

DIAGNOSA PENYAKIT GIGI DAN MULUT DENGAN MENGGUNAKAN PENDEKATAN SISTEM PAKAR

Khodijah Hulliyah, Fitri Mintarsih, Budi Kurniawan

budi.kurniawan51@gmail.com, khodijah2@yahoo.com, mintarsihfr@gmail.com

0813 16 000 190, 0812 894 7537, 0815 130 335 03

Jurusan Informatika, Universitas Islam Negeri, Jakarta.

ABSTRAK

Komputer pada era globalisasi saat ini menjadi kebutuhan utama dalam menunjang kerja-kerja manusia. Salah satu cabang ilmu komputer yang banyak dimanfaatkan oleh manusia untuk membantu kerjanya adalah pembentukan sistem pakar yang merupakan salah satu sub bidang ilmu kecerdasan buatan [1]. Salah satu pemanfaatan sistem pakar adalah dalam bidang kedokteran gigi. Terbukti dengan munculnya penelitian **I Nyoman Kusuma Wardana** bertajuk perancangan sistem pakar untuk diagnosa penyakit mulut dan gigi menggunakan bahasa pemrograman CLIPS yang dipublikasikan pada *Seminar Nasional Aplikasi Teknologi Informasi* yang diadakan di Universitas Gajah Mada, Yogyakarta 21 Juni 2008. Aplikasi ini tampil dengan interface berupa pertanyaan-pertanyaan tertutup tentang gejala yang dirasakan oleh *user* sehingga tidak memaksimalkan hasil diagnosa yang didapat. Sistem pakar ini juga menjadi pengembangan dari penelitian **Bambang Suyono** (Seminar Nasional Informatika UPN Yogyakarta 22 Mei 2010) yang memiliki kekurangan pada penggunaan metode penelusuran *depth first search* sehingga tidak mampu menampilkan dua atau beberapa solusi [3], padahal dalam mendiagnosa penyakit terkadang seorang dokter menetapkan diagnosa banding. Hal inilah yang mendorong penulis mengembangkan sistem aplikasi tersebut dengan memperbaiki segala kekurangan yang ada. Metode yang digunakan penelitian ini adalah *Extreme Programming (XP)* yang merupakan bagian dari metode AGILE [7], terdiri dari lima tahap yaitu, *Planning, Design, Coding, Test dan release* [8]. Penelitian ini menggunakan *software* PHP versi 5.3.5 sebagai bahasa pemrograman, MYSQL versi 5.0.7 sebagai *database* dan keduanya terdapat dalam satu paket *localhost* XAMPP versi 1.7.4, sementara *design* aplikasi ini menggunakan Microsoft Office Front Page 2003. Sistem operasi yang digunakan penulis adalah Microsoft Windows Vista dan *Personal Computer (PC)* dengan spesifikasi, prosesor Intel Pentium Dual Core, RAM 1 GB dan Hard disk 160 GB. Kelebihan sistem pakar ini dari sistem sebelumnya adalah cakupan diagnosa penyakit gigi dan mulut yang lebih banyak, tampilan lebih interaktif serta mampu menentukan diagnosa banding dari kedekatan gejala yang menyertai penyakit gigi dan mulut.

Kata kunci: *Extreme Programming (XP), Sistem Pakar, Penyakit Gigi dan Mulut*

1. PENDAHULUAN

Komputer pada era globalisasi saat ini menjadi kebutuhan utama dalam menunjang kerja-kerja manusia. Peran komputer kini pun menjadi lebih meluas, tidak hanya menjadi alat bantu hitung (seperti penggunaan awal komputer [5] tapi juga menjadi alat bantu penyelesaian masalah-masalah yang dihadapi manusia. Salah satu cabang ilmu komputer yang banyak dimanfaatkan oleh manusia untuk membantu kerjanya adalah pembentukan sistem pakar yang merupakan salah satu sub bidang ilmu kecerdasan buatan [1]. Kecerdasan buatan (*Artificial Intelligence*) adalah kegiatan menyediakan mesin seperti komputer dengan kemampuan untuk menampilkan

perilaku yang dianggap cerdas jika diamati oleh manusia.

Salah satu pemanfaatan sistem pakar adalah dalam bidang kedokteran atau kesehatan. Terbukti dengan munculnya penelitian **I Nyoman Kusuma Wardana** bertajuk perancangan sistem pakar untuk diagnosa penyakit mulut dan gigi menggunakan bahasa pemrograman CLIPS yang dipublikasikan pada *Seminar Nasional Aplikasi Teknologi Informasi* yang diadakan di Universitas Gajah Mada, Yogyakarta 21 Juni 2008. Penelitian tersebut menghasilkan sistem pakar dengan 18 diagnosa penyakit mulut dan gigi dengan 40 gejala yang menyertainya. Penelitian tersebut memiliki kelemahan, karena ditampilkan berupa

pertanyaan tertutup berkait dengan gejala yang dirasakan sehingga tidak mampu mencakup keseluruhan diagnosa. Pertanyaan tertutup artinya, hanya memunculkan pertanyaan yang menyudut pada satu bagian penyelesaian atau diagnosa yang tidak memungkinkan hasil diagnosa lain diproses. Pertanyaan-pertanyaan tersebut muncul dengan kombinasi berbeda saat dijalankan sehingga lagi-lagi tidak memfasilitasi penanganan dari keseluruhan diagnosa. Kekurangan yang lain terdapat pada penggunaan teknik penelusuran *depth first search* sehingga tidak mampu menampilkan dua atau beberapa solusi [3], padahal dalam diagnosa penyakit terkadang seorang dokter menetapkan diagnosa banding. Kekurangan terakhir adalah data yang digunakan sebagai parameter diagnosa belum merupakan data diagnosa penyakit yang merepresentasikan penyakit di daerah tropis khususnya Indonesia.

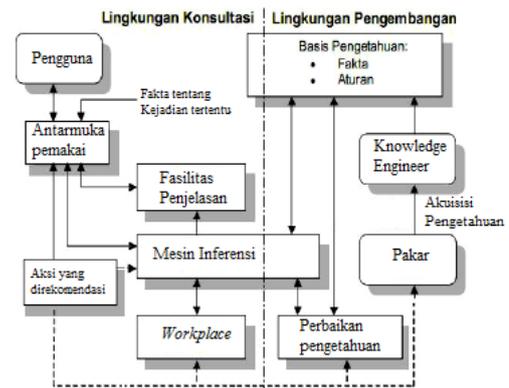
Oleh karena itu, dibutuhkan pengembangan dari sistem sebelumnya dan penelitian ini menjadi pengembangan dari sistem tersebut. Metode yang digunakan penelitian ini adalah *Extreme Programming* (XP) yang merupakan bagian dari metode AGILE [7], terdiri dari lima tahap yaitu, *Planning, Design, Coding, Test dan release* [8]. Penelitian ini menggunakan *software PHP* versi 5.3.5 sebagai bahasa pemrograman, *MYSQL* versi 5.0.7 sebagai *database* dan keduanya terdapat dalam satu paket *localhost XAMPP* versi 1.7.4, sementara *design* aplikasi ini menggunakan *Microsoft Office Front Page* 2003.

2. LANDASAN TEORI

2.1 Sistem Pakar

Sistem pakar (*expert sistem*) merupakan paket perangkat lunak atau paket program komputer yang ditujukan sebagai penyedia nasihat dan sarana bantu dalam memecahkan masalah dalam bidang-bidang spesialisasi tertentu seperti sains, perkerjasama matematika, kedokteran, pendidikan dan sebagainya[1]. Komputer yang demikian dapat dijadikan seperti konsultan atau tenaga ahli di bidangnya [6]. Sistem pakar sendiri merupakan program *artificial intelligence* yang menggabungkan pangkalan pengetahuan (*knowledge base*) dengan sistem inferensi. Ini merupakan bagian *software* spesialisasi tingkat tinggi yang berusaha menduplikasi fungsi seorang pakar dalam suatu bidang keahlian [9].

Sistem pakar tersusun dari dua bagian utama, yaitu lingkungan pengembangan (*development environment*) dan lingkungan konsultasi (*consultation environment*) [1]. Lingkungan pengembangan berisi komponen-komponen yang digunakan untuk memasukkan pengetahuan pakar kedalam lingkungan sistem pakar, sedangkan lingkungan konsultasi berisi komponen yang akan digunakan oleh *user* dalam memperoleh pengetahuan pakar.



Gambar 1. Arsitektur Sistem Pakar

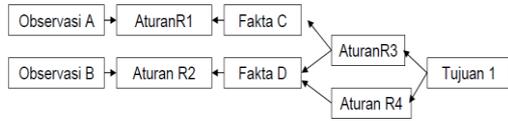
Hal terpenting dari sistem pakar adalah mesin inferensi yang menjadi otak dari sistem pakar itu sendiri. Mesin inferensi adalah program komputer yang memberikan metodologi untuk penalaran tentang informasi yang ada dalam basis pengetahuan dan dalam *workplace*, dan untuk memformulasikan kesimpulan [1]. Selama proses konsultasi antar sistem dan pemakai, mekanisme inferensi menguji aturan satu demi satu sampai kondisi aturan itu benar sampai [6]. Kesimpulannya mesin inferensi berisi pola penalaran dan penelusuran masalah yang ada sehingga menemukan kesimpulan.

2.2 Pola Penalaran

Secara umum ada dua teknik utama yang digunakan dalam mekanisme inferensi untuk pengujian aturan, yaitu penalaran maju (*forward chaining*) dan penalaran mundur (*backward chaining*).

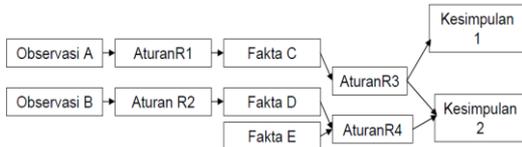
Pelacakan atau penalaran kebelakang (*backward chaining*) adalah pendekatan yang dimotori tujuan (*goal-driven*) [1]. Penalaran ini juga biasa disebut penalaran dari atas ke bawah [1] yaitu penalaran yang dimulai dari level tertinggi membangun suatu hipotesis,

turun ke fakta level paling bawah yang dapat mendukung hipotesis.



Gambar 2. Pola Bacward Chaining

Pelacakan atau penalaran kedepan (*forward chaining*) adalah metode pencarian atau penarikan kesimpulan yang berdasarkan pada data atau fakta yang ada menuju ke kesimpulan, penelusuran dimulai dari fakta yang ada lalu bergerak maju melalui premis-premis untuk menuju kesimpulan atau dapat dikatakan bottom up reasoning.

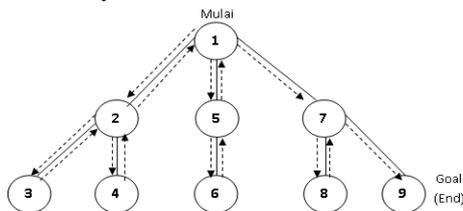


Gambar 3. Pola Forward Chaining

2.3 Pola Penelusuran

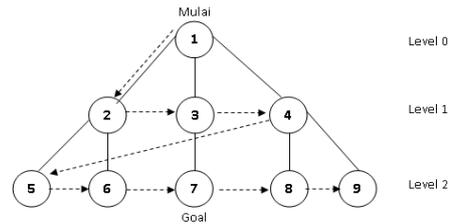
Dua pola pencarian di atas akan bergantung pula pada pola penelusuran yang diterapkan. Ada tiga jenis pola penelusuran yang dapat digunakan yaitu *depth-first search*, *breadth-first search* dan *best first search*.

Depth-first search adalah teknik penelusuran data pada node-node secara vertikal dan sudah terdefiniskan, misalnya dari kiri ke kanan [6].



Gambar 4. Pola Penelusuran *Depth-first search*

Breadth-first search adalah teknik penelusuran data pada semua node dalam satu level atau satu tingkatan sebelum ke level atau tingkatan dibawahnya [6].



Gambar 5. Pola Penelusuran *Breadth-first search*

Best-first search adalah penelusuran yang menggunakan pengetahuan akan suatu masalah untuk melakukan panduan pencarian kearah node tempat dimana solusi berada. Pencarian jenis ini dikenal juga sebagai heuristik [6]. Dapat dikatakan pula *best-first search* bekerja berdasarkan kombinasi kedua metode sebelumnya[1].

2.4 XAMPP

XAMPP awalnya dikembangkan dari sebuah tim proyek yang bernama Apache Friends. XAMPP ini merupakan sebuah *software* yang mendukung banyak sistem operasi dan merupakan kompilasi dari beberapa program. Fungsi dari XAMPP adalah sebagai server yang berdiri sendiri (*localhost*) yang terdiri atas program Apache, my SQL, PHP, dan Perl.

Nama aplikasi ini sendiri merupakan sebuah singkatan dari X (empat sistem operasi), Apache, MySQL, PHP, dan Perl.

2.5 PHP

PHP (Personal Home Page) Pertama kali ditemukan tahun 1995 oleh seorang Software Developer bernama Rasmus Lerdorf [4]. PHP hadir sebagai usaha mendinamisasikan halaman website. Hal inilah yang membedakan PHP dengan HTML. HTML merupakan bahasa statis yang apabila kita ingin merubah konten/isinya maka yang harus dilakukan pertama kali nya adalah, membuka file-nya terlebih dahulu, kemudian menambahkan isi kedalam file tersebut [4]. Sementara, jika menggunakan PHP, hanya perlu merubah bagian yang dituju saja.

2.6 MYSQL

Berbicara mengenai mysql, tidak lepas dari berbicara mengenai sql (*structured query language*) yakni

bahasa yang berisi perintah-perintah untuk memanipulasi database, mulai dari melakukan perintah *select* untuk menampilkan isi database, *insert* atau menambahkan isi kedalam *database*, *delete* atau menghapus isi *database* dan *edit database* [4].

MySQL berawal dari proyek yang dimulai oleh kedua orang *developer*, yakni Michael Widenius dan David Axmark di tahun 1994. Proyek ini didasari karena ingin membuat suatu sistem database yang murah, meskipun ketika itu ada *database* yang *powerfull* yakni oracle, namun database ini bersifat komersil yang harganya super mahal, dan begitu menguasai pasar.

3. METODOLOGI

Pertama yang dilakukan dalam penelitian ini adalah mengumpulkan data-data yang berkaitan dengan penyakit gigi dan mulut yang sering muncul di Indonesia yang berisi tentang diagnosa dan gejala yang menyertainya. Pengumpulan data ini dilakukan melalui wawancara dan studi literatur. Hasil pengumpulan data ini dapat dilihat pada tabel-tabel berikut.

Tabel 1. Data Diagnosa Penyakit Gigi dan Mulut

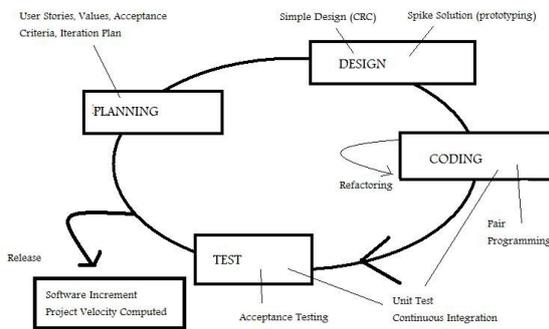
No	Diagnosa
1	Abses periodontal
2	Bau mulut (Halitosis)
3	Cheilitis
4	Eritema Multiformis
5	Gingivitis
6	Herpes simpleks (Primary Herpetic Gingivostomatitis)
7	Kalkulus
8	Kandidiasis
9	Kanker
10	Karies Media
11	Karies profunda
12	Karies Superfisial
13	Kelainan Sendi Temporomandibuler
14	Liken planus
15	Masalah kelenjar ludah
16	Nekrosis pulpa
17	Periodontitis
18	Pulpitis akut
19	Pulpitis kronis
20	Sariawan atau Stomatitis Aphthosa recurrent (SAR/ sariawan berulang)

Tabel 2. Data Gejala yang Mengiringi Diagnosa yang Ada

Gejala	
1.	Bau mulut tak sedap
2.	Benjolan putih pada bagian dalam bibir
3.	Bercak kecoklatan mendatar pada bibir
4.	Bercak putih berlendir pada mulut
5.	Bercak putih pada lidah
6.	Bercak putih pada rongga mulut
7.	Bibir kemerahan
8.	Bibir pecah-pecah
9.	Bibir teras kering dan keras
10.	Bibir terasa panas seperti terbakar
11.	Bintik putih pada gigi
12.	Bintik-bintik merah bersisik pada daerah mulut
13.	Demam
14.	Dentin terlihat
15.	Disertai rasa gatal
16.	Disertai sakit kepala
17.	Gigi berlubang
18.	Gigi keluar darah
19.	Gigi nyeri saat terkena rangsangan (panas atau dingin)
20.	Nyeri pada TMJ (Sendi Temporo Mandibuler)
21.	Gusi bengkak
22.	Gusi licin dan mengkilap
23.	Gusi merah muda
24.	Gusi mudah berdarah
25.	Infeksi pada kelenjar ludah
26.	Kemerahan pada sudut-sudut mulut
27.	Lubang sangat besar pada gigi
28.	Luka mudah berdarah
29.	Luka terbuka berwarna biru keputihan
30.	Luka terbuka pada lidah
31.	Muncul benjolan kemerahan pada lubang gigi
32.	Nanah pada pangkal gusi
33.	Nyeri pada kelenjar ludah
34.	Nyeri pada otot pengunyah
35.	Nyeri saat berbaring
36.	Nyeri saat gigi tertekan makanan
37.	Nyeri saat mengunyah
38.	Pembengkakan kelenjar getah bening
39.	Pembengkakan pada kelenjar ludah
40.	Pembusukan gigi
41.	Pulpa mati rasa
42.	Pulpa terinfeksi / radang pada pulpa
43.	Resesi gusi
44.	Sakit berdenyut tanpa rangsangan
45.	Sudut mulut terasa nyeri
46.	Sudut-sudut mulut bersisik
47.	Terasa perih saat makan dan minum
48.	Terbentuk kantong antara gigi dan gusi
49.	Terdapat benjolan ungu dengan pinggiran bersisik pada daerah mulut
50.	Terdapat endapan plak
51.	Terdapat ulkus (luka) pada mukosa mulut
52.	Terjadi daerah kemerahan pada mulut

- 53. Ulkus (luka) pada sudut mulut
- 54. Vesikel (gelembung) pada mukosa mulut
- 55. Terdapat karang gigi
- 56. Perkembangan penyakit cepat (bulanan)
- 57. Ruang pulpa terbuka
- 58. Luka atau leci atau benjolan yang muncul berulang

Sementara itu, proses pengembangan sistem menggunakan metodologi Extreme Programming (XP). Metodologi XP memiliki lima tahap pengembangan [8] seperti terlihat pada gambar berikut;



Gambar 6. Siklus XP dengan pola sederhana [8]

Penjelasan tahap-tahapnya ada pada uraian berikut:

3.1. Planning

Bagian ini melakukan pengumpulan informasi kemudian menyeleksi (yang dibantu oleh *user* ahli atau pakar) sehingga sesuai dengan kebutuhan aplikasi. Pada penelitian ini pakar yang dimaksud adalah dokter gigi.

3.2. Design

Aktivitas ini memiliki prinsip dasar sederhana yaitu menggunakan kartu CRC (Class-Responsibility-Collaborator) untuk identifikasi dan pengaturan kelas-kelas dalam konsep object oriented. Jika dalam prosesnya menemui kesulitan maka dilakukan langkah spike solution yaitu pembuatan *prototype*.

3.3. Coding

Aktivitas utama adalah refactoring yaitu pengembangan *design* menuju implementasi. Pair program yang

menjadi bagian dari fase ini, dilakukan untuk *real time program solving* dan *real time quality assurance*. Sebelum *coding* dilakukan siapkan unit *test* yang akan digunakan pada fase selanjutnya (*test*).

3.4. Test

Fase ini merupakan fase pengujian hasil dari aplikasi yang telah dibangun. Pengujian yang dilakukan mengacu pada unit *test* yang telah disusun mulai dari awal fase *coding*.

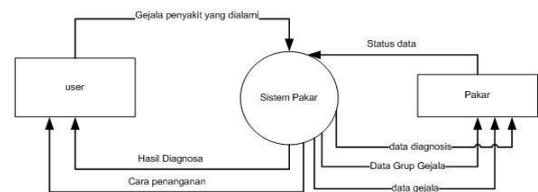
3.5. Release

Fase ini merupakan tahap dimana sudah tidak ada lagi stories (data) dari *user* yang diimplementasikan sehingga semua kebutuhan sistem dianggap telah terpenuhi. Tidak ada lagi perubahan arsitektur, *design* atau pelaksanaan *coding* [2].

4. PEMBAHASAN

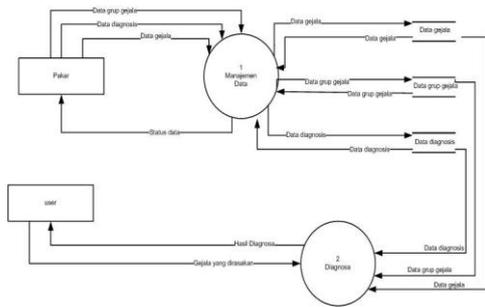
4.1 Data Flow Diagram (DFD)

Data pada sistem ini berjalan sesuai dengan rancangan DFD (Data Flow Diagram) yang sudah disusun. DFD yang ada terdiri dari diagram konteks, diagram level 1 dan diagram level 2. Diagram konteks sistem pakar ini pada gambar 7 berikut ini



Gambar 7. Diagram Konteks

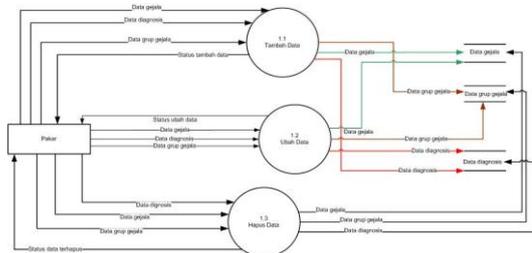
Dari diagram konteks di atas, akan dijelaskan ke dalam diagram level 1 pada gambar 8 berikut ini.



Gambar 8. Diagram level 1

Diagram level 1 di atas menggambarkan bahwa dalam sistem pakar ini terdapat dua proses yaitu proses 1 manajemen basis data dan proses 2 diagnosa. Setiap proses diperjelas dengan diagram rincian kecuali, proses 2 karena user tidak memanipulasi data hanya menerima hasil olah data yang telah dibatasi oleh sistem pakar ini.

Proses 1 manajemen basis data terdiri dari tiga bagian proses yaitu proses 1.1 tambah data, proses 1.2 ubah data, dan proses 1.3 hapus data. Semua proses ini dapat dilihat pada gambar 9.



Gambar 9. Diagram Level 2

4.2 Basis Data

Penyusunan basis data sistem pakar ini menggunakan teknik ERD (Entity Relational Diagram). Langkah yang dilakukan adalah

1. menentukan entitas dengan atributnya beserta primary keynya,
2. menggambarkan kardinalitas dari diagram ER
3. membentuk skema basis data berdasarkan relasi yang terlihat pada poin 2 (dua)
4. membentuk tabel-tabel berdasar primary key yang terpilih. [10]

Pada sistem pakar ini, basis data yang ada disusun dengan langkah seperti uraian tersebut di atas.

4.2.1. Menentukan Entitas dengan Atributnya

Entitas yang ada pada sistem pakar ini hanya tiga buah yaitu diagnosa, gejala dan grup gejala. Atribut awal yang ditentukan terlihat pada tabel 3. berikut.

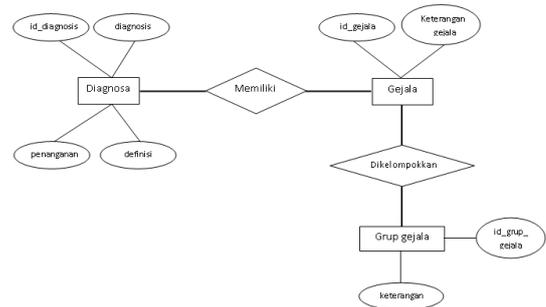
Entitas	Atribut
Diagnosa	- <u>Id diagnosis</u> - Diagnosis - Definisi - Penanganan
Gejala	- <u>Id gejala</u> - Keterangan gejala
Grup gejala	- <u>Id grup gejala</u> - Keterangan

Tabel 3. Entitas dan Atribut Awal Sebelum diolah

Seperti terlihat pada tabe 4.1 bahwa entitas Diagnosa memiliki atribut id_diagnosis (*primary key*), diagnosis, definisi dan penanganan. Entitas gejala memiliki atribut id_gejala (*primary key*) dan keterangan_gejala. Entitas Grup Gejala memiliki atribut id_grup_gejala (*primary key*) dan keterangan. Data inilah yang selanjutnya akan diproses pada tahap pembentukan ERD.

4.2.2. Menyusun ERD

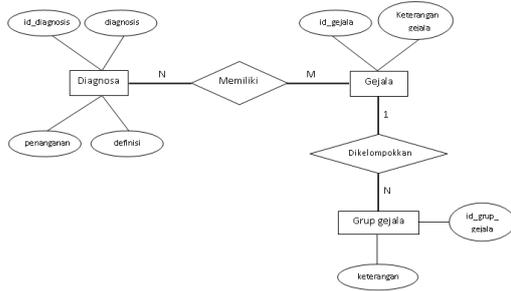
Langkah pertama pada pembentukan ERD adalah memasukkan data entitas dan atribut serta menentukan relasi antar entitas. Entitas dengan atribut serta relasi antar entitas ditunjukkan pada gambar 10 berikut ini,



Gambar 10 ERD dengan atribut tiap entitas

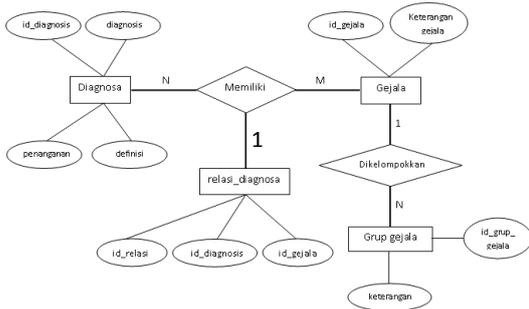
Langkah selanjutnya adalah menentukan kardinalitas. Berdasar keterangan di atas bahwa tiap gejala dapat merepresentasikan beberapa diagnosa penyakit, tiap diagnosa penyakit memiliki beberapa gejala dan tiap grup gejala memiliki

beberapa gejala maka, nilai kardinalitas yang muncul ditunjukkan pada gambar 11 berikut,



Gambar 11 ERD sementara dengan keterangan kardinalitas

Pada gambar 11 terlihat kardinalitas banyak ke banyak dari relasi antara entitas diagnosa dengan entitas gejala, sehingga menghasilkan entitas tambahan dengan komponen atribut berisi atribut kunci dari kedua entitas tersebut (diagnosa dan gejala). Gambar 12 menunjukkan hasil perubahan ERD,



Gambar 12 ERD lengkap

4.2.3. Struktur Basis Data

Tabel 4 Struktur Tabel Diagnosa

Nama Field	Data Type	File Size	Keterangan
id_diagnosis	Int	5	Primary Key
diagnosis	Varchar	100	Diagnosa
definisi	Varchar	100	Definisi diagnosa yang ada
penanganan	Varchar	150	Cara penanganan

Tabel 5 Struktur Tabel Gejala

Nama Field	Data Type	File Size	Keterangan
id_gejala	Int	5	Primary Key
keterangan_gejala	varchar	100	Gejala
id_group	int	5	Foreign key

Tabel 6 Struktur Tabel Grup Gejala

Nama Field	Data Type	File Size	Keterangan
id_grup_gejala	Int	5	Primary Key
keterangan_gejala	varchar	100	Gejala

Tabel 4.5 Struktur Tabel Relasi Diagnosa

Nama Field	Data Type	File Size	Keterangan
id_relsa	Int	5	Primary Key
id_diagnosis	Int	5	Primary Key
id_gejala	Int	5	Primary Key

4.3 Implementasi

Tampilan interface dari sistem pakar diagnosa penyakit gigi dan mulut ini terlihat pada gambar 12 berikut ini.



Gambar 12. Halaman Utama atau Home

Pada menu utama tersebut kita dapat melihat tersedia beberapa pilihan halaman yaitu; Home, Diagnosa, Edit, Help dan About.

Halaman Home merupakan halaman utama atau pembuka. Halaman diagnosa adalah halaman dimana user melakukan proses diagnosa penyakit dengan memilih gejala yang tersedia. Halaman edit adalah halaman dimana pakar atau admin memanipulasi knowledge base dari sistem pakar ini. Halaman help berisi tentang keterangan bantuan dari pengaplikasian sitem pakar ini. Halaman about us berisi tentang biodata singkat penulis.

Halaman inti bagi user adalah halaman diagnosa. Tampilannya terlihat seperti gambar 13.



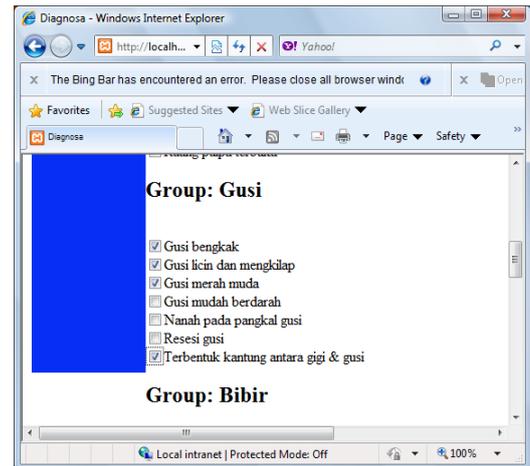
Gambar 13. Halaman Diagnosa

Menu diagnosa (gambar 13) merupakan halaman memproses hasil *inputan* gejala dari *user*. Menu diagnosa di desain dengan menampilkan gejala dalam lima kelompok yaitu, group gigi, group gusi, group bibir, group lidah dan mulut, dan group lain-lain. Kelompok ini dimaksudkan agar tidak membingungkan *user* dalam proses memilih gejala. Pada bagian bawah halaman ini terdapat dua tombol bantu yaitu tombol proses dan tombol reset. Tombol proses untuk menampilkan hasil diagnosa sedangkan tombol reset digunakan untuk mengosongkan kembali atau membatalkan semua pilihan sebelumnya.

Kita ambil contoh, jika user memilih gejala misalnya;

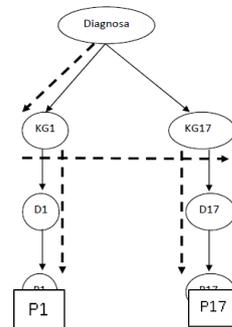
- Gejala 1: gusi bengkak,**
- Gejala 2: gusi licin dan mengkilap,**
- Gejala 3: gusi merah muda**
- Gejala 4: Terbentuk kantong antara gigi dan gusi**

Seperti terlihat pada gambar 14.



Gambar14. Pemilihan Diagnosa

Setelah itu *user* menekan tombol proses pada bagian bawah tampilan diagnosa. Saat tombol ini ditekan maka, program akan mengecek data yang telah di *input* oleh *user* dengan *database* atau *knowledge base* yang ada. Dengan penelusuran *best first search*, maka akan ditemukan dua diagnosa yaitu, abses periodontal dan periodontitis. Pohon penelusurannya diperlihatkan pada gambar 15.



Gambar15. Pohon Penelusuran Best First Search

Tampilan hasil dari pencarian diagnosa tersebut ditampilkan pada gambar 16.



Gambar 16. Tampilan hasil diagnosa

5. KESIMPULAN

Kesimpulan yang dapat ditarik pada penelitian ini adalah:

1. Pembuatan aplikasi dengan konsep sistem pakar yang menggunakan *interface* web dalam pendiagnosaan penyakit gigi dan mulut dapat membantu pendiagnosaan awal penyakit yang berkaitan dengan gigi dan mulut.
2. Aplikasi sistem pakar ini dirancang beserta keluarannya yaitu berupa diagnosa beserta penanganan dan pengobatan dari penyakit gigi dan mulut.
3. Dengan adanya aplikasi sistem pakar ini dapat menjadi *database* pengetahuan mengenai hal-hal yang berhubungan dengan gejala dan diagnosa penyakit-penyakit gigi dan mulut beserta solusi dari diagnosanya.
4. Aplikasi ini telah mampu menampilkan diagnosa banding yang menjadi penambal kekurangan aplikasi sebelumnya (aplikasi yang disusun oleh **I Nyoman Kusuma Wardana**).

DAFTAR PUSTAKA

- [1] Arhami, Muhammad. 2005. *Konsep Dasar Sistem Pakar*. Yogyakarta : ANDI
- [2] Awad, M.A. 2005. *A Comparison Between Agile and Traditional Software Development Methodologies* [online]. Tersedia: [http://pds10.egloos.com/pds/200808/13/85/A comparison between Agile and Traditonal SW development methodologies.pdf](http://pds10.egloos.com/pds/200808/13/85/A%20comparision%20between%20Agile%20and%20Traditonal%20SW%20development%20methodologies.pdf). [17 Februari 2011]
- [3] Desiani, Anita. 2006. *Konsep Kecerdasan Buatan*. Yogyakarta : ANDI
- [4] Dwiartara, Loka. *Perkenalan Installasi dan Konfigurasi*. [online] tersedia: <http://www.ilmuwebsite.com> Akses 10 September 2010 pkl. 10:25

- [5] Hartono, Jogiyanto. 2002. *Pengenalan Komputer Edisi III*. Yogyakarta : ANDI
- [6] Hartono, Jogiyanto. 2003. *Pengembangan Sistem Pakar Menggunakan Visual Basic*. Yogyakarta : ANDI
- [7] Hunt, John. 2006. *Agile Software Construction*. London : Springer
- [8] Proboyekti, Umi. 2008. *Bahan Ajar Rekayasa Perangkat Lunak*. [online]. Tersedia: http://lecturer.ukdw.ac.id/othie/agile_model.pdf
- [9] Suparman. 1991. *Mengenal Artificial Intelligent cetakan I*. Yogyakarta: ANDI OFFSET
- [10] Simarmata, Janner. 2007. *Perancangan Basis Data*. Yogyakarta : ANDI