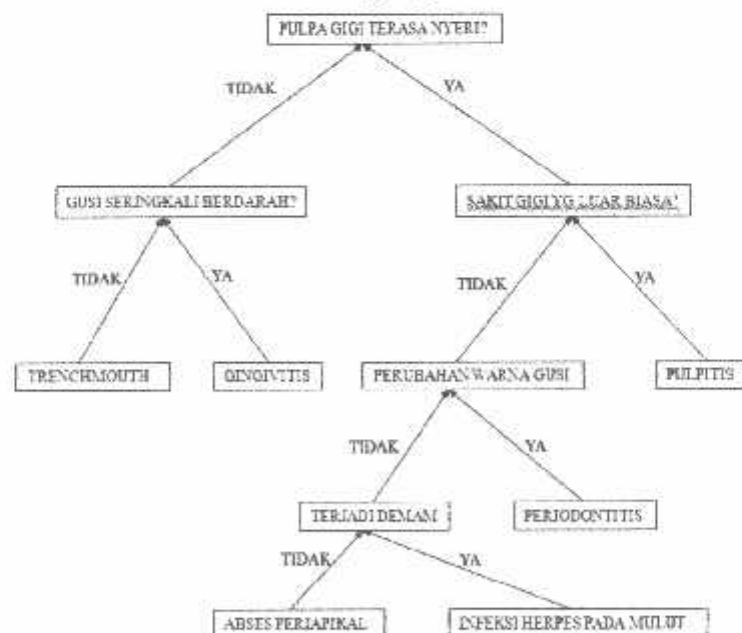
- Metode *Backward chaining* adalah suatu metode untuk menemukan suatu fakta dengan cara menelusuri subgoals yang ada secara rekursif. Cara kerjanya inference engine memulai dari goal yang telah ditentukan kemudian berjalan mundur untuk membuktikan kebenaran goal tersebut berdasarkan rule-rule apa saja yang dapat membentuk goal tersebut.

Atau dengan kata lain merupakan proses pencarian solusi dari kesimpulan kemudian menelusuri fakta-fakta yang ada hingga menemukan solusi yang sesuai dengan fakta-fakta yang diberikan oleh user (*goal-driven*)

### 3.2. *Forward Chaining*

- Proses dimulai dengan pencarian fakta-fakta awal dahulu sebelum menentukan sebuah kesimpulan.
- Fakta-fakta dalam dalam sistem disimpan dalam memori kerja dan secara kontinyu diperbarui. Memori kerja diartikan sebagai tatanan aturan yang sudah di tentukan didalam sistem.
- Aktivitas sistem dilakukan berdasarkan siklus mengenal-beraksi (*recognise-act*). Mula-mula, sistem mencari semua aturan yang kondisinya terdapat di memori kerja. Kemudian mencocokkan dengan aturan yang sudah ditanamkan didalam sistem.

- Struktur Penelusuran Backward Chaining Gigi Dan Mulut



Gambar 3.2 Struktur Penelusuran Backward Chaining Gigi dan Mulut

**Penjelasan :**

JIKA pertanyaan = “pulpa gigi terasa nyeri?”

DAN Jawaban = “Tidak”

MAKA pertanyaan = “Gusi seringkali berdarah?”

JIKA pertanyaan = “pulpa gigi terasa nyeri?”

DAN Jawaban = “YA”

MAKA pertanyaan = “Sakit gigi yang luar biasa?”

Dan seterusnya untuk node-node lainnya node daun akan memberikan jawaban sebagai respons dari pertanyaan.

**3.2 Algoritma Metode Backward Chaining**

Penggunaan strategi pencarian runut balik untuk membangun mesin inferensi memerlukan suatu algoritma tertentu sehingga bias diimplementasikan menggunakan bahasa pemrograman. Salah satu algoritma runut balik yang diambil dari buku *Introduction To Expert Systems : The Development and Implementation Of Rule-Based Expert Systems* (Ignizio, 1991) adalah sebagai berikut :

**3.2.1 Inisialisasi**

3.2.1.1 Buat 3 tabel yaitu :

1. Tabel Working Memory, untuk menyimpan pasangan atribut-nilai hasil dari proses inferensi.
2. Tabel Goal, untuk menyimpan atribut yang nilainya sedang dicari.
3. Tabel Rule/Premise Status, untuk menyimpan nomor kaidah, status dari kaidah yang bersesuaian, pasangan atribut-nilai dari klausa premis kaidah-kaidah dalam basis pengetahuan, nomor klausa premis dalam kaidah yang bersesuaian, dan status dari klausa premis tersebut.

3.2.1.2 Semua klausa premis dalam tabel Rule/Premise Status diberi status free.

3.2.1.3 Semua kaidah dalam tabel Rule/Premise Status diberi status active.

**3.2.2 Mulai inferensi**

- a) Sebutkan kesimpulan akhir (solusi dari permasalahan) yang ingin dicapai.
  - b) Letakkan atribut dari klausa kesimpulan yang merupakan solusi dari permasalahan pada puncak tabel Goal.
-



### 3.2.3 Pengecekan kaidah

- a) Jika tabel Goal kosong maka STOP.
- b) Jika tabel Goal tidak kosong maka cari kaidah-kaidah berstatus active yang atribut dari klausa kesimpulannya bersesuaian dengan atribut yang berada pada puncak tabel Goal.
  - 1) Jika hanya 1 kaidah yang ditemukan, lakukan langkah 6. Jika ada beberapa kaidah yang ditemukan, cari yang berstatus triggered, lakukan langkah 6. Jika tidak ada kaidah yang berstatus triggered, pilih salah satu kaidah lalu proses dengan langkah 6.
  - 2) Jika tidak ada kaidah yang ditemukan, lakukan langkah 4.

### 3.2.4 Query

Ambil salah satu data dari sekumpulan data yang diberikan oleh pemakai yang bersesuaian dengan atribut yang berada pada puncak tabel Goal.

- a) Jika tidak ada maka STOP.
- b) Jika ada maka pindahkan atribut yang berada pada puncak tabel Goal lalu tempatkan pada tabel Working Memory beserta nilainya, yaitu data yang diambil tadi.

### 3.2.5 Pembaharuan status kaidah/premis

- a) Gunakan isi dari tabel Working Memory untuk membaharui tabel Rule/Premise Status.
  - b) Jika ada klausa premis yang berstatus false pada suatu kaidah maka beri status discard pada kaidah tersebut. Tetapi jika seluruh klausa premis pada suatu kaidah semuanya berstatus true maka beri status triggered pada kaidah tersebut.
- b) Kembali ke langkah 3.

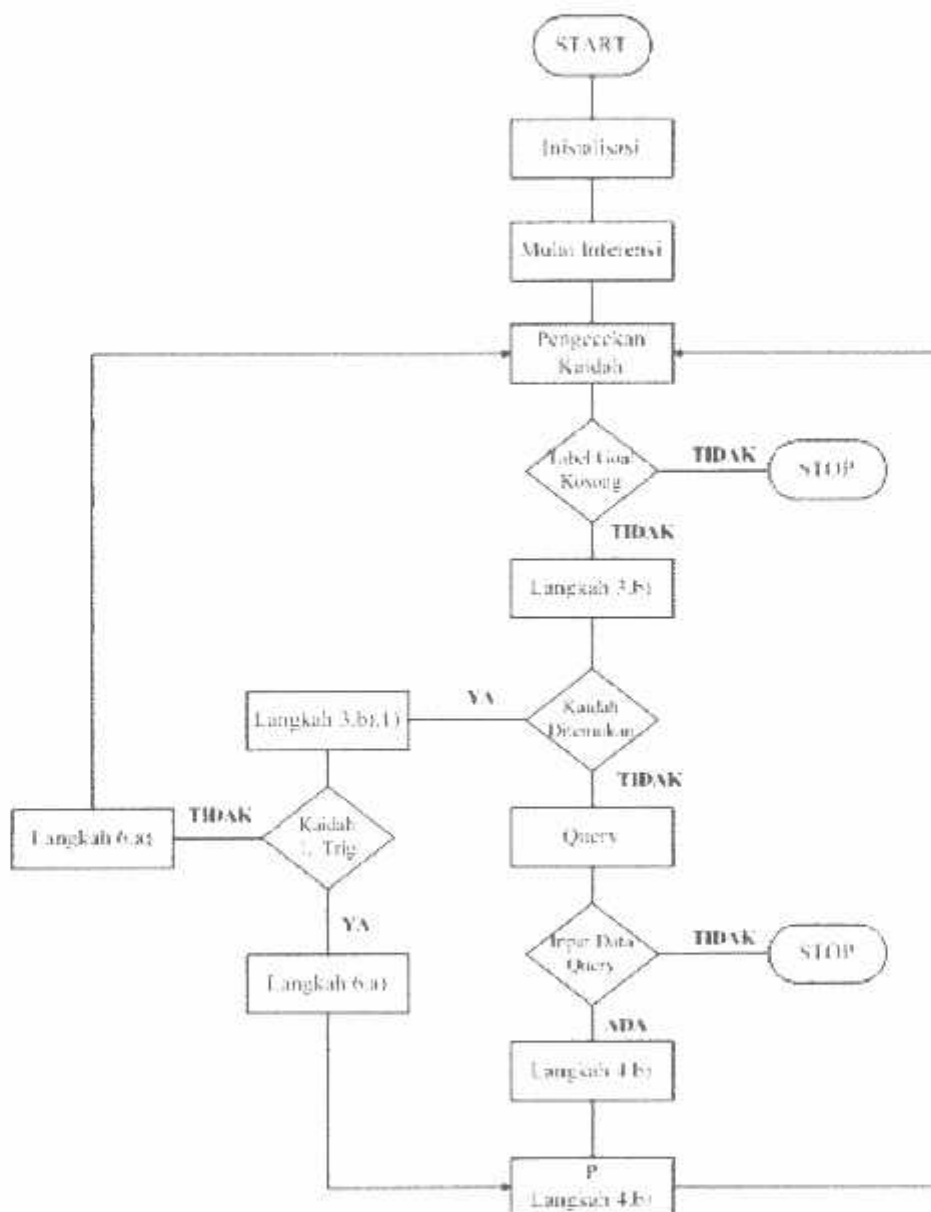
### 3.2.6 Evaluasi kaidah

- a) Jika kaidah berstatus triggered maka pindahkan atribut yang berada pada puncak tabel Goal ke tabel Working Memory beserta nilainya yang terdapat dalam klausa kesimpulan pada kaidah tersebut. Ubah status kaidah tersebut menjadi fired. Kembali ke langkah 5.
-

b) Jika kaidah tidak berstatus triggered maka pilih atribut dari klausa premis pertama yang berstatus free lalu tempatkan pada puncak tabel Goal. Kembali ke langkah 3.

Algoritma runut balik di atas mempunyai 2 keadaan yang menyebabkan proses inferensi berhenti, yaitu :

1. Pada saat tabel Goal kosong, berarti kesimpulan yang merupakan solusi dari permasalahan sudah diperoleh.
2. Pada saat data dari pemakai yang diminta oleh sistem pakar untuk memenuhi prompt kaidah tidak ada, berarti kesimpulan yang merupakan solusi dari permasalahan tidak ditemukan.



Gambar 3.3 Diagram Algoritma Backward Chaining

Algoritma ini bisa dipakai sebagai salah satu pilihan, apabila sistem pakar yang dikembangkan memang menggunakan backward chaining.

### 3.3. DIAGRAM BLOK SISTEM

Diagram blok sistem menggambarkan setiap blok atau bagian dari dari sistem aplikasi. Sistem Pakar untuk Mendiagnosa penyakit Gigi dan Mulut pada manusia dirancang untuk dapat memberikan informasi kepada pengguna, gangguan kesehatan yang diderita. Pengguna memberi masukkan berupa gejala-gejala gangguan kesehatan. Dari gejala-gejala tersebut sistem pakar akan memberikan informasi mengenai gangguan kesehatan yang diderita pasien. Sistem pakar tersebut dapat digambarkan dalam diagram blok seperti yang diperlihatkan dalam Gambar 3.3.



Gambar 3.4. Diagram Blok Sistem Pakar untuk Mendiagnosa penyakit Gigi dan Mulut .

### 3.4. Metode Perancangan

#### 3.4.1 Perancangan Sistem Aplikasi

Perancangan sistem adalah penggambaran, perencanaan dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah ke dalam satu kesatuan yang utuh dan berfungsi. Tujuan dari perancangan secara umum adalah untuk memberikan gambaran secara umum kepada pengguna tentang *implementasi* sistem pakar dalam bidang gigi dan mulut sebagai pendukung pengambilan keputusan dengan metode *backward chaining*. Karena aplikasi sistem pakar ini merupakan proses pencarian solusi dari kesimpulan kemudian menelusuri fakta-fakta yang ada hingga menemukan solusi yang sesuai dengan fakta-fakta yang diberikan oleh user.

#### 3.4.2 Teknik Perancangan

Teknik yang digunakan pada desain implementasi sistem pakar gigi dan mulut sebagai pendukung pengambilan keputusan adalah teknik *prototyping*. Mengingat Gigi dan Mulut merupakan suatu cabang ilmu yang terus berkembang, maka perancangan

sistem yang akan dibuat menuntut keluwesan. Adanya penambahan data-data terkomputerisasi maupun perubahan data-data input diharapkan dapat diantisipasi oleh sistem pakar dalam bidang Gigi dan Mulut sebagai pendukung pengambilan keputusan tanpa pemrograman ulang.

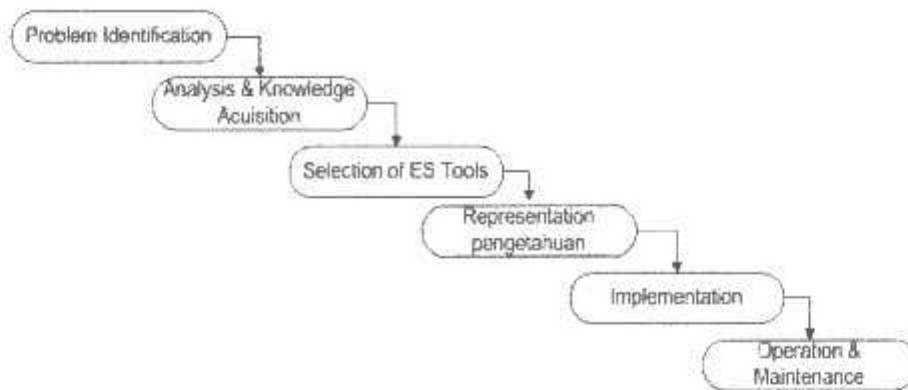
Sistem pakar dikembangkan dengan metode yang disebut dengan Expert System Development Life Cycle (ESDLC), terdiri dari tahapan[5] :

- **Problem Identification (Identifikasi Masalah)**  
Mengidentifikasi masalah yang ingin diangkat untuk diselesaikan dengan sistem pakar yang akan dibuat
- **Preliminary Requirement Analysis & Knowledge Acquisition (Analisis Kebutuhan dan Akuisisi Pengetahuan)**  
Pencarian sumber-sumber pengetahuan yang bersangkutan dengan hal yang akan dibahas, meliputi : wawancara dengan sang pakar, buku-buku, artikel serta informasi-informasi lainnya yang memang dibutuhkan untuk menyempurnakan sistem pakar yang dibuat.
- **Selection of ES Tools (Pemilihan Tools Sistem Pakar )**  
Meliputi pemilihan software apa saja yang akan digunakan untuk membangun aplikasi ini.
- **Representation (Representasi Pengetahuan)**  
Penanaman pengetahuan didalam sistem pakar yang dibuat, atau dikatakan sebagai basis aturan yang kelak akan digunakan dalam sistem pakar.
- **Verification & Validation (Verifikasi dan validasi)**
- **Implementation (Implementasi )**  
Pengimplementasian aplikasi sistem pakar, meliputi desain dan pemrograman didalamnya juga pengujian program yang dibuat.
- **Operation & Maintenance ( Pengoperasian dan Pemeliharaan)**  
Pengoperasian dan perawatan meliputi pengoptimalan pengoperasian serta pemeliharaan setelah aplikasi selesai dibuat. Pemeliharaan bisa diartikan sebagai perbaikan serta penambahan pengetahuan baru didalam aplikasi.

### **3.4.3 Proses sistem Pakar**

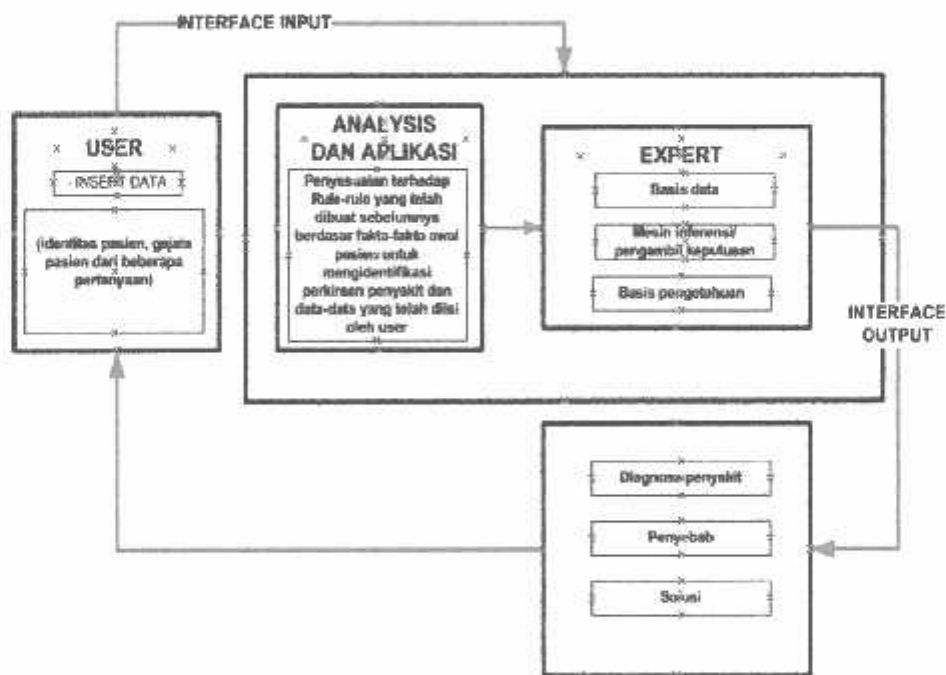
Aplikasi Sistem Pakar Gigi dan mulut ini mengacu kepada model proses pengembangan Perangkat Lunak waterfall.

---



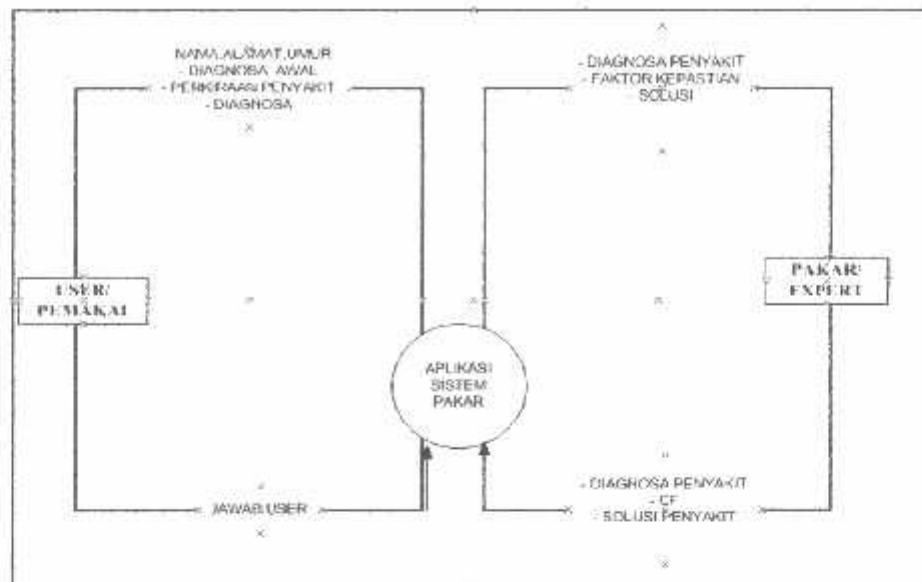
Gambar 3.5 Model Proses PL *Waterfall*

Aplikasi sistem pakar ini terbagi atas 3 komponen dasar, diantaranya : *User*, *Analysis* dan *expert*. Bagian-bagian ini mempunyai fungsi-fungsi yang berbeda tetapi saling mendukung komponen yang lain diantaranya berfungsi sebagai pintu gerbang masuk sekaligus sebagai pemroses dan sebagai sistem pakar itu sendiri, berikut gambaran untuk lebih jelasnya :



Gambar 3.6 konsep dasar Sistem Pakar Gigi dan Mulut

Berikut juga dengan perancangan proses dilakukan dengan memakai alat bantu berupa diagram konteks, berikut gambar beserta penjelasannya :



Gambar 3.7 Diagram konteks perancangan sistem pakar Gigi dan Mulut

Keterangan :

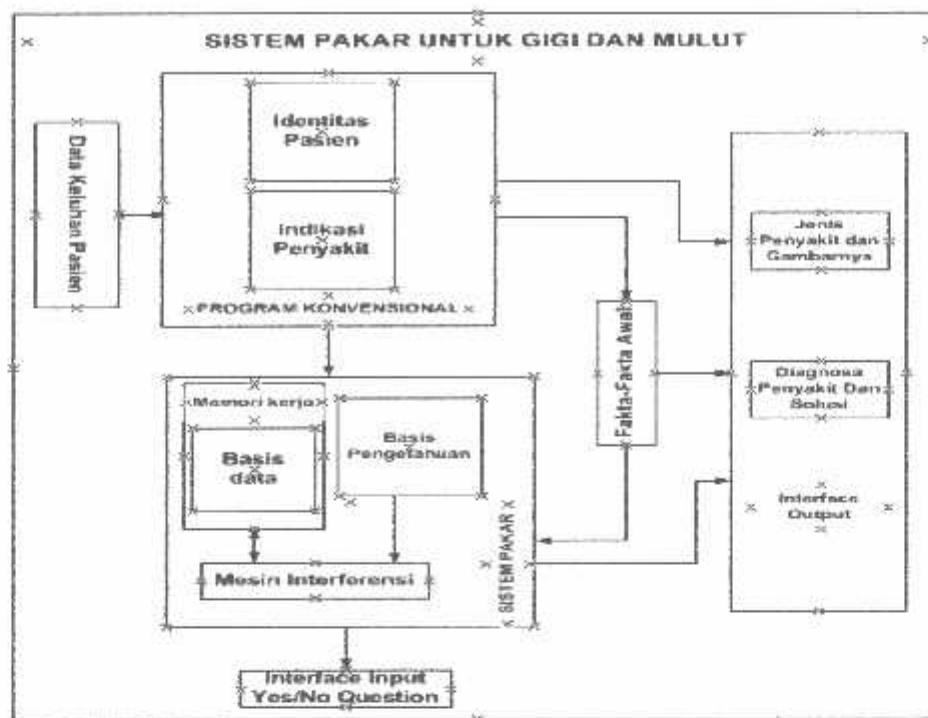
- Proses diawali oleh User/pemakai aplikasi ini
- User akan menginputkan data-data yang diperlukan untuk diproses dalam aplikasi system pakar ini
  - Data-data berupa :
    - Nama,Alamat,Umur
    - Diagnosa Awal
    - Perkiraan Penyakit
    - Diagnosa
- Data-data inilah yang harus diisi/jawab oleh user sebelum memproses lebih lanjut
  - Data-data tersebut akan di inputkan ke aplikasi system pakar ini
  - Begitu juga hal yang sebelumnya, dari aplikasi ini data-data ini akan diproses lebih lanjut dengan rule-rule yang telah dibuat sebelumnya
  - Dari data-data tersebut akan diperoleh beberapa kesimpulan dasar atas :
    - Diagnosa Penyakit
    - Faktor Kepastian
    - Solusi

- Selanjutnya dari kesimpulan dasar tersebut akan diproses lebih lanjut oleh pakar dari system ini melewati pengidentifikasian system hingga diperoleh data-data akurat yang diperlukan oleh user, diantaranya :
  - Diagnosa Penyakit
  - CF
  - Solusi Penyakit
- Selanjutnya data-data ini akan dikembalikan ke aplikasi system pakar ini berupa informasi yang dibutuhkan oleh user dan juga solusinya.

Secara garis besar sistem pakar dalam bidang Gigi dan mulut dibuat dengan tuntutan untuk melakukan tugas sebagai berikut : (1). Mengambil data-data keluhan user, (2). Memasukan dan membandingkan data-data tersebut ke dalam rule-rule yang telah dituliskan dalam basis pengetahuan, (3). Mendeskripsikan kondisi user berdasarkan kesimpulan yang didapat dari hasil membandingkan seperti yang telah dilakukan pada tugas (2). Deskripsi kondisi user sebagai output sistem pakar dalam bidang Gigi dan mulut memuat kondisi umum user, diagnosis penyakit, factor kepastian dan solusi .

Implementasi sistem pakar dalam bidang Gigi dan mulut sebagai pendukung pengambilan keputusan dengan metode *Backward Chaining* merupakan suatu program yang terdiri dari dua jenis program. Kedua jenis program tersebut adalah program konvensional dan program sistem pakar. Program konvensional digunakan untuk proses iterasi dan untuk mengolah basis data, sedangkan program sistem pakar digunakan dalam proses inferensial dan untuk mengolah basis pengetahuan. Adapun gambaran sistem secara konseptual sebagai berikut :

---



Gambar 3.8 Konseptual Sistem Pakar Gigi Dan Mulut

Keterangan :

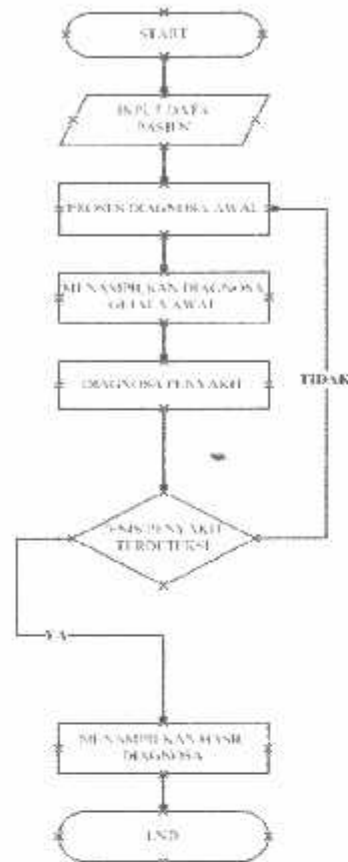
- Program Konvensional digunakan untuk proses iterasi dan untuk mengolah basis data .
- Program sistem pakar digunakan dalam proses inferensial dan untuk mengolah basis pengetahuan.
- Basis pengetahuan merupakan bentuk konfigurasi tertentu dari semua pengetahuan yang didapatkan selama proses akuisisi pengetahuan.
- Akuisisi pengetahuan diartikan yang menjembatani antara pakar dan basis pengetahuan dengan tujuan untuk mendapatkan pengetahuan dari pakar.
- Interface output terdiri dari pada informasi-informasi apa saja yang akan diberikan oleh system pakar meliputi diagnosa penyakit, beserta gambar dan solusinya.

### 3.5 Proses Penelusuran Sistem Pakar

Proses penelusuran pakar gigi dan mulut dimulai dengan input data pasien kemudian gejala awal yang dirasakan pasien setelah itu diagnosa gejala penyakit kemudian jenis penyakit terdeteksi jika "YA" Proses selanjutnya ke hasil diagnosa sebaliknya jika tidak looping (perulangan) ke gejala.



Berikut flowchart penelusuran untuk lebih jelasnya :



Gambar 3.9 Flowchart sistem pakar gigi dan mulut

### 3.6 Perancangan Struktur Basis Data (*data base*)

*Database* yang digunakan dalam aplikasi ini adalah *Xampp For Windows version 1.7.3* dan *connector* nya *MySQL ODBC 3.51* dengan pertimbangan Aplikasi tersebut lebih mudah didapat dan mudah diaplikasikan dalam pemrograman.

Perancangan struktur *database* tidak terlepas dari perancangan masukan (*input*) dan keluaran (*output*), perancangan *input* dan *output* akan berpengaruh besar terhadap kelengkapan informasi yang akan dibuat. Berikut adalah gambaran dari rancangan tabel yang akan dibuat untuk memenuhi kebutuhan *database* aplikasi yang akan dibuat nantinya.

#### 3.6.1 Perancangan Tabel

Adapun tabel yang digunakan adalah sebagai berikut :

### 1. Tabel Diagnosa

Tabel Diagnosa berisi tentang beberapa macam ID untuk lebih jelasnya berikut desain tabel tersebut :

Tabel 3.1 Diagnosa

<i>Field Name</i>	Jenis	<i>Length</i>
ID_DIAGNOSA	INT	11
ID_PENYAKIT	VARCHAR	5
ID_PASIEN	VARCHAR	5
TANGGAL	DATETIME	-
MD	DOUBLE	-
MB	DOUBLE	-
CF	DOUBLE	-

### 2. Tabel Gejala

Tabel Gejala digunakan untuk menampilkan gejala – gejala penyakit melalui id\_gejala berikut desain tabel nya.

Tabel 3.2 Gejala

<i>Field Name</i>	Jenis	<i>Length</i>
ID_GEJALA	VARCHAR	5
GEJALA	VARCHAR	255

### 3. Tabel Gejala Penyakit

Tabel Gejala Penyakit digunakan untuk menampung data-data ID Gejala penyakit.

Tabel 3.3 Gejala Penyakit

<i>Field Name</i>	<i>Data Type</i>	<i>Length</i>
ID_GP	INT	11
ID_GEJALA	VARCHAR	5
ID_PENYAKIT	VARCHAR	5

## 4. Tabel Pasien

Tabel Pasien digunakan untuk mencatat data-data pasien untuk diagnosa penyakit gigi dan mulut.

Tabel 3.4 Pasien

<i>Field Name</i>	<i>Data Type</i>	<i>Length</i>
ID_PASIEN	VARCHAR	5
NAMA	VARCHAR	100
ALAMAT	VARCHAR	255
UMUR	INT	11

## 5. Tabel Penyakit

Tabel Penyakit digunakan untuk menampung data-data penyakit gigi dan mulut.

Tabel 3.5 Penyakit

<i>Field Name</i>	<i>Data Type</i>	<i>Length</i>
ID_PENYAKIT	VARCHAR	5
NAMA	VARCHAR	100
KETERANGAN	VARCHAR	255
SOLUSI	TEXT	-

## 6. Tabel Rule

Tabel Rule digunakan untuk menampung data-data Rule penyakit .

Tabel 3.6 Rule

<i>Field Name</i>	<i>Jenis</i>	<i>Length</i>
ID_RULE	INT	11
ID_RULE SEBELUM	VARCHAR	5
ID_GP	VARCHAR	5
ID_RULE_YA	VARCHAR	5
ID RULE TIDAK	VARCHAR	5
MB	FLOAT	-
MD	FLOAT	-

## 7. Tabel User

Tabel User digunakan untuk menampung data-data akses User penyakit gigi dan mulut.

Pada tabel yang dibuat, tidak terdapat *primary key* untuk menghindari *error*, pada aplikasi ini nama-namanya banyak yang hampir mirip, jadi untuk mengatasinya *primary key* tidak dipergunakan karena jika yang dimasukkan datanya ada yang sama maka tidak akan bisa masuk ke *database*.

Tabel 3.7 User

<i>Field Name</i>	<i>Data Type</i>	<i>Length</i>
ID_USER	VARCHAR	10
PASS	VARCHAR	255
NAMA	VARCHAR	50
JABATAN	VARCHAR	20

## 3.7 Perancangan Form Aplikasi

Perancangan form aplikasi sistem pakar ini terdiri dari beberapa form yang saling terhubung dengan form lainnya, dibuat beberapa form karena tiap-tiap form mempunyai fungsi yang berbeda.

### 3.7.1 Form Menu Utama

Form Utama merupakan form awal saat kita membuka aplikasi, terdiri atas 2 button diantaranya button pasien untuk melanjutkan ke form diagnose pasien dan button pakar untuk melanjutkan ke form insert,update,delete data,gigi dan mulut pada aplikasi ini.

Gambar 3.10 Form Menu Utama

### 3.7.2 Form Registrasi Pasien

Pada form ini user akan disuguhkan fields untuk diisi, meliputi nama, alamat dan umur untuk ID sudah otomatis di tampilkan oleh aplikasi. Data yang dimasukkan nantinya akan diproses pada form report. Form ini merupakan form pertama yang berhubungan dengan form report. Berikut gambar untuk lebih jelasnya :

Gambar 3.11 Form Registrasi Pasien

### 3.7.3 Form Diagnosa Awal

Disini user disuguhkan form diagnosa awal yang terdiri dari berbagai macam pertanyaan dengan tujuan untuk mendapatkan hasil diagnosa awal penyakit untuk lebih jelasnya seperti gambar ini.

Gambar 3.12 Form Diagnosa Awal

### 3.7.4 Form Hasil Diagnosa Awal

Form ini merupakan lanjutan dari form diagnosa awal dimana sudah didapatkan hasil diagnosa awal suatu penyakit dan masih di proses ke form selanjutnya untuk lebih jelasnya lihat gambar dibawah :

New Next

**Hasil Diagnosa Awal, Penyakit Anda  
Abses Periapikal**

Gambar

Keterangan :

Gambar 3.13 Form Hasil Diagnosa Awal

### 3.7.5 Form Diagnosa Penyakit

Form ini prosesnya hampir sama dengan form diagnosa awal penyakit yang membedakan cuma dari segi pertanyaan dimana pertanyaan nya lebih mengacu ke jenis penyakit yang sudah ditemukan pada form sebelumnya.

New Next

Apakah Anda Mengalami :  
Mengunyah Timbul Nyeri

Jawab :

YA       TIDAK

SIPAGI

Gambar 3.14 Diagnosa Penyakit

### 3.7.6 Form Hasil Diagnosa Penyakit

Form ini merupakan hasil dari semua pertanyaan yang telah di inputkan user sebelumnya yang menghasilkan diagnosa penyakit beserta solusi dan faktor kepastian nya.

New

**Hasil Diagnosa Penyakit Anda**  
[CF : 60%]  
**Trenchmouth**

Gambar

Keterangan :

Solusi :

Gambar 3.15 Form Hasil Diagnosa Penyakit

### 3.7.7 Form Login User

Form ini untuk user sebagai pakar atau admin dimana pakar bisa mengakses semuanya misalnya insert,update,delete data klo admin cuma bisa akses data user.

Login User

ID

PASSWORD

Clear

**SIPAGI**

Gambar 3.16 Form Login User

### 3.7.8 Form Pakar

Form ini merupakan inti dari aplikasi pakar gigi dan mulut berfungsi menambah atau mau menghapus data penyakit login sebagai pakar aja yang bias mengaksesnya.

The screenshot shows a window titled "File Master Report About". Inside, there is a menu structure:

- Periyakit
  - Pakar — Rule Base
  - Pasien — Gejala
  - Penanyaan

At the bottom of the window, the text "SIPAGI" is displayed.

Gambar 3.17 Form Pakar

### 3.7.9 Form Penyakit

Form ini berfungsi untuk menambah data – data nama penyakit yang baru beserta keterangan dan solusinya

The screenshot shows a form titled "Pengolahan Data Penyakit". It has four input fields: ID, Nama, Keterangan, and Solusi. Below the form is a table with the following data:

ID	Nama	Keterangan
P01	Tweachmouth	
P02	Abes Periapikal	
P03	Bau Mulut	
P04	Gingivitis	
P05	Infeksi Herpes Pada Mulut	
P06	Parulis	
P07	Periodontitis	

Gambar 3.18 Form Penyakit

### 3.7.10 Form Rule Base

Form ini nantinya berfungsi untuk menampilkan berbagai macam jenis pertanyaan system pakar gigi dan mulut pada aplikasi ini dan juga untuk menentukan faktor kepastian nya..



Pengolahan Data Gejala Penyakit			
Penyakit	ID	Nama	Gejala
Pertanyaan sekarang			
Pertanyaan Sebelumnya			
Pertanyaan Jawaban Ya			
Pertanyaan Jawaban Tidak			
Measure of Belief (MB)	Measure of Belief (MB)		
<input type="button" value="Hapus"/> <input type="button" value="Edit"/> <input type="button" value="Clear"/>			

Gambar 3.19 Form Rule Base

### 3.7.11 Form Gejala

Form ini digunakan untuk memasukkan gejala – gejala penyakit yang ada pada aplikasi pakar untuk lebih jelasnya gambar dibawah ini.

Pengolahan Data Gejala	
ID	<input type="text" value="G21"/>
Gejala	<input type="text" value="Bau Mulut"/>
<input type="button" value="Hapus"/> <input type="button" value="Edit"/> <input type="button" value="Clear"/>	
ID	Gejala
G01	
G02	
G03	
G04	
G05	
G06	

Gambar 3.20 Form Gejala

### 3.7.12 Form Pertanyaan

Form ini digunakan untuk memilih berbagai macam gejala penyakit supaya nantinya bisa dibuat untuk menampilkan pertanyaan pada aplikasi sistem pakar gigi dan mulut

Pengolahan Data Gejala Penyakit			
Penyakit	ID	Nama	Keterangan
	P01	Trenchmouth	.....
Gejala	P02	Abses Periapikal	.....
<input type="checkbox"/> G07	P03	Halitosis	.....
<input type="checkbox"/> G08	P04	Gingivitis	.....
<input type="checkbox"/> G09	P05	Infeksi Herpes	.....
<input type="checkbox"/> G10	P06	Pulpitis	.....
<input type="checkbox"/> G11	P07	Periodontitis	.....
<input type="checkbox"/> G12			
<input type="checkbox"/> G13			
<input type="checkbox"/> G14			
<input type="checkbox"/> G15			
<input type="checkbox"/> G16			
<input type="checkbox"/> G17			
<input type="checkbox"/> G18			
<input type="checkbox"/> G19			
<input type="checkbox"/> G20			
<input type="checkbox"/> G21			
<input type="checkbox"/> G22			
<input type="checkbox"/> G23			
<input type="checkbox"/> G24			
<input type="checkbox"/> G25			
<input type="checkbox"/> G26			
<input type="checkbox"/> G27			
<input type="checkbox"/> G28			
<input type="checkbox"/> G29			

Gambar 3.21 Form Pertanyaan

### 3.8 Struktur Basis Pengetahuan (*Knowledge Base*)

Basis pengetahuan terdiri dari kumpulan data atau fakta-fakta tertentu yang dibutuhkan untuk menyelesaikan permasalahan yang spesifik. Dengan membentuk basis pengetahuan berarti memasukkan fakta-fakta yang dibutuhkan oleh sistem. Basis pengetahuan yang digunakan dalam sistem ini adalah tentang aturan nama penyakit dan gejala penyakit. Representasi penyakit dan gejala penyakit dinyatakan pada tabel 3.8 dibawah ini.

Tabel 3.8 Pengolahan Data Penyakit Gigi dan Mulut

NO	ID	Nama Penyakit	Keterangan	Solusi
1	P01	Trench Mouth	Trench Mouth (Infeksi Vincent, Gingivitis ulserativa nekrotikan akut) adalah suatu infeksi gusi yang tidak menular dan terasa nyeri, menyebabkan nyeri, demam, dan kelelahan.	Pengobatan dilakukan dengan pembersihan, dimana semua jaringan gusi yang mati dan karang gigi dibuang. Karena pembersihan ini menimbulkan nyeri, maka digunakan obat bius lokal. Beberapa hari pertama setelah pembersihan, penderita

				<p>diharuskan berkumur-kumur dengan larutan hidrogen peroksida (setengah bagian hidrogen peroksida 3% dicampur dengan setengah bagian air), beberapa kali dalam sehari. Selama 2 minggu, penderita mengunjungi dokter gigi setiap 1-2 hari. Pembersihan berlanjut sampai terjadinya penyembuhan. Jika bentuk dan posisi gusi tidak kembali normal, dokter gigi akan melakukan pembedahan untuk kembali membentuk gusi sebagai pencegahan terhadap kekembuhan dan pencegahan terhadap periodontitis. Antibiotik diberikan jika trench mouth sangat berat atau tidak dapat dilakukan perawatan gigi.</p>
2	P02	Abses Periapikal	Abses Periapikal adalah pengumpulan nanah	Abses atau selulitis diatasi dengan

			yang telah menyebar dari sebuah gigi ke jaringan di sekitarnya, biasanya bersal dari suatu infeksi.	menghilangkan infeksi dan membuang nanah melalui bedah mulut atau pengobatan saluran akar. Untuk membantu menghilangkan infeksi seringkali diberikan antibiotik. Tindakan yang terpenting adalah mencabut pulpa yang terkena dan mengeluarkan nanahnya.
3	P03	Bau Mulut	Halitosis (Bau Mulut) adalah bau nafas yang tidak enak, tidak menyenangkan dan menusuk hidung.	Daun parsley segar atau permen mint bisa menghasilkan bau mulut yang bersifat sementara.
4	P04	Gingivitis	Gingivitis adalah peradangan pada gusi (gingiva). Gingivitis sering terjadi dan bisa timbul kapan saja setelah tumbuhnya gigi.	Kondisi media yang menyebabkan atau memperburuk gingivitis harus diatasi. Jika penyebabnya adalah obat-obatan, maka pertumbuhan gusi yang berlebihan harus diangkat melalui pembedahan.
5	P05	Infeksi Herpes pada mulut	Infeksi Herpes Mulut Primer (Gingivostomatitis Herpetik Primer, Herpes	Tujuan pengobatan pada herpes primer adalah untuk mengurangi rasa sakit,

			<p>Labialis) adalah suatu infeksi awal oleh virus herpes simpleks yang dengan segera bisa menyebabkan terbentuknya luka yang terasa nyeri digusi dan bagian mulut lainnya.</p>	<p>sehingga penderita bisa tidur, makan dan minum secara normal. Rasa nyeri bisa menyebabkan anak tidak mau makan dan tidak mau minum, bila disertai demam, hal ini bisa dengan segera menyebabkan dehidrasi (kekurangan cairan tubuh). Karena itu anak yang sakit harus minum cairan sebanyak mungkin. Untuk mengurangi nyeri pada penderita dewasa atau anak yang lebih besar, bisa digunakan obat kumur anestetik (misalnya lidokain) atau bisa juga digunakan obat kumur yang mengandung baking soda.</p>
6	P06	Pulpitis	<p>Pulipitis adalah peradangan pada pulpa gigi yang menimbulkan rasa nyeri.</p>	<p>Peradangan mereda jika penyebabnya diobati. Jika pulpitis diketahui pada stadium dini, maka penambalan sementara yang mengandung obat penenang saraf bisa</p>

				<p>menghilangkan nyeri. Tambalan ini bisa dibiarkan sampai 6-8 minggu dan kemudian diganti dengan tambalan permanen. Jika terjadi kerusakan pulpa yang luas dan tidak dapat diperbaiki, satu-satunya cara untuk menghilangkan nyeri adalah dengan mencabut pulpa, baik melalui pengobatan saluran akar maupun dengan pencabutan gigi.</p>
7	P07	Periodontitis	<p>Periodontitis (piore) terjadi jika gingivitis menyebar ke struktur penyangga gigi.</p>	<p>Seorang dokter gigi bisa membersihkan kantong sampai kedalaman 0,5 cm dengan alat khusus, yang dapat membuang seluruh karang gigi dan permukaan akar gigi yang sakit. Untuk kantong yang dalamnya mencapai 0,6 cm atau lebih, seringkali diperlukan pembedahan. Seorang dokter gigi juga dapat</p>

				<p>mengangkat sebagian gusi yang terpisah sehingga gusi yang tertinggal bisa direkatkan lagi dengan lebih erat ke gigi dan penderita bisa membersihkan plaknya di rumah. Jika terbentuk abses, diberikan antibiotik. Ke dalam kantong yang dalam bisa dimasukkan filamen yang mengandung antibiotik sehingga obat bisa mencapai daerah yang sakit dalam konsentrasi yang tinggi. Abses Periodontal menyebabkan serangan pengrusakan tulang, tetapi pengobatan segera dengan pembedahan dan antibiotik memungkinkan tulang yang rusak untuk tumbuh kembali. Jika setelah pembedahan timbul luka terbuka di mulut, diberikan obat</p>
--	--	--	--	---

---

				kumur klorheksidin selama 1 menit, 2 kali/hari untuk sementara waktu, menggantikan gosok gigi dan pemakaian benang gigi.
--	--	--	--	--

Tabel 3.9 Pengurutan Data Gejala

No	Kode Gejala	Gejala
1	G01	Nyeri gusi
2	G02	Bau mulut yang tidak enak
3	G03	Gusi mudah berdarah
4	G04	Mengunyah dan menelan terasa nyeri
5	G05	Timbul demam ringan
6	G06	Mengunyah timbul nyeri
7	G07	Kemungkinan demam
8	G08	Terjadi pembengkakan di daerah rahang
9	G09	Bau nafas tercium tidak enak
10	G10	Gusi membengkak
11	G11	Gusi seringkali berdarah
12	G12	Gusi dan bagian mulut lainnya terasa nyeri
13	G13	Terdapat banyak luka terbuka yang berwarna putih atau kuning didalam mulut
14	G14	Peradangan Gusi Biasa Dan Sakit Mulut Yang Luar Biasa
15	G15	Terjadi Demam
16	G16	Tidak Enak Badan
17	G17	Sakit Gigi Yang Luar Biasa
18	G18	Pulpa Gigi Terasa Nyeri
19	G19	Perdarahan Gusi
20	G20	Perubahan Warna Gusi
21	G21	Bau Mulut



22	G00	Selamat datang, Next untuk mulai proses..
23	G99	Proses Selesai, Next untuk melihat hasil.

---

## BAB IV IMPLEMENTASI DAN PENGUJIAN SISTEM

### 4.1 Implementasi Sistem

Tahap implementasi merupakan tahap akhir dari proses membangun sebuah sistem. Pada tahap ini semua rancangan yang sudah dihasilkan diterjemahkan dengan menggunakan bahasa pemrograman sehingga menjadi sebuah sistem.

Pada implementasi program aplikasi diagnosa penyakit gigi dan mulut ini memiliki tampilan sebagai berikut :

#### 4.1.1 Form Menu Utama

Pada jendela utama ini terdapat 2 button menu utama, yaitu : button pasien dan button pakar. Berikut gambarnya :



Gambar 4.1 Form Menu Utama

#### 4.1.2 Form Registrasi Pasien

Pada halaman Registrasi Pasien, sebelum proses diagnosa user akan disuguhkan fields-fields nama, alamat, umur yang nantinya akan diproses ke form report, untuk membatalkan data yang sudah di inputkan tekan button clear berikut gambaran lebih jelasnya :



Registrasi Pasien

Registrasi Data Medis (Pasien)    Diagnosa    Clear

ID: U028  
 Nama: HASHRUL HAQ  
 Alamat: MALANG  
 Umur: 25

**SIPAGI**  
 Aplikasi Sistem Pakar untuk Diagnosa Penyakit Gigi dan Mulut

Gambar 4.2 Form Registrasi Pasien

#### 4.1.3 Form Diagnosa Awal

Disini user disuguhkan form diagnosa awal yang terdiri dari berbagai macam pertanyaan dengan tujuan untuk mendapatkan hasil diagnosa awal penyakit, ada 2 button, button new untuk kembali ke pertanyaan baru sedangkan next untuk ke pertanyaan selanjutnya dan jawaban ya atau tidak berfungsi untuk mencari goal driven backward chaining atau jenis penyakit yang terdeteksi lebih spesifik.



Proses Diagnosa Awal

New    Next

**Apakah Anda Mengalami:**  
 Mengunyah timbul nyeri

Jawab:  
 Ya     Tidak

**SIPAGI**  
 Aplikasi Sistem Pakar untuk Diagnosa Penyakit Gigi dan Mulut

Gambar 4.3 Form Diagnosa Awal

#### 4.1.4 Form Hasil Diagnosa Awal

Form ini merupakan lanjutan dari form diagnosa awal dimana sudah didapatkan hasil diagnosa awal suatu penyakit dan masih di proses ke form selanjutnya dengan

mengklik button next untuk ke pertanyaan selanjutnya, lebih jelasnya lihat gambar dibawah :



Gambar 4.4 Form Hasil Diagnosa Awal

#### 4.1.5 Form Diagnosa Penyakit

Form ini prosesnya hampir sama dengan form diagnosa awal penyakit begitu juga dengan menu atau button yang ada pada form ini yang membedakan cuma dari segi pertanyaan dimana pertanyaan nya lebih mengacu ke jenis penyakit yang sudah ditemukan pada form sebelumnya.



Gambar 4.5 Form Diagnosa Penyakit

#### 4.1.6 Form Hasil Diagnosa Penyakit

Form ini merupakan hasil dari semua pertanyaan yang telah di inputkan user sebelumnya dimana outputnya telah menghasilkan faktor keyakinan nya 25% nama penyakit beserta gambar dan solusinya.



Gambar 4.6 Form Hasil Diagnosa Penyakit

#### 4.1.7 Form Login User

Pada Form ini user disuguhkan fields ID dan Password yang berfungsi untuk pengolahan data pada aplikasi pakar gigi dan mulut, terdiri dari 2 macam login : 1. Login sebagai User, 2. Login sebagai Pakar pada analisa aplikasi ini di contohkan login sebagai pakar.



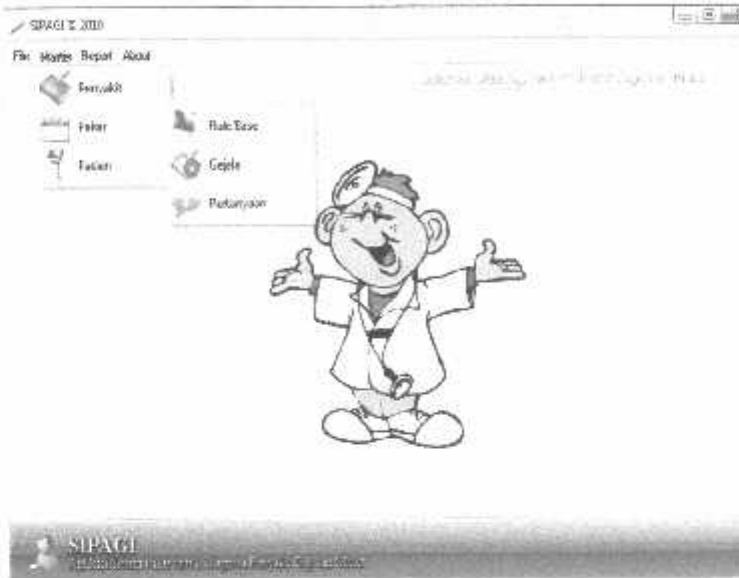
Gambar 4.7 Form Login User

#### 4.1.8 Form Pakar

Form Pakar ini terdiri dari berbagai menu misalnya file->logout->keluar. Untuk menu master terdiri Penyakit->Pakar->Pasien.

Klo isi dari Pakar sendiri ada 3 item 1.Rule Base->2.Gejala

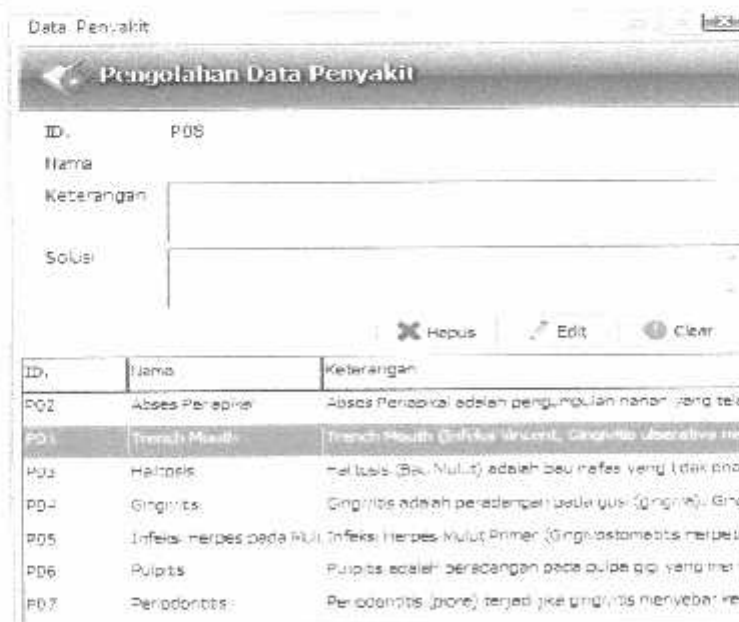
3.->Pertanyaan, untuk lebih jelasnya seperti gambar yang ada dibawah ini :



Gambar 4.8 Form Pakar

#### 4.1.9 Form Penyakit

Form ini yang nantinya digunakan untuk menghapus atau menambah data jenis penyakit baru yang belum ada pada aplikasi ini, terdiri dari beberapa fields seperti nama-keterangan dan solusi serta button hapus, edit, clear.



Gambar 4.9 Form Penyakit

#### 4.1.10 Form Rule Base

Form ini nantinya berfungsi untuk menampilkan berbagai macam jenis pertanyaan system pakar gigi dan mulut pada aplikasi ini dan juga untuk menentukan faktor kepastian nya

The screenshot shows a window titled 'Data Rule Patar' with a sub-header 'Pengolahan Data Gejala Penyakit'. It contains a table with three columns: 'Penyakit', 'ID', 'Nama', and 'Gejala'. The table lists various symptoms and their corresponding disease IDs and names.

Penyakit	ID	Nama	Gejala
102   Abses Periapikal	101	Trench Mouth	Selamat datang, he-
Pertanyaan Sekarang	102	Trench Mouth	Nyeri gusi
192   Proses Selesai, Next untuk melihat hasil.	103	Trench Mouth	Bau mulut yang total
Pertanyaan Sebelumnya	104	Trench Mouth	Gusi mudah berdarah
192   Proses Selesai, Next untuk melihat hasil.	105	Trench Mouth	Mengunyah dan men-
Pertanyaan Jawaban Ya	106	Trench Mouth	Timbul demam ringan
192   Proses Selesai, Next untuk melihat hasil.	107	Abses Periapikal	Selamat datang, he-
Pertanyaan Jawaban Tidak	108	Abses Periapikal	Mengunyah timbul nyeri
192   Proses Selesai, Next untuk melihat hasil.	109	Abses Periapikal	Kemungkinan demam
Measure of Belief (MB) Measure of Disbelief (MD)	110	Abses Periapikal	Terjadi pembengkakan
	111	Halitosis	Selamat datang, he-

Gambar 4.10 Form Rule Base

#### 4.1.11 Form Gejala

Form ini digunakan untuk memasukkan gejala – gejala penyakit yang ada pada aplikasi pakar untuk lebih jelasnya gambar dibawah ini.

The screenshot shows a window titled 'Data Gejala' with a sub-header 'Pengolahan Data Gejala'. It contains a form with two input fields: 'ID' (containing 'G21') and 'Gejala' (containing 'Bau mulut'). Below the form is a table listing symptoms with their IDs.

ID	Gejala
G01	Nyeri gusi
G02	Bau mulut yang tidak enak
G03	Gusi mudah berdarah
G04	Mengunyah dan menelan terasa nyeri
G05	Timbul demam ringan
G06	Mengunyah timbul nyeri
G07	Kemungkinan demam
G08	Terjadi pembengkakan di daerah rahang
G09	Bau nafas terdium tidak enak

Gambar 4.11 Form Gejala

#### 4.1.12 Form Pertanyaan

Form ini digunakan untuk memilih berbagai macam gejala penyakit supaya nantinya bisa dibuat untuk menampilkan pertanyaan pada aplikasi sistem pakar gigi dan mulut.

The screenshot shows a window titled "Pengolahan Data Gejala Penyakit". On the left, there is a list of symptoms (Gejala) with checkboxes, including "Demam", "Terjadi pembengkakan di daerah rahang", "Bau nafas berbau tidak sedap", "Gusi memblek", "Gusi berdarah", "Gusi dan bagian mulut lain nya terasa nyeri", "Terdapat banyak luka terbuka yang berwarna putih atau", "Peradangan gusi biasa dan sakit mulut yang luar biasa", "Terjadi demam", "Tidak enak badan", "Sakit gigi yang luar biasa", "Pulpa gigi terasa nyeri", "Perdarahan gusi", "Perubahan warna gusi", "Bau mulut", "Selamat datang, Next untuk mulai proses.", and "Proses Selesai, Next untuk melihat hasil.". On the right, there is a table with columns "ID", "Nama", and "Keterangan".

ID	Nama	Keterangan
P01	Alnus Periapikal	Alnus Periapikal adalah
P01	Trench Mouth	Trench Mouth (Infeksi Vi
P02	Halitosis	Halitosis (Bau Tidak) ada
P04	Gingivitis	Gingivitis adalah peradai
P05	Infeksi Herpes pada	Infeksi Herpes (Herpes) ada
P06	Palpitasi	Palpitasi adalah peradang
P07	Periodontitis	Periodontitis (gusi) ada

Gambar 4.12 Form Pertanyaan

## 4.2 Pengujian Program

Tahap terakhir dalam perancangan sistem adalah pengujian sistem. Pengujian sistem ini dilakukan untuk menguji dan mengetahui apakah sistem berjalan dengan baik dan benar sesuai dengan basis pengetahuan sistem pakar. Sistem ini menggunakan jenis pengujian :

### 4.2.1 Alpha Test

*Alpha test* yaitu pengujian program dengan cara mengundang beberapa *user* untuk menjalankan program tersebut, kemudian masing-masing *user* akan diberikan *form quisioner* untuk memberi penilaian terhadap program.

Tabel 4.1 Pertanyaan *Alpha Test*

NO	NAMA	JABATAN	KRITERIA		
			A	B	C
1	Drg.Yuliana ratna kumala.SPKG	Kepala klinik dokter gigi Tlogomas	4	3	-
2	Azwari Irwansyah	Alumni Teknik Sipil Unisma	3	3	-



3	Ummi Syarofah	Mahasiswi Fk Fakultas Kedokteran Unisma	4	4	-
4	Deni Nanang Arif	Pemilik Warkop AGP	3	4	-
5	Rahman Aliansyah	Pegawai Warkop AGP	3	3	-
	Rata-Rata	(+)/5	3,4	3,4	-

Evaluasi terhadap aplikasi program ini dilakukan pada lima (5) responden, satu (1) Dokter gigi tlogomas malang, dua (2) alumni teknik sipil dan mahasiswi fakultas kedokteran unisma malang, dua (2) Pegawai warung kopi AGP dinoyo malang.

**a. Keterangan Kriteria :**

- o Kriteria A : Kemudahan penggunaan program
- o Kriteria B : Desain tampilan program
- o Kriteria C :-

**b. Keterangan Skala Penilaian :**

- Nilai 1 : Sangat Kurang
- Nilai 2 : Kurang
- Nilai 3 : Cukup
- Nilai 4 : Baik
- Nilai 5 : Sangat Baik

**c. Penilaian secara keseluruhan terhadap kelayakan program :**

a) Kemudahan penggunaan program :

$$(4+4+4+4+4)/25 * 100\% = 80\%$$

b) Desain tampilan program :

$$(3+5+4+4+4)/25 * 100\% = 80\%$$

Dari hasil penilaian terhadap sistem, maka dapat disimpulkan bahwa sistem layak dipergunakan untuk mendiagnosa penyakit gigi dan mulut.

## BAB V

### KESIMPULAN DAN SARAN

#### 5.1 Kesimpulan

Dari perancangan dan implementasi Aplikasi Sistem Pakar Untuk Diagnosa Penyakit Gigi Dan Mulut yang telah dilakukan, dapat diambil kesimpulan sebagai berikut :

Kesimpulan yang dapat diambil berdasarkan hasil implementasi jaringan saraf tiruan antara lain :

1. Sistem pakar yang dibuat merupakan sistem yang berdasarkan pada aturan –aturan dimana program disimpan dalam bentuk aturan-aturan sebagai prosedur pemecahan masalah. Aturan tersebut biasanya berbentuk IF –THEN
2. Berdasarkan proses diagnosa yang telah dilakukan user sebelumnya sistem pakar mendiagnosis penyakit gigi dan mulut dapat mengenali pola – pola masukan dan menghasilkan keluaran berupa hasil diagnose penyakit beserta gambar dan solusinya yang sesuai pada saat proses diagnosa.
3. Berdasarkan pengujian yang dilakukan dengan *alpha test*, maka aplikasi sudah bisa digunakan untuk mendeteksi jenis penyakit gigi dan mulut.

#### 5.2 Saran

Aplikasi ini masih jauh dari sempurna dan masih banyak kekurangan. Oleh karena itu perlu dilakukan pengembangan dan penyempurnaan lebih lanjut. Adapun saran yang dapat dikemukakan agar aplikasi ini bisa berfungsi dengan lebih optimal adalah:

1. Aplikasi sistem pakar gigi dan mulut ini berbasis dekstop ditujukan ke dokter untuk kedepannya diharapkan berbasis web agar mudah di akses masyarakat umum khususnya orang awam.
2. *User* diharapkan membaca dengan teliti keluhan apa saja yang pernah dialami oleh *user*, dari berbagai macam pertanyaan yang telah disediakan pada form diagnosa, agar mendapatkan hasil diagnosa penyakit yang optimal beserta faktor kepastian nya.

3. Untuk daftar jenis-jenis penyakit, diharapkan bisa dikembangkan ke yang lebih banyak lagi dengan syarat sudah diketahui oleh penulis dan rekomendasi dari Dokter Gigi dan Mulut bagaimana mengatasi penyakit dan solusinya.
4. Agar aplikasi dapat berjalan dengan optimal, sebaiknya control panel xampp dari aplikasi ini diaktifkan dahulu.
5. Untuk tampilan pada tiap form diharapkan bisa lebih bagus lagi dengan efek-efek grafis yang lebih bagus lagi, tentunya dengan mencari komponen-komponen yang berhubungan dengan pengaturan tampilan pada program yang dipakai untuk membuat aplikasi ini yaitu *Borland Delphi 7*.



## LAMPIRAN CODING

### CODING FORM REGISTRASI PASIEN

```

unit Unit_reg;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, DB, ADODB, ImgList, acAlphaImageList, StdCtrls, Buttons,
  sBitBtn, ExtCtrls, sLabel, sPanel, Spin, sEdit, sSkinManager,
  sSkinProvider;
type
  TFormReg = class(TForm)
    sPanel1: TsPanel;
    sLabelFX1: TsLabelFX;
    sLabelFX2: TsLabelFX;
    Image1: TImage;
    sPanel2: TsPanel;
    btnDiag: TsBitBtn;
    sAlphaImageList1: TsAlphaImageList;
    con: TADOQuery;
    sLabelFX3: TsLabelFX;
    Nama: TsEdit;
    sLabel4: TsLabel;
    sLabel1: TsLabel;
    alamat: TsEdit;
    sLabel2: TsLabel;
    umur: TSpinEdit;
    sLabel3: TsLabel;
    IdUser: TsEdit;
    BtnClear: TsBitBtn;
    sSkinProvider1: TsSkinProvider;
    sSkinManager1: TsSkinManager;
    procedure IdUserChange(Sender: TObject);
    procedure btnDiagClick(Sender: TObject);
    procedure BtnClearClick(Sender: TObject);
    procedure FormClose(Sender: TObject; var Action: TCloseAction);
    procedure FormCreate(Sender: TObject);
  private
    procedure getCode();
  end;

```

---

```

public
  kodePasien :string;{ Public declarations }
end;
var
  FormReg: TFormReg;
implementation
uses Unit_diagnosa , Unit_Main;
{$R *.dfm}
procedure TFormReg.gctCode();
var
  Kodes : String;
  i : Integer;
  found : boolean;
begin
  i := 1;
  Kodes := FormatCurr('U000',1);
  IdUser.Text := Kodes;
  found := false;
  While Not found do
  begin
    Kodes := FormatCurr('U000',i);
    con.close;
    con.sql.clear;
    con.sql.add('SELECT * FROM 'tb_pasien' WHERE id_pasien="'+kodes+' ');
    con.Prepared;
    con.open;
    if con.recordcount >0 then
      found := false
    Else
      begin
        iduser.Text := Kodes;
        found := True;
      end;
    i := i + 1;
  end;

end;

end;
procedure TFormReg.IdUserChange(Sender: TObject);
begin

```

---

```

If (trim(nama.Text) <> "") and (trim(alamat.Text) <> "") then
    btnDiag.Visible := true
else
    btnDiag.Visible := false;
end;

procedure TFormReg.btnDiagClick(Sender: TObject);
begin
    if (MessageDlg('Yakin Untuk Simpan data?', mtConfirmation, [mbYes, mbNo], 0)) = mrYes then
        begin
            try
                con.close;
                con.sql.clear;
                con.sql.add('INSERT INTO tb_pasien '+
                    '(id_pasien, nama, alamat, umur) '+
                    'VALUES (''+trim(IdUser.Text)+'',''+trim>Nama.Text)+'
                    '''+trim(alamat.Text)+'',''+trim(IntToStr(umur.Value)+'')');
                con.ExecSQL;
                // MessageDlg('Proses Penyimpanan Berhasil', mtInformation, [ mbOK], 0 );
                kodePasien := IdUser.Text;
                if (Formmain.fdiag = nil) then
                    Formmain.fdiag := TFormDiagnosa.Create(Application);
                Formmain.fdiag.Show;
                Formmain.freg.top := -2000;
                // Formmain.freg.WindowState := wsMinimized;
                BtnClearClick(self);
            except
                MessageDlg('Proses Penyimpanan Gagal', mtError, [ mbOK], 0 );
            end;
        end;
    else
        begin
            BtnClearClick(self);
        end;
    end;
end;

procedure TFormReg.BtnClearClick(Sender: TObject);
begin
    IdUser.Text := '';

```

---

```

    btnDiag.Visible := False;
    getCode;
    nama.Text := "";
    alamat.Text := "";
end;
procedure TFormReg.FormClose(Sender: TObject; var Action: TCloseAction);
begin
    Formmain.freg := nil;
    FormMain.mode := 0;
    Action := caFree;
end;
procedure TFormReg.FormCreate(Sender: TObject);
begin
    BtnClearClick(sender);
end;
end.

```

### ***CODING FORM UJI (HASIL DIAGNOSA)***

```

unit Unit_uji;
interface
uses
    Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
    Dialogs, DB, ADODB, ImgList, acAlphaImageList, StdCtrls, sLabel, Buttons,
    sBitBtn, ExtCtrls, sPanel, Math, jpeg;
type
    TForm_Uji = class(TForm)
        sPanel1: TsPanel;
        sLabelFX1: TsLabelFX;
        sLabelFX2: TsLabelFX;
        Image1: TImage;
        sPanel2: TsPanel;
        BtnNext: TsBitBtn;
        btnNew: TsBitBtn;
        Panel1: TPanel;
        lb2: TsLabel;
        lb1: TsLabel;
        tanya: TsLabel;
        jawabya: TRadioButton;
        jawabtidak: TRadioButton;
        pnlHasil: TPanel;
    end;

```

---



```

lbKetsakit: TslLabel;
lbSakit: TsLabelFX;
sAlphaImageList1: TsAlphaImageList;
con: TADOQuery;
lbket: TMemo;
Imagehasil: TImage;
Image2: TImage;
procedure jawabyaClick(Sender: TObject);
procedure btnNewClick(Sender: TObject);
procedure BtnNextClick(Sender: TObject);
procedure FormShow(Sender: TObject);
procedure FormClose(Sender: TObject; var Action: TCloseAction);
procedure FormCloseQuery(Sender: TObject; var CanClose: Boolean);
procedure FormCreate(Sender: TObject);
private
    currInde : integer;
    currGp   : String;
    maxInde  : integer;
    curRule  : string;
    countTanya : integer;
    procedure getData(kode :String);
public
    kodePenyakit :string;
end;
var
    Form_Uji: TForm_Uji;
    MD, MB, CF,tMD, tMB, tCF : Double;
implementation
uses Unit_diagnosa, Unit_Main;
{$R *.dfm}
procedure TForm_Uji.jawabyaClick(Sender: TObjcet);
begin
    if jawabya.Checked <> jawabtidak.Checked then
        BtnNext.Visible := true;
end;
procedure TForm_Uji.getData(kode:String);
var i:integer;
begin
    con.close;
    con.sql.clear;

```

---

```

con.sql.add('SELECT MD,MB,id_rule, tb_gejala.gejala gejala,tb_gejala_penyakit.id_gp idgp ' +
' FROM tb_rule, tb_gejala_penyakit, tb_gejala, ' +
' tb_gejala_penyakitI tb_gejala_penyakitI ' +
' WHERE tb_rule.id_gp = tb_gejala_penyakitI.id_gp AND ' +
' tb_gejala_penyakitI.id_gejala = tb_gejala.id_gejala AND ' +
' tb_rule.id_rule_sebelum = tb_gejala_penyakitI.id_gp AND ' +
' (tb_gejala_penyakitI.id_penyakit = '"+trim(kode)+"') AND ' +
' (tb_gejala_penyakitI.id_gejala <> "G00") AND ' +
' (tb_gejala_penyakitI.id_gejala = "G00");
con.Prepared;
con.open;
if con.recordcount >0 then
begin
    for i := 1 to con.RecordCount do
    begin
        begin
            tanya.Caption:= con.fieldByName('gejalas').AsString;
            curRule := con.fieldByName('id_rule').AsString;
            currGp := con.fieldByName('idgp').AsString;
            con.Next;
        end;
    end
end;
end;
procedure TForm_Uji.btnNewClick(Sender: TObject);
begin
    close;
end;
procedure TForm_Uji.BtnNextClick(Sender: TObject);
var i,tempIdx:integer;
    tempIdY,tempIdSblm : string;
    tempMax,CF : Double;
begin
    if (count(Tanya>maxInde) then
    begin
        CF := MB- MD/(1-min( MB, MD));
        con.close;
        con.sql.clear;
        con.sql.add('SELECT MB, MD, ' +
' nama, keterangan,solusi ' +

```

---

```

' FROM tb_rule, tb_gejala_penyakit, tb_penyakit ' +
' WHERE tb_rule.id_rule_ya = tb_gejala_penyakit.id_gp AND ' +
'     tb_gejala_penyakit.id_penyakit = tb_penyakit.id_penyakit and (tb_penyakit.id_penyakit =
    "+kodePenyakit+" ) );
con.Prepared;
con.open;
if con.recordcount >0 then
begin
    for i := 1 to con.RecordCount do
        begin
            tMD := tMd + con.fieldByName('MD').AsFloat;
            tMB := tMB + con.fieldByName('MB').AsFloat;
            con.Next;
        end;
        tCF := tCF + (tMB - tMD/(1-min( tMB, tMD)));

        lbketsakit.Caption := 'Hasil Diagnosa Penyakit Anda ' + #13#10+ ' [CF: ' FloatToStr( ((CF/tCF) *
            100) )+'% ]';
        lbSakit.Caption := con.fieldByName('nama').AsString;
        lbket.Text := 'Keterangan : ' + #13#10+con.fieldByName('keterangan').AsString + #13#10+#13#10+
            'Solusi : ' + #13#10+con.fieldByName('solusi').AsString;
        Imagehasil.Picture.LoadFromFile(ExtractFilePath(Application.ExeName)
            +'images\'+ kodePenyakit+'.jpg' );
        BtnNext.Hide;
    end
else
begin
    lbSakit.Caption := 'Tidak Sakit';
    lbket.Text := 'Lakukan diagnosa ulang untuk memastikan';
    Imagehasil.Picture.LoadFromFile(ExtractFilePath(Application.ExeName)
        +'images\2.jpg' );
end;
con.close;
con.sql.clear;
con.sql.add('INSERT INTO tb_diagnosa ' +
    '(id_penyakit, id_pasien, tanggal, MD, MB, CF) ' +
    ' VALUES ("'+kodePenyakit+'","'+formmain.freg_kodePasien+'",current_date,"'+FloatToStr(MD)+'
    ","'+FloatToStr(MB)+'","'+FloatToStr(CF)+'")');
con.ExecSQL;
pnlHasil.Show;

```

```

con.close;
con.sql.clear;
con.sql.add('SELECT id_rule_ya, id_rule,MD,MB, '+
' gejala, tb_rule.id_gp AS idp ' +
' FROM tb_rule, tb_gejala_penyakit, tb_gejala ' +
' WHERE tb_rule.id_rule_ya = tb_gejala_penyakit.id_gp AND ' +
' tb_gejala_penyakit.id_gejala = tb_gejala.id_gejala AND ' +
' (tb_rule.id_rule = '"+curRule+"'");
con.Prepared;
con.open;
if con.recordcount >0 then
begin
for i := 1 to con.RecordCount do
begin
begin
tanya.Caption := con.fieldByName('gejala').AsString;
tempIdY := con.fieldByName('id_rule_ya').AsString;
currGp := con.fieldByName('idp').AsString;
con.Next;
end;
end
end;
con.close;
con.sql.clear;
con.sql.add('SELECT id_rule, id_rule_sebelum ,MD,MB FROM tb_rule ' +
' WHERE (id_gp = '"+tempIdY+"' ) and (id_rule_sebelum = '"+currGp+"' ) );
con.Prepared;
con.open;
if con.recordcount >0 then
begin
MD:= MD + con.fieldByName('MD').AsFloat;
curRule := con.fieldByName('id_rule').AsString;
end;
end;
Inc(countTanya);
jawabya.Checked := false;
jawabtidak.Checked := false;
BtnNext.Hide;
end;
procedure TForm_Uji.FormShow(Sender: TObject);

```

---

```

begin
    pnlHasil.Hide;
    jawabya.Checked := false;
    jawabtidak.Checked := false;
    MD :=0;
    MB :=0;
    CF :=0;
    con.close;
    con.sql.clear;
    con.sql.add('SELECT tb_gejala_penyakit.id_penyakit as idp,MD,MB '+
        ' FROM tb_rule, tb_gejala_penyakit '+
        ' WHERE tb_rule.id_gp = tb_gejala_penyakit.id_gp AND ' +
        ' (tb_gejala_penyakit.id_gejala <> "G00") and ' +
        ' tb_gejala_penyakit.id_penyakit="" + kodePenyakit+""');
    con.Prepared;
    con.open;
    if con.recordcount >0 then
        begin
            currInde := 0;
            maxInde := con.RecordCount-1;
            countTanya := 1;
            end;
        getData( kodePenyakit);
    end;
    procedure TForm_Uji.FormClose(Sender: TObject; var Action: TCloseAction);
    begin
        FormMain.fuji.FormShow(self);
        FormMain.fdiag.WindowState := wsNormal;
        FormMain.fdiag.btnNewClick(sender);
        // FormMain.fuji := nil;
        // Action := caFree;
    end;

    procedure TForm_Uji.FormCloseQuery(Sender: TObject; var CanClose: Boolean);
    begin
        // if (MessageDlg('Yakin Untuk Mengakhiri Proses?',mtConfirmation,[mbYes,mbNo],0)) = mrNo then
        // CanClose := False;
    end;
    procedure TForm_Uji.FormCreate(Sender: TObject);
    begin

```

---

```

    tCF := 0;
    tMB := 0;
    tMD := 0;
end;

end.

```

### ***CODING FORM LOGIN***

```

unit Unit_login;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, sSkinManager, sSkinProvider, DB, ADODB, ImgList,
  acAlphaImageList, StdCtrls, sEdit, Buttons, sBitBtn, ExtCtrls, sLabel,
  sPanel;
type
  TFormLogin = class(TForm)
    sLabel4: TLabel;
    sLabel3: TLabel;
    sPanel1: TPanel;
    sLabelFX1: TLabelFX;
    sLabelFX2: TLabelFX;
    Image1: TImage;
    sPanel2: TPanel;
    sLabelFX3: TLabelFX;
    pass: TEdit;
    id: TEdit;
    sAlphaImageList1: TAlphaImageList;
    con: TADOQuery;
    sSkinProvider1: TSkinProvider;
    sSkinManager1: TSkinManager;
    sPanel3: TPanel;
    btncancel: TBitBtn;
    btnlogin: TBitBtn;
    procedure btncancelClick(Sender: TObject);
    procedure idChange(Sender: TObject);
    procedure btnloginClick(Sender: TObject);
    procedure passKeyPress(Sender: TObject; var Key: Char);
  private
    { Private declarations }
  end;

```

---

```

public
  { Public declarations }
end;
var
  FormLogin: TFormLogin;
implementation
uses Unit_Main, DateUtils;
{$R *.dfm}
procedure TFormLogin.btncancelClick(Sender: TObject);
begin
  close;
  formmain.show;
  FormMain.mode := 0;
end;
procedure TFormLogin.idChange(Sender: TObject);
begin
  If (trim(id.Text) <> '') and (trim(pass.Text) <> '') then
    btnlogin.Visible := true
  else
    btnlogin.Visible := false;
end;
procedure TFormLogin.btnloginClick(Sender: TObject);
begin
  con.close;
  con.sql.clear;
  con.sql.add('SELECT `id_user`, `pass`, `nama`, `jabatan` FROM `tb_user` WHERE
    id_user="' + trim(id.Text) + '" ');
  con.Prepared;
  con.open;
  if con.recordcount > 0 then
  begin
    With FormMain do
    begin
      ToolBarMain.Show;
      Gejala1.Visible := false;
      User1.Visible := false;
      Pertanyaan1.Visible := false;
      RuicBase1.Visible := false;
      Pasien1.Visible := false;
      Penyakit1.Visible := false;
    end;
  end;
end;

```

---

```

    Pakar1.Visible := false;
    Pasien2.Visible := false;
    Diagnosa1.Visible := false;
    Report1.Enabled := false;
    if (con.FieldName('jabatan').AsString='Admin') then
    begin
        Report1.Enabled := false;
        User1.Visible := True;
        mode := 4;
    end
    else
    begin
        Report1.Enabled := true;
        Pasien2.Visible := true;
        Diagnosa1.Visible := true;
        Gejala1.Visible := true;
        Pertanyaan1.Visible := true;
        RuleBase1.Visible := true;
        Pasien1.Visible := true;
        Penyakit1.Visible := true;
        Pakar1.Visible := true;
        mode := 5;
    end;
    show;
end;
hide;
end
Else
    MessageDlg('Login Gagal, Cek User ID dan Password', mtError, [mbOk], 0);
end;
procedure TFormLogin.passKeyPress(Sender: TObject; var Key: Char);
begin
    if key=#13 then btnloginClick(sender);
end;
end.

```

### ***CODING FORM PENYAKIT***

```

unit Unit_Penyakit;
interface
uses

```



*Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms, Dialogs, DB, ADODB, ImgList, acAlphaImageList, StdCtrls, Buttons, sBitBtn, Grids, sComboBox, sLabel, sEdit, ExtCtrls, sPanel, sSkinProvider, sSkinManager;*

type

```
TFormPenyakit = class(TForm)
  sPanel2: TsPanel;
  slabelFX1: TsLabelFX;
  Image1: TImage;
  idpeny: TsEdit;
  sLabel3: TsLabel;
  Nama: TsEdit;
  sLabel4: TsLabel;
  sLabel1: TsLabel;
  Data: TStringGrid;
  sPanel1: TsPanel;
  BtnDel: TsBitBtn;
  BtnClear: TsBitBtn;
  BtnSave: TsBitBtn;
  sAlphaImageList1: TsAlphaImageList;
  con: TADOQuery;
  kode: TsLabel;
  ket: TMemo;
  sLabel2: TsLabel;
  solusi: TMemo;
  sSkinManager1: TsSkinManager;
  sSkinProvider1: TsSkinProvider;
  procedure FormCreate(Sender: TObject);
  procedure BtnClearClick(Sender: TObject);
  procedure idpenyChange(Sender: TObject);
  procedure FormClose(Sender: TObject; var Action: TCloseAction);
  procedure DataSelectCell(Sender: TObject; ACol, ARow: Integer;
    var CanSelect: Boolean);
  procedure BtnDelClick(Sender: TObject);
  procedure BtnSaveClick(Sender: TObject);
private
  procedure getCode();
  procedure getData();
public
  { Public declarations }
```

---

```

    end;
var
    FormPenyakit: TFormPenyakit;
implementation
uses Unit Main;
{$R *.dfm}
procedure TFormPenyakit.getCode();
var
    Kodes : String;
    i : Integer;
    found : boolean;
begin
    i := 1;
    Kodes := FormatCurr('P00',1);
    idpeny.Text := Kodes;
    found := false;
    While Not found do
    begin
        Kodes := FormatCurr('P00',i);
        con.close;
        con.sql.clear;
        con.sql.add('SELECT * FROM `tb_penyakit` WHERE id_penyakit="'+kodes+'" ');
        con.Prepared;
        con.open;
        if con.recordcount >0 then
            found := False
        else
            begin
                idpeny.Text := Kodes;
                found := True;
            end;
        i := i + 1;
    end;
end;
procedure TFormPenyakit.FormCreate(Sender: TObject);
begin
    with data do
        begin
            ColCount := 4;
            Cells[0,0] := 'ID.';

```

---

```

idpeny.Text := '';
BtnDel.Visible := False;
BtnSave.Visible := false;
idpeny.Text := '';
BtnSave.Caption := 'Simpan';
getCode;
nama.Text := '';
ket.Clear;
solusi.Clear;
kode.Caption := '';
end;
procedure TFormPenyakit.IdpenyChange(Sender: TObject);
begin
  If (trim(nama.Text) <> '') and (trim(ket.Text) <> '') then
    BtnSave.Visible := true
  else
    BtnSave.Visible := false;
  if (BtnSave.Caption = 'Edit') then
    begin
      BtnDel.Visible := false;
      BtnSave.Caption := 'Simpan';
    end
end;
procedure TFormPenyakit.FormClose(Sender: TObject;
  var Action: TCloseAction);
begin
  formmain.fpenyakit := nil;
  Action := caFree;
end;
procedure TFormPenyakit.DataSelectCell(Sender: TObject; ACol,
  ARow: Integer; var CanSelect: Boolean);
begin
  kode.Caption := IntToStr(ARow);
  BtnSave.Caption := 'Edit';
  BtnSave.Visible := True;
  BtnDel.Visible := true
end;
procedure TFormPcnyakit.BtnDelClick(Sender: TObject);
begin
  if (MessageDlg('Yakin Untuk Menghapus data?', mtConfirmation, [mbYes, mbNo], 0) = mrYes) then

```

---

```

begin
  try

    con.close;
    con.sql.clear;
    con.sql.add('Delete      from      tb_penyakit      where      id_penyakit=      "'+
      data.Cells[0,strtoint(kode.Caption)] +' " ');
    con.ExecSQL;

    MessageDlg('Proses Penghapusan Berhasil',mtInformation,[ mbOK],0 );
    BtnClearClick(self);

  except
    MessageDlg('Proses Penghapusan Gagal',mtError,[ mbOK],0 );
  exit;
  end;
end

end;
procedure TFormPenyakit.BtnSaveClick(Sender: TObject);
var aRow : integer;
begin
  if (MessageDlg('Yakin Untuk '+ BtnSave.Caption +' data?',mtConfirmation,[mbYes,mbNo],0))=
    mrYes then
    begin
      try
        if BtnSave.Caption='Simpan' then
          begin

            con.close;
            con.sql.clear;
            con.sql.add('INSERT INTO tb_penyakit ( 'id_penyakit', 'nama',
              'keterangan','solusi') VALUES (''+trim(idpeny.Text )+'', '''+trim(nama.Text )+'',
              '''+trim(ket.Text )+'', '''+trim(solusi.Text )+'')');
            con.ExecSQL;

          end
          else If BtnSave.Caption ='Edit' then
            begin
              aRow := StrToInt(kode.Caption);
              with data do
                begin
                  idpeny.Text := Cells[0,aRow] ;
                end
              end;
            end;
          end;
        end;
      end;
    end;
  end;
end;

```

---

```

        Nama.Text := Cells[1,arow] ;
        ket.Text := Cells[2,arow] ;
        solusi.Text := Cells[3,arow] ;
    end;
    BtnSave.Caption := 'Update';
    BtnDel.Visible := False;
    exit;
end
else if BtnSave.Caption='Update' then
begin
    con.close;
    con.sql.clear;
    con.sql.add('UPDATE tb_penyakit SET nama = "'-trim(nama.Text )-"' , keterangan =
    "'+trim(ket.Text )+"' , solusi = "'-trim(solusi.Text )+"' WHERE ('id_penyakit'=
    "'+idpeny.Text +'");
    con.ExecSQL;
end;
    MessageDlg('Proses Penyimpanan Berhasil',mtInformation,[ mbOK],0 );
    BtnClearClick(self);
except
    MessageDlg('Proses Penyimpanan Gagal',mtError,[ mbOK],0 );
    exit;
end;
end
else
begin
    BtnClearClick(self);
end;
end;
end.

```

### ***CODING RULE BASE***

```

unit Unit_Ru;
interface
uses
    Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
    Dialogs, DB, ADODB, ImgList, acAlphaImageList, StdCtrls, sComboBox,
    sAlphaListBox, sCheckListBox, Buttons, sBitBtn, Grids, ExtCtrls, sLabel,
    sPanel, sEdit, sSkinProvider, sSkinManager;

```

---

type

```

TFormRule = class(TForm)
  sLabel4: TLabel;
  kode: TLabel;
  sPanel2: TPanel;
  sLabelFX1: TLabelFX;
  Image1: TImage;
  Data: TStringGrid;
  sPanel1: TPanel;
  BtnDel: TBitBtn;
  BtnClear: TBitBtn;
  BtnSave: TBitBtn;
  penyakit: TComboBox;
  sAlphalmageList1: TsAlphalmageList;
  con: TADOQuery;
  gejala_sekarang: TComboBox;
  sLabel2: TLabel;
  rule_sebelum: TComboBox;
  sLabel3: TLabel;
  rule_ya: TComboBox;
  sLabel5: TLabel;
  rule_tidak: TComboBox;
  sLabel6: TLabel;
  sLabel1: TLabel;
  tEB: TEdit;
  tMD: TEdit;
  sLabel7: TLabel;
  sSkinManager1: TSSkinManager;
  sSkinProvider1: TSSkinProvider;
  procedure FormCreate(Sender: TObject);
  procedure BtnClearClick(Sender: TObject);
  procedure penyakitChange(Sender: TObject);
  procedure BtnSaveClick(Sender: TObject);
  procedure DataSelectCell(Sender: TObject; ACol, ARow: Integer;
    var CanSelect: Boolean);
  procedure BtnDelClick(Sender: TObject);
  procedure FormClose(Sender: TObject; var Action: TCloseAction);
private
  procedure getData();
public

```

---

```

    { Public declarations }
end;
var
    FormRule: TFormRule;
implementation
uses Unit_splash, Unit_Main;
{$R *.dfm}
procedure TFormRule.getData();
var i:integer;
    kodePenyakit : string;
begin
    con.close;
    con.sql.clear;
    con.sql.add('SELECT * FROM `tb_penyakit` WHERE `id_penyakit` in (SELECT `id_penyakit`
        FROM `tb_gejala_penyakit`)');
    con.Prepared;
    con.open;
    penyakit.Clear;
    kodePenyakit := "";
    if con.recordcount >0 then
    begin
        for i := 1 to con.RecordCount do
        begin
            penyakit.Items.Add(con.fieldByName('id_penyakit').AsString+' |
                | con.fieldByName('nama').AsString);
            con.Next;
        end;
        penyakit.ItemIndex := 0;
    end;
    con.close;
    con.sql.clear;
    con.sql.add('SELECT id_rule, nama, gejala, ' +
        ' id_rule_ya, id_rule_tidak ,id_rule_sebelum,MD,MB ' +
        ' FROM tb_rule, tb_gejala_penyakit, tb_gejala, tb_penyakit ' +
        ' WHERE tb_rule.id_gp = tb_gejala_penyakit.id_gp AND ' +
        ' tb_gejala_penyakit.id_gejala = tb_gejala.id_gejala AND ' +
        ' tb_gejala_penyakit.id_penyakit = tb_penyakit.id_penyakit');
    con.Prepared;
    con.open;
    Data.RowCount := 1;

```

---

```

Data.RowCount := 2;
Data.FixedRows := 1;
if con.recordcount >0 then
begin
  for i := 1 to con.RecordCount do
  begin
    with data do
    begin
      Cells[0,i] := con.fieldByName('id_rule').AsString;
      Cells[1,i] := con.fieldByName('nama').AsString;
      Cells[2,i] := con.fieldByName('gejala').AsString;
      Cells[3,i] := con.fieldByName('id_rule_sebelum').AsString;
      Cells[4,i] := con.fieldByName('id_rule_ya').AsString;
      Cells[5,i] := con.fieldByName('id_rule_tidak').AsString;
      Cells[6,i] := con.fieldByName('MB').AsString;
      Cells[7,i] := con.fieldByName('MD').AsString;
      RowCount :=i+1;
      con.Next;
    end;
  end
end;
penyakitChange(self);
end;
procedure TFormRule.FormCreate(Sender: TObject);
begin
  with data do
  begin
    ColCount := 8;
    Cells[0,0] := 'ID.';
    Cells[1,0] := 'Nama';
    Cells[2,0] := 'Gejala';
    Cells[3,0] := 'Rule Sebelum';
    Cells[4,0] := 'Rule Ya';
    Cells[5,0] := 'Rule Tidak';
    Cells[6,0] := 'MB';
    Cells[7,0] := 'MD';
    ColWidths[1] := 120;
    ColWidths[2] := 500;
    ColWidths[2] := 1500;
  end;
end;

```

---



```

    BtnClearClick(self);
end;
procedure TFormRule.BtnClearClick(Sender: TObject);
begin
    getData;
    BtnDel.Visible := False;
    BtnSave.Visible := false;
    BtnSave.Caption := 'Simpan';
    kode.Caption := "";
    tFB.Clear;
    tMD.Clear;
    penyakit.Enabled := true;
end;
procedure TFormRule.penakitChange(Sender: TObject);
var kodePenyakit : string;
    i :integer;
begin
    If (trim(penyakit.Text) <> "") then
        BtnSave.Visible := true
    else
        BtnSave.Visible := false;
    if (BtnSave.Caption = 'Edit') then
        begin
            BtnDel.Visible := false;
            BtnSave.Caption := 'Simpan';
        end;

    kodePenyakit := trim(copy(penyakit.Text,0,pos(' ',penyakit.Text)));
    if trim(kodePenyakit) <> "" then
        begin
            con.close;
            con.sql.clear;
            con.sql.add('SELECT id_gp, gejala ' +
                ' FROM tb_gejala, tb_gejala_penyakit ' +
                ' WHERE     tb_gejala.id_gejala =     tb_gejala_penyakit.id_gejala     and
                id_penyakit="'+kodePenyakit+'"');
            con.Prepared;
            con.open;
            gejala_sekarang.Clear;
            if con.recordcount >0 then

```

---

```

begin
  for i := 1 to con.RecordCount do
  begin
    gejala_sekarang.Items.Add(con.fieldByName('id_gp').AsString+'|'+
      + con.fieldByName('gejala').AsString);
    con.Next;
  end;
  gejala_sekarang.ItemIndex := 0;
end;
con.close;
con.sql.clear;
con.sql.add('SELECT id_gp, gejala'+
  ' FROM tb_gejala, tb_gejala_penyakit'+
  ' WHERE tb_gejala.id_gejala = tb_gejala_penyakit.id_gejala and
  id_penyakit="'+kodePenyakit+'");
con.Prepared;
con.open;
rule_sebelum.Clear;
if con.recordcount >0 then
begin
  for i := 1 to con.RecordCount do
  begin
    rule_sebelum.Items.Add(con.fieldByName('id_gp').AsString+'|'+
      + con.fieldByName('gejala').AsString);
    con.Next;
  end;
  rule_sebelum.ItemIndex := 0;
end;
con.close;
con.sql.clear;
con.sql.add('SELECT id_gp, gejala'+
  ' FROM tb_gejala, tb_gejala_penyakit'+
  ' WHERE tb_gejala.id_gejala = tb_gejala_penyakit.id_gejala and
  id_penyakit="'+kodePenyakit+'");
con.Prepared;
con.open;
rule_ya.Clear;
if con.recordcount >0 then
begin
  for i := 1 to con.RecordCount do

```

---

```

begin
    rule_ya.Items.Add(con.fieldByName('id_gp').AsString+' |'
        + con.fieldByName('gejala').AsString);
    con.Next;
end;
rule_ya.ItemIndex := 0;
end;
con.close;
con.sql.clear;
con.sql.add('SELECT id_gp, gejala ' +
    ' FROM tb_gejala, tb_gejala_penyakit ' +
    ' WHERE tb_gejala.id_gejala = tb_gejala_penyakit.id_gejala and
    id_penyakit="'+kodePenyakit+'');
con.Prepared;
con.open;
rule_tidak.Clear;
if con.recordcount > 0 then
begin
    for i := 1 to con.RecordCount do
    begin
        rule_tidak.Items.Add(con.fieldByName('id_gp').AsString+' |'
            + con.fieldByName('gejala').AsString);
        con.Next;
    end;
    rule_tidak.ItemIndex := 0;
end;
end;
end;
procedure TFormRule.BtnSaveClick(Sender: TObject);
var ij : integer;
    tempA,tempB,kodepcnyakit :string;
begin
    if (MessageDlg('Yakin Untuk '+ BtnSave.Caption +' data?',mtConfirmation,[mbYes,mbNo],0))=
        mrYes then
    begin
        try
            if BtnSave.Caption='Simpan' then
            begin
                con.close;
            con.sql.clear;

```

---

```

con.sql.add('SELECT * ' +
' FROM tb_rule ' +
' WHERE id_gp="' + trim(copy(gejala_sekarang.Text,0,pos(' ',gejala_sekarang.Text))) + '" ');
con.Prepared;
con.open;
if con.recordcount >0 then
begin
    MessageDlg('Data Sudah Ada',mtError,[ mbOK],0 );
    exit;
end;

con.close;
con.sql.clear;
con.SQL.Clear;
con.sql.add('INSERT INTO tb_rule (MB,MD,id_rule_sebelum, id_gp, id_rule_ya,
id_rule_tidak) ' +
' VALUES (' + trim(tLB.Text) + ',' + trim(tMD.Text) + ',' +
trim(copy(rule_sebelum.Text,0,pos(' ',rule_sebelum.Text))) + ',' +
trim(copy(gejala_sekarang.Text,0,pos(' ',gejala_sekarang.Text))) +
' ,'+ trim(copy(rule_ya.Text,0,pos(' ',rule_ya.Text))) + ',' +
trim(copy(rule_tidak.Text,0,pos(' ',rule_tidak.Text))) + '");
con.ExecSQL;
end
else If BtnSave.Caption = 'Edit' then
begin

con.close;
con.sql.clear;
con.sql.add('SELECT id_penyakit ' +
' FROM tb_rule, tb_gejala_penyakit ' +
' WHERE tb_rule.id_gp = tb_gejala_penyakit.id_gp and
id_rule="' + data.Cells[0,strtoint(kode.Caption)] + '" ');
con.Prepared;
con.open;
tLB.Text := data.Cells[6,strtoint(kode.Caption)];
tMD.Text := data.Cells[7,strtoint(kode.Caption)];
if con.recordcount >0 then
begin
    kodePenyakit := con.fieldByName('id_penyakit').AsString;
end;
if trim(kodePenyakit) <> '' then

```

```

begin
con.close;
con.sql.clear;
con.sql.add('SELECT id_gp, gejala ' +
' FROM tb_gejala, tb_gejala_penyakit ' +
' WHERE tb_gejala.id_gejala = tb_gejala_penyakit.id_gejala and
id_penyakit="'+kodePenyakit+'"');
con.Prepared;
con.open;
gejala_sekarang.Clear;
if con.recordcount >0 then
begin
for i := 1 to con.RecordCount do
begin
gejala_sekarang.Items.Add(con.fieldByName('id_gp').AsString+' |
+ con.fieldByName('gejala').AsString);
con.Next;
end;
gejala_sekarang.ItemIndex := 0;
end;
con.close;
con.sql.clear;
con.sql.add('SELECT id_gp, gejala ' +
' FROM tb_gejala, tb_gejala_penyakit ' +
' WHERE tb_gejala.id_gejala = tb_gejala_penyakit.id_gejala and
id_penyakit="'+kodePenyakit+'"');
con.Prepared;
con.open;
rule_sebelum.Clear;
if con.recordcount >0 then
begin
for i := 1 to con.RecordCount do
begin
rule_sebelum.Items.Add(con.fieldByName('id_gp').AsString+' |
+ con.fieldByName('gejala').AsString);
con.Next;
end;
rule_sebelum.ItemIndex := 0;
end;
con.close;

```

---

```

con.sql.clear;
con.sql.add('SELECT id_gp, gejala ' +
  ' FROM tb_gejala, tb_gejala_penyakit ' +
  ' WHERE tb_gejala.id_gejala = tb_gejala_penyakit.id_gejala and
  id_penyakit="'+kodePenyakit+'");
con.Prepared;
con.open;
rule_ya.Clear;
if con.recordcount >0 then
begin
  for i := 1 to con.RecordCount do
  begin
    rule_ya.Items.Add(con.fieldByName('id_gp').AsString+' '
      + con.fieldByName('gejala').AsString);
    con.Next;
  end;
  rule_ya.ItemIndex := 0;
end;
con.close;
con.sql.clear;
con.sql.add('SELECT id_gp, gejala ' +
  ' FROM tb_gejala, tb_gejala_penyakit ' +
  ' WHERE tb_gejala.id_gejala = tb_gejala_penyakit.id_gejala and
  id_penyakit="'+kodePenyakit+'");
con.Prepared;
con.open;
rule_tidak.Clear;
if con.recordcount >0 then
begin
  for i := 1 to con.RecordCount do
  begin
    rule_tidak.Items.Add(con.fieldByName('id_gp').AsString+' '
      + con.fieldByName('gejala').AsString);
    con.Next;
  end;
  rule_tidak.ItemIndex := 0;
end;
end;

penyakit.Enabled :=False;
BtnSave.Caption := 'Update';

```

---

```

        BtnDel.Visible := False;
        exit;
    end
else if BtnSave.Caption = 'Update' then
begin
    con.close;
    con.sql.clear;
    con.SQL.Clear;
    con.sql.add('UPDATE tb_rule ' +
        ' SET MB="'+trim(tEB.Text)+'",MD="'+trim(tMD.Text)+'",id_rule_sebelum = "' +
        trim(copy(rule_sebelum.Text,0,pos('!',rule_sebelum.Text)))+
        '" , id_gp = "' +trim(copy(gejala_sekarang.Text,0,pos('!',gejala_sekarang.Text)))+
        '" , id_rule_ya = "' + trim(copy(rule_ya.Text,0,pos('!',rule_ya.Text)))+
        '" , id_rule_tidak = "' + trim(copy(rule_tidak.Text,0,pos('!',rule_tidak.Text)))+ "' ' +
        ' WHERE (id_rule = "' + data.Cells[0, strtoint(kode.Caption)] + "')');
    con.ExecSQL;
end;
MessageDlg('Proses Penyimpanan Berhasil',mtInformation,[ mbOK],0 );
BtnClearClick(self);
except
    MessageDlg('Proses Penyimpanan Gagal',mtError,[ mbOK],0 );
exit;
end;
end
else
begin
    BtnClearClick(self);
end;
end;
procedure TFormRule.DataSelectCell(Sender: TObject; ACol, ARow: Integer;
var CanSelect: Boolean);
begin
    kode.Caption := IntToStr(ARow);
    BtnSave.Caption := 'Edit';
    BtnSave.Visible := True;
    BtnDel.Visible := true;
end;
procedure TFormRule.BtnDelClick(Sender: TObject);
begin
    if (MessageDlg('Yakin Untuk Menghapus data?',mtConfirmation,[mbYes,mbNo],0)) = mrYes then

```

---

```

begin
  try
    con.close;
    con.sql.clear;
    con.sql.add('DELETE FROM tb_rule WHERE id_rule = '+
      data.Cells[0,strtoint(kode.Caption)]+' ');
    con.ExecSQL;
    MessageDlg('Proses Penghapusan Berhasil',mtInformation,[ mbOK],0 );
    BtnClearClick(self);
  except
    MessageDlg('Proses Penghapusan Gagal',mtError,[ mbOK],0 );
  exit;
end;
end
end;
procedure TFormRule.FormClose(Sender: TObject; var Action: TCloseAction);
begin
  FormMain.frule := nil;
  Action := caFree;
end;
end.

```

### ***CODING FORM PASIEN***

```

unit Unit_Pasien;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, DB, ADODB, ImgList, acAlphaImageList, StdCtrls, Buttons,
  sBitBtn, Grids, sComboBox, sEdit, ExtCtrls, sLabel, sPanel;
type
  TFormPasien = class(TForm)
    kode: TLabel;
    sPanel2: TPanel;
    sLabelFX1: TLabelFX;
    Image1: TImage;
    Data: TStringGrid;
    sPanel1: TPanel;
    BtnDel: TBitBtn;
    sAlphaImageList1: TAlphaImageList;

```

---



```

con: TADOQuery;
sLabel1: TLabel;
jenis: TsComboBox;
keyword: TsEdit;
procedure FormCreate(Sender: TObject);
procedure keywordChange(Sender: TObject);
procedure DataSelectCell(Sender: TObject; ACol, ARow: Integer;
  var CanSelect: Boolean);
procedure BtnDelClick(Sender: TObject);
procedure FormClose(Sender: TObject; var Action: TCloseAction);
private
  { Private declarations }
public
  { Public declarations }
end;
var
  FormPasien: TFormPasien;
implementation
uses Unit Main;
{$R *.dfm}
procedure TFormPasien.FormCreate(Sender: TObject);
begin
  Data.RowCount := 2;
  with data do
  begin
    ColCount := 4;
    Cells[0,0] := 'ID.';
    Cells[1,0] := 'Nama';
    Cells[2,0] := 'Alamat';
    Cells[3,0] := 'Umur';
    ColWidths[1] := 100;
    ColWidths[2] := 280;
    ColWidths[3] := 80;
  end;
  jenis.Items.Add('Nama');
  jenis.Items.Add('ID Pasien');
  jenis.Items.Add('Alamat');
  jenis.ItemIndex := 0;
end;
procedure TFormPasien.keywordChange(Sender: TObject);

```

---

```

var i:integer;
begin
    con.close;
    con.sql.clear;
    con.sql.add('select * from tb_pasien where ' + StringReplace(jenis.Text, ' ', '_',[rfIgnoreCase]) + "
        like '%" + trim(keyword.Text) + "%' ");
    con.Prepared;
    con.open;
    Data.RowCount := 1;
    Data.RowCount := 2;
    Data.FixedRows := 1;
    if con.recordcount > 0 then
    begin
        for i := 1 to con.RecordCount do
        begin
            with data do
            begin
                Cells[0,i] := con.fieldByName('id_pasien').AsString;
                Cells[1,i] := con.fieldByName('nama').AsString;
                Cells[2,i] := con.fieldByName('alamat').AsString;
                Cells[3,i] := con.fieldByName('umur').AsString;
                RowCount := i+1;
                con.Next;
            end;
        end
    end;
end;
procedure TFormPasien.DataSelectCell(Sender: TObject; ACol, ARow: Integer;
    var CanSelect: Boolean);
begin
    kode.Caption := IntToStr(ARow);
    BtnDel.Visible := true
end;
procedure TFormPasien.BtnDelClick(Sender: TObject);
begin
    if (MessageDlg('Yakin Untuk Menghapus data?', mtConfirmation, [mbYes, mbNo], 0)) = mrYes then
    begin
        try
            con.close;
            con.sql.clear;

```

---

```

con.sql.add('Delete      from      tb_pasien      where      id_pasien      '+
data.Cells[0,strtoint(kode.Caption)] +' ');
con.ExecSQL;
MessageDlg('Proses Penghapusan Berhasil',mtInformation,[ mbOK],0 );
except
    MessageDlg('Proses Penghapusan Gagal',mtError,[ mbOK],0 );
exit;
end;
end :
Data.RowCount := 1;
Data.RowCount := 2;
Data.FixedRows := 1;
keyword.Text := ' ';
keywordChange(self);
BtnDel.Visible := false;
end;
procedure TFormPasien.FormClose(Sender: TObject; var Action: TCloseAction);
begin
    FormMain.fpasien := nil;
    Action := caFree;
end;
end.

```

---

```
con.sql.add('Delete      from      tb_pasien      where      id_pasien="
data.Cells[0,StrToInt(kode.Caption)] +"' ');
con.ExecSQL;
MessageDlg('Proses Penghapusan Berhasil',mtInformation,[ mbOK],0 );
except
    MessageDlg('Proses Penghapusan Gagal',mtError,[ mbOK],0 );
exit;
end;

end ;
Data.RowCount := 1;
Data.RowCount := 2;
Data.FixedRows := 1;
keyword.Text := ' ';
keywordChange(self);
BtnDel.Visible := false;

end;
procedure TFormPasien.FormClose(Sender: TObject; var Action: TCloseAction);
begin
    FormMain.fpasien := nil;
    Action := caFree;
end;
end.
```

---



FORMULIR BIMBINGAN SKRIPSI

Nama : Nashrul Haq  
Nim : 05.12.583  
Masa Bimbingan : 14 September 2010 s/d 14 Pebruari 2011  
Judul Skripsi : Pengembangan Sistem Pakar Untuk Untuk Diagnosa Penyakit Gigi Dan Mulut Dengan Menggunakan Metode Backward Chaining

No	Tanggal	Uraian	Paraf Pembimbing
1	21 Desember 2010	Demo Program	
2	06 Januari 2011	Revisi Makalah	
3	27 Januari 2011	Revisi Penulisan & Abstraksi	
4	02 Pebruari 2011	Revisi Nomor Halaman & Nomor Tabel	
5	10 Pebruari 2011	Maju Bab I,II,III,IV,V dan daftar pustaka, cover dan lembar persetujuan	
6			
7			
8			
9			
10			

Malang,  
Dosen pembimbing I

Ir. Yusuf Ismail Nakhoda, MT  
NIP. Y. 1018800189

Form S-4b



**FORMULIR BIMBINGAN SKRIPSI**

Nama : Nashrul Haq  
Nim : 05.12.583  
Masa Bimbingan : 14 September 2010 s/d 14 Pebruari 2011  
Judul Skripsi : Pengembangan Sistem Pakar Untuk Untuk Diagnosa Penyakit Gigi Dan Mulut Dengan Menggunakan Metode Backward Chaining

No	Tanggal	Uraian	Paraf Pembimbing
1	21 Desember 2010	Demo Program	
2	06 Januari 2011	Revisi Makalah Seminar Hasil	
3	11 Januari 2011	Revisi Uji Responden	
4	18 Januari 2011	Revisi Uji Keakuratan Kelayakan	
5	26 Januari 2011	ACC Makalah Seminar Hasil	
6			
7			
8			
9			
10			

Malang,  
Dosen pembimbing II

**Dr. Eng. Aryanto Soetedjo, ST, MT.**  
NIP.Y.1030800417

Form S-4b



**BERITA ACARA UJIAN SKRIPSI**  
**FAKULTAS TEKNOLOGI INDUSTRI**


Nama : NASHRUL HAQ  
Nim : 05.12.583  
Jurusan : Teknik Elektro S-1  
Konsentrasi : Teknik Komputer dan Informatika S-1  
Judul : **PENGEMBANGAN SISTEM PAKAR UNTUK DIAGNOSA  
PENYAKIT GIGI DAN MULUT DENGAN MENGGUNAKAN  
METODE BACKWARD CHAINING**

Dipertahankan dihadapan Tim Pengujian Skripsi jenjang Program Strata Satu (S-1)

Pada Hari : Senin  
Tanggal : 21 Februari 2011  
Dengan Nilai : 78,00 (B+) *✓*

Panitia Ujian Skripsi :

Ketua Majelis Penguji



Ir. Yusuf Ismail Nakhoda, MT.  
NIP.Y.1018800189

Sekretaris Majelis Penguji



Dr. Eng. Aryanto S, ST, MT.  
NIP.P.1030800417

Anggota Penguji :

Penguji I,



Sandy Nataly M, S, Kom.  
NIP.P.1030800418

Penguji II,



Sotyo Hadi, ST, MSc.  
NIP.Y.1039700309