

# Aplikasi Diagnosa Penyakit Gigi dan Pengobatannya Menggunakan Metode Forward Chaining dan Association Rule Analysis

**Andi Muh. Lukman**

Fakultas Teknik UPRI Makassar

Lulu2k7@gmail.com

## **Abstrak**

*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, 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.*

*Teknik pengumpulan data yang digunakan antara lain observasi, wawancara, dan studi literature. Untuk teknik analisis data menggunakan metode Forward Chaining. Pengujian Basis Part sebagai alternatif pengembangan white box testing, penulis menggunakan metode Basis Path Testing, Penginputan data gejala/penyakit untuk dibuatkan suatu rule diagnosis ke dalam knowledge base. Dimana penginputan ini dilakukan oleh pakar pada Rumah Sakit Umum Daya Makassar. Berdasarkan hasil penggunaan metode forward chaining dan association rule analysis tersebut dapat disimpulkan bahwa informasi mengenai gejala-gejala yang menjadi penyebab penyakit gigi beserta penanganannya berupa tindakan atau terapi.*

**Kata kunci:** *Diagnosa penyakit menggunakan metode forward chaining dan association rule analysis.*

## **Abstract**

*Patients with dental disease can also be affected by another, suffered from dental disease, as based on medical science, dental disease may spread and can cause organs of the body experiencing pain. It is of course also have a way of handling, namely to diagnose whether the disease suffered by the patient's teeth may propagate or not based on the characteristics that are applicable to the theory of the medical science.*

*Data collection techniques used include observation, interview, and literature study. For data analysis techniques using methods Forward Chaining. Part Base testing as an alternative to white box testing, the authors use the method Basis Path Testing, inputting the data symptom / disease diagnosis to be made a rule to the knowledge base. Where inputting is done by the experts at the General Hospital Makassar Power. Based on the results of the use of forward chaining method and association rule analysis, we can conclude that the information about the symptoms that cause dental disease and its treatment in the form of action or treatment.*

**Keywords:** *Diagnosis of diseases using forward chaining and association rule analysis.*

## **1. Pendahuluan**

Ilmu kedokteran yang ada saat ini sudah mengalami perkembangan dengan sangat pesat. Banyak obat-obat baru yang ditemukan untuk mengatasi berbagai penyakit yang bermunculan di dunia. Demikian pula dengan salah satu bidang ilmu kedokteran, yaitu bidang kedokteran gigi. Masalah dan penyakit gigi saat ini merupakan masalah yang dihadapi oleh

masyarakat banyak, dan varian penyakit dan kelainan gigi juga semakin berkembang dari waktu ke waktu. Hal ini menuntut para dokter gigi, baik itu yang sudah lama berprofesi dalam dunia kedokteran, para dokter muda dan mahasiswa yang akan dilantik sebagai dokter harus mengikuti perkembangan dan update terbaru tentang masalah-masalah yang ada pada gigi beserta cara penanganannya.

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, 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. Tidak mungkin seorang ahli dapat menyimpan semua informasi yang dia butuhkan hanya dalam memori otaknya saja, apalagi untuk kasus-kasus khusus yang jarang terjadi. Kendala lain, selain begitu banyaknya informasi, adalah waktu. Seandainya ada ahli yang mengklaim bahwa ahli tersebut mampu menghafal semua informasi, tidak mungkin informasi tersebut akan bertahan selamanya, karena seorang ahli bagaimanapun juga hanyalah seorang manusia yang tidak mungkin menang melawan waktu.

Berdasarkan uraian dari latar belakang maka penulis tertarik untuk membuat sistem yang memadukan teknologi komputer dengan pengetahuan seorang pakar ke dalam suatu aplikasi yang dibangun dengan teknologi komputer yang menerapkan salah satu bidang metode pencarian yaitu metode forward chaining yang memiliki cara pikir mirip dengan seorang pakar dalam melakukan anamnesa dalam proses diagnosa. Dimana aplikasi yang dibuat bukan bertujuan sebagai pengganti posisi seorang dokter atau seorang pakar dalam bidang tertentu, melainkan sebagai alat pelengkap atau alat bantu yang bisa digunakan seorang dokter. Sistem ini penulis buat dengan judul "Aplikasi Diagnosa Penyakit Gigi Dan Pengobatannya Menggunakan Metode Forward Chaining Dan Association Rule Analysis".

## **2. Tinjauan Pustaka**

### **Pengertian Aplikasi**

Menurut Tim DINASTINDO [6] "aplikasi adalah masalah yang memakai teknik pemrosesan data, aplikasi biasanya mengacu pada komputasi, seperti keperluan kapasitas komputasi yang diinginkan, atau pemrosesan data". Aplikasi komputer biasanya ditulis dalam bahasa pemrograman dan dipergunakan untuk menyelesaikan masalah tertentu dengan kata lain aplikasi dapat melakukan pekerjaan-pekerjaan sesuai kebutuhan user, seperti: program dalam dunia perbankan untuk menyediakan informasi rekening bulanan para nasabah, program dalam dunia pendidikan (universitas) yang menyediakan informasi akademik baik melalui PC (*Personal Computer*) maupun melalui *mobile device (handphone)*.

## Pengertian Diagnosa

Kata diagnosa berasal dari bahasa Yunani yang berarti penentuan jenis penyakit dengan meneliti (memeriksa) gejala dan proses pemeriksaan terhadap hal yang dipandang tidak beres. Diagnosa merupakan suatu usaha mencari sumber-sumber penyebab terjadinya sesuatu yang dilakukan berupa penelusuran dan penelitian yang terencana, sistematis dan berkesinambungan atau sampai tercapainya tujuan yang diharapkan.

## Sistem Pakar

Menurut Kusriani [4] sistem pakar adalah aplikasi berbasis komputer yang digunakan untuk menyelesaikan masalah sebagaimana yang dipikirkan oleh pakar. Pengertian lain menurut Anita Desiani & Muhammad Arhami [2] sistem pakar adalah sistem yang berbasis pengetahuan, yaitu sistem yang meniru penalaran dari seorang pakar dalam bidang tertentu. Seorang pakar dengan sistem pakar mempunyai banyak perbedaan. Menurut Anita Desiani & Muhammad Arhami [2] mengemukakan perbandingan kemampuan antara seorang pakar dengan sebuah sistem pakar seperti pada Tabel 1.

**Tabel 1:** Perbandingan kemampuan pakar dengan sistem pakar

<b>Faktor</b>	<b>Human Expert</b>	<b>Expert System</b>
<i>Time Availability</i>	<i>Hari kerja</i>	<i>Setiap saat</i>
<i>Geografis</i>	<i>Lokal/tertentu</i>	<i>Dimana saja</i>
<i>Keamanan</i>	<i>Tidak tergantikan</i>	<i>Dapat diganti</i>
<i>Perishable/dapat habis</i>	<i>Ya</i>	<i>Tidak</i>
<i>Perfomansi</i>	<i>Variabel</i>	<i>Konsisten</i>
<i>Kecepatan</i>	<i>Variabel</i>	<i>Konsisten</i>
<i>Biaya</i>	<i>Tinggi</i>	<i>Terjangkau</i>

## Ciri-Ciri dan Kategori Masalah Sistem Pakar

Sistem pakar merupakan program-program praktis yang menggunakan strategi heuristik yang dikembangkan oleh manusia untuk menyelesaikan permasalahan yang spesifik, disebabkan oleh keheuristikannya dan sifatnya yang berdasarkan pada pengetahuan. Anita Desiani & Muhammad Arhami [2] mengemukakan berdasarkan pengetahuan, yaitu:

1. Memiliki informasi yang handal, baik dalam menampilkan langkah-langkah maupun dalam menjawab pertanyaan-pertanyaan tentang proses penyelesaian.
2. Mudah dimodifikasi, yaitu dengan menambah atau menghapus suatu kemampuan dari basis pengetahuannya.
3. Heuristik dalam menggunakan pengetahuan (yang sering kali tidak sempurna) untuk mendapatkan penyelesaiannya.
4. Dapat digunakan dalam berbagai jenis komputer.
5. Memiliki kemampuan untuk beradaptasi.

### Keuntungan Sistem Pakar

Sistem pakar (*expert system*) merupakan paket perangkat lunak atau paket program komputer yang ditujukan sebagai penyedia nasihat dan sarana bantu dalam memecahkan masalah di bidang-bidang spesialisasi tertentu seperti sains, perikanan, matematika, kedokteran, pendidikan dan sebagainya. Sistem pakar merupakan subset dari *Artificial Intelligence*.

Ada beberapa keunggulan sistem pakar, di antaranya dapat:

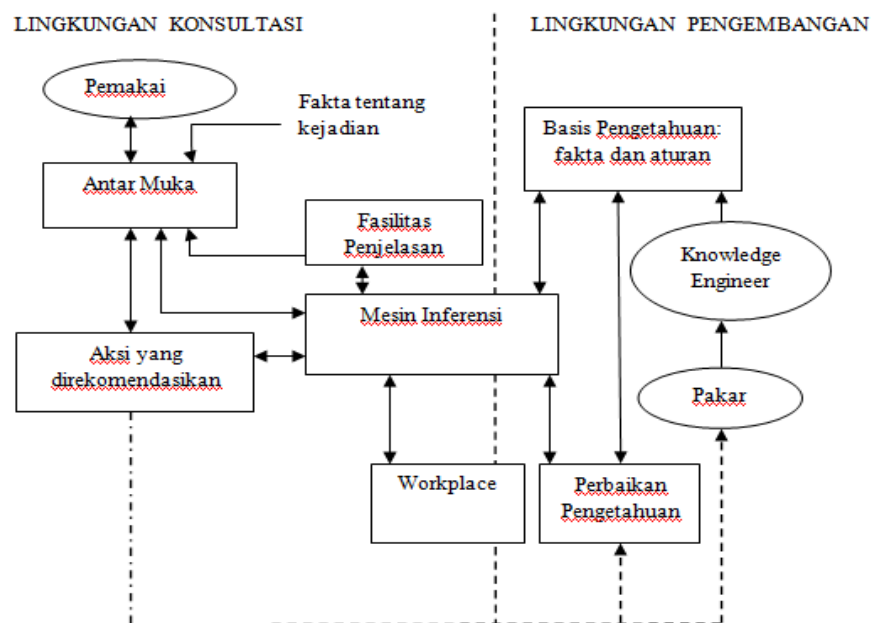
1. Menghimpun data dalam jumlah yang sangat besar.
2. Menyimpan data tersebut untuk jangka waktu yang panjang dalam suatu bentuk tertentu.
3. Mengerjakan perhitungan secara cepat dan tepat dan tanpa jemu mencari kembali data yang tersimpan dengan kecepatan tinggi.

Sementara kemampuan sistem pakar di antaranya adalah:

1. Menjawab berbagai pertanyaan yang menyangkut bidang keahliannya.
2. Bila diperlukan dapat menyajikan asumsi dan alur penalaran yang digunakan untuk sampai ke jawaban yang dikehendaki.
3. Menambah fakta kaidah dan alur penalaran sah yang baru ke dalam otaknya.

### Struktur Sistem Pakar

Menurut Turban [2], sistem pakar disusun oleh dua bagian utama, yaitu lingkungan pengembangan (*development environment*) dan lingkungan konsultasi (*consultation environment*). Komponen-komponen sistem pakar dalam kedua bagian tersebut dapat dilihat pada gambar berikut yaitu *user interface* (antarmuka pengguna), basis pengetahuan, akuisisi pengetahuan, mesin inferensi, *workplace*, fasilitas penjelasan, perbaikan pengetahuan.



Gambar 1: Arsitektur sistem pakar

#### a. Representase Pengetahuan

Representase pengetahuan dan kemampuan untuk melakukan penalaran dalam bidang kecerdasan buatan merupakan hal yang penting. Meskipun suatu sistem memiliki banyak pengetahuan, namun jika tidak memiliki kemampuan untuk menalar akan percuma saja. Sebaliknya jika sistem memiliki kemampuan yang handal untuk menalar tapi basis pengetahuan yang dimilikinya tidak cukup, maka solusi yang diperolehnya menjadi tidak maksimal. Menurut Kusri (2008: 6) pengertian representase pengetahuan adalah metode yang digunakan untuk mengkodekan pengetahuan dalam sebuah sistem pakar. Representasi dimaksudkan untuk menangkap sifat-sifat penting masalah dan membuat informasi itu dapat diakses oleh prosedur pemecahan masalah.

#### b. Kaidah Produksi

Pengetahuan dalam sistem produksi boleh juga direpresentasikan oleh himpunan kaidah dalam bentuk: IF [kondisi] THEN [aksi]

Dengan sebuah kontrol sistem dan basis data. Kontrol sistem memberikan aturan penerjemahan dan pengurutan. Basis data beraksi sebagai konteks cadangan untuk record yang kondisinya di evaluasi oleh kaidah dan informasi dimana kaidah akan beraksi. Berikutnya untuk sintaks IF-THEN, kaidah produksi juga sering digambarkan **sebagai pasangan-pasangan berikut kondisi-aksi, antecedent-konsequent, pola-aksi, situasi-respons.**

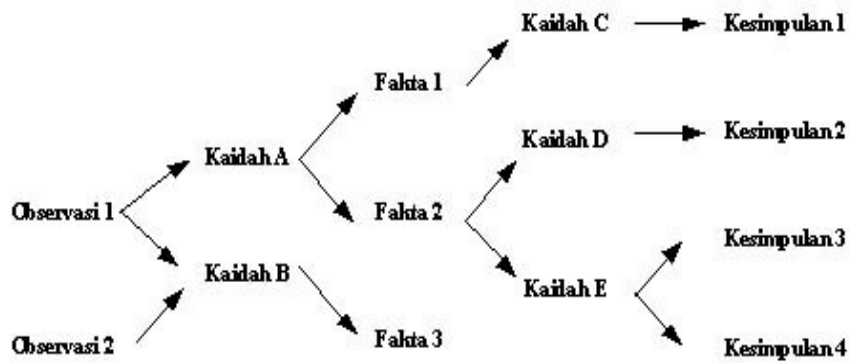
#### c. Metode Inferensi

Menurut Kusri (2008: 8) inferensi merupakan proses untuk menghasilkan informasi dari fakta yang diketahui atau diasumsikan. Inferensi adalah konklusi logis (*logical conclusion*) atau implikasi berdasarkan informasi yang tersedia. Dalam sistem pakar proses inferensi dilakukan dalam suatu modul yang disebut *inference Engine* (Mesin inferensi). Ketika representasi pengetahuan (RP) pada bagian *knowledge base* telah lengkap, atau paling tidak telah berada pada level yang cukup akurat, maka RP tersebut telah siap digunakan. *Inference engine* merupakan modul yang berisi program tentang bagaimana mengendalikan proses *reasoning*.

Ada dua metode inferensi yang penting dalam sistem pakar yaitu: runut maju (*forward chaining*) dan runut balik (*backward chaining*).

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

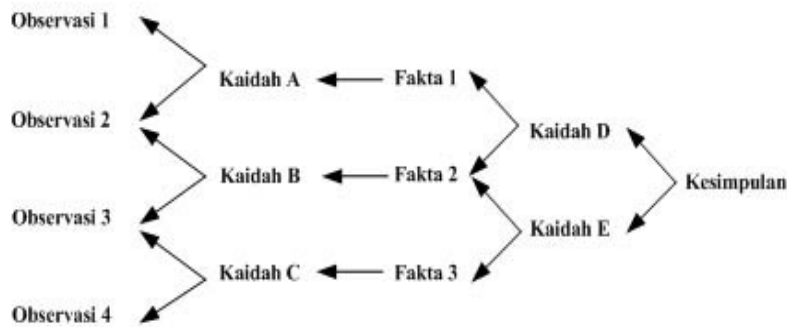
Menurut Wilson (Kusri, 2008: 8) Runut maju berarti menggunakan himpunan aturan kondisi-aksi. Dalam metode ini, data digunakan untuk menentukan aturan mana yang akan dijalankan, kemudian aturan tersebut dijalankan. Mungkin proses menambahkan data ke memori kerja. Proses diulang sampai ditemukan suatu hasil. Pada teknik *forward chaining*, penalaran akan dimulai dari sejumlah fakta-fakta atau data-data untuk menguji *hipotesis*, seperti tampak pada Gambar 2.



Gambar 2: Diagram runut maju (forward chaining)

**e. Runut Balik (Backward Chaining)**

Menurut Giarratano dan Riley (Kusrini, 2008: 11) runut balik merupakan metode penalaran kebalikan dari runut maju. Runut balik disebut juga sebagai *goal-driven reasoning*, merupakan cara yang efisien untuk memecahkan masalah yang dimodelkan sebagai masalah pemilihan terstruktur. Tujuan inferensi adalah mengambil pilihan terbaik dari banyak kemungkinan. Metode inferensi runut balik ini cocok digunakan untuk memecahkan masalah diagnosis.



Gambar 3: Diagram runut balik (backward chaining)

**f. Association Rule Analysis**

*Association analysis* didefinisikan sebagai suatu proses untuk menemukan semua *association rule* yang memenuhi syarat minimum *support* dan syarat minimum *confidence*. Dalam pencarian *association rule*, diperlukan suatu variabel ukuran yang dapat ditentukan oleh *user*, untuk mengatur batasan sejauh mana dan sebanyak apa hasil *output* yang diinginkan oleh *user*.

Rumus *support* dan *confidence*:

$$Support, s(X \rightarrow Y) = \sigma(X \cup Y) / N \dots\dots\dots (1)$$

$$Confidence, c(C \rightarrow Y) = \sigma(X \cup Y) / \sigma(X) \dots\dots\dots (2)$$

**g. Faktor Kepastian**

Dalam menghadapi suatu masalah, sering ditemukan jawaban yang tidak memiliki kepastian penuh. Ketidakpastian ini bisa berupa probabilitas atau kebolehjadian yang

bergantung pada hasil suatu kejadian. Hasil yang tidak pasti disebabkan oleh dua faktor, yaitu aturan yang tidak pasti dan jawaban pengguna yang tidak pasti atas suatu pertanyaan yang diajukan oleh sistem.

#### h. Menentukan *Certainty Factor* (CF) Pararel

CF Paralel merupakan CF yang diperoleh dari beberapa premis pada sebuah aturan. Besarnya CF sequensial dipengaruhi oleh CFUser untuk masing-masing premis dan operator dari premis. Rumus untuk masing-masing operator dapat dilihat pada rumus 3.2 3.3 dan 3.4.

$$CF(x \text{ Dan } y) = \text{Min} (CF(x), CF(y)) \dots\dots\dots(3.2)$$

$$CF(x \text{ Atau } y) = \text{Max} (CF(x), CF(y)) \dots\dots\dots(3.3)$$

$$CF(\text{Tidak } x) = - CF(x)$$

#### i. Menentukan CF Sequensial

Bentuk dasar rumus *certainty factor* sebuah aturan JIKA E MAKA H ditunjukkan oleh Rumus 3.5

$$CF(H,e) = CF(E,e)*CF(H,E) \dots\dots\dots(3.5)$$

di mana :

CF(E,e) : *certainty factor* evidence E yang mempengaruhi oleh evidence e

CF(H,E) : *certainty factor* hipotesis dengan asumsi *evidence* diketahui dengan pasti, yaitu ketika CF(E,e) = 1

CF(E,e) : *certainty factor* hipotesis yang mempengaruhi oleh *evidence* e

Jika semua *evidence* pada *antecedent* diketahui dengan pasti, maka rumusnya ditunjukkan oleh Rumus 3.6.

#### j. Perancangan Sistem

Pengertian perancangan sistem menurut Jogiyanto (2001: 196) adalah sebagai penggambaran, perencanaan dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah ke dalam kesatuan yang utuh dan berfungsi. Pengertian perancangan sistem yang lain menurut Jogiyanto [3] adalah menentukan bagaimana suatu sistem akan menyelesaikan apa yang mesti diselesaikan.

#### k. Diagram Arus Data

Menurut Jogiyanto (2001: 700) diagram arus data (*Data Flow Diagram* atau DFD) adalah diagram yang menggunakan notasi-notasi untuk menggambarkan arus dari suatu sistem. DFD sering digunakan untuk menggambarkan suatu sistem yang telah ada atau sistem yang baru dikembangkan secara logika tanpa mempertimbangkan lingkungan fisik dimana data tersebut mengalir (misalnya lewat telepon, surat dan sebagainya) atau lingkungan fisik dimana data tersebut akan tersimpan (misalnya file kartu, *microhone*, *hardisk*, *tape*, *diskkette*, dan sebagainya). DFD merupakan alat yang digunakan pada metodologi pengembangan sistem yang terstruktur (*Structur Analysis and Design*).

#### Kamus Data

Menurut Jogiyanto [3], Kamus data (KD) atau *data dictionary* (DD) atau disebut juga dengan istilah *system data dictionary* adalah katalog fakta tentang data dan kebutuhan-

kebutuhan informasi dari suatu sistem informasi. Kamus data dibuat berdasarkan arus data yang ada pada diagram arus data. Arus data pada diagram arus data sifatnya global, hanya ditunjukkan nama arus datanya saja. Keterangan lebih lanjut tentang struktur dari suatu arus data pada diagram arus data secara lebih terinci dapat dilihat pada kamus data.

### **Pengujian White Box**

Teknik pengujian sistem yang digunakan pada perancangan ini dititik beratkan pada perangkat lunak atau aplikasi. Menurut Pressman [5] mengatakan bahwa “Pengujian perangkat lunak adalah elemen kritis dari jaminan kualitas perangkat lunak dan merepresentasikan kajian pokok dari spesifikasi, desain, dan pengkodean”. Roger R.Pressman [5] mengatakan bahwa Pengujian *white-box* adalah metode desain *test case* yang menggunakan struktur control desain procedural untuk memperoleh test case.

## **3. Metode Penelitian**

### **Metode Pengumpulan Data**

Dasar penelitian yang digunakan sebagai bahan untuk kelengkapan data dan informasi:

1. Penelitian keputusan (*library research*)
2. Penelitian laporan (*file reseach*)

- a. Observasi

Yaitu dengan mengamati secara langsung kegiatan diagnosa penyakit gigi yang dilakukan oleh dokter gigi untuk menarik kesimpulan bagaimana kerja seorang pakar.

- b. Wawancara

Yaitu dengan penjelasan-penjelasan dan keterangan dengan jalan mengadakan tanya jawab kepada pihak yang ada hubungannya dengan objek penelitian, dalam hal ini melakukan wawancara dengan dokter gigi.

### **Metode Pengujian Perangkat Lunak**

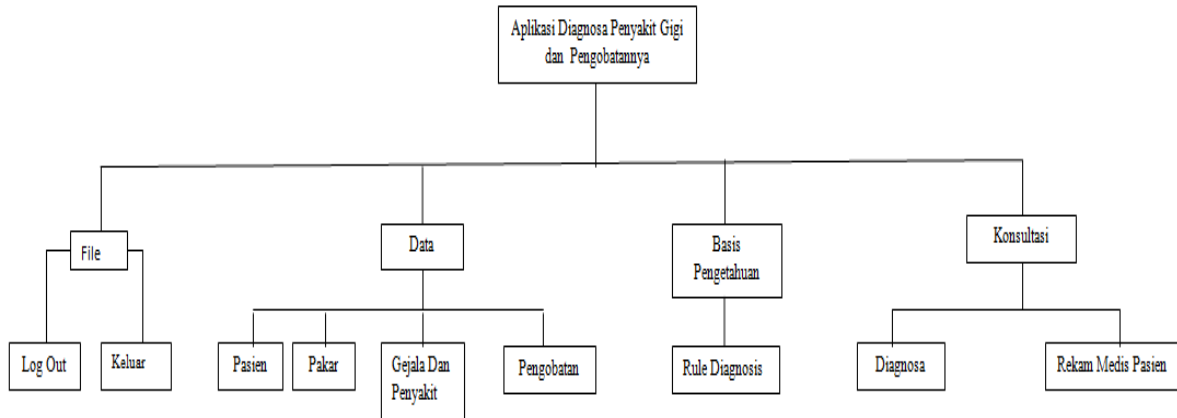
Untuk menguji program aplikasi yang dirancang, penulis menggunakan pendekatan *white box* yang merupakan salah satu metode pengujian yang menggunakan struktur *control design procedure* untuk memperoleh *test case*. Jadi, dengan menggunakan metode ini, penulis dapat mengetahui cara kerja dari aplikasi yang dirancang secara terperinci sesuai spesifikasi dan menilai apakah setiap fungsi atau prosedur yang dirancang sudah sesuai dengan baik dan benar.

Dalam merancang aplikasi diagnosa penyakit gigi dan pengobatannya menggunakan metode *forwardchaining* dan *association rule analysis*, penulis melakukan beberapa tahap penelitian, antara lain: pengumpulan data, pembuatan sistem, dan pengujian sistem.

### **Model Arsitektur Aplikasi**

Rencana/Model arsitektur aplikasi yang akan dibuat tampak seperti pada Gambar 4:





**Gambar 4: Model arsitektur aplikasi**

Pada gambar di atas terlihat bahwa aplikasi yang dibuat terdiri dari empat pilihan menu utama, yaitu:

1. File
  - a. Log Out, untuk berpindah dari user yang satu ke user yang lain.
  - b. Keluar, untuk menutup aplikasi.
2. Data
  - a. Pasien, berisi data pasien yang nantinya akan melakukan konsultasi.
  - b. Pakar, berisi data pakar dalam hal ini tenaga medis yang asli dalam bidang gigi yang nantinya akan membuat aturan diagnosis dan terapi yang akan dimasukkan ke dalam basis pengetahuan.
  - c. Gejala dan Penyakit, berisi data gejala dan penyakit gigi.
  - d. Pengobatan, berisi data pengobatan penyakit gigi yang mungkin akan diberikan kepada pasien
3. Basis Pengetahuan
  - a. *Rule diagnosis*, merupakan kumpulan aturan yang ditetapkan dari pakar ahli yang nantinya menjadi bahan acuan pengambilan keputusan.
4. Konsultasi
  - a. Diagnosa, merupakan pilihan yang disediakan aplikasi untuk pasien yang ingin melakukan konsultasi.
  - b. Rekam medis pasien, sebagai tempat penyimpanan informasi riwayat pasien yang telah berkonsultasi.

#### 4. Perancangan Sistem Dan Pengujian Sistem

##### Rancangan Program

Aplikasi diagnosa penyakit gigi dan pengobatannya dapat digolongkan sebagai aplikasi sistem tingkat satu, yaitu dengan kriteria:

1. Menggunakan *internal knowledge based* diagnosa penyakit gigi dan pengobatannya ini terintegrasi langsung dengan database dan dapat ditambahkan, dihapus dengan *rule* baru jika suatu saat *rule-rule* lama tidak menunjang lagi dengan penyakit dan pengobatan yang

sudah tersedia sebelumnya dalam aplikasi ini.

2. Dapat melakukan penambahan data dan membuat laporan data yang ada pada *knowledge base*.
3. *Inference engine* yang digunakan dengan menggunakan metode *forward chaining*.

Perancangan aplikasi ini dibagi menjadi 5 bagian utamaya itu:

1. Pengumpulan data dan Perancangan *rules*.
2. Perancangan *database*, dan Metode *Inference*
3. Perancangan *user interface*,

### **Pengumpulan Data**

Pengumpulan data mengenai jenis-jenis penyakit gigi yang ada, mulai dari gejala-gejala umum, gejala-gejala yang spesifik, dan cara pengobatannya. Pengumpulan data ini dilakukan melalui buku penunjang, referensi dari beberapa situs di internet, serta informasi yang didapat dari hasil wawancara dengan para dokter gigi di tempat penelitian dan calon dokter gigi yang nantinya akan menjadi *user* aplikasi ini.

### **Perancangan Rules**

Setelah pengumpulan data, langkah berikutnya adalah membuat *rules* atau aturan-aturan rule diagnosis. Klausa di antara *IF* dan *THEN*, berisi tentang premis dan *value* premis yang benar, sedangkan klausa sesudah *THEN* merupakan hasil kesimpulan dari *rule* tersebut.

### **Rancangan Metode Inferensi**

*Inference engine* pada Aplikasi Diagnosa Penyakit Gigi Dan Pengobatannya ini menggunakan metode *forward chaining*. Penggunaan semacam sistem query pada awal pemakaian program ditujukan untuk mengumpulkan semua premis yang memiliki keterkaitan dengan kesimpulan penyakit yang diderita oleh pasien. Setelah tahap tersebut baru kemudian mesin inferensi *forward chaining* dapat dijalankan dengan menampilkan premis tersebut sebagai pemeriksaan objektif dan subjektif dalam proses diagnosa. Inisialisasi tabel kerja dilakukan setiap memulai program atau bila akan memulai suatu sesi konsultasi. Tabel kerja yang digunakan adalah tabel *working memory*. Tabel *working memory* berfungsi untuk menyimpan semua *variable* pada suatu *rule* saat proses pengambilan keputusan.

### **Penentuan Nilai Faktor Kepastian**

Penentuan nilai faktor kepastian sangat ditentukan oleh dua faktor, yaitu:

1. Faktor kepastian dari rule yang dibuat oleh pakar yang besarnya antara -1 sampai dengan 1. Nilai faktor kepastian ini diberikan oleh pakar pada saat penginputan rule diagnosis.
2. Faktor Keyakinan Premis, merupakan faktor keyakinan yang didapat dari jawaban user pada saat proses diagnosa atau pada saat sesi tanya jawab.

### **Rancangan Implementasi**

Dalam merancang implementasi aplikasi ini perlu diperhatikan beberapa hal yaitu:

1. Pengadaan perangkat keras dan perangkat lunak sesuai dengan kebutuhan sistem.

2. Penginputan data gejala/penyakit untuk dibuatkan suatu *rule diagnosis* ke dalam *knowledge base*. Dimana penginputan ini dilakukan oleh pakar pada Rumah Sakit Umum Daya Makassar.
3. Pada layanan Rekam Medis merupakan kunci dari semua aplikasi ini, karena dapat merubah ataupun menghapus data pada aplikasi ini.

#### **A. Pengujian Basis Path**

Sebagai alternatif pengembangan *white box testing*, penulis menggunakan metode *Basis Path Testing* yang berguna untuk:

1. Mengukur kompleksitas *logic* dari desain prosedur dan menggunakannya sekaligus sebagai pedoman untuk mendapatkan konsisten dari jalur aplikasi.
2. Pengujian yang dilakukan dijamin menggunakan *statement* dalam program minimal satu kali selama pengujian.
3. Menghitung *cyclometris complexity* sebagai ukuran kognitif untuk menentukan jumlah *independent path* sebagai jalur yang perlu diuji.

#### **B. Hasil Pengujian Perangkat Lunak**

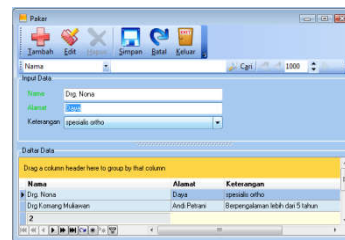
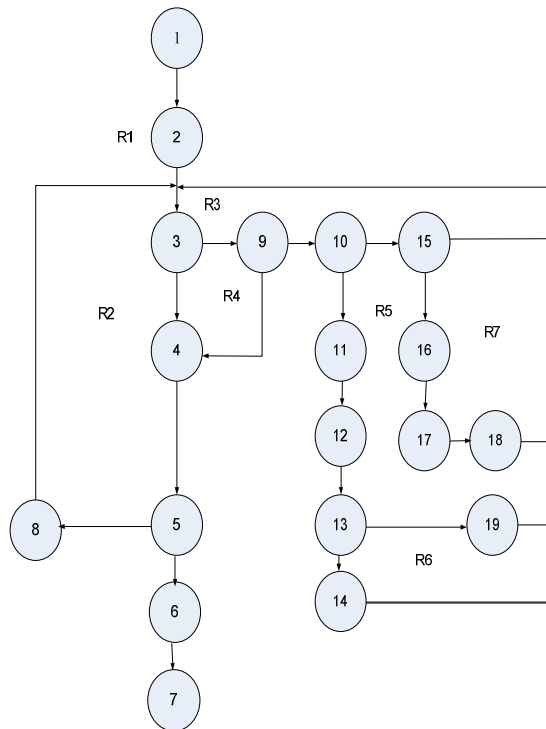
Pengujian perangkat lunak yang dilakukan pada sistem informasi akademik Politeknik Kesehatan Makassar Jurusan Farmasi sebagai berikut :

##### **1. Flowgraph Input Data Pakar**

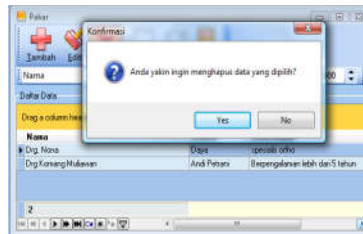
Dari *flowchart* input data pakar yang digunakan untuk pengujian perangkat lunak, maka ditentukan *flowgraph* sebagai berikut:

Notasi *Flowgraph* :

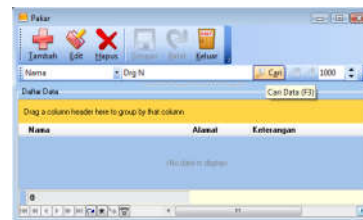
1. Mulai
2. Tampilkan data dalam grid
3. Apakah mau ditambah, jika ya input nama dan identitas pakar, apakah mau disimpan jika ya proses penyimpanan. Jika tidak proses batal dan looping, jika tidak di tambah apakah mau diedit
4. Jika ya, input perubahan data pasien apakah mau disimpan jika ya proses simpan perubahan., Jika tidak disimpan proses batal dan looping, jika tidak di edit apakah mau di cari
5. Jika ya, input indeks pencarian, apakah ditemukan jika ya tampilkan hasil pencarian dalam grid jika tidak lakukan refresh. Jika tidak di cari apakah mau di hapus
6. Jika ya tampilkan pesan penghapusan jika tidak lakukan looping
7. Selesai



Tampilan proses edit data



Tampilan pesan hapus data



Tampilan proses pencarian

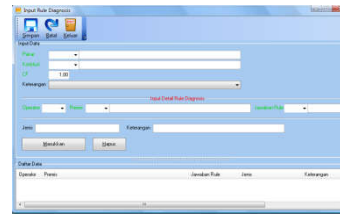
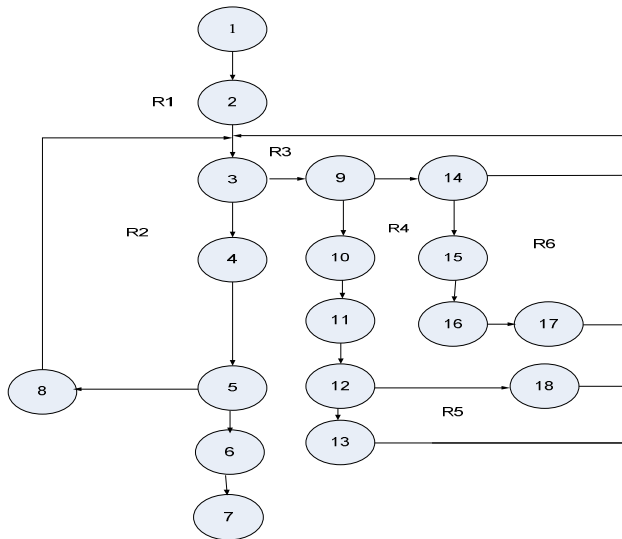
Gambar 5. Flowgraph Input Data Pakar

## 2. Flowgraph Input Data Rule Diagnosis

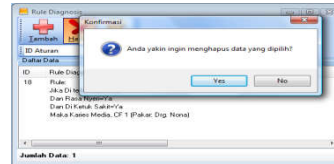
Dari *flowchart* input data rule diagnosis yang digunakan untuk pengujian perangkat lunak, maka ditentukan *flowgraph* sebagai berikut:

Notasi *Flowgraph* :

1. Mulai
2. Tampilkan data dalam grid
3. Apakah mau ditambah, jika ya input data rule diagnosis, apakah mau disimpan jika ya proses penyimpanan. Jika tidak proses batal dan looping, jika tidak di tambah apakah mau cari
4. Jika ya, input indeks pencarian, apakah ditemukan jika ya tampilkan hasil pencarian dalam grid jika tidak lakukan refresh. Jika tidak di cari apakah mau di hapus
5. Jika ya tampilkan pesan penghapusan jika tidak lakukan loopin
6. Selesai



Tampilan proses tambah data



Tampilan pesan hapus data



Tampilan cari data

**Gambar 6.** Flowgraph input rule diagnosis

Perhitungan *Cyclomatic Complexity* dari Flowgraph di atas memiliki Region = 6

1. Menghitung *Cyclomatic Complexity* dari Edge dan Node

Dengan rumus :  $V(G) = (E - N) + 2$

Dimana : E (jumlah edge pada flowgraph) = 22

N (jumlah node pada flowgraph) = 18

Penyelesaian :  $V(G) = (22-18) + 2$

$V(G) = 6$

Jadi jumlah path dari flowgraph di atas sebanyak 6 path

2. Menghitung *Cyclomatic Complexity* dari P

P adalah jumlah titik yang menyatakan logika dalam diagram alir dengan rumus  $V(G) = P + 1$  dimana P =

Penyelesaian :  $V(G) = 5 + 1$

$V(G) = 6$

3. Independent Path pada flowgraph di atas adalah :

Path 1 = 1-2-3-4-5-6-7

Path 2 = 1-2-3-4-5-8-3-4-5-6-7

Path 3 = 1-2-3-9-14-3-4-5-6-7

Path 4 = 1-2-3-9-10-11-12-4-5-6-7

Path 5 = 1-2-3-9-10-11-12-13-3-4-5-6-7

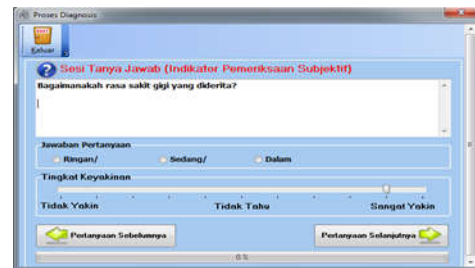
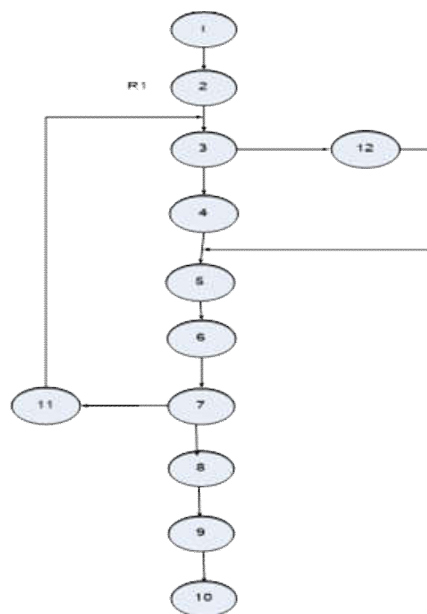
Path 6 = 1-2-3-9-14-15-16-17-3-4-5-6-7

### 3. Flowgraph Proses Diagnosa

Dari *flowchart* proses diagnosa yang digunakan untuk pengujian perangkat lunak, maka ditentukan *flowgraph* sebagai berikut:

Notasi *Flowgraph* :

1. Mulai
2. Tampilkan pertanyaan dalam memo
3. Apakah bukan pertanyaan awal dan ingin kembali ke pertanyaan sebelumnya?
4. Jika ya, ke pertanyaan sebelumnya, jika tidak maka ke pertanyaan selanjutnya.
5. Apakah pertanyaan telah habis?, jika ya jalan kan search engine, jika tidak kembali ke nomor 3
6. Selesai



Tampilan sesi pertanyaan



Tampilan jalankan search

**Gambar 7.** *Flowgraph* Proses diagnosa

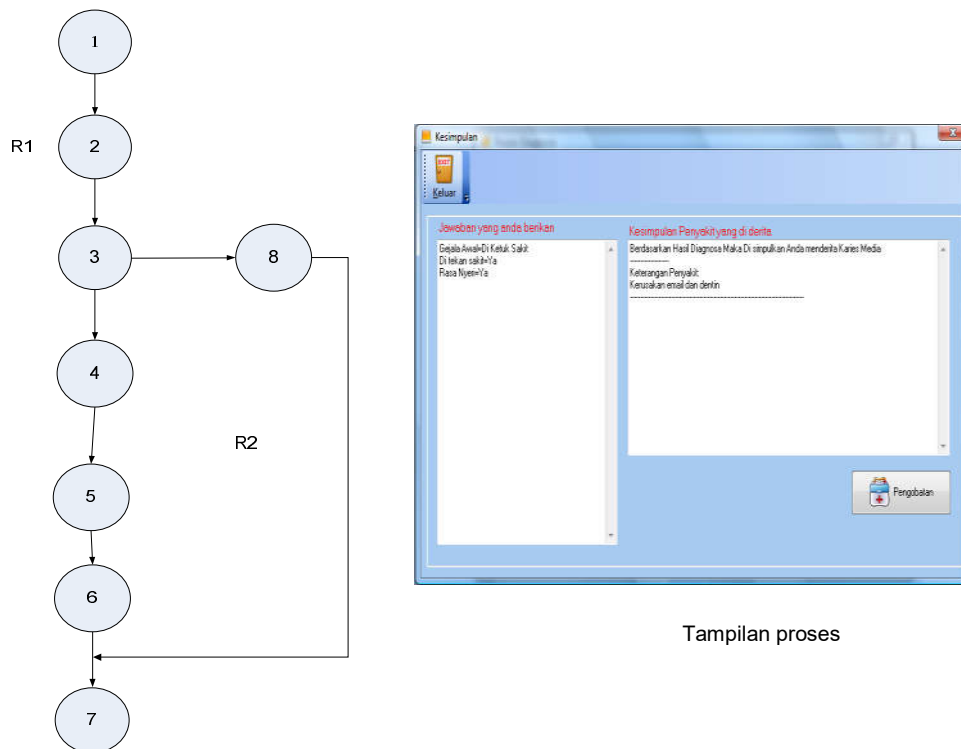
### 4. Flowgraph Proses Kesimpulan

Dari *flowchart* proses diagnosa yang digunakan untuk pengujian perangkat lunak, maka ditentukan *flowgraph* sebagai berikut:

Notasi *Flowgraph* :

1. Mulai
2. Tampilkan kembali pertanyaan dan jawaban sebelumnya
3. Apakah ada kesimpulan?
4. Jika ya, proses kesimpulan dan tampilkn pada memo kesimpulan. Jika tidak maka ke nomor 6
5. Jika terdapat kesimpulan aktifkan button pengobatan dan munculkan form pengobatan, jika tidak button pengobatan *disable*

## 6. Selesai



**Gambar 8.** Flowgraph proses kesimpulan

Perhitungan *Cyclomatic Complexity* dari *Flowgraph* di atas memiliki *Region* = 2

1. Menghitung *Cyclomatic Complexity* dari *Edge* dan *Node*

Dengan rumus :  $V(G) = (E - N) + 2$

Dimana : E (jumlah edge pada flowgraph) = 8

N (jumlah node pada flowgraph) = 8

Penyelesaian :  $V(G) = (8-8) + 2$

$V(G) = 2$

Jadi jumlah *path* dari *flowgraph* di atas sebanyak 2 *path*

2. Menghitung *Cyclomatic Complexity* dari P

P adalah jumlah titik yang menyatakan logika dalam diagram alir dengan rumus  $V(G) = P + 1$  dimana P = 1

Penyelesaian :  $V(G) = 1 + 1$

$V(G) = 2$

3. *Independent Path* pada *flowgraph* di atas adalah :

Path 1 = 1-2-3-4-5-6-7

Path 2 = 1-2-3-8-7

## Tabel Pengujian Perangkat Lunak

Tabel 2. Hasil Pengujian White Box

NO	Flowgraph	Independent Path	Region	Kompleksitas Siklomatis
1	Setting Database	3	3	3
2	User Login	2	2	2
3	Menu Utama	7	7	7
4	Menu Input Data	5	5	5
5	Menu Basis Pengetahuan	2	2	2
6	Menu Konsultasi	3	3	3
7	Menu Laporan	7	7	7
8	Menu Tools	8	8	8
9	Menu Bantuan	3	3	3
10	Input Data Pasien	7	7	7
11	Input Data Pakar	7	7	7
12	Input Data Gejala Dan Penyakit	7	7	7
13	Input Data Pengobatan	7	7	7
14	Input Data Rule Diagnosis	6	6	6
15	Proses Diagnosa Awal	3	3	3
16	Proses Diagnosa	3	3	3
17	Proses Kesimpulan	3	3	3
18	Proses Pengobatan	2	2	2
19	Rekam Medis Pasien	2	2	2
20	Cetak Data Pasien	2	2	2
21	Cetak Data Pakar	2	2	2
22	Cetak Data Gejala dan Penyakit	2	2	2
23	Cetak Data Pengobatan	2	2	2
24	Cetak Data Rule Diagnosis	2	2	2
25	Cetak Data Rekam Medis Pasien	3	3	3

Berdasarkan hasil pengujian perangkat lunak yang terdapat pada Tabel 2, maka sistem dikatakan sudah terbukti benar, karena *Cyclomatic Complexity*, *Region* dan *Independent Path* adalah sama.

Jadi kesimpulannya, hasil dari pengujian yang telah dilakukan telah membuktikan bahwa program telah bebas dari logika kesalahan.

## 5. Penutup

Pada bab akhir penulis memberikan kesimpulan dan saran yang berhubungan dengan pelaksanaan dan hasil penelitian yang telah dilakukan.

### Kesimpulan

1. Menggunakan metode *forward chaining* dan *association rule analysis*, dapat diperoleh informasi mengenai gejala-gejala yang menjadi penyebab penyakit gigi beserta penanganannya berupa tindakan atau terapi.
2. Dengan adanya perangkat lunak berupa aplikasi diagnosa penyakit gigi, dapat mendiagnosa penyakit gigi yang memiliki cara kerja mirip dengan dokter gigi dalam proses dianosa penyakit gigi dan mampu memberikan solusi cerdas berupa cara-cara



penanganan dan tindakan yang harus dilakukan terhadap suatu penyakit yang diderita berdasarkan basis pengetahuan yang tersimpan dalam *knowledge base*.

### Saran

Agar memperoleh hasil yang lebih baik dalam pengembangan dunia teknologi yang semakin pesat saat ini maka penulis memberikan saran sebagai berikut:

1. Perlu dilakukan survei yang lebih mendalam, dan penyempurnaan dari rule-rule yang telah ada, sehingga aplikasi ini benar-benar menyerupai hasil pemikiran seorang ahli dalam bidang gigi.
2. Diharapkan software aplikasi ini ke depannya dikembangkan dalam bentuk web, sehingga semakin memperkaya basis pengetahuan program aplikasi karena program ini menerima input dari pengguna berupa gejala-gejala baru yang mungkin muncul dan dapat digunakan oleh khalayak umum.
3. Dengan dibangunnya aplikasi ini, diharapkan bidang teknologi komputer mampu mengambil peranan yang besar dalam bidang medis, khususnya bidang kesehatan gigi.

### 6. Referensi

- [1] Arhami, Muhammad. 2005, "*Konsep Dasar Sistem Pakar*", Andi Offset: Yogyakarta.
- [2] Desiani, Anita dan Arhami, Muhammad. 2006, "*Konsep Kecerdasan Buatan*", Andi Offset: Yogyakarta.
- [3] Jogiyanto. 2001, *Analisis dan Desain Sistem Informasi Pendekatan Terstruktur*, Edisi Kedua Cetakan Keempat, Andi Offset: Yogyakarta.
- [4] Kusrini. 2008, "*Aplikasi Sistem Pakar*", Andi Offset: Yogyakarta.
- [5] Pressman, Roger S. 2002, *Rekayasa Perangkat Lunak Pendekatan Praktisi (Buku Satu)*, Andi Offset: Yogyakarta.
- [6] Tim DINASTINDO. 1993, "*Kamus Komputer Eksekutif*", DINASTINDO: Jakarta.
- [7] <http://lukas21.wordpress.com/pengertian-penyakit/>
- [8] <http://nugroz.com/pemrograman/sistem-pakar/sedikit-teori-sistem-pakar/>
- [9] <http://www.dianprima.com/diagnosa-kesulitan-belajar.jsp>
- [10] [http://id.wikipedia.org/wiki/Aplikasi\\_komputer](http://id.wikipedia.org/wiki/Aplikasi_komputer)<http://kesehatangigi.blogspot.com/2008/01/pengertian-sakit-gigi.html>
- [11] <http://yayanakhyar.files.wordpress.com/2009/02/files-of-drsmed-tutorial-gimul.pdf>