

**SISTEM PAKAR DIAGNOSA PENYAKIT GIGI DAN JARINGAN
PENYANGGA GIGI MENGGUNAKAN METODE FORWARD
CHAINING BERBASIS WEB**

(Skripsi)

Oleh

PUTRI DWI PANGESTU



**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2017**

ABSTRACT

EXPERT SYSTEM OF DIAGNOSE DENTAL DISEASE AND PERIODONTAL TISSUE DISEASE USING FORWARD CHAINING METHOD BASED ON WEBSITE

BY

PUTRI DWI PANGESTU

Dental and oral health is important to note and an integral part of health that requires immediate treatment before it is too late and can affect a person's condition. One of the obstacles is the difficulty of diagnosing dental disease in remote areas caused by the spread of experts or dentists are less evenly. Therefore this study built an expert system that can diagnose dental disease and dental support network based on the knowledge obtained from the expert directly. Expert system uses Php programming language and Mysql database. It's based on website and using Forward Chaining as an Inference method. This method can diagnose the disease by adjusting the facts experienced by the rules that have been made. The expert system will display the value of accuracy using classic probability calculations after the diagnosis is obtained that adjusted with the calculation of normalization. The system consisted of 20 types of disease , 91 types of symptom, and 20 rules. It can diagnose 20 diseases and provide some information such as explanations, related symptoms, and immediate action of curing diseases. The result of this research were: (1) functional testing using black box method with Equivalence Partitioning (EP) technique got results as expected, (2) the expertise testing got average result of 86,45% (very good), and (3) the assessment using questionnaires with three categories of respondents produced an average of 86.28% for first respondents , 88.85% for second respondents, and 85.71% for third respondents where all three were classified as very good. it can be concluded that the system can diagnose dental disease well.

Keywords: expert system, forward chaining, classic probability, dental diagnosis

ABSTRAK

SISTEM PAKAR DIAGNOSA PENYAKIT GIGI DAN JARINGAN PENYANGGA GIGI MENGGUNAKAN METODE *FORWARD CHAINING* BERBASIS WEB

OLEH

PUTRI DWI PANGESTU

Kesehatan gigi dan mulut penting untuk diperhatikan dan merupakan bagian integral dari kesehatan secara keseluruhan yang memerlukan penanganan segera sebelum terlambat dan dapat mempengaruhi kondisi kesehatan seseorang. Salah satu kendala yang dialami adalah sulitnya mendiagnosa penyakit gigi di daerah pelosok disebabkan oleh persebaran pakar atau dokter gigi yang kurang merata. Oleh karena itu pada penelitian ini dibangun suatu sistem pakar yang dapat mendiagnosa penyakit gigi dan jaringan penyangga gigi berdasarkan pengetahuan yang diperoleh dari pakar langsung. Sistem pakar yang dibangun berbasis web dengan menggunakan bahasa pemrograman *Php* dan *database Mysql*. Metode inferensi yang digunakan adalah metode *Forward Chaining*. Metode ini dapat mendiagnosa penyakit dengan menyesuaikan fakta yang dialami dengan aturan yang telah dibuat. Setelah diagnosa didapatkan maka sistem pakar akan menampilkan nilai akurasi menggunakan perhitungan *classic probability* yang disesuaikan dengan perhitungan normalisasi. Sistem terdiri dari 20 data penyakit, 91 data gejala, dan 20 aturan. Sistem pakar ini dapat mendiagnosa 20 penyakit dan memberikan informasi terkait penyakit seperti penjelasan, gejala-gejala terkait, serta tindakan awal penyembuhan penyakit. Hasil dari penelitian ini adalah: (1) pengujian fungsional menggunakan metode *black box* dengan teknik *Equivalence Partitioning* (EP) memperoleh hasil sesuai harapan, (2) sistem melalui pengujian kepakaran dan memperoleh hasil rata-rata sebesar 86,45% (sangat baik), (3) penilaian menggunakan kuisioner dengan tiga kategori responden menghasilkan rata-rata sebesar 86,28% untuk responden I, 88,85% bagi responden II, dan 85,71% bagi responden III dimana ketiganya tergolong sangat baik. Berdasarkan nilai akurasi tersebut maka dapat disimpulkan bahwa sistem dapat mendiagnosa penyakit gigi dengan baik.

Kata Kunci: sistem pakar, *forward chaining*, *classic probability*, diagnosa penyakit gigi.

**SISTEM PAKAR DIAGNOSA PENYAKIT GIGI DAN JARINGAN
PENYANGGA GIGI MENGGUNAKAN METODE *FORWARD CHAINING*
BERBASIS WEB**

Oleh

PUTRI DWI PANGESTU

Skripsi

Sebagai Salah Satu Syarat untuk Mencapai Gelar
SARJANA KOMPUTER

pada

Jurusan Ilmu Komputer
Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung



**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG**

2017

Judul Skripsi

**: SISTEM PAKAR DIAGNOSA PENYAKIT
GIGI DAN JARINGAN PENYANGGA GIGI
MENGUNAKAN METODE FORWARD
CHAINING BERBASIS WEB**

Nama Mahasiswa

: Putri Dwi Pangestu

Nomor Pokok Mahasiswa

: 1317051048

Jurusan

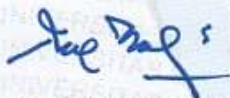
: Ilmu Komputer

Fakultas

: Matematika dan Ilmu Pengetahuan Alam

MENYETUJUI

1. Komisi Pembimbing



Dr. Ir. Kurnia Muludi, M.S.Sc.
NIP 19640616 198902 1 001



drg. Ayu Jembar Sari
SIP 503/0413/SIP/DG/IV.03/IX/2017

2. Ketua Jurusan Ilmu Komputer



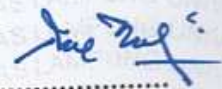
Dr. Ir. Kurnia Muludi, M.S.Sc.
NIP 19640616 198902 1 001

MENGESAHKAN

1. Tim Penguji

Ketua

: **Dr. Ir. Kurnia Muludi, M.S.Sc.**



Sekretaris

: **drg. Ayu Jembar Sari**



Penguji

Bukan Pembimbing : **Aristoteles, S.Si., M.Si.**



2. Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam



Prof. Warsito, S.Si., D.E.A., Ph.D.

NIP 19710212 199512 1 001

Tanggal Lulus Ujian Skripsi : **07 November 2017**

PERNYATAAN

Saya yang bertanda tangan di bawah ini, menyatakan bahwa skripsi saya yang berjudul "Sistem Pakar Diagnosa Penyakit Gigi dan Jaringan Penyangga Gigi menggunakan Metode *Forward Chaining* Berbasis Web" merupakan karya saya sendiri dan bukan karya orang lain. Semua tulisan yang tertuang di skripsi ini telah mengikuti kaidah penulisan karya ilmiah Universitas Lampung. Apabila dikemudian hari terbukti skripsi saya merupakan hasil penjiplakan atau dibuat orang lain, maka saya bersedia menerima sanksi berupa pencabutan gelar yang saya terima.

Bandar Lampung, 7 November 2017



PUTRI DWI PANGESTU
NPM. 1317015048

RIWAYAT HIDUP



Penulis dilahirkan pada tanggal 20 November 1994 di Bandar Lampung, Lampung sebagai anak kedua dari dua bersaudara dengan Ayah yang bernama Edi Santoso dan Ibu bernama Triningsih. Penulis menyelesaikan pendidikan formal pertama kali di TK Dharma Wanita pada tahun 2001, kemudian melanjutkan pendidikan dasar di SDN 2 Rawa Laut Teladan Bandar Lampung dan selesai pada tahun 2007. Pendidikan menengah pertama pada SMPN 2 Bandar Lampung diselesaikan penulis pada tahun 2010, kemudian melanjutkan ke pendidikan menengah atas di SMAN 10 Bandar Lampung yang diselesaikan pada tahun 2013.

Pada tahun 2013 penulis terdaftar sebagai mahasiswa di Jurusan Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung. Selama dalam masa perkuliahan, penulis mengikuti organisasi internal jurusan yaitu Himakom (Himpunan Mahasiswa Jurusan Ilmu Komputer) periode 2013/2014 hingga periode 2015/2016. Selama menjadi mahasiswa beberapa kegiatan yang dilakukan penulis antara lain pada bulan Januari 2016 penulis melaksanakan Kuliah Kerja Nyata (KKN) di Desa Tri Mukti Jaya Kecamatan Banjar Agung Tulang Bawang, dan pada bulan Juli 2016 penulis melaksanakan Kerja Praktik (KP) di Koperasi Karyawan (KOPKAR) Ruwa Jurai PTPN VII Bandar Lampung.

PERSEMBAHAN

*Segala puji Syukur atas berkah dan rahmat dari Allah Subhanallah Wata'ala,
Kupersembahkan Skripsi Ini Untuk Orang-Orang Yang Selalu Kuharapkan
Cinta dan Kasih Sayangnya...*

Teruntuk Papa dan Mama yang tidak pernah memutus doa untuk anak-anaknya, terimakasih untuk menopang jatuhku dan menuntunku, terimakasih untuk jadi saksi langkah-langkahku...

Teruntuk kakakku, terima kasih untuk teguran dan motivasimu...

Teruntuk teman-temanku tercinta, terimakasih untuk selalu ada menunggu dan menemani...

*Almamater Tercinta,
UNIVERSITAS LAMPUNG*

MOTTO

“Remember Me, I will remember you”
(QS: Al-Baqarah : 152)

“Melangkahlah, Allah akan berikan jalan. Semua karena Allah”
(Triningsih)

“Hajar. Jangan takut gagal”
(Edi Santoso)

“Jangan takut. Takutlah hanya kepada Allah SWT”
(Penulis)

SANWACANA

Assallamu'alaikum Warahmatullahi Wabarakatu..

Puji syukur kehadiran Allah Subhanallahu Wata'ala karena atas berkah dan rahmat-Nya penulis dapat menyelesaikan skripsi ini yang berjudul “Sistem Pakar Diagnosa Penyakit Gigi dan Jaringan Penyangga Gigi menggunakan Metode *Forward Chaining* Berbasis Web”. Skripsi ini merupakan salah satu syarat untuk memperoleh gelar Sarjana Komputer di Jurusan Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung.

Dalam pelaksanaan dan penyusunan skripsi ini penulis sangat berterima kasih dan memberikan penghargaan yang sedalam-dalamnya kepada seluruh pihak yang membantu penulis menyelesaikan skripsi ini. Penulis ingin mengucapkan terima kasih dengan setulus hati terutama kepada:

1. Saya. Terima kasih kepada saya sendiri yang selalu mencoba walau gagal, bangkit setiap jatuh dan tetap berjalan sampai garis akhir meski terluka.
2. Kedua Orangtua tercinta, Mama, Papa, dan kakakku serta saudara-saudaraku yang selalu kusayangi dan ku kasihi yang selalu memberikan dukungan, motivasi dan doanya yang tak terhingga.
3. Bapak Prof. Drs. Warsito, S.Si., D.E.A., Ph.D. selaku Dekan FMIPA Universitas Lampung.
4. Bapak Dr. Ir. Kurnia Muludi, M.S.Sc. selaku ketua Jurusan Ilmu Komputer FMIPA Universitas Lampung, pembimbing akademik, serta pembimbing skripsi atas kesediaannya, kesabaranan dan keikhlasannya

untuk memberikan dukungan, bimbingan, nasihat, saran, dan kritik dalam proses penyelesaian skripsi ini.

5. Ibu drg. Ayu Jembar Sari selaku pembimbing pembantu yang telah ikut memberikan bimbingan, saran dan masukan guna penyempurnaan dalam penulisan skripsi ini.
6. Bapak Aristoteles S.Si., M.Si. selaku dosen pembahas skripsi, yang telah memberikan saran dan masukan guna penyempurnaan dalam penulisan skripsi ini.
7. Bapak dan Ibu Dosen Jurusan Ilmu Komputer yang telah memberikan ilmu dan pengalaman hidup selama penulis menjadi mahasiswa.
8. Ibu Ade Nora Maela, Mas Irsan, Mas Nofal dan Mas Zai, Mbak Lia yang telah membantu memudahkan segala urusan administrasi penulis di Jurusan Ilmu Komputer.
9. Seluruh keluarga dan saudaraku yang telah memberikan dukungan selama proses perkuliahan yang tidak bisa disebutkan satu persatu.
10. Teman- temanku tercinta, Rita, Kidiwiw, Upeh, Mpril, Ujank, Adiba, Vandu Uhuy, Danjen, Pani Ulala dan Nadya terimakasih banyak untuk selalu ada dan menemani baik suka maupun duka.
11. Teman-temanku tercinta, Dini, Ulino, Dyah, Kancut, Annisa tetanggaku, Nana, Jokowi, Faiq, Webe, Wika, Mas Paldi, Mita, Ici, Zakiah, Nci, Yeni, dan Fitria terimakasih banyak sudah mewarnai hari-hari selama ini.

12. Rekan-rekan Ilmu Komputer 2013 yang tidak bisa disebutkan satu persatu, terima kasih untuk segala dukungan, bantuan serta kebersamaannya selama ini .

13. Semua pihak yang secara langsung maupun tidak langsung yang telah membantu dalam penyelesaian skripsi ini.

Penulis telah berusaha semaksimal mungkin dalam penulisan skripsi ini untuk mencapai suatu kelengkapan dan kesempurnaan. Penulis juga mengharapkan saran dan kritik yang bersifat membangun dari semua pihak. Akhirnya dengan segala kerendahan hati penulis berharap skripsi ini dapat memberikan manfaat, baik kepada penulis khususnya maupun kepada pembaca pada umumnya.

Bandar Lampung, 07 November 2017

Penulis,

Putri Dwi Pangestu

DAFTAR ISI

	Halaman
DAFTAR ISI	xi
DAFTAR TABEL	xv
DAFTAR GAMBAR	xvi
DAFTAR LAMPIRAN	xix
 I. PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	4
1.3 Batasan Masalah.....	5
1.4 Tujuan.....	5
1.5 Manfaat.....	5
 II. TINJAUAN PUSTAKA	6
2.1 Kecerdasan Buatan.....	6
2.2 Sistem Pakar.....	7
2.2.1 Definisi Sistem Pakar	7
2.2.2 Ciri-Ciri Sistem Pakar	7
2.2.3 Struktur Sistem Pakar.....	8
2.2.4 Manfaat Sistem Pakar.....	10

2.2.5 Konsep Dasar Sistem Pakar	11
2.2.5.1 Kepakaran	11
2.2.5.2 Pakar	12
2.2.5.3 Pemindahan Kepakaran	12
2.2.5.4 Inferensi	13
2.2.5.5 Aturan-aturan	13
2.2.5.6 Kemampuan Menjelaskan.....	13
2.2.6 Bentuk Sistem Pakar	14
2.2.7 Klasifikasi Sistem Pakar.....	14
2.3 Teknik Inferensi	15
2.4 Metode <i>Forward Chaining</i>	16
2.5 Pengujian.....	19
2.5.1 <i>Black Box Testing</i>	19
2.5.2 Metode <i>Classic Probability</i>	20
2.5.3 Skala Likert	21
2.6 Penyakit Gigi.....	22
2.6.1 Pengertian Penyakit Gigi dan Mulut	22
2.6.2 Pengertian Oral Diagnosis.....	23
2.6.3 Jenis-jenis Penyakit Gigi	23
2.6.3.1 Penyakit Jaringan Keras Gigi	23
2.6.3.2 Penyakit Jaringan Pulpa.....	24
2.6.3.3 Penyakit jaringan penyangga gigi.....	25
2.7 <i>Hypertext Preprocessor</i> (PHP)	27
2.8 Database dan MySQL	28
2.9 <i>Unified Modeling Language</i> (UML).....	29

III. METODOLOGI PENELITIAN	36
3.1 Waktu dan Tempat Penelitian	36
3.2 Alat Pendukung	36
3.3 Tahapan Penelitian	37
3.3.1 Tahap Identifikasi Masalah	38
3.3.2 Tahap Perumusan Masalah.....	39
3.3.3 Tahap Pengumpulan Data	39
3.3.4 Tahap Pembuatan Sistem	40
3.3.4.1 <i>Software Requirement Analysis</i>	40
3.3.4.2 <i>Design</i>	41
3.3.4.2.1 <i>Flowchart</i>	41
3.3.4.2.2 <i>Activity Diagram</i>	44
3.3.4.2.2.1 <i>Kelola Daftar Rule</i>	45
3.3.4.2.2.2 <i>Kelola Level</i>	46
3.3.4.2.2.3 <i>Kelola Daftar Penyakit</i>	47
3.3.4.2.2.4 <i>Kelola Daftar Gejala</i>	48
3.3.4.2.2.5 <i>Akses Daftar Penyakit</i>	49
3.3.4.2.2.6 <i>Kelola Deteksi Penyakit Gigi</i>	50
3.3.4.2.3 <i>Sequence Diagram</i>	51
3.3.4.2.3.1 <i>Kelola Daftar Rules</i>	51
3.3.4.2.3.2 <i>Kelola Level</i>	52
3.3.4.2.3.3 <i>Kelola Daftar Penyakit</i>	53
3.3.4.2.3.4 <i>Kelola Daftar Gejala</i>	54
3.3.4.2.3.5 <i>Akses Daftar Penyakit</i>	55
3.3.4.2.3.6 <i>Kelola Deteksi Penyakit Gigi</i>	56
3.3.4.2.4 <i>Class Diagram</i>	57
3.3.4.2.5 <i>Rancangan Interface (Antarmuka)</i>	59
3.3.4.3 <i>Coding</i>	76

3.3.4.3.1 Metode <i>Forward Chaining</i>	76
3.3.4.4 Pengujian (<i>Testing</i>)	78
3.3.4.4.1 Pengujian Internal	79
3.3.4.4.2 Pengujian Eksternal.....	83
3.3.5 Penyusunan Laporan	84
IV. HASIL DAN PEMBAHASAN	85
4.1 Analisa Kebutuhan Data	85
4.2 Representasi Pengetahuan.....	87
4.3 Implementasi Sistem	87
4.3.1 Tampilan Halaman <i>Input</i> Data (Pakar)	88
4.3.2 Tampilan Halaman Proses Diagnosa (Pengguna)	91
4.3.3 Tampilan Halaman Hasil Diagnosa (Pengguna)	99
4.3.4 Analisa Presentase Penyakit (Pengguna)	100
4.4 Pengujian Sistem.....	101
4.4.1 Pengujian <i>Internal</i>	102
4.4.1.1 Pengujian Fungsional.....	102
4.4.1.2 Pengujian Kepakaran	107
4.4.2 Pengujian <i>Eksternal</i>	114
4.4.2.1 Analisa Hasil Kuisisioner	118
V. PENUTUP	129
5.1 Kesimpulan	129
5.2 Saran.....	130
DAFTAR PUSTAKA	131
LAMPIRAN	134

DAFTAR TABEL

Tabel	Halaman
Tabel 2.1. Tabel Kemungkinan.....	21
Tabel 2.2. Notasi <i>Activity Diagram</i>	33
Tabel 2.3. Notasi <i>Sequence Diagram</i>	35
Tabel 3.1. Daftar pengujian <i>black box</i> untuk pakar	80
Tabel 3.2. Daftar pengujian <i>black box</i> untuk pengguna (<i>user</i>)	83
Tabel 4.1. Daftar Nama Penyakit Gigi dan Jaringan Penyangga Gigi.....	85
Tabel 4.2. Hasil pengujian fungsional sistem untuk pakar	102
Tabel 4.3. Hasil pengujian fungsional sistem untuk pengguna.....	106
Tabel 4.4. Hasil pengujian kepakaran sistem.....	107
Tabel 4.5. Hasil penilaian kuisisioner oleh dokter gigi (Responden I).....	116
Tabel 4.6. Hasil penilaian kuisisioner oleh pasien (Responden II)	117
Tabel 4.7. Hasil penilaian kuisisioner oleh Mahasiswa Ilmu Komputer Unila (Responden III).....	118
Tabel 4.8. Kriteria <i>index</i> penilaian hasil kuisisioner	119

DAFTAR GAMBAR

Gambar	Halaman
Gambar 2.1. Komponen dalam sebuah sistem pakar	9
Gambar 2.2. Iterasi ke – 1	17
Gambar 2.3. Iterasi ke – 2	18
Gambar 2.4. Iterasi ke – 3	18
Gambar 2.5. Contoh aktor	31
Gambar 2.6. Contoh <i>use case</i>	31
Gambar 3.1. Tahapan penelitian	38
Gambar 3.2. <i>Use case diagram</i> sistem pakar diagnosa penyakit gigi	40
Gambar 3.3. Proses bisnis oleh pakar.....	42
Gambar 3.4. Proses bisnis identifikasi	43
Gambar 3.5. <i>Activity diagram</i> kelola daftar rules	45
Gambar 3.6. <i>Activity diagram</i> kelola <i>level</i>	46
Gambar 3.7. <i>Activity diagram</i> kelola daftar penyakit	47
Gambar 3.8. <i>Activity diagram</i> kelola daftar gejala.....	48
Gambar 3.9. <i>Activity diagram</i> akses daftar penyakit.....	49
Gambar 3.10. <i>Activity diagram</i> kelola deteksi penyakit gigi	50

Gambar 3.34. Rancangan halaman lihat data detail penyakit gigi pengguna	74
Gambar 3.35. Rancangan halaman deteksi penyakit gigi pengguna.....	75
Gambar 3.36. Rancangan halaman deteksi penyakit gigi pengguna.....	75
Gambar 3.37. Rancangan halaman hasil deteksi penyakit gigi pengguna	76
Gambar 4.1. Halaman tambah data gejala	88
Gambar 4.2. Halaman tambah data penyakit	89
Gambar 4.3. Halaman tambah relasi penyakit-gejala (<i>rule</i> /aturan)	90
Gambar 4.4. Halaman tambah level	91
Gambar 4.5. Halaman beranda pengguna	92
Gambar 4.6. Halaman identitas pengguna	92
Gambar 4.7. Halaman pertanyaan Iterasi-1	93
Gambar 4.8. Halaman pertanyaan Iterasi -2.....	95
Gambar 4.9. Halaman pertanyaan Iterasi- 3.....	96
Gambar 4.10. Halaman pertanyaan Iterasi- 4.....	97
Gambar 4.11. Halaman pertanyaan Iterasi- 5.....	98
Gambar 4.12. Halaman hasil diagnosa.....	100
Gambar 4.13. Grafik penilaian kuisisioner terhadap pernyataan 1	120
Gambar 4.14. Grafik penilaian kuisisioner terhadap pernyataan 2	121
Gambar 4.15. Grafik penilaian kuisisioner terhadap pernyataan 3	122
Gambar 4.16. Grafik penilaian kuisisioner terhadap pernyataan 4	123
Gambar 4.17. Grafik penilaian kuisisioner terhadap pernyataan 5	124
Gambar 4.18. Grafik penilaian kuisisioner terhadap pernyataan 6	126
Gambar 4.19. Grafik penilaian kuisisioner terhadap pernyataan 7	127

DAFTAR LAMPIRAN

Lampiran	Halaman
Lampiran 1. Data Penyakit dan Data Gejala	135
Lampiran 2. Data Penyakit.....	141
Lampiran 3. Data Gejala	142
Lampiran 4. Pohon Keputusan.....	146
Lampiran 5. Tabel Keputusan	147
Lampiran 6. Basis Aturan	150

I. PENDAHULUAN

1.1 Latar Belakang

Gigi merupakan salah satu organ terpenting yang ada di dalam tubuh manusia. Sebagai satu satunya organ yang tidak bisa menyembuhkan dirinya sendiri, gigi menjadi organ tubuh yang sangat dijaga dan dirawat kondisinya selama kehidupan seseorang berlangsung. Gigi juga termasuk organ yang sangat penting dalam proses pengolahan makanan. Seseorang dapat dipastikan akan sulit mengunyah makanan tanpa adanya gigi. Kesehatan gigi dan mulut penting untuk diperhatikan dan merupakan bagian integral dari kesehatan secara keseluruhan yang memerlukan penanganan segera sebelum terlambat dan dapat mempengaruhi kondisi kesehatan seseorang. Setiap orang perlu mengetahui cara menjaga kesehatan gigi dan mulut mereka. Pengetahuan tentang kesehatan gigi dan mulut merupakan salah satu usaha untuk mencegah dan menanggulangi masalah kesehatan gigi melalui pendekatan pendidikan kesehatan gigi dan mulut (Azhary, 2016). Kondisi gigi dan mulut yang tidak terjaga dengan baik dapat menyebabkan masalah yang lain di sekitar mulut, diantaranya timbul gigi yang berlubang, sakit gigi, karang gigi, plak gigi, peradangan pada gusi, sariawan, dan kelainan-kelainan yang lain disekitar gigi (Setyaningsih, 2007).

Berdasarkan data yang diperoleh dari analisis lanjut Riskesdas tahun 2007, karakteristik seseorang (umur, pendidikan, tempat tinggal, serta sosial ekonomi) responden memengaruhi terjadinya penyakit gigi. Semakin tinggi pendidikan dan semakin tinggi tingkat sosial ekonomi responden semakin kecil risiko terjadinya penyakit gigi, responden yang tinggal di kota beresiko untuk terjadinya karies lebih besar dibandingkan responden yang tinggal di desa (Oktarina, 2010).

Data informasi kesehatan gigi di Indonesia yang diperoleh dari Pusat Data dan Informasi (Pusdatin) Kementerian Kesehatan RI tahun 2014 mengungkapkan bahwa presentase penduduk Indonesia yang memiliki masalah kesehatan gigi tahun 2007 dan 2013 meningkat dari 23,2% menjadi 25,9%. Penduduk yang menerima perawatan medis gigi meningkat dari 29,7% tahun 2007 menjadi 31,1% pada tahun 2013 (Kemenkes RI, 2014).

Pusdatin dan Badan PPSDM (Pengembangan dan Pemberdayaan Sumber Daya Manusia) melakukan analisis sederhana untuk melihat persebaran tenaga kesehatan dokter gigi di puskesmas berdasarkan data olahan yang diperoleh pada tahun 2013. Hasil dari analisis Pusdatin dan PPSDM menunjukkan bahwa sebagian besar puskesmas di 33 Provinsi masih ada yang “kurang” tenaga dokter gigi sedangkan provinsi lain memiliki tenaga kesehatan gigi (dokter gigi/ perawat gigi) berlebih (Kemenkes RI, 2014).

Hasil prevalensi yang telah dilakukan oleh Kemenkes dan PPSDM menunjukkan bahwa Indonesia memiliki peningkatan pada masalah kesehatan gigi serta penyebaran tenaga kerja dokter gigi yang ternyata kurang seimbang di setiap provinsi. Hal ini bisa berdampak pada sulitnya jangkauan dokter gigi pada

sebagian daerah pedesaan yang sebetulnya memiliki tingkat kesehatan gigi yang lebih buruk dibanding di kota.

Idealnya masyarakat seperti ini membutuhkan langkah cepat dalam memeriksakan gigi mereka sebelum pergi ke dokter gigi yang lokasinya dapat terbilang jauh dari jangkauan. Hal ini diperuntukan dalam penanggulangan awal dalam mengatasi penyakit gigi yang mungkin diderita. Oleh karena itu dibutuhkan cara cepat dalam mendiagnosa penyakit gigi yang dapat dilakukan oleh setiap individu.

Teknologi sudah semakin berkembang pada era sekarang. Sebagian besar pekerjaan telah melibatkan teknologi. Sistem pakar menjadi salah satu bagian dari ilmu kecerdasan buatan yang berkembang pesat dalam dunia teknologi. Dengan sistem pakar, seseorang dapat terbantu sesuai dengan kebutuhan disaat sulit untuk menemui para pakar secara langsung. Sistem pakar dapat membantu dalam menyelesaikan masalah dan juga sebagai perwakilan dari para pakar. Dengan adanya teknologi kecerdasan semacam ini seseorang dapat menghemat waktu serta biaya. Sistem pakar menyediakan solusi yang disederhanakan dalam kasus rumit yang berulang – ulang. Implementasinya dapat dibangun dalam bentuk *web based* ataupun *mobile*.

Silvia dkk (2015) telah melakukan penelitian untuk mendiagnosa karies seseorang melalui aplikasi berbasis web yang ditulis dalam sebuah jurnal yang berjudul “Aplikasi Diagnosis Karies pada Gigi Manusia Berbasis Web”. Pengguna diminta untuk menjawab pertanyaan dan jawaban yang diperoleh akan disesuaikan dengan *rules* dalam sistem untuk memberikan kesimpulan.

Sedangkan pada jurnal lain yang berjudul “Penerapan Metode *Forward Chaining* untuk Mendeteksi Penyakit THT” (Wiwik Verina, 2015), telah dibangun sistem pakar untuk mendeteksi penyakit THT yang peningkatannya semakin tinggi dan tidak diiringi oleh jumlah tenaga ahlinya. Dalam analisisnya sistem pakar ini menggunakan metode *forward chaining* sebagai mesin inferensi. Akurasi yang dihasilkan sesuai dengan data yang didapat dari pakar THT untuk menentukan penyakit berdasarkan gejala yang ada.

Maka didasarkan oleh fakta – fakta yang ada serta penelitian-penelitian sebelumnya yang berkaitan, penulis berinisiatif untuk membuat suatu sistem pakar diagnosa penyakit gigi dan jaringan penyangga gigi berbasis web menggunakan metode *forward chaining* yang berguna bagi penderita penyakit gigi dalam mendiagnosa penyakit giginya sendiri. Data mengenai nama penyakit, gejala, serta penanganannya didapatkan langsung dari pakar yang memahami ilmu penyakit gigi. Diharapkan dengan adanya sistem ini masyarakat bisa memperoleh gambaran akan penyakit gigi yang sedang dialami dan dengan mudah dapat mengonsultasikan kepada dokter gigi setelahnya.

1.2 Rumusan Masalah

Adapun rumusan masalah pokok yang menjadi dasar dalam pengerjaan skripsi ini adalah “Bagaimana cara membangun dan merancang sistem pakar diagnosa penyakit gigi dan jaringan penyangga gigi berbasis web yang dapat membantu masyarakat pelosok dalam diagnosa penyakit gigi secara individu serta memberi solusi pertolongan pertama dengan menerapkan metode *forward chaining*”.

1.3 Batasan Masalah

Adapun batasan masalah dari penelitian ini adalah sebagai berikut.

1. Sistem pakar dikhususkan untuk mendiagnosa penyakit gigi.
2. Penyakit gigi yang didiagnosa berjumlah 20 penyakit.
3. Gejala yang digunakan berjumlah 91 gejala.
4. Sistem pakar ini dibangun pada web.
5. Metode penalaran yang dipakai adalah metode *forward chaining*.

1.4 Tujuan

Adapun tujuan dari penelitian ini adalah sebagai berikut.

1. Membuat sistem pakar pendiagnosa penyakit gigi pada masyarakat.
2. Mendeteksi penyakit gigi pada individu.
3. Menentukan solusi pertolongan pertama yang dilakukan setelah mengetahui penyakit gigi.

1.5 Manfaat

Adapun manfaat yang diperoleh dari penelitian sebagai berikut.

1. Memberikan informasi penyakit gigi dan memberikan solusi pertolongan pertama.
2. Mempermudah dan mempercepat waktu seseorang untuk mendeteksi penyakit gigi sendiri.

II. TINJAUAN PUSTAKA

2.1 Kecerdasan Buatan

Kecerdasan buatan berasal dari bahasa Inggris “Artificial Intelligence” atau singkatan AI, yaitu *intelligence* adalah kata sifat yang berarti cerdas, sedangkan *artificial* artinya buatan. Kecerdasan buatan yang dimaksud di sini merujuk pada mesin yang mampu berpikir, menimbang tindakan yang akan diambil dan mampu mengambil keputusan seperti yang dilakukan oleh manusia (Sutojo dkk, 2011).

Menurut Handojo dan Irawan (2009), kecerdasan buatan adalah cabang ilmu komputer yang bertujuan untuk membuat sebuah komputer dapat berpikir dan bernalar seperti manusia. Kecerdasan buatan dapat membantu manusia dalam membuat keputusan, mencari informasi secara lebih akurat, atau membuat komputer lebih mudah digunakan dengan tampilan yang menggunakan bahasa natural sehingga mudah dipahami. Salah satu bagian dari sistem kecerdasan buatan adalah sistem pakar dimana sistem pakar adalah bagian dari ilmu kecerdasan buatan yang secara spesifik berusaha mengadopsi kepakaran seseorang di bidang tertentu ke dalam suatu sistem atau program komputer.

2.2 Sistem Pakar

2.2.1 Definisi Sistem Pakar

Sistem pakar merupakan sebuah program yang berfungsi untuk menirukan seorang pakar atau ahli sehingga program tersebut dapat melakukan apa yang dikerjakan oleh seorang pakar. Pembentukan sistem pakar didasarkan pada suatu ide dengan mentransfer pengetahuan seorang pakar ke dalam komputer. Pengetahuan yang tersimpan itu digunakan untuk menyelesaikan masalah yang sesuai dengan pengetahuan yang telah diperoleh. Dengan adanya sistem pakar, maka keluaran yang akan dihasilkan lebih efektif, terorganisir, dan tepat dibandingkan dengan keluaran yang dihasilkan oleh manusia (Hartati dkk, 2008).

Sistem pakar merupakan cabang dari Artificial Intelligence (AI) yang dikembangkan pada pertengahan 1960. Sistem pakar berasal dari istilah knowledge-based expert system, yaitu sebuah sistem yang menggunakan pengetahuan manusia dimana pengetahuan tersebut dimasukkan kedalam komputer dan kemudian digunakan untuk menyelesaikan masalah-masalah yang biasanya membutuhkan kepakaran atau keahlian manusia (Sutojo dkk, 2011).

2.2.2 Ciri-Ciri Sistem Pakar

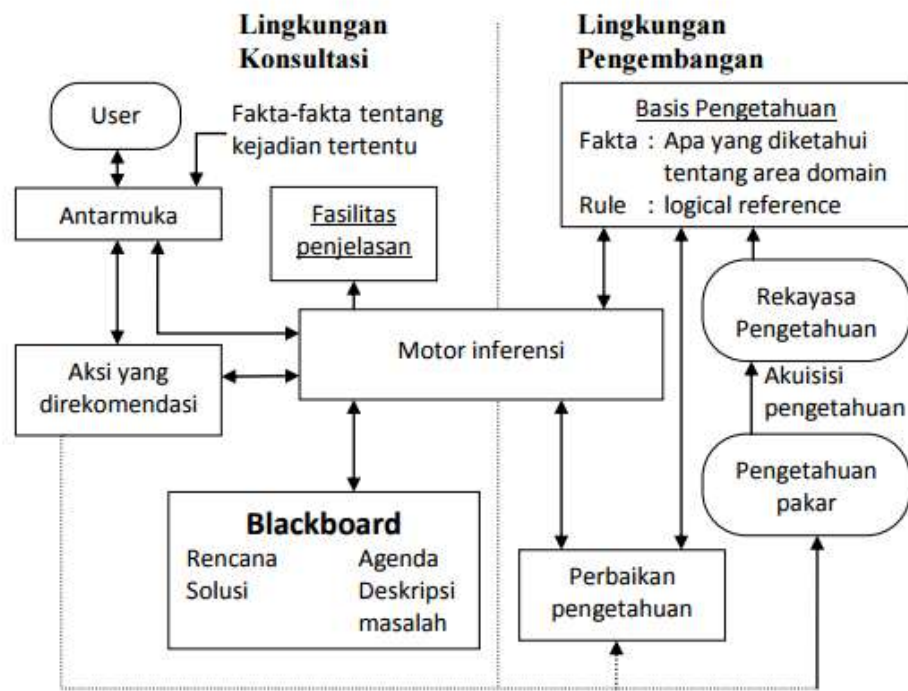
Menurut Sutojo dkk (2011), ciri-ciri sistem pakar adalah :

1. terbatas pada domain keahlian tertentu,
2. dapat memberikan penalaran-penalaran untuk data-data yang tidak lengkap atau tidak pasti,

3. dapat mengemukakan rangkaian alasan-alasan yang diberikannya dengan cara yang dapat dipahami,
4. berdasarkan pada kaidah/ketentuan/*rule* tertentu,
5. dirancang untuk dapat dikembangkan secara bertahap,
6. pengetahuan dan mekanisme penalaran (inference) jelas terpisah,
7. keluarannya bersifat anjuran,
8. sistem dapat mengaktifkan kaidah secara searah yang sesuai dituntun oleh dialog dengan user.

2.2.3 Struktur Sistem Pakar

Menurut Sutojo dkk. (2011) dan Siswanto (2010), sistem pakar disusun oleh dua bagian utama, yaitu bagian lingkungan pengembangan (development environment) dan lingkungan konsultasi (consulation environment). Lingkungan pengembang digunakan oleh pembuat sistem pakar untuk membangun komponen komponennya dan memperkenalkan pengetahuan ke dalam *knowledge base* (basis pengetahuan). Sedangkan lingkungan konsultasi digunakan oleh pengguna untuk berkonsultasi sehingga pengguna mendapatkan pengetahuan dan nasihat dari sistem pakar layaknya berkonsultasi dengan seorang pakar.



Gambar 2.1 Komponen dalam sebuah sistem pakar (Sutojo dkk., 2011)

Pada Gambar 2.1 dapat dijelaskan sebagai berikut.

1. Akuisisi Pengetahuan

Subsistem ini digunakan untuk memasukkan pengetahuan dari seorang pakar dengan cara merekayasa pengetahuan agar bisa diproses oleh komputer dan meletakkannya ke dalam basis pengetahuan dengan format tertentu.

2. Basis pengetahuan (*knowledge base*)

Basis pengetahuan berisi pengetahuan yang diperlukan untuk memahami, memformulasikan dan menyelesaikan masalah. Basis pengetahuan terdiri dari dua elemen dasar yaitu fakta dan *rule* atau aturan.

3. Mesin inferensi (*Interference Engine*)

Mesin inferensi adalah sebuah program yang berisi metodologi yang digunakan untuk melakukan penalaran terhadap informasi-informasi dalam basis pengetahuan untuk memformulasikan konklusi.

4. Daerah kerja (*Blackboard*)

Daerah kerja yaitu area pada memori yang berfungsi sebagai basis data. Ada tiga tipe keputusan yang dapat direkam pada blackboard yaitu rencana, agenda dan solusi.

5. Antarmuka (*User Interface*)

Antarmuka digunakan sebagai media komunikasi antara pengguna dan sistem pakar. Program akan mengajukan pertanyaan-pertanyaan dan sistem pakar akan mengambil kesimpulan berdasarkan jawaban dari user.

6. Subsistem penjelasan (*Explanation Subsystem*)

Subsistem penjelasan berfungsi memberi penjelasan kepada user, bagaimana suatu kesimpulan dapat diambil.

2.2.4 Manfaat Sistem Pakar

Menurut Sutojo dkk (2011) sistem pakar menjadi populer karena sangat banyak kemampuan dan manfaat yang diberikannya, di antaranya sebagai berikut.

1. Meningkatkan produktivitas, karena sistem pakar dapat bekerja lebih cepat daripada manusia.
2. Membuat seorang yang awam bekerja seperti layaknya seorang pakar.
3. Meningkatkan kualitas, dengan memberi nasihat yang konsisten dan mengurangi kesalahan.

4. Mampu menangkap pengetahuan dan kepakaran seseorang.
5. Dapat beroperasi di lingkungan yang berbahaya.
6. Memudahkan akses pengetahuan seorang pakar.
7. Handal, sistem pakar tidak pernah menjadi bosan dan kelelahan atau sakit.
8. Meningkatkan kapabilitas sistem komputer. Integrasi sistem pakar dengan sistem komputer lain membuat sistem lebih efektif dan mencakup lebih banyak aplikasi.
9. Mampu bekerja dengan informasi yang tidak lengkap atau tidak pasti.
10. Bisa digunakan sebagai media pelengkap dalam pelatihan. Pengguna pemula yang bekerja dengan sistem pakar akan menjadi lebih berpengalaman karena adanya fasilitas penjelas yang berfungsi sebagai guru.
11. Meningkatkan kemampuan untuk menyelesaikan masalah karena sistem pakar mengambil sumber pengetahuan dari banyak pakar.

2.2.5 Konsep Dasar Sistem Pakar

Menurut Sutojo dkk (2011) konsep dasar sistem pakar meliputi enam hal sebagai berikut.

2.2.5.1 Kepakaran

Kepakaran merupakan suatu pengetahuan yang diperoleh dari pelatihan, membaca, dan pengalaman. Kepakaran inilah yang memungkinkan para ahli dapat mengambil keputusan lebih cepat dan lebih baik daripada seseorang yang bukan pakar. Kepakaran itu sendiri meliputi pengetahuan sebagai berikut.

1. Fakta-fakta pada lingkup permasalahan tertentu.
2. Teori-teori pada lingkup permasalahan tertentu.
3. Prosedur-prosedur dan aturan-aturan berkenaan dengan lingkup permasalahan tertentu.
4. Strategi-strategi global untuk menyelesaikan masalah.
5. *Meta-knowledge* (pengetahuan tentang pengetahuan).

2.2.5.2 Pakar

Pakar adalah seseorang yang mempunyai pengetahuan, pengalaman, dan metode khusus serta mampu menerapkannya untuk memecahkan masalah atau memberi nasihat. Seorang pakar harus mampu menjelaskan dan mempelajari hal-hal baru yang berkaitan dengan topik permasalahan, jika perlu harus mampu menyusun kembali pengetahuan-pengetahuan yang didapatkan, dan dapat memecahkan aturan-aturan serta menentukan relevansi kepakarannya.

2.2.5.3 Pemindahan Kepakaran

Tujuan dari sistem pakar adalah memindahkan kepakaran dari seorang pakar ke dalam komputer, kemudian ditransfer kepada orang lain yang bukan pakar. Proses ini melibatkan empat kegiatan, yaitu :

1. akuisisi pengetahuan (dari pakar atau sumber lain),
2. representasi pengetahuan (pada komputer),
3. inferensi pengetahuan,
4. pemindahan pengetahuan ke pengguna.

2.2.5.4 Inferensi

Inferensi adalah sebuah prosedur yang mempunyai kemampuan dalam melakukan penalaran. Inferensi ditampilkan pada suatu komponen yang disebut mesin inferensi yang mencakup prosedur-prosedur mengenai pemecahan masalah. Semua pengetahuan yang dimiliki oleh seorang pakar disimpan pada basis pengetahuan oleh sistem pakar. Tugas mesin inferensi adalah mengambil kesimpulan berdasarkan basis pengetahuan yang dimilikinya.

2.2.5.5 Aturan-aturan

Kebanyakan aplikasi sistem pakar komersial adalah sistem yang berbasis *rule* (*rulebased system*), yaitu pengetahuan disimpan terutama dalam bentuk *rule*, sebagai prosedur-prosedur pemecahan masalah.

2.2.5.6 Kemampuan Menjelaskan

Fasilitas lain dari sistem pakar adalah kemampuannya untuk menjelaskan saran atau rekomendasi yang diberikannya. Penjelasan dilakukan dalam subsistem yang disebut subsistem penjelasan (*explanation*). Bagian dari sistem ini memungkinkan sistem untuk memeriksa penalaran yang dibuatnya sendiri dan menjelaskan operasioperasinya.

2.2.6 Bentuk Sistem Pakar

Sistem pakar dibagi menjadi empat bentuk, yaitu sebagai berikut.

1. Berdiri Sendiri

Sistem pakar jenis ini merupakan perangkat lunak yang berdiri sendiri tidak bergabung dengan software lainnya.

2. Tergabung

Sistem pakar jenis ini merupakan bagian program yang terkandung di dalam suatu algoritma (konvensional), atau merupakan program dimana di dalamnya memanggil algoritma subrutin lain (konvensional).

3. Menghubungkan ke perangkat lunak lain

Bentuk ini biasanya merupakan sistem pakar yang menghubungkan ke suatu paket program tertentu, misalnya DBMS.

4. Sistem Mengabdi

Sistem pakar merupakan bagian dari komputer khusus yang dihubungkan dengan suatu fungsi tertentu. Misalnya sistem pakar yang digunakan untuk membantu menganalisis data radar.

2.2.7 Klasifikasi Sistem Pakar

Menurut Siswanto (2005), klasifikasi sistem pakar berdasarkan kegunaannya adalah sebagai berikut.

1. Diagnosis

Diagnosis digunakan untuk merekomendasikan obat untuk orang sakit, kerusakan mesin, kerusakan rangkaian elektronik. Bertujuan menemukan

masalah/kerusakan yang terjadi. Biasanya menggunakan pohon keputusan (*decision tree*) sebagai representasi pengetahuannya.

2. Pengajaran

Pengajaran digunakan untuk pengajaran, mulai dari SD s/d PT. Membuat diagnosa apa kekurangan dari siswa, kemudian memberikan solusi atau cara untuk memperbaikinya.

3. Interpretasi

Untuk menganalisa data yang tidak lengkap, tidak teratur, dan data yang kontradiktif. Misal : untuk interpretasi citra.

4. Prediksi

Misalnya bagaimana seorang pakar meteorologi memprediksi cuaca besok berdasarkan data-data sebelumnya. Untuk melakukan peramalan suatu kasus.

5. Perencanaan

Digunakan untuk merencanakan suatu pekerjaan misalnya, perencanaan mesinmesin dan manajemen bisnis. Untuk menghemat biaya, waktu, & material, sebab pembuatan model sudah tidak diperlukan. Contoh : sistem konfigurasi komputer.

6. Kontrol

Digunakan untuk mengontrol kegiatan yang membutuhkan presisi waktu tinggi. Misal: pengontrolan pada industri-industri berteknologi tinggi.

2.3 Teknik Inferensi

Mesin inferensi adalah bagian yang mengandung mekanisme fungsi berpikir dan pola-pola penalaran sistem yang digunakan seorang pakar. Mekanisme ini akan

menganalisa suatu masalah tertentu dan selanjutnya mencari kesimpulan yang terbaik (Siswanto, 2010).

Menurut Sutojo dkk. (2011) dan Siswanto (2010) Ada dua teknik penalaran (inference) yaitu sebagai berikut.

- *Forward Chaining* (Pelacakan ke depan)

Teknik ini memulai pencarian dengan fakta yang diketahui untuk menguji kebenaran hipotesa, kemudian mencocokkan fakta tersebut dengan bagian IF dari *rule* IF-THEN. Teknik ini cocok digunakan untuk menangani masalah peramalan (prognosis) dan pengendalian (*controlling*).

- *Backward Chaining* (Pelacakan ke belakang)

Teknik ini memulai pencarian dari kesimpulan (*goal*) dengan mencari sekumpulan hipotesa-hipotesa yang mendukung menuju fakta-fakta yang mendukung sekumpulan hipotesa tersebut.

2.4 Metode *Forward Chaining*

Menurut Sutojo dkk (2011) metode *forward chaining* merupakan teknik pencarian yang dimulai dengan fakta yang diketahui, kemudian mencocokkan fakta-fakta tersebut dengan bagian IF dari *rules* IF-THEN. Bila ada fakta yang cocok dengan bagian IF, maka *rule* tersebut dieksekusi. Bila sebuah *rule* dieksekusi, maka sebuah fakta baru (bagian THEN) ditambahkan ke dalam database. Setiap kali pencocokan berhenti bila tidak ada lagi *rule* yang bisa dieksekusi. Contoh:

misalkan diketahui sistem pakar menggunakan 5 buah *rule* berikut.

R1 : IF (Y AND D) THEN Z

R2 : IF (X AND B AND E) THEN Y

R3 : IF A THEN X

R4 : IF C THEN L

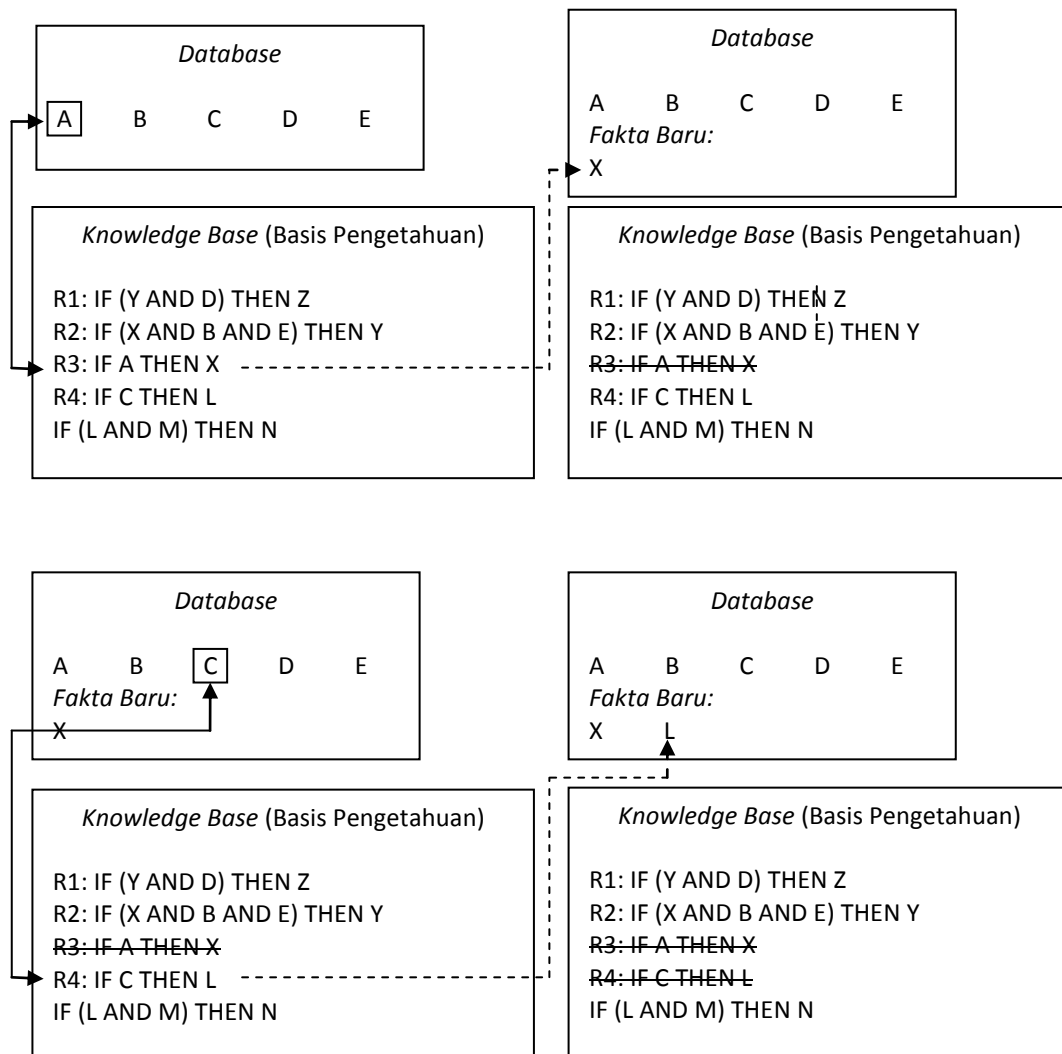
R5 : IF (L AND M) THEN N

Fakta-fakta : A, B, C, D, dan E bernilai benar.

Goal : menentukan apakah Z bernilai benar atau salah.

Iterasi ke-1

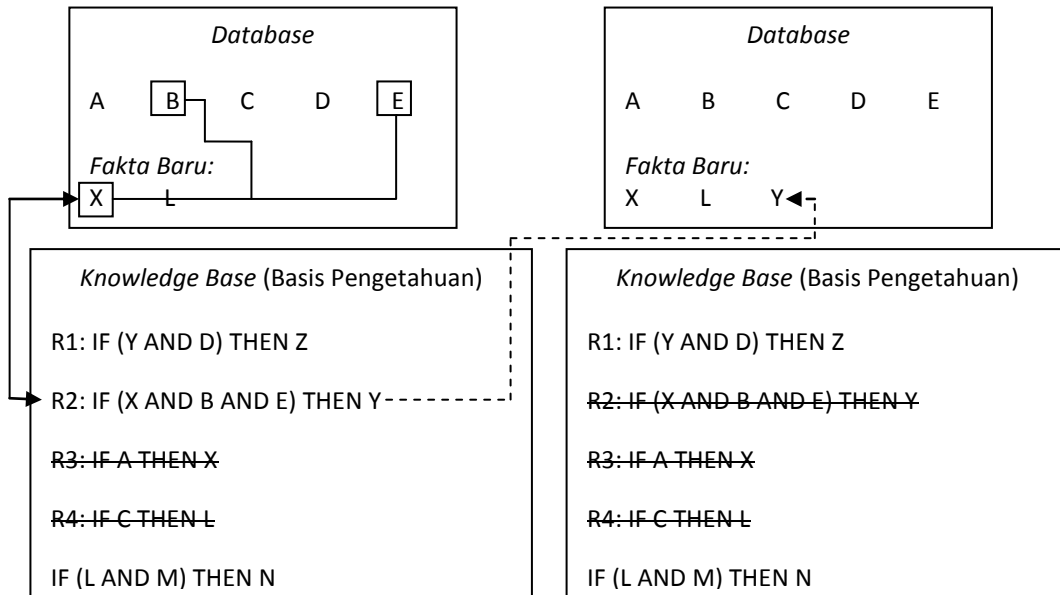
Iterasi ke-1 dapat dilihat pada Gambar 2.2.



Gambar 2.2. Iterasi ke – 1

Iterasi ke-2

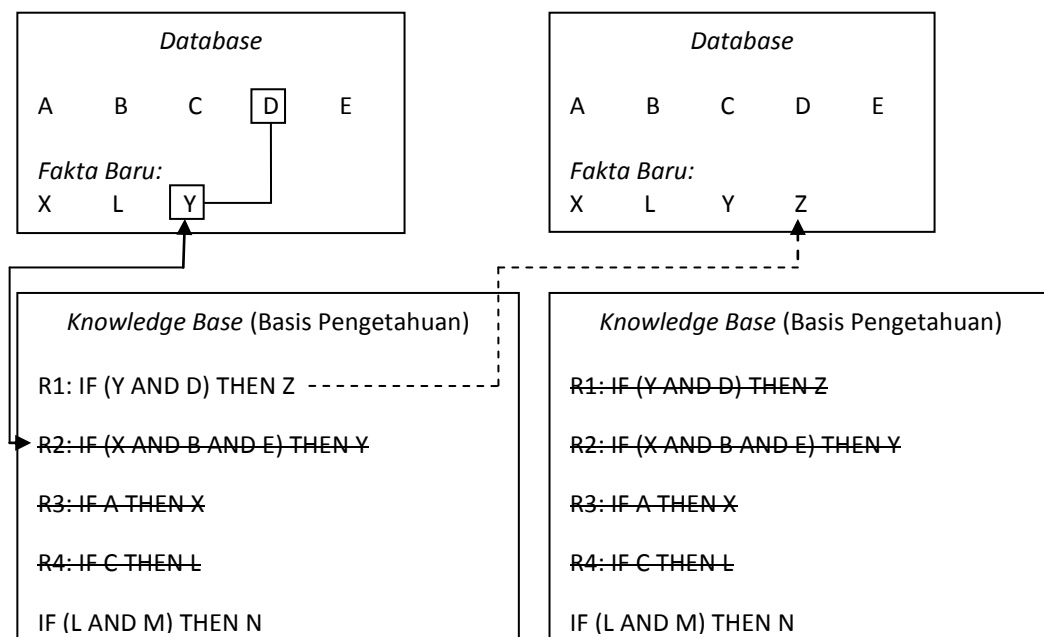
Berdasarkan Gambar 2.3 iterasi ke-2 adalah sebagai berikut.



Gambar 2.3. Iterasi ke-2

Iterasi ke-3

Iterasi ke-3 dijelaskan dalam Gambar 2.4 sebagai berikut.



Gambar 2.4. Iterasi ke-3

Sampai di sini proses dihentikan karena sudah tidak ada lagi *rule* yang bisa dieksekusi. Hasil pencarian adalah Z bernilai benar (lihat database di bagian fakta baru).

2.5 Pengujian

Pengujian perangkat lunak merupakan proses eksekusi suatu program atau sistem dengan tujuan menemukan ada atau tidaknya kekurangan atau masalah pada sistem dengan melibatkan setiap aktivitas. Pada pengujian sistem ini dilakukan evaluasi pada setiap atribut atau kemampuan suatu sistem sehingga diketahui apakah sistem yang dibuat sudah sesuai dengan kebutuhan pengguna (Zulkifli, 2013).

2.5.1 Black Box Testing

Black box testing merupakan salah satu metode pengujian sistem yang dilakukan berdasarkan spesifikasi kebutuhan sistem dan tidak perlu memeriksa *coding*. Dengan *black box testing*, pengujian yang dilakukan hanya berdasarkan pandangan pengguna untuk mengetahui apakah fungsi yang dibutuhkan berjalan sesuai harapan atau tidak. Keuntungan penggunaan metode ini adalah pengujian tidak memerlukan pengetahuan yang spesifik mengenai bahasa pemrograman yang digunakan dalam pembuatan sistem tersebut dan juga pengetahuan pada implementasinya (Nidhra dan Dondeti, 2012).

Pada *black box testing*, ada beberapa teknik yang dapat digunakan, salah satunya adalah *Equivalence Class Partioning*. Pengujian dengan *Equivalence Class*

Partioning didasarkan pada asumsi bahwa *input* dan *output* program dapat dibagi menjadi kelas dengan jumlah terbatas (*valid dan non-valid*) sehingga semua kasus yang sudah dipartisi ke dalam kelas-kelasnya akan diuji dengan perilaku yang sama (Nidhra dan Dondeti, 2012).

2.5.2 Metode *Classic Probability*

Probabilitas merupakan suatu cara kuantitatif yang berhubungan dengan ketidakpastian yang telah ada. Teori probabilitas klasik pertama kali diperkenalkan oleh Pascal dan Fermat pada tahun 1654. Kemudian banyak kerja yang telah dilakukan untuk mengerjakan probabilitas dan ada beberapa cabang baru dari probabilitas yang dikembangkan. Probabilitas klasik disebut juga *a priori probability* karena hubungan dengan suatu permainan (*games*) atau sistem. Seperti yang telah disebutkan sebelumnya, istilah *a priori* berarti “sebelum” (Arhami, 2005).

Rumus umum untuk probabilitas klasik didefinisikan sebagai peluang $P(A)$ dengan n adalah banyaknya kejadian, $n(A)$ merupakan banyaknya hasil mendapatkan A. Frekuensi relatif A adalah maka (Arhami, 2005) :

$$P(A) = \frac{n(A)}{N} \dots\dots\dots (1)$$

Keterangan:

A : Nama penyakit

$P(A)$: Peluang penyakit

n : Jumlah seluruh gejala yang berkaitan dengan penyakit

n(A) : Jumlah gejala penyakit yang dipilih/ sesuai

Probabilitas klasik ini digunakan untuk mendapatkan peluang kemungkinan suatu penyakit, sehingga presentase yang dihitung adalah:

$$\text{Presentase (A)} = P(A) \times 100\% \dots\dots\dots (2)$$

Tabel 2.1. Tabel Kemungkinan

Kondisi	Presentase
Pasti Tidak	<10%
Tidak Tahu	10-19%
Hampir Mungkin	20-39%
Mungkin	40-59%
Kemungkinan Benar	60-79%
Hampir Pasti	80-99%
Pasti	100%

2.5.3 Skala Likert

Metode ini merupakan metode penskalaan pernyataan sikap yang menggunakan distribusi *respons* sebagai dasar penentuan nilai skalanya. Nilai skala setiap pernyataan tidak ditentukan oleh derajat *favourable* masing-masing, akan tetapi ditentukan oleh distribusi *respons* setuju dan tidak setuju dari sekelompok responden yang bertindak sebagai kelompok uji coba (Azwar, 2011).

Skala Likert, yaitu skala yang berisi lima tingkat preferensi jawaban dengan pilihan sebagai berikut : 1= sangat setuju; 2 = tidak setuju; 3 = ragu-ragu atau netral; 4 = setuju; 5 = sangat setuju. Persentase penilaian berdasarkan kriteria

skala likert akan diperoleh dengan rumus aritmatika *mean*, yaitu (Djarwanto, 1996) :

$$P = \frac{X_i}{n \times N} \times 100 \% \quad \dots\dots\dots (3)$$

Keterangan :

P = Persentase pernyataan

X_i = Nilai kuantitatif total

n = Jumlah responden

N = Nilai kategori pernyataan terbaik

Selanjutnya, penentuan interval per kategori digunakan rumus sebagai berikut :

$$I = \frac{100\%}{K} \quad \dots\dots\dots (4)$$

Keterangan :

I = Interval

K = Kategori interval

2.6 Penyakit Gigi

2.6.1 Pengertian Penyakit Gigi dan Mulut

Ilmu penyakit gigi dan mulut atau oral patologi adalah ilmu yang mempelajari penyakit-penyakit dan kelainan yang terjadi pada rongga mulut, tanda-tanda atau gejalanya, penyebabnya serta perawatannya (Kristiani dkk, 2010).

2.6.2 Pengertian Oral Diagnosis

Menurut Kerr (1974) dalam Kristiani dkk (2010) diagnosis dalam kedokteran gigi dapat diartikan sebagai penentuan jenis penyakit yang diderita pasien. Pengertian lainnya adalah cara-cara pemeriksaan untuk menentukan suatu diagnosa. Mengidentifikasi kelainan–kelainan yang berhubungan dengan gigi dan jaringan sekitarnya dengan jalan menanyakan, memeriksa, dan menyatukan gambaran penyakit yang terlihat dengan faktor-faktor yang diperoleh dari wawancara tersebut yang dapat membedakan dari penyakit yang lain.

2.6.3 Jenis-jenis Penyakit Gigi

Berdasarkan modul Penyakit Gigi dan Mulut oleh Kristiani dkk (2010) dipaparkan beberapa jenis penyakit gigi diantaranya adalah sebagai berikut.

2.6.3.1 Penyakit Jaringan Keras Gigi

1. Kehilangan Jaringan Keras Gigi/ Keausan
 - a. EROSI : Hilangnya jaringan keras gigi sebagai akibat dari proses kimia yang tidak melibatkan bakteri.
 - b. ATRISI : Hilangnya jaringan keras gigi pada bagian incisal dan oklusal dari permukaan gigi yang berlawanan, dapat disebabkan karena fungsi pengunyahan ataupun karena kebiasaan buruk (*bruxism*).

- a. ABRASI: Hilangnya jaringan keras gigi yang disebabkan karena proses mekanis seperti pada penggunaan sikat gigi, pemakaian tusuk gigi yang salah, pada kebiasaan pangur/ gusar.
- c. ABFRAKSI : Hilangnya jaringan keras gigi yang terjadi pada daerah servikal labial/ bukal gigi orang dewasa, penyebabnya biasanya karena *fatigue* (kelelahan gigi), fraktur dan deformasi dari struktur gigi sebagai akibat dari tekanan biomekanis.

2. Kehilangan Jaringan Keras Gigi Karena Karies

Karies adalah penyakit pada jaringan keras gigi yang disebabkan oleh kerja mikroorganisme pada karbohidrat yang dapat diragikan. Karies ditandai dengan adanya demineralisasi mineral-mineral email dan dentin, diikuti kerusakan bahan organik.

2.6.3.2 Penyakit Jaringan Pulpa

Berikut merupakan macam-macam penyakit gigi pada jaringan pulpa.

- 1. Peradangan Jaringan Pulpa Reversibel: suatu kondisi inflamasi pulpa ringan sampai sedang yang disebabkan oleh rangsangan, pulpa bisa kembali pada keadaan tidak terinflamasi setelah rangsangan dihilangkan.
- 2. Peradangan Jaringan Pulpa Ireversibel: Suatu keadaan inflamasi pulpa yang persisten, dapat simtomatik atau asimtomatik dan disebabkan oleh suatu rangsangan.

3. Degenerasi Pulpa : Degenerasi umumnya ditemukan pada gigi orang tua, tetapi dapat juga ditemukan pada gigi orang muda yang disebabkan adanya iritasi ringan yang persisten, seperti pada degenerasi kalsifik pulpa.
4. Nekrosis Pulpa: Nekrosis adalah kematian pulpa, dapat sebagian atau seluruhnya. Nekrosis meskipun suatu akibat inflamasi, dapat juga terjadi setelah injuri traumatik yang pulpanya rusak sebelum terjadi reaksi inflamasi.

2.6.3.3 Penyakit/ kelainan jaringan penyangga gigi dan jaringan lunak mulut

Berikut merupakan penyakit pada jaringan penyangga gigi dan jaringan lunak mulut.

1. Pembesaran Gingiva

Salah satu bentuk perubahan patologis dari gingiva yang mengalami *Hyperplasia* (bertambah besarnya jaringan karena terjadi proliferasi atau bertambahnya sel) atau *Hypertropi* (pembesaran jaringan disebabkan ukuran sel yang membesar).

2. Peradangan Gingiva (Gingivitis)

Peradangan pada gingiva yang menunjukkan adanya tanda-tanda penyakit/ kelainan pada gingival. Gingivitis disebabkan oleh plak dan dipercepat adanya faktor iritasi lokal dan sistemik.

Gusi yang berdarah akibat sikat gigi atau napas yang berbau kurang sedap dapat merupakan gejala gingivitis. Penyakit ini diakibatkan terselipnya plak atau sisa-sisa makanan di celah antara gigi sehingga terjadi peradangan pada gusi. Luka-luka kecil yang terjadi di permukaan gusi yang meradang tersebut

menyebabkan gusi mulai berdarah. Bakteri-bakteri yang hidup dengan subur dalam plak tersebut akan mengeluarkan gas sewaktu waktu mencerna makanannya. Jika dihasilkan dalam jumlah yang cukup banyak gas ini akan tercium sebagai napas yang kurang sedap. Gingivitis yang dibiarkan berlarut-larut dapat juga berkembang menjadi penyakit periodontitis (Hutapea, 2006).

3. Peradangan jaringan penyangga (Periodontitis)

Peradangan dari jaringan penyangga gigi yang meliputi gingiva, serabut-serabut periodontal, sementum dan tulang alveolar sebagai akibat lanjut dari gingivitis yang tidak dirawat. Periodontitis disebabkan oleh iritasi local dan traumatik oklusi.

Periodontitis berarti peradangan pada jaringan yang mendukung gigi. Periodontitis disebabkan oleh bakteri yang sebelumnya telah menyebabkan gingivitis. Peradangan ini akan menghancurkan serat-serat pengikat gigi sehingga gigi akan tidak terpancang lagi dengan kuat dan akhirnya terlepas. Selanjutnya pembengkakan pada jaringan gusi yang disertai pembentukan nanah bisa menyebar ke sekitarnya sehingga gigi lain juga bisa ikut terlepas (Hutapea, 2006).

Sedangkan dalam buku Gigi dan Mulut oleh Julianti (2008) dijelaskan beberapa penyakit gigi yaitu sebagai berikut.

1. Karies : kerusakan jaringan keras gigi yang disebabkan oleh asam yang ada dalam karbohidrat melalui perantara mikroorganisme yang ada dalam saliva.
2. Kelainan jaringan penyangga (jaringan periodontal):
 - a. gingival,

- b. ligament,
 - c. periodontal,
 - d. sementum,
 - e. tulang alveolus.
3. Kalkulus : karang gigi yang disebut juga kalkulus atau tartar adalah lapisan kerak berwarna kuning yang menempel pada gigi dan terasa kasar, yang dapat menyebabkan masalah pada gigi.
4. Gingivitis : penyakit periodontal stadium awal berupa peradangan pada gingiva, termasuk penyakit paling umum yang sering ditemukan pada jaringan mulut.
5. Periodontitis: periodontitis terjadi jika gingivitis menyebar ke struktur penyangga gigi. Periodontitis merupakan salah satu penyebab utama lepasnya gigi pada dewasa dan pada lanjut usia.
6. Kelainan gingiva akibat penyakit darah : hemofilia.
7. Erupsi gigi : erupsi gigi adalah proses perkembangan gigi yang bergerak dari posisi benih gigi menembus alveolar kedalam rongga mulut, dan beroklusi dengan gigi antagonisnya.

2.7 *Hypertext Preprocessor (PHP)*

PHP merupakan akronim dari PHP yaitu *Hypertext Preprocessor*. PHP merupakan bahasa pemrograman berbasis *web* yang sudah sering digunakan. Terdapat perbedaan antara *web* yang menggunakan PHP dan *web* yang hanya sekedar menggunakan HTML saja. Hal tersebut dapat dilihat pada proses saat *web server* memenuhi permintaan client untuk menampilkan halaman *web*.

Pada halaman *web* yang hanya menggunakan HTML, server langsung mengirimkan halaman yang diminta oleh client dalam bentuk script HTML. Sedangkan pada *web* yang menggunakan PHP sebelum *server* mengirimkan script HTML kepada *client*, *server* membaca terlebih dahulu *script* PHP yang ada pada *server* tersebut kemudian mengirimkan hasil dari *script* PHP tersebut kepada client berupa HTML (Kadir, 2007).

2.8 Database dan MySQL

Database adalah kumpulan data yang tersusun secara sistematis sehingga dapat memudahkan pengguna dalam mengakses dan mengelola serta untuk mendapatkan informasi yang efektif dan efisien. *Database* dapat berdiri sendiri dan dapat juga digabung menjadi satu kesatuan, penggabungan antar *database* disebut dengan relasi, sebagai contoh data *costumer* digabung dengan data *order*. Untuk menggabungkan data harus ada penghubung yang disebut dengan indeks. Syarat dari indeks adalah tidak boleh ada data yang sama dan data yang digabung masing-masing harus mempunyai indeks yang sama.

DBMS (*Database Management System*) merupakan *software* yang banyak digunakan dalam pengolahan data. Contoh manajemen *database* adalah sistem manajemen *database* relasi. Komponen dalam sistem manajemen *database* relasi adalah sebagai berikut.

1. *Database* sebagai tempat untuk menyimpan data yang terstruktur, data tersimpan dalam tabel, tabel tersebut terdiri dari baris dan kolom yang di dalamnya terdapat *field* dan *record*.

2. *System management* adalah *software* yang digunakan untuk mengelola *database* tersebut.

Relasi adalah hubungan antara tabel- tabel yang ada di dalam database. Fungsi MySQL dapat dikatakan sebagai *interpreter query*, karena setiap menggunakan *query* SQL (perintah SQL) harus meletakkannya di dalam fungsi ini. Dengan kata lain, SQL tidak dapat dijadikan tanpa adanya fungsi MySQL. MySQL termasuk jenis *relational database management system* (RDBMS). Sehingga istilah seperti tabel, baris dan kolom tetap digunakan dalam MySQL. Pada MySQL, SQL merupakan kependekan *Structured Query Language*. SQL digunakan untuk berkomunikasi dengan sebuah *database*. SQL adalah bahasa yang meliputi perintah-perintah untuk menyimpan, menerima, memelihara, dan mengatur akses ke *database* serta digunakan untuk memanipulasi dan menampilkan data dari *database* (Puspitasari, 2011).

2.9 Unified Modeling Language (UML)

Unified Modeling Language (UML) adalah keluarga notasi grafis yang didukung oleh meta-model tunggal, yang membantu pendeskripsian dan desain sistem perangkat lunak, khususnya sistem yang dibangun menggunakan pemrograman berorientasi objek (OO). Definisi ini merupakan definisi yang sederhana. Pada kenyataannya, pendapat orang – orang tentang UML berbeda satu sama lain. Hal ini dikarenakan oleh sejarahnya sendiri dan oleh perbedaan persepsi tentang apa yang membuat sebuah proses rancang – bangun perangkat lunak efektif. *Unified Modeling Language* (UML) merupakan standar yang relatif terbuka yang

dikontrol oleh *Object Management Group* (OMG), sebuah konsorsium terbuka yang terdiri dari banyak perusahaan. OMG dibentuk untuk membuat standar – standar yang mendukung interoperabilitas, khususnya interoperabilitas sistem berorientasi objek. OMG mungkin lebih dikenal dengan standar – standar COBRA (*Common Object Request Broker Architecture*). UML lahir dari penggabungan banyak bahasa permodelan grafis berorientasi objek yang berkembang pesat pada akhir 1980-an dan awal 1990-an. UML dibuat oleh Grady Booch, James Rumbaugh, dan Ivar Jacobson di bawah bendera Rational Software Corp. UML menyediakan notasi-notasi yang membantu memodelkan sistem dari berbagai perspektif. UML tidak hanya digunakan dalam pemodelan perangkat lunak, namun hampir dalam semua bidang yang membutuhkan pemodelan (Fowler, 2004).

UML dipaparkan dalam beberapa diagram sebagai berikut.

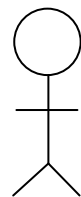
1. *Use Case Diagram*

Menurut Fowler (2004) *Use case Diagram* digunakan untuk menggambarkan sistem dari sudut pandang pengguna sistem tersebut (*user*), sehingga pembuatan *use case diagram* lebih dititikberatkan pada fungsionalitas yang ada pada sistem, bukan berdasarkan alur atau urutan kejadian. Sebuah *use case diagram* merepresentasikan sebuah interaksi antara aktor dengan sistem.

Berikut merupakan komponen yang ada dalam *use case diagram*.

a. Aktor

Pada dasarnya aktor bukanlah bagian dari *use case diagram*, namun untuk dapat terciptanya suatu *use case diagram* diperlukan aktor, dimana aktor tersebut mempresentasikan seseorang atau sesuatu (seperti perangkat atau sistem lain) yang berinteraksi dengan sistem yang dibuat. Sebuah aktor mungkin hanya memberikan informasi inputan pada sistem, hanya menerima informasi dari sistem atau keduanya menerima dan memberi informasi pada sistem. Aktor hanya berinteraksi dengan *use case*, tetapi tidak memiliki kontrol atas *use case*. Aktor digambarkan dengan *stick pan* seperti yang terdapat pada Gambar 2.5.

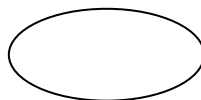


Actor

Gambar 2.5 Contoh aktor

b. *Use Case*

Use case adalah gambaran fungsionalitas dari suatu sistem, sehingga pengguna sistem paham dan mengerti kegunaan sistem yang akan dibangun. Bentuk *use case* dapat terlihat pada Gambar 2.6.



Use Case

Gambar 2.6. Contoh *use case*

Ada beberapa relasi yang terdapat pada *use case* diagram yaitu sebagai berikut.

1. *Association*, menghubungkan link antar element.
2. *Generalization*, disebut juga pewarisan (*inheritance*), sebuah elemen dapat merupakan spesialisasi dari elemen lainnya.
3. *Dependency*, sebuah element bergantung dalam beberapa cara ke element lainnya.
4. *Aggregation*, bentuk *association* dimana sebuah elemen berisi elemen lainnya.

Berikut ini merupakan tipe relasi yang mungkin akan terjaadi pada *use case* diagram.

1. <<*include*>>, yaitu kelakuan yang harus terpenuhi agar sebuah *event* dapat terjadi, dimana pada kondisi ini sebuah *use case* adalah bagian dari *use case* lainnya.
2. <<*extends*>>, yaitu kelakuan yang hanya berjalan di bawah kondisi tertentu seperti menggerakkan peringatan.
3. <<*communicates*>>, merupakan pilihan selama asosiasi hanya tipe *relationship* yang dibolehkan antara aktor dan *use case*.

2. Activity Diagram

Activity Diagram menggambarkan rangkaian aliran dari aktivitas, digunakan untuk mendeskripsikan aktivitas yang dibentuk dalam suatu operasi sehingga

dapat juga digunakan untuk aktivitas lainnya (Fowler, 2004). Notasi *Activity Diagram* dapat dilihat pada Tabel 2.2.

Tabel 2.2 Notasi *Activity Diagram* (Meildy, 2014).






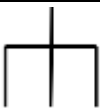


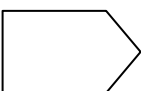

Simbol	Keterangan
	Titik Awal
	Titik Akhir
	<i>Activity</i>
	Pilihan untuk mengambil keputusan
	<i>Fork</i> : digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau untuk menggabungkan dua kegiatan parallel menjadi satu
	<i>Rake</i> : menunjukkan adanya dekomposisi
	Tanda Waktu
	Tanda penerimaan
	Tanda pengiriman
	Aliran akhir (<i>flow final</i>)

Diagram ini sangat mirip dengan *flowchart* karena memodelkan *workflow* dari satu aktivitas ke aktivitas lainnya atau dari aktivitas ke status. Pembuatan *activity diagram* pada awal pemodelan proses dapat membantu memahami keseluruhan proses. *Activity diagram* juga digunakan untuk menggambarkan interaksi antara beberapa *use case* (Fowler, 2004).

3. *Class Diagram*

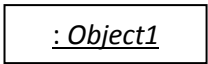
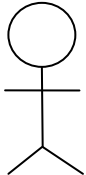
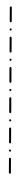
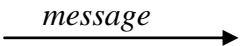
Class adalah sebuah spesifikasi yang akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class Diagram* menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungan satu sama lain seperti pewarisan, asosiasi, dan lain-lain. Hubungan antar *class*: Asosiasi, yaitu hubungan statis antar *class*. Umumnya menggambarkan *class* yang memiliki atribut berupa *class* lain, atau *class* yang harus mengetahui eksistensi *class* lain.

1. Agregasi, yaitu hubungan yang menyatakan bagian (“terdiri atas”).
2. Pewarisan, yaitu hubungan hirarki antar *class*. *Class* dapat diturunkan dari *class* lain dan mewarisi semua atribut dan metode *class* asalnya serta bisa menambahkan fungsionalitas baru. Sehingga *class* tersebut disebut anak dari *class* yang diwarisinya.
3. Hubungan dinamis, yaitu rangkaian pesan (*message*) yang di- *class* dari satu *class* kepada *class* lain. Hubungan dinamis dapat digambarkan dengan menggunakan *sequence* diagram yang akan dijelaskan kemudian (Fowler, 2004).

4. *Sequence Diagram*

Sequence Diagram menggambarkan interaksi antara sejumlah objek dalam urutan waktu. Kegunaannya untuk menunjukkan rangkaian pesan yang dikirim antara objek juga interaksi antar objek yang terjadi pada titik tertentu dalam eksekusi sistem. (Fowler, 2014). Notasi *Sequence Diagram* dapat dilihat pada Tabel 2.3.

Tabel 2.3 Notasi *Sequence Diagram* (Meildy, 2014).

Simbol	Nama	Keterangan
	<i>Object</i>	<i>Object</i> merupakan <i>instance</i> dari sebuah <i>class</i> dan dituliskan tersusun secara horizontal. Digambarkan sebagai sebuah <i>class</i> (kotak) dengan nama obyek didalamnya yang diawali dengan sebuah titik koma
	<i>Actor</i>	<i>Actor</i> juga dapat berkomunikasi dengan <i>object</i> , maka <i>actor</i> juga dapat diurutkan sebagai kolom. Simbol <i>Actor</i> sama dengan simbol pada <i>Actor Use Case Diagram</i> .
	<i>Lifeline</i>	<i>Lifeline</i> mengindikasikan keberadaan sebuah <i>object</i> dalam basis waktu. Notasi untuk <i>Lifeline</i> adalah garis putus-putus vertikal yang ditarik dari sebuah obyek
	<i>Message</i>	<i>Message</i> , digambarkan dengan anak panah horizontal antara <i>Activation</i> . <i>Message</i> mengindikasikan komunikasi antara objek-objek.

III. METODOLOGI PENELITIAN

3.1 Waktu dan Pelaksanaan

Penelitian ini dilakukan di Jurusan Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung, Rumah Sakit Umum Pusat Mohammad Hoesin Palembang dan Klinik Kosasih Rawat Inap Pasar Tugu Bandar Lampung. Waktu penelitian dilakukan pada periode semester genap tahun ajaran 2016/2017.

3.2 Alat Pendukung

Adapun alat pendukung yang dibutuhkan dalam penelitian ini adalah sebagai berikut.

A. Perangkat Keras

Perangkat keras yang digunakan selama penelitian adalah berupa laptop yang memiliki spesifikasi sebagai berikut.

- a. Hp Pavilion g4 Notebook PC.
- b. *Processor Intel ® CoreTM i3-2350M CPU@ 2.30GHz.*
- c. *RAM 1990 MB.*

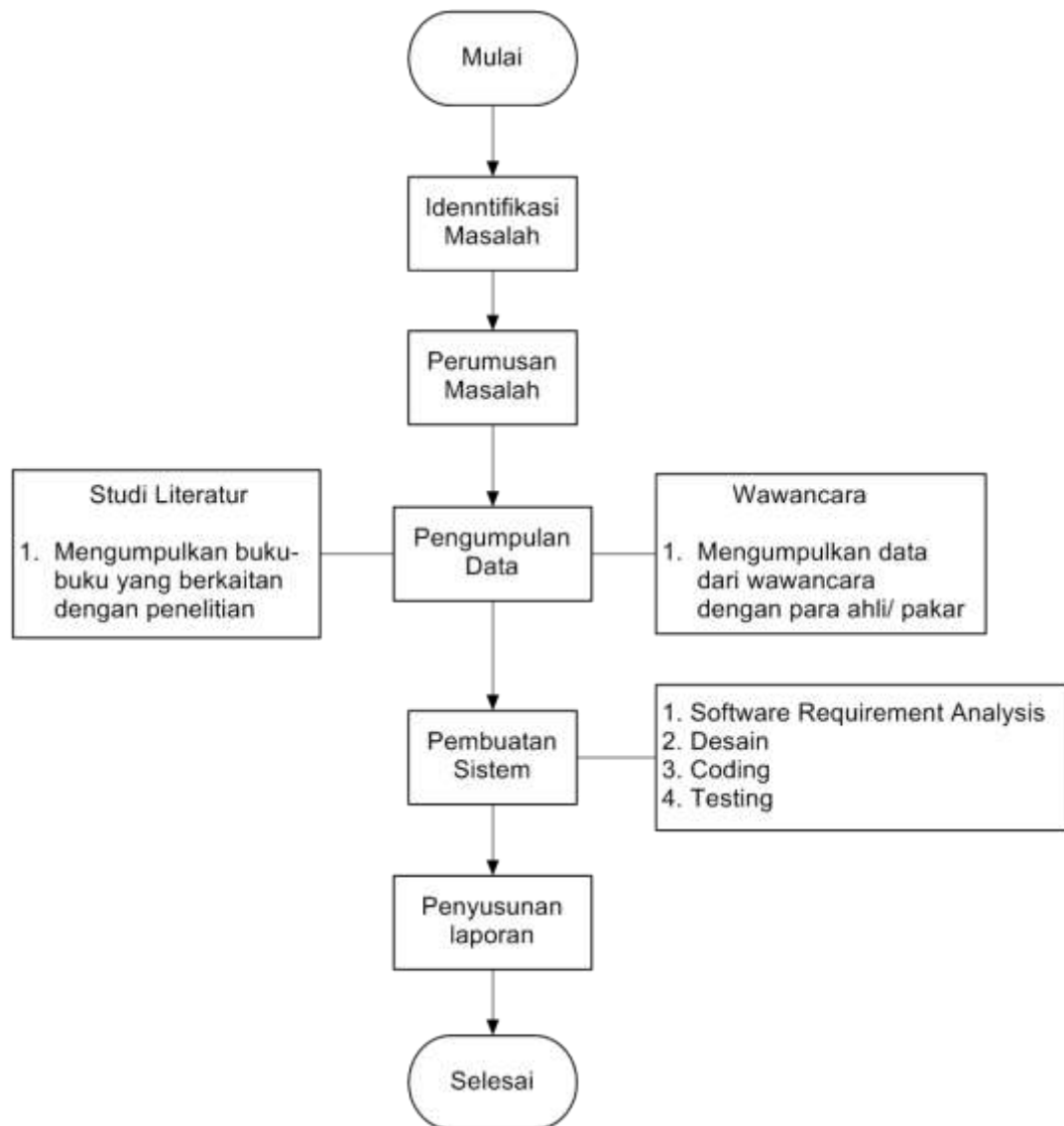
B. Perangkat Lunak

Perangkat lunak yang digunakan dalam pembuatan sistem informasi monitoring adalah sebagai berikut.

- a. Sistem operasi: *Windows 7 Ultimate 32 bit*.
- b. Aplikasi:
 - *Notepad++ v6.7.7* sebagai *editor source code*,
 - *XAMPP 1.8.3* sebagai *database server*,
 - Web Browser *Google Chrome* untuk menguji coding yang telah dibuat,
 - *Adobe Photoshop CS6 [Portable]*, digunakan untuk pembuatan tampilan user interface dan editing atribut gambar.

3.3 Tahapan Penelitian

Tahapan penelitian merupakan langkah-langkah yang akan dilakukan peneliti selama penelitian berlangsung. Tahapan penelitian dimulai dari identifikasi masalah, perumusan masalah, pengumpulan data, pembuatan sistem, analisis hasil dan pengujian yang disajikan dalam Gambar 3.1.



Gambar 3.1 Tahapan Penelitian

3.3.1 Tahap Identifikasi Masalah

Identifikasi masalah adalah salah satu proses penelitian yang boleh dikatakan paling penting diantara proses lainnya. Identifikasi masalah adalah pengenalan masalah atau inventarisir masalah. Masalah dalam penelitian akan menentukan kualitas dari penelitian, bahkan juga menentukan apakah sebuah kegiatan bisa disebut penelitian atau tidak. Tahapan identifikasi masalah

merupakan tahapan dasar dimana pengidentifikasian dan analisa pada masalah-masalah yang ada. Tahap ini akan menghasilkan rumusan masalah, tujuan penelitian, manfaat penelitian dan juga batasan – batasan masalah.

3.3.2 Tahap Perumusan Masalah

Tahapan ini merupakan proses merumuskan dan membatasi masalah yang ingin diteliti. Perumusan masalah dibutuhkan untuk memfokuskan peneliti pada penelitiannya. Sehingga sistem yang dikerjakan tidak melebihi batas yang sebelumnya telah ditentukan.

3.3.3 Tahap Pengumpulan Data

Pada tahapan pengumpulan data terdapat dua metode yaitu sebagai berikut.

1. Studi Literatur

Metode studi literatur dipakai melalui berbagai literatur yang ada. Beberapa contoh literatur berasal dari jurnal, buku, atau dokumen yang berkaitan dengan penelitian yang dilakukan.

2. Wawancara

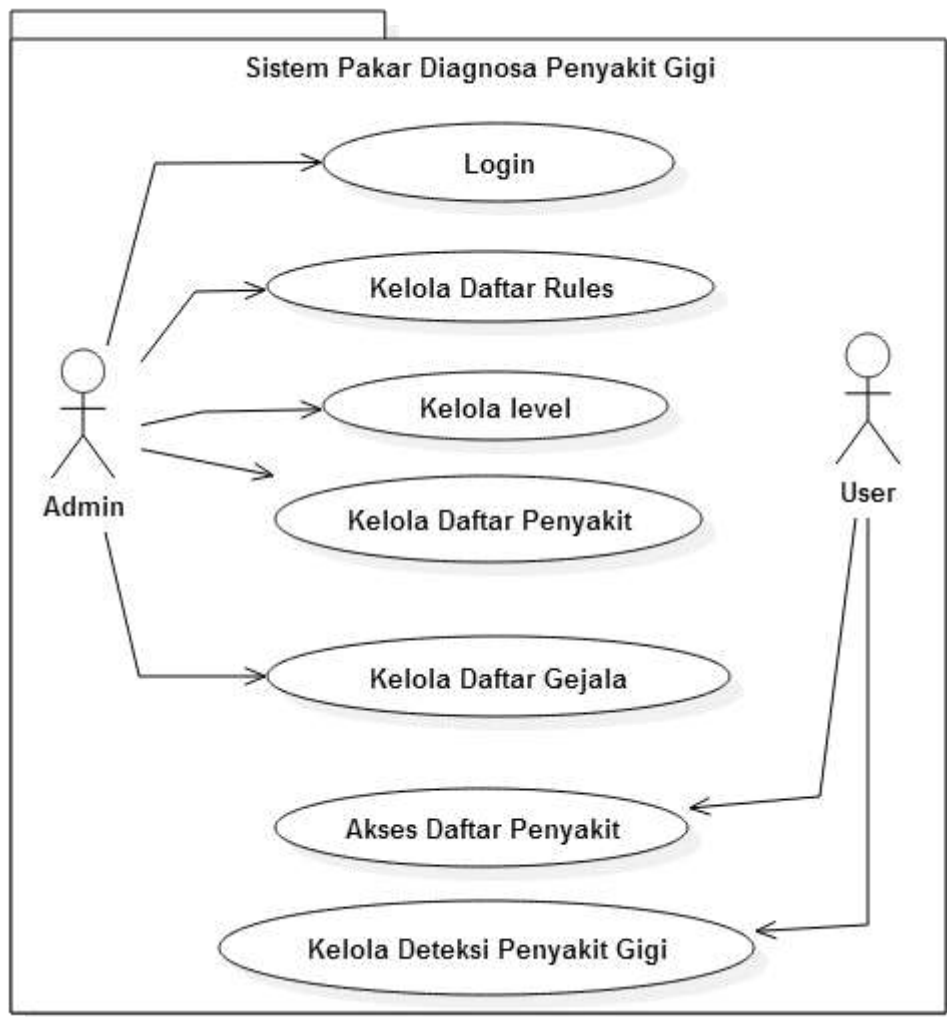
Metode ini dilakukan melalui proses tanya jawab atau *interview* dengan para ahli/pakar di bidangnya. Proses ini bertujuan untuk mendapatkan data yang lebih mendalam dan valid. Data yang didapatkan bisa jadi tidak terdapat pada tahapan studi literatur. Data yang didapatkan pada metode ini kemudian dikumpulkan menjadi satu untuk kemudian disusun menjadi baris aturan yang digunakan dalam sistem pakar.

3.3.4 Tahap Pembuatan Sistem

Dalam tahap pembuatan sistem digunakan empat tahapan sebagai berikut.

3.3.4.1 *Software Requirement Analysis*

Pada proses analisis ini dilakukan penjabaran kebutuhan para *user* yang mungkin akan dibutuhkan dalam membangun sistem pakar penyakit gigi ini. Dalam melakukan analisis maka dirancang sebuah *usecase* untuk menggambarkan suatu *interface* dari sisi *user* yang dijabarkan dalam Gambar 3.2.



Gambar 3.2 Use Case Diagram Sistem Pakar Diagnosa Penyakit Gigi

Berdasarkan gambar diatas maka dapat dijelaskan bahwa terdapat dua *level user* pada sistem ini yaitu seorang admin yaitu seorang pakar, dan juga seorang pengguna yaitu masyarakat publik. Untuk mengakses sistem, admin perlu melakukan *login* dalam sistem. Kemudian admin dapat melakukan kelola daftar aturan-aturan yang digunakan dalam mendiagnosa penyakit, serta dapat mengelola daftar penyakit yang ada maupun yang belum dibuat sama sekali. Pada setiap gejala dalam sebuah penyakit, pakar dapat mengelola urutan *level* pada gejala dalam suatu penyakit. Disamping itu juga admin dapat mengelola daftar gejala dalam sistem. Sedangkan masyarakat umum sebagai *user* dapat mengakses daftar penyakit dan dapat mengelola deteksi penyakit gigi untuk mengetahui penyakit yang mereka alami.

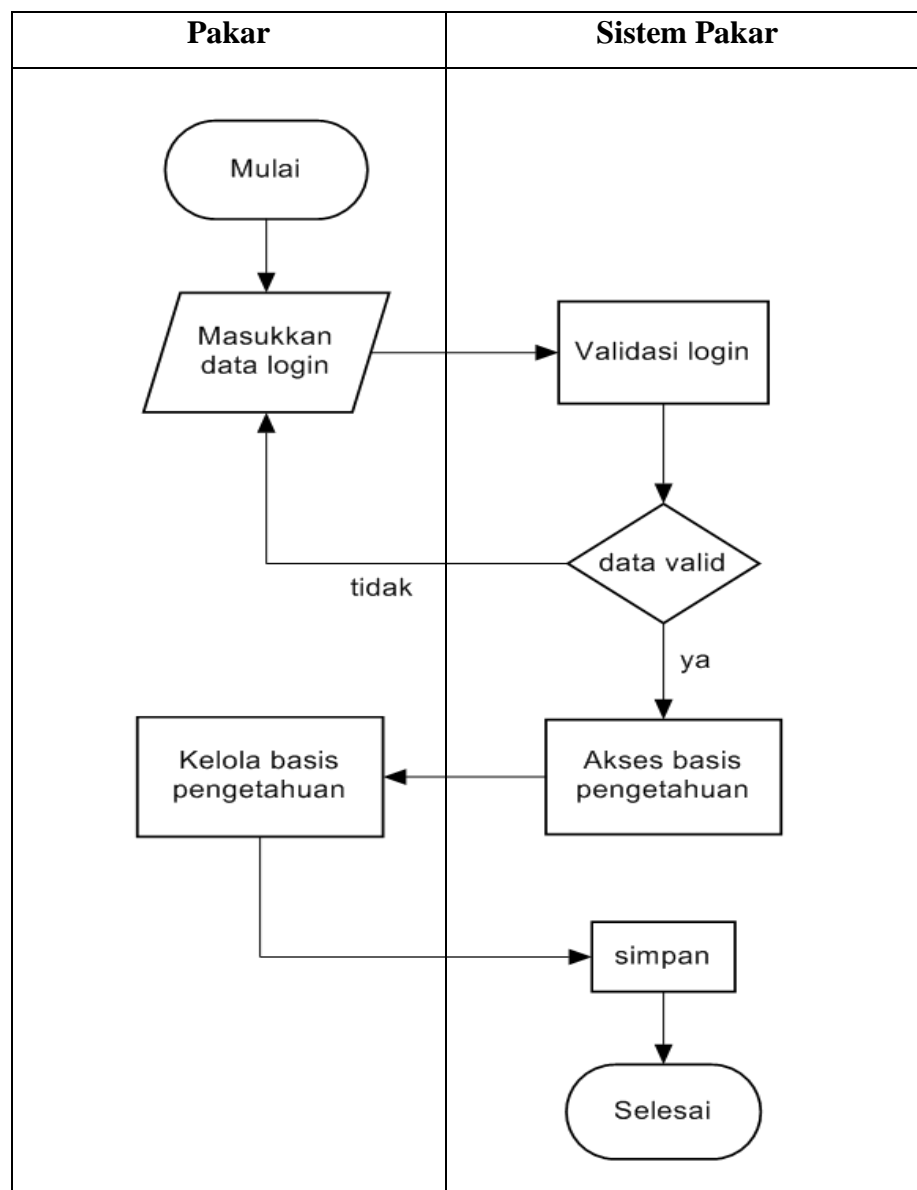
3.3.4.2 Design

Proses ini akan dimulai dengan suatu perancangan sistem dimana desain disini dibuat untuk mengimplementasikan kebutuhan yang sebelumnya telah disebutkan dalam *usecase diagram*. Desain akan dibuat menggunakan UML (*Unified Modelling Language*) yang terdiri atas empat model yaitu *flowchart*, *activity diagram*, *sequence diagram*, dan *class diagram*.

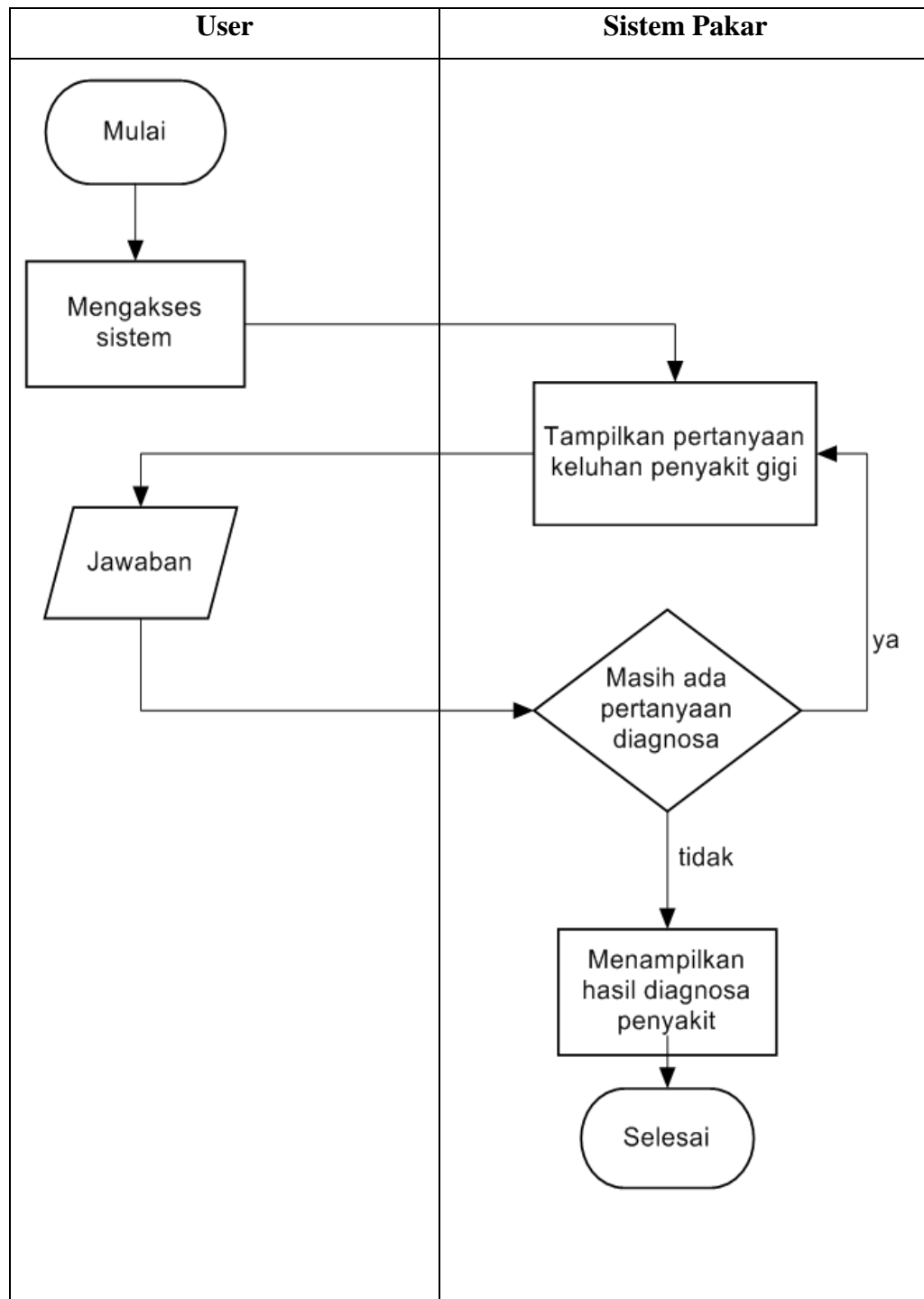
3.3.4.2.1 Flowchart

Sistem pakar diagnosa penyakit gigi memiliki dua aktor yaitu seorang admin dan juga pengguna. Secara garis besar, segala kegiatan yang dilakukan oleh seorang aktor dalam sistem pakar dapat digambarkan dalam bentuk *flowchart*. Suatu bisnis

proses dapat ditunjukkan juga dalam penggambaran *flowchart*. Proses bisnis merupakan suatu kumpulan proses yang berisi kumpulan aktifitas yang saling berelasi satu dengan yang lainnya untuk menghasilkan penyelesaian masalah suatu produk. Proses bisnis yang baik harus mempermudah pemahaman proses-proses di dalamnya. Proses bisnis dalam sistem pakar diagnosa penyakit gigi terfokus pada proses bisnis pakar sebagai admin dan juga proses identifikasi. Proses bisnis keduanya dapat dilihat pada Gambar 3.3 dan Gambar 3.4.



Gambar 3.3 Proses bisnis oleh pakar



Gambar 3.4 Proses bisnis identifikasi

Berdasarkan Gambar 3.3 dijelaskan bahwa pakar disini sebagai administrator dapat melakukan proses bisnis dengan mengakses situs web pakar terlebih dahulu dan menginputkan data login berupa *username* dan *password* yang nantinya akan

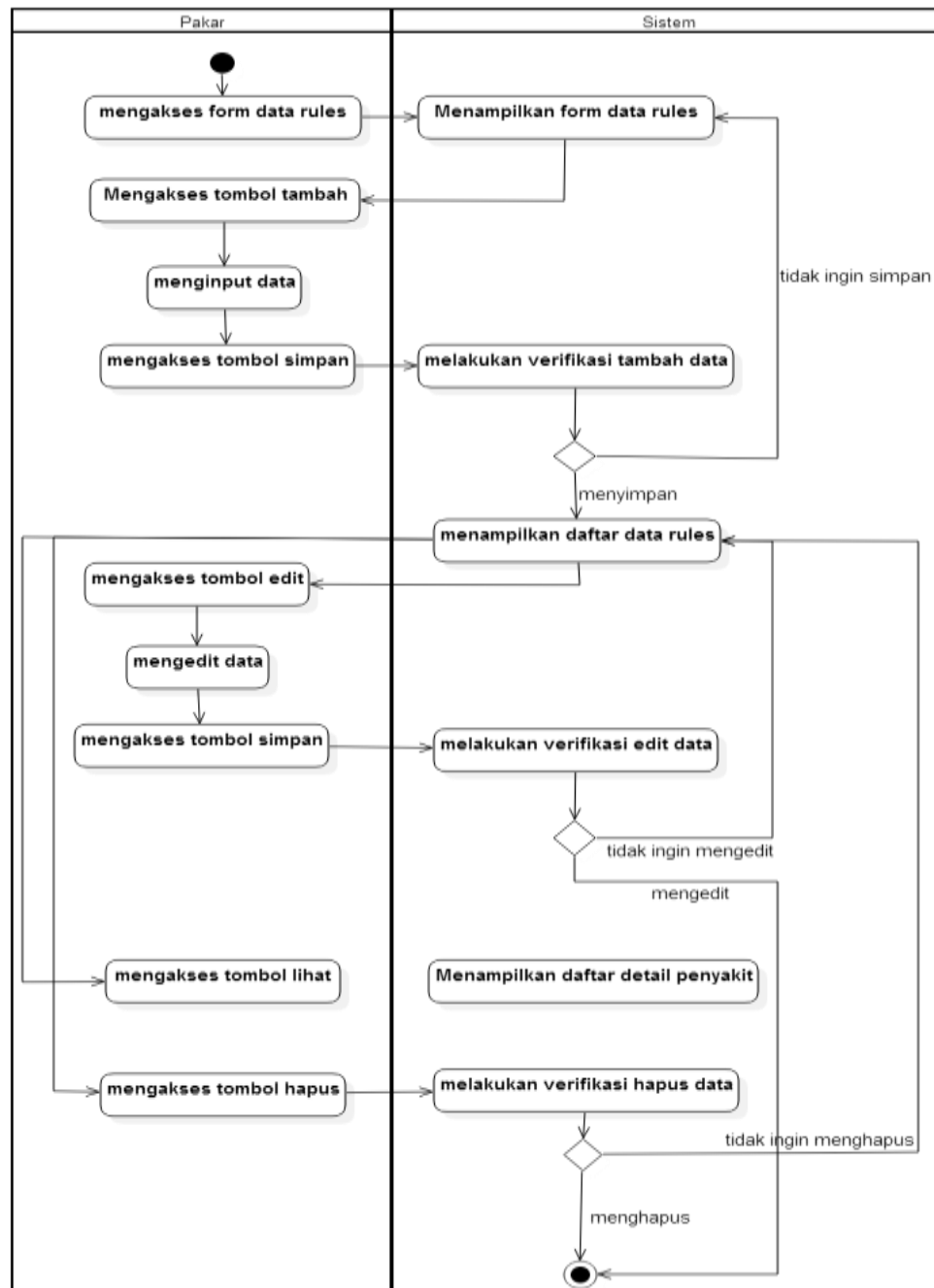
divalidasi oleh sistem pakar sendiri. Selanjutnya sistem akan menampilkan hal yang berkaitan dengan basis pengetahuan kepada pakar. Disini pakar dapat mengelola basis pengetahuan yang dimilikinya untuk kemudian dapat disimpan dalam sistem.

Sedangkan pada proses bisnis identifikasi oleh *user* dalam Gambar 3.4 dapat dijelaskan bahwa *user* dapat mengakses sistem pakar penyakit gigi yang kemudian sistem akan menampilkan berbagai pertanyaan berkaitan dengan gejala penyakit gigi yang mungkin dialami oleh *user*. Selanjutnya *user* akan menjawab setiap pertanyaan yang ada hingga selesai untuk kemudian sistem akan menampilkan hasil diagnosa penyakit gigi yang sedang dialami *user*. Sistem pakar akan menampilkan gambar, deskripsi penyakit, serta gejala yang memang berkaitan dengan penyakit tersebut, serta solusi yang dapat diperoleh *user*.

3.3.4.2.2 Activity Diagram

Activity diagram adalah diagram yang menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. Diagram ini sangat mirip dengan *flowchart* karena memodelkan *workflow* dari satu aktivitas ke aktivitas lainnya atau dari aktivitas ke status. Pembuatan *activity diagram* pada awal pemodelan proses dapat membantu memahami keseluruhan proses. Dalam sistem pakar diagnosa penyakit gigi disajikan empat *activity* yang dipaparkan pada Gambar 3.5 sampai dengan Gambar 3.8.

3.3.4.2.1 Activity Diagram Kelola Daftar Rules

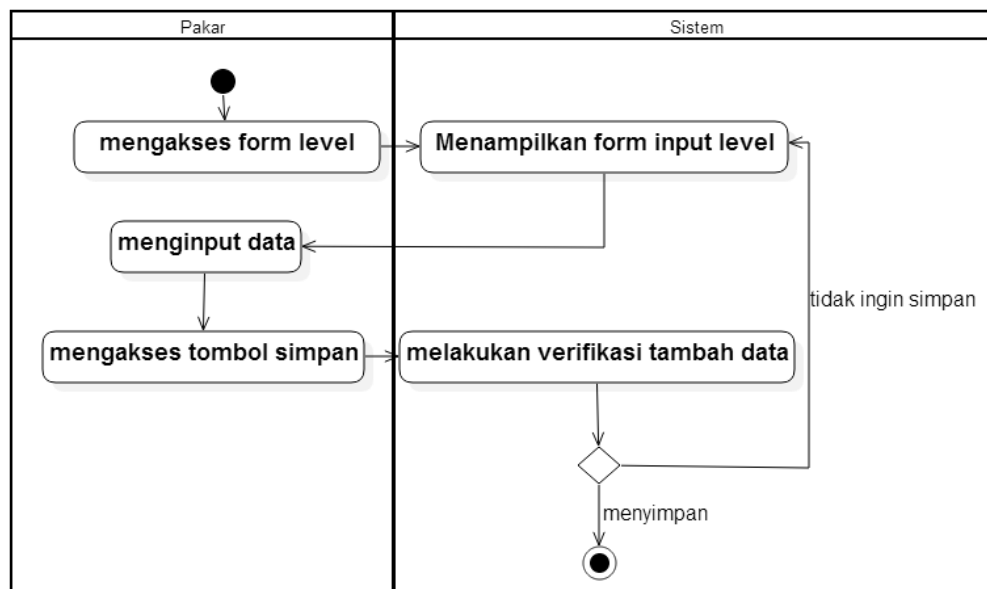


Gambar 3.5 Activity Diagram Kelola Daftar Rules

Diagram aktifitas kelola daftar *rules* berisi pakar sebagai admin yang dapat mengakses form data aturan-aturan pada sistem yang kemudian sistem akan menampilkan formnya. Selanjutnya pakar dapat mengakses tombol tambah untuk

memasukkan data aturan berbasis pengetahuannya. Maka setelah itu pakar dapat menyimpannya dan sistem akan melakukan verifikasi penambahan data. Jika pakar ingin mengedit salah satu aturan maka ia dapat mengakses tombol edit untuk kemudian merubah data yang diinginkan dan kemudian menyimpannya dalam sistem. Untuk menghapus, pakar dapat mengakses tombol hapus dan sistem akan melakukan verifikasi penghapusan data.

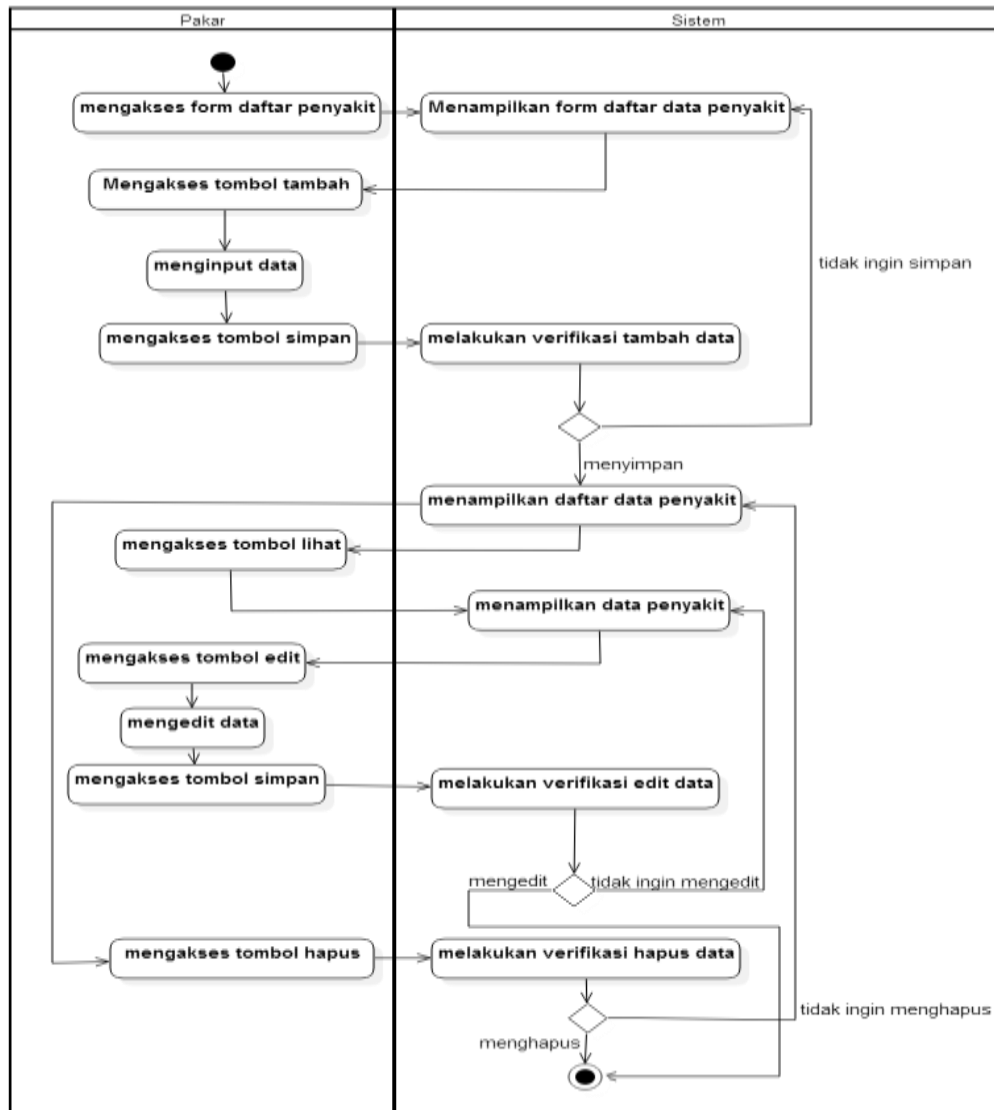
3.3.4.2.2 Activity Diagram Kelola Level



Gambar 3.6 Activity Diagram Kelola Level

Gambar 3.6 menjelaskan tentang Activity Diagram pakar saat mengelola urutan *level* dengan mengakses form *level* terlebih dahulu. Sistem akan menampilkan form input *level*. Selanjutnya pakar dapat menginputkan urutan *level* untuk setiap gejala dalam sebuah penyakit yang diinginkan. Kemudian admin dapat mengakses tombol simpan dan sistem akan melakukan verifikasi penyimpanan data *level*.

3.3.4.2.3 Activity Diagram Kelola Daftar Penyakit

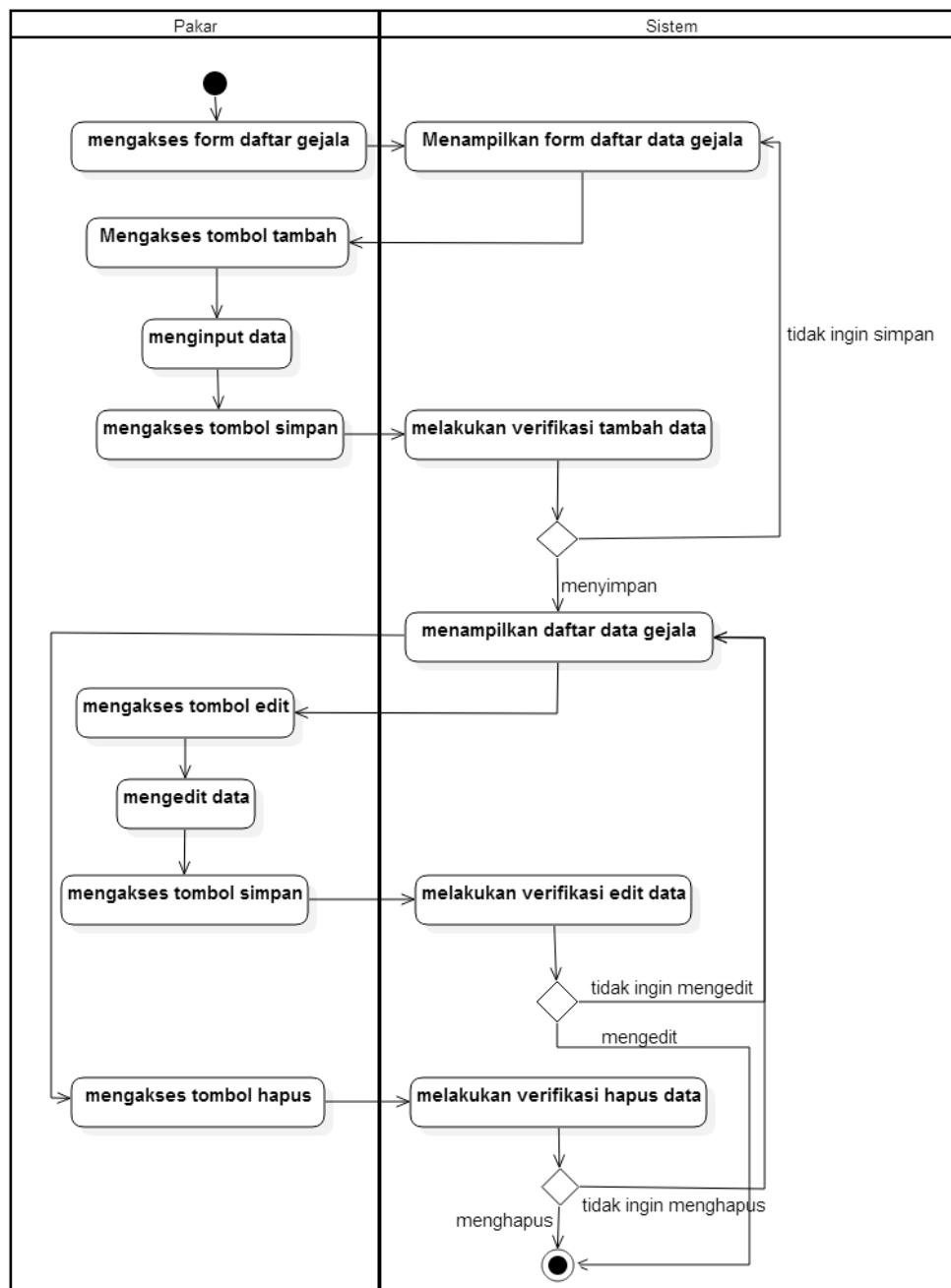


Gambar 3.7 Activity Diagram Kelola Daftar Penyakit

Gambar 3.7 menjelaskan bahwa diagram aktifitas kelola daftar penyakit meliputi pakar yang dapat mengakses daftar penyakit. Sistem akan menampilkannya dan pakar dapat menambahkan data melalui tombol tambah yang ada dan memasukkan data penyakit yang diinginkan. Selanjutnya pakar dapat menyimpan untuk kemudian sistem akan memverifikasi penambahan data pada sistem. Pada tampilan daftar penyakit, pakar dapat melihat penjelasan suatu penyakit gigi

dengan mengakses tombol lihat dan dapat merubahnya dengan mengakses tombol edit. Untuk menghapus daftar penyakit yang diinginkan maka pakar dapat mengakses tombol hapus. Sistem akan langsung memverifikasi penghapusan data.

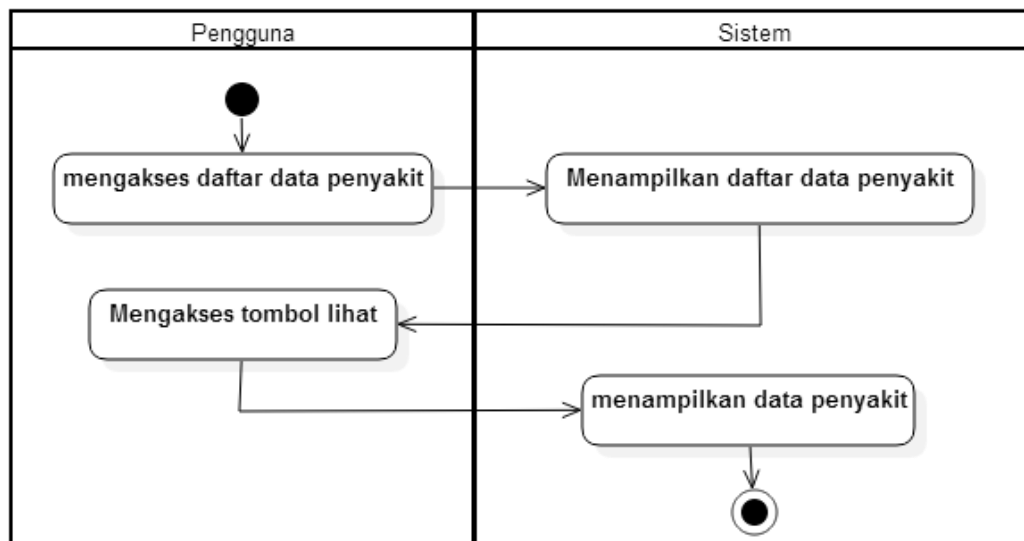
3.3.4.2.4 Activity Diagram Kelola Daftar Gejala



Gambar 3.8 Activity Diagram Kelola Daftar Gejala

Berdasarkan Gambar 3.8 diatas dijelaskan bahwa pakar dapat mengakses daftar gejala dan dapat mengakses tombol tambah setiap ingin menambah gejala baru. Setelah menginputkan gejala, pakar dapat mengakses tombol simpan. Maka sistem akan melakukan verifikasi tambah data dan menampilkan daftar semua gejala. Selanjutnya pakar dapat mengakses tombol edit dan mengedit data yang diinginkan. Pakar pun dapat mengakses tombol simpan dan sistem akan melakukan verifikasi edit data. Disamping itu juga pakar dapat menghapus gejala dengan menggunakan tombol hapus dan sistem akan menghapus data gejala yang diinginkan.

3.3.4.2.2.5 Activity Diagram Akses Daftar Penyakit

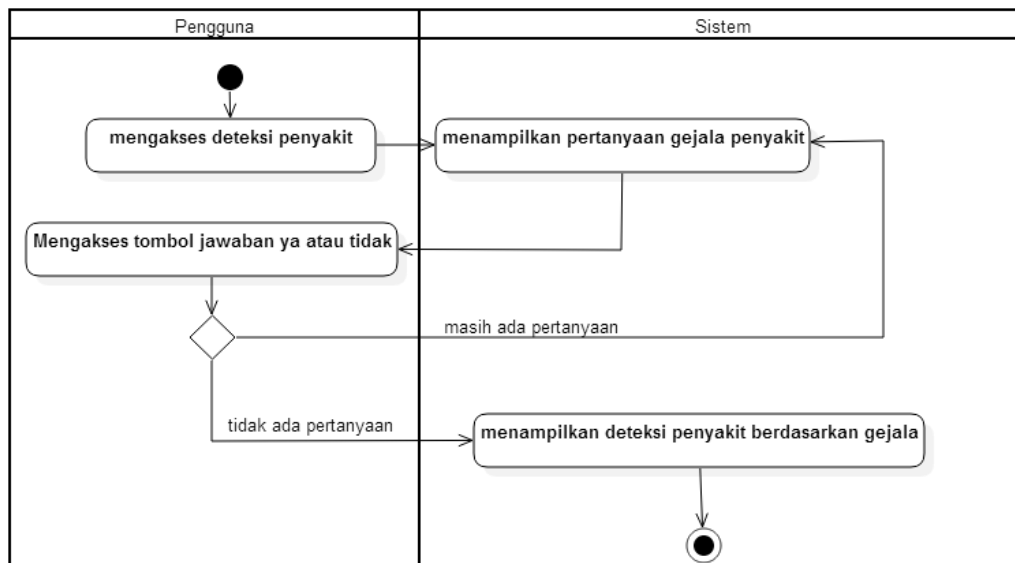


Gambar 3.9 Activity Diagram Akses daftar penyakit

Gambar 3.9 menjelaskan bahwa pengguna memiliki hak akses terhadap daftar penyakit. Sistem akan langsung menampilkan daftar data penyakit kepada pengguna *web*. Selanjutnya untuk melihat penjelasan detail tentang penyakit maka

pengguna dapat mengakses tombol lihat. Maka sistem akan menampilkan data penyakit secara terinci mulai dari nama penyakitnya, gambar, deskripsi, gejala, dan solusi penyembuhannya.

3.3.4.2.2.6 Activity Diagram Kelola Deteksi Penyakit Gigi



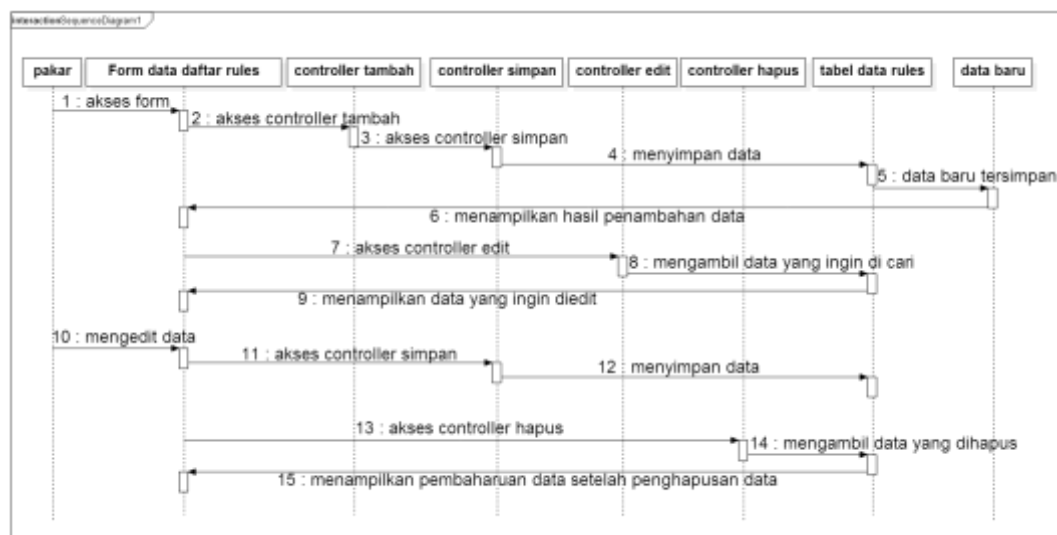
Gambar 3.10 Activity Diagram Kelola Deteksi Penyakit Gigi

Gambar 3.10 menjelaskan bahwa pengguna dapat mengakses deteksi penyakit gigi pada sistem. Sistem akan menampilkan pertanyaan mengenai gejala penyakit paling umum kepada pengguna. Pengguna dapat menjawab pertanyaan yang ada dengan mengakses tombol jawaban ya atau tidak. Dari jawaban yang ada, sistem akan mengidentifikasi aturan yang berlaku di dalamnya. Jika masih ada aturan maka sistem akan kembali menampilkan pertanyaan berbeda sesuai alur aturan yang ada. Jika aturan telah sampai tahap akhir, maka sistem akan berhenti menampilkan pertanyaan dan langsung menampilkan deteksi penyakit gigi berdasarkan gejala yang dialami oleh pengguna.

3.3.4.2.3 Sequence Diagram

Sequence Diagram adalah diagram yang menggambarkan interaksi antar objek dan mengindikasikan komunikasi diantara banyak objek tersebut. Diagram ini menunjukkan serangkaian pesan yang dipertukarkan oleh objek yang melakukan tugas atau aksi tertentu.

3.3.4.2.3.1 Sequence Diagram Kelola Daftar Rules

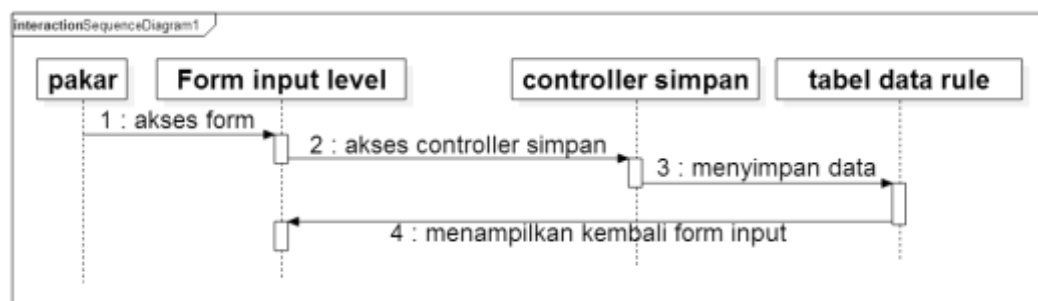


Gambar 3.11 *Sequence Diagram Kelola Daftar Rules*

Berdasarkan Gambar 3.11 dapat dijelaskan bahwa pakar dapat mengelola daftar *rules* dengan cara mengakses form data *rules* dan mengakses *controller* tambah. Setelah itu *controller* simpan dapat diakses untuk kemudian menyimpan data dalam tabel data *rules* sehingga data baru tersimpan. Maka form data *rules* dapat menampilkan hasil penambahan data baru. Pada form data *rules*, pakar juga dapat mengakses *controller* edit untuk merubah data *rules* yang ada. Maka data akan diambil dari tabel data *rules* dan kemudian data akan ditampilkan pada form.

Pakar bisa melakukan pengeditan dan mengakses *controller* simpan untuk menyimpan data *rules* yang telah diedit dalam tabel data *rules*. Selain itu pakar juga dapat menghapus *rules* yang ada dengan mengakses *controller* hapus dan data yang ingin dihapus langsung diambil dari tabel untuk kemudian dihapus dan form akan menampilkan pembaharuan data setelah penghapusan.

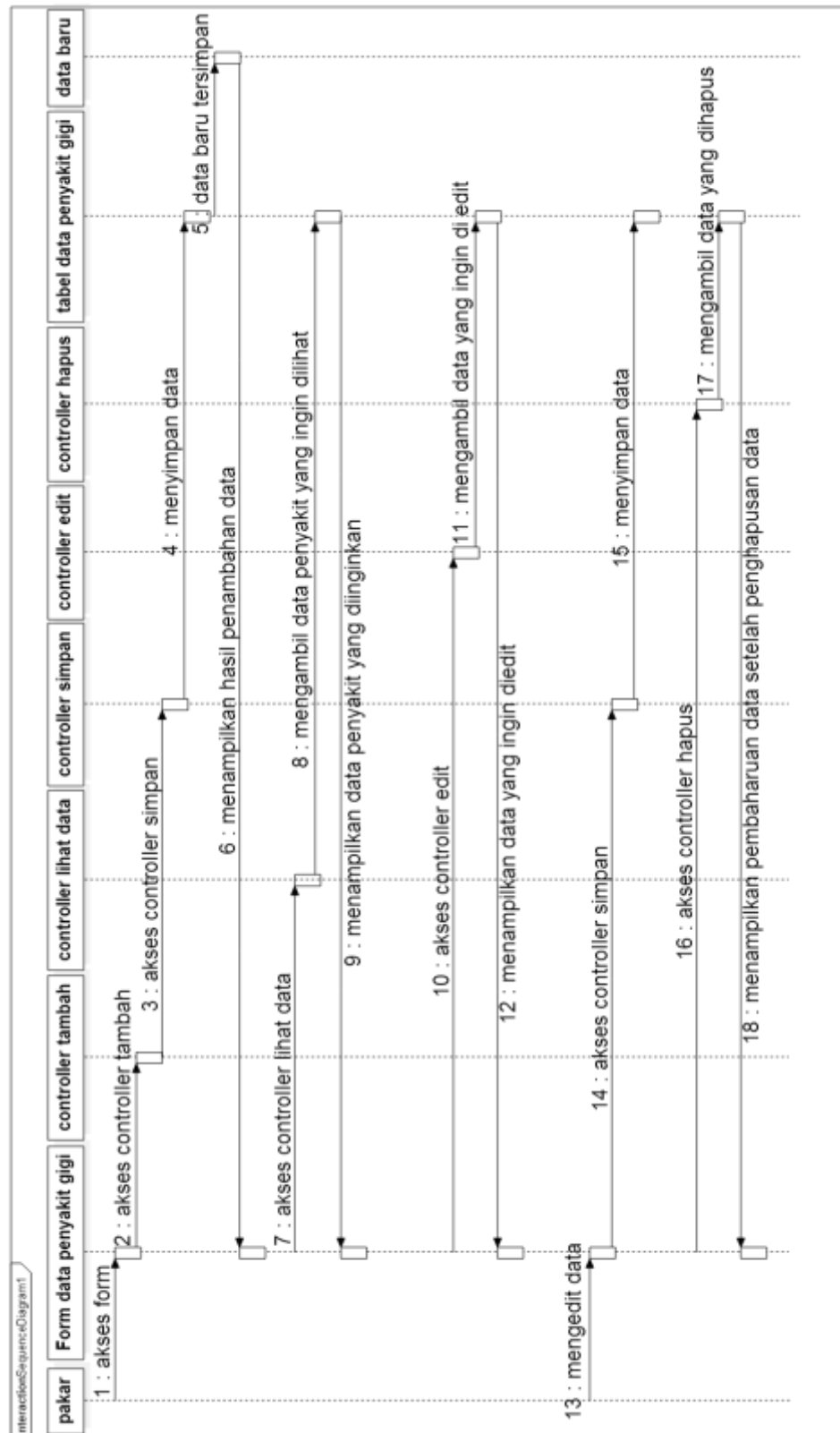
3.3.4.2.3.2 *Sequence Diagram Kelola Level*



Gambar 3.12 *Sequence Diagram Kelola Level*

Berdasarkan Gambar 3.12 dapat dijelaskan bahwa pakar dapat mengelola pengurutan *level* dalam sistem yang dimulai dari pengaksesan input *level*. Disini pakar dapat melakukan input urutan *level* pada setiap gejala dari sebuah penyakit. Selanjutnya pakar dapat mengakses *controller* simpan untuk kemudian data *level* dapat disimpan dalam tabel *rule* dan kembali pada inputan data *level*.

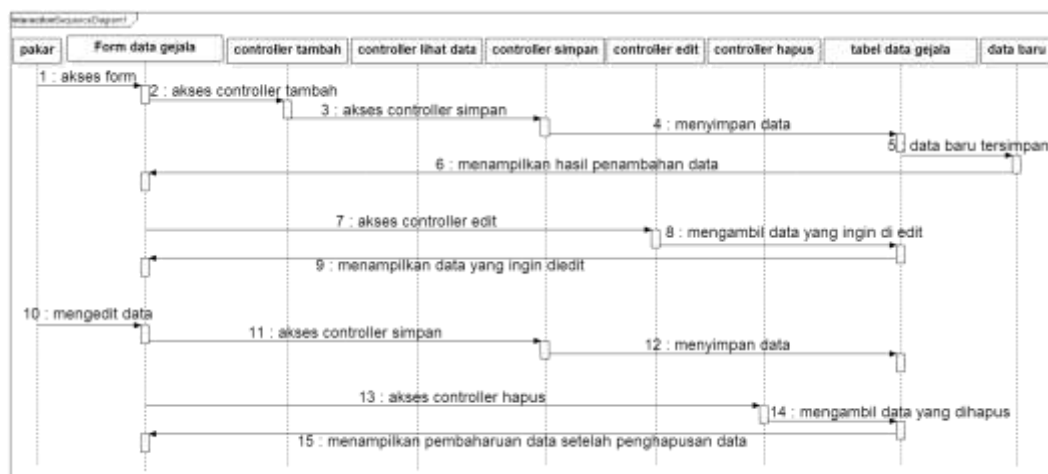
3.3.4.2.3.3 Sequence Diagram Kelola Daftar Penyakit



Gambar 3.13 Sequence Diagram Kelola Daftar Penyakit

Gambar 3.13 menjelaskan bahwa pakar dapat mengelola daftar penyakit gigi yang ada dengan mengakses form daftar penyakit gigi. Selanjutnya untuk menambah data baru pakar dapat mengakses *controller* tambah dan *controller* simpan untuk menyimpan data baru dalam tabel daftar penyakit gigi sehingga terbentuk data baru. Selanjutnya data baru akan ditampilkan dalam form daftar penyakit gigi. Untuk melihat data penyakit gigi, pakar dapat mengakses *controller* lihat dan data penyakit yang ingin dilihat akan diambil dari tabel data penyakit gigi. Data penyakit gigi akan disajikan dalam form daftar penyakit. Selain itu pakar dapat mengedit data penyakit dengan mengakses *controller* edit dan menyimpan pembaharuan melalui akses pada *controller* simpan. *Controller* hapus dapat diakses pula untuk menghapus data penyakit yang ada. Form daftar penyakit akan menampilkan pembaharuan data setelah penghapusan data.

3.3.4.2.3.4 Sequence Diagram Kelola Daftar Gejala

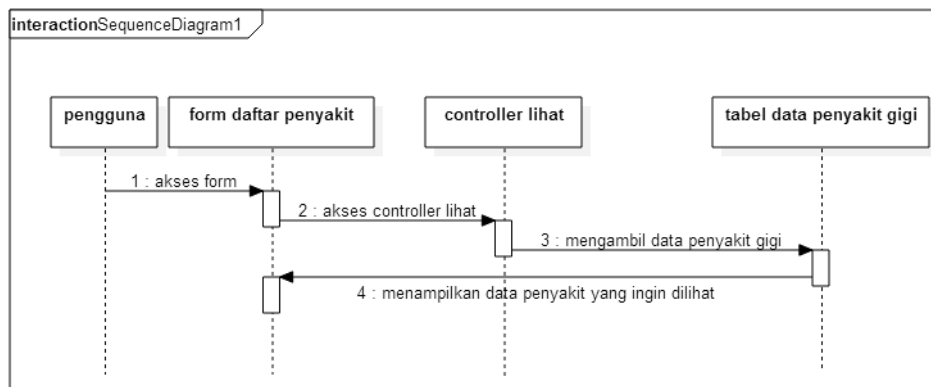


Gambar 3.14 Sequence Diagram Kelola Daftar Gejala

Gambar 3.14 menjelaskan bahwa pakar mengakses daftar gejala dan menambah gejala baru melalui *controller* tambah. Selanjutnya data gejala baru dapat

disimpan melalui akses *controller* simpan. Maka sistem akan menyimpan data gejala dalam tabel data gejala dan sistem akan menampilkan daftar gejala penyakit secara keseluruhan. Selanjutnya untuk mengedit gejala penyakit, pakar dapat mengakses *controller edit* dan sistem akan mengambil data yang ingin diedit dalam tabel data gejala. Setelah merubah data yang diinginkan, pakar dapat mengakses *controller* simpan untuk menyimpan data gejala ke dalam tabelnya. Sedangkan untuk menghapus gejala, maka pakar dapat mengakses *controller* hapus. Sistem akan mengambil data untuk dihapus dan akan kembali menampilkan pembaharuan data setelah penghapusan data.

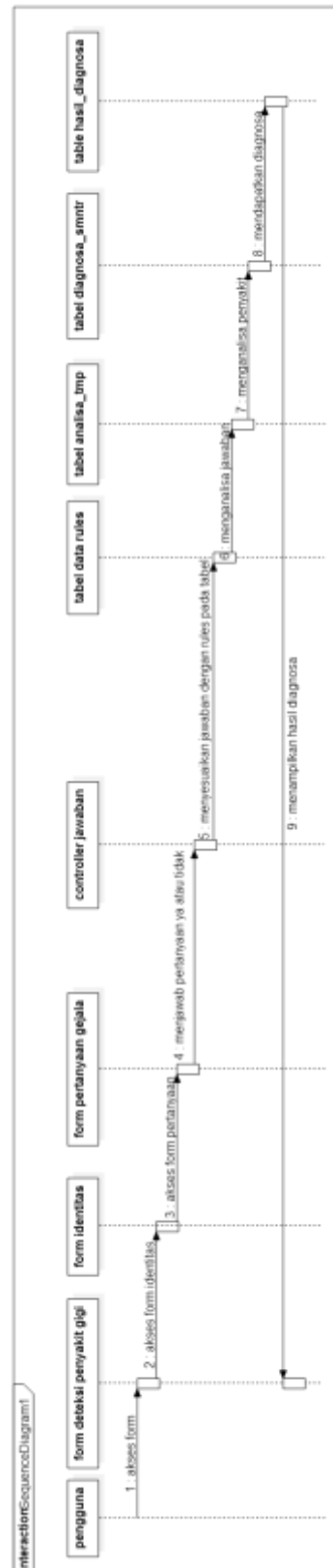
3.3.4.2.3.5 Sequence Diagram Akses Daftar Penyakit



Gambar 3.15 Sequence Diagram Akses Daftar Penyakit Gigi

Gambar 3.15 menjelaskan bahwa pengguna memiliki hak akses pada daftar penyakit gigi dengan mengakses form daftar penyakit. Pengguna dapat melihat penjelasan penyakit gigi lebih detail maka ia dapat mengakses *controller* lihat dan data penyakit yang diinginkan akan diambil dari tabel data penyakit gigi. Kemudian data penyakit gigi yang ingin dilihat akan ditampilkan pada form daftar penyakit gigi.

3.3.4.2.3.6 Sequence Diagram Kelola Deteksi Penyakit Gigi

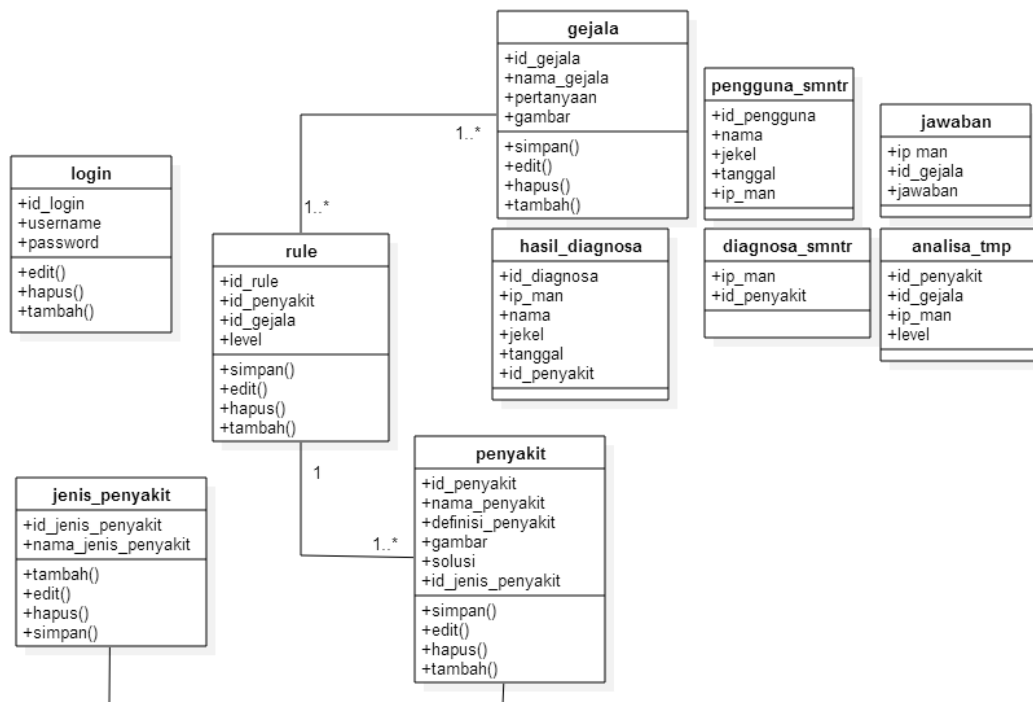


Gambar 3.16 Sequence Diagram Kelola Deteksi Penyakit Gigi

Gambar 3.16 menjelaskan rangkaian kegiatan pengguna dalam mengelola deteksi penyakit gigi. Pertama, pengguna dapat mengakses form deteksi penyakit gigi dan dapat langsung mengakses form identitas terlebih dulu, selanjutnya akan langsung ke *form* pertanyaan mengenai gejala-gejala yang dirasakan saat menderita sakit pada gigi. Pengguna dapat menjawab pertanyaan berupa ya atau tidak menggunakan *controller* jawaban yang tersedia untuk setiap pertanyaan gejala. Setiap jawaban yang telah diakses oleh pengguna akan disesuaikan dengan *rules* yang ada pada tabel. Selanjutnya akan dilakukan analisa diagnosa dan melalui proses diagnosa sementara sampai pada akhirnya dapat disimpulkan penyakit gigi berdasarkan *rules* yang kemudian ditampilkan dalam form deteksi penyakit gigi.

3.3.4.2.4 Class Diagram

Class diagram merupakan suatu diagram untuk memberikan gambaran sistem secara statis dan relasi antar mereka. Beberapa diagram akan menampilkan kelas-kelas dan relasinya yang dapat dibuat menjadi beberapa diagram sesuai dengan keinginan terhadap sistem yang dibangun. Seperti *class diagram* Sistem Pakar Diagnosa Penyakit Gigi yang ditujukan pada Gambar 3.17 di bawah ini.



Gambar 3.17 Class Diagram Sistem Pakar Diagnosa Penyakit Gigi dan Jaringan Penyangga Gigi

Gambar 3.17 menjelaskan bahwa Sistem Pakar Diagnosa Penyakit Gigi dan Jaringan Penyangga Gigi memiliki *diagram class* yang terdiri atas kelas login, rule, penyakit, gejala, diagnosa, dan jenis penyakit.

1. Database memiliki kelas login untuk menyimpan username dan password saat sistem diakses.
2. Kelas gejala memiliki hubungan asosiasi terhadap rule dengan nilai kardinalitas banyak ke banyak. Hal ini berarti banyaknya suatu gejala penyakit bisa memiliki banyak rule di dalamnya.
3. Kelas jenis penyakit memiliki hubungan asosiasi dengan kelas penyakit. Dapat diartikan bahwa suatu penyakit bisa memiliki jenis penyakit yang berbeda-beda.

4. Kelas penyakit memiliki hubungan asosiasi dengan kelas *rule* dengan nilai kardinalitas banyak ke satu. Bisa ada banyak penyakit yang memiliki satu *rule* dalam *database*.
5. Kelas *rule* memiliki asosiasi terhadap kelas gejala dan penyakit. Kelas *rule* akan mengumpulkan berbagai gejala dan penyakit untuk disatukan dalam *rule* yang ada.
6. Kelas *pengguna_smntr* merupakan kelas yang menampung identitas sementara pengguna yang berisi nama, jenis kelamin, tanggal akses, dan ip *user*.
7. Kelas jawaban merupakan kelas yang menampung jawaban dari pertanyaan yang dijawab pengguna. Isi tabel ini berupa id gejala dan jawaban untuk setiap gejala.
8. Kelas *analisa_tmp* merupakan kelas yang menampung ip *user*, id penyakit, id gejala, dan level untuk sementara.
9. Kelas *diagnosa_smntr* menampung jawaban hasil diagnosa sementara sampai semua pertanyaan habis berupa kolom ip *user* dan id penyakit.
10. Kelas *hasil_diagnosa* menampung hasil akhir dari proses diagnosa.

3.3.4.2.5 Rancangan *Interface* (Antarmuka)

Rancangan antarmuka dibuat sebagai penghubung antara pakar dengan pengguna (*user*). Dalam sistem pakar diagnosa penyakit gigi, seorang pakar dan pengguna memiliki hak akses halaman yang berbeda dalam web. Gambar 3.18 sampai dengan Gambar 3.37 memaparkan halaman akses pakar dan pengguna.

1. Pakar

a. Halaman *Login*

Halaman *login* merupakan halaman yang pertama kali muncul ketika pakar sebagai admin mengakses sistem. Sebelum masuk admin dapat memasukkan *username* dan *password* terlebih dahulu. Sistem akan mengecek apakah *username* dan *password* yang dimasukkan sudah *valid* atau belum. Jika *valid* maka admin dapat mengakses sistem, jika tidak *valid* maka sistem akan kembali menampilkan halaman login. Perancangan form *login* ditampilkan pada Gambar 3.18.

The image shows a login form for a specialist (Pakar). It has a teal header with the word "login" in white. Below the header is a light blue background. In the center, there is a black silhouette of a person's head and shoulders inside a black circle. Below this, there are two input fields: "Username :" followed by a text box, and "Password :" followed by a text box. At the bottom, there is a dark grey button with the word "Login" in white.

Gambar 3.18 Rancangan halaman login pakar

b. Halaman Beranda

Halaman beranda merupakan halaman yang akan dituju pakar pertama kali saat mengakses sistem setelah halaman *login*. Rancangan beranda pakar disajikan dalam Gambar 3.19 di bawah ini. Beranda akan menampilkan beberapa menu *side bar* diantaranya adalah tentang, aturan, penyakit,gejala, *level* dan logout.

Untuk pakar yang baru saja mengakses sistem ini dapat mengunduh buku panduan penggunaan sistem yang tersedia pada halaman ini.



Gambar 3.19 Rancangan halaman beranda pakar

c. Halaman Tentang

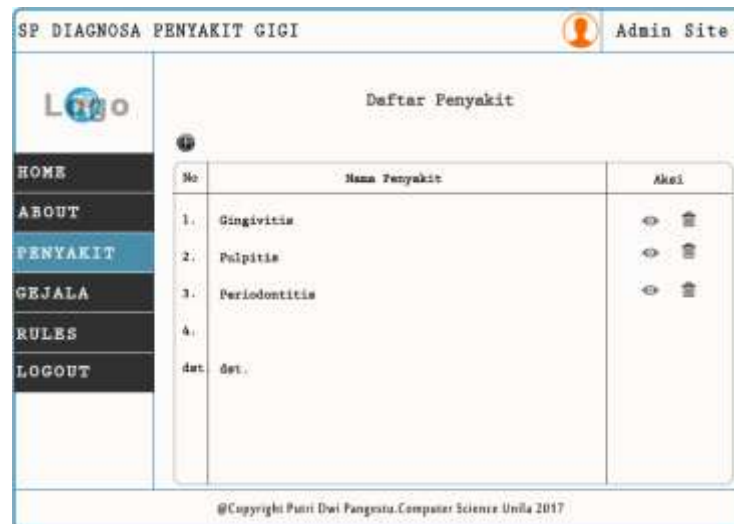
Halaman tentang akan menampilkan informasi secara umum mengenai sistem pakar diagnosa penyakit gigi dan pengembang sistem. Perancangan halaman tentang disajikan dalam Gambar 3.20.



Gambar 3.20 Rancangan halaman tentang pakar

d. Halaman Penyakit

Halaman penyakit berisi daftar nama-nama penyakit gigi yang disajikan dalam sebuah tabel berisi nomor, nama penyakit serta aksi yang dapat dilakukan oleh admin. Admin dapat menambahkan daftar penyakit baru dengan mengakses tombol tambah dan melihat data dalam setiap penyakit yang ada dengan mengakses tombol lihat. Selain itu admin dapat mengakses tombol hapus untuk menghapus daftar penyakit yang ada pada sistem. Perancangan halaman penyakit disajikan dalam Gambar 3.21.



Gambar 3.21 Rancangan halaman daftar penyakit pakar

e. Halaman Tambah Penyakit

Untuk menambahkan daftar penyakit gigi baru, admin dapat mengakses tombol tambah yang nantinya akan membawa admin pada halaman tambah data penyakit baru. Perancangan tampilan tambah penyakit disajikan dalam Gambar 3.22. Pakar sebagai admin dapat mengisi nama penyakit baru, definisinya, dan solusi, serta meng-*upload* gambar sesuai dengan penyakit. Setelah selesai mengisi *form*, admin dapat mengakses tombol simpan untuk menyimpan data dalam *database* atau *cancel* untuk kembali ke halaman awal penyakit.

Gambar 3.22 Rancangan halaman tambah penyakit gigi baru pakar

f. Halaman Lihat Data Penyakit

Halaman lihat data penyakit dapat diakses oleh admin dengan menekan tombol lihat pada setiap daftar penyakit yang ada. Perancangan tampilan detail penyakit gigi disajikan dalam Gambar 3.23 dibawah. Admin dapat mengetahui penjelasan secara detail dari setiap penyakit gigi mulai dari gambar, nama penyakit, definisi, solusi, dan gejala dalam penyakit. Jika admin sudah selesai maka dapat menekan tombol ok untuk kembali pada halaman daftar penyakit gigi. Jika ingin merubah data penyakit yang ada maka admin dapat mengakses tombol edit.



Gambar 3.23 Rancangan halaman lihat data penyakit pakar

g. Halaman Edit Data Penyakit

Halaman edit data penyakit gigi akan muncul saat admin mengakses tombol edit setelah melihat data penyakit dalam Gambar 3.23 sebelumnya. Perancangan halaman edit penyakit disajikan dalam Gambar 3.24. Admin dapat merubah data penyakit mulai dari gambar, nama penyakit, definisi, solusi dan gejala yang sebelumnya dipilih. Jika pengeditan telah selesai maka admin dapat menekan tombol simpan dan akan kembali pada halaman lihat detail penyakit.

Gambar 3.24 Rancangan halaman edit data penyakit gigi pakar

h. Halaman Gejala

Halaman gejala akan menampilkan daftar gejala penyakit yang sudah diinputkan sebelumnya. Setiap gejala memiliki id masing-masing. Selain itu admin dapat mengedit ataupun menghapus gejala penyakit yang ada. Rancangan halaman gejala digambarkan pada Gambar 3.25.

No	Nama Penyakit	Pertanyaan	Gambar	Aksi
1.	Terasa ngilu			
2.	Gigi terkikis			
3.	Bau mulut			
4.	det. det.			

Gambar 3.25 Rancangan Halaman Gejala

i. Halaman Tambah Gejala

Halaman tambah gejala menampilkan form yang berisi id gejala, nama gejala, serta pertanyaan yang berkaitan dengan nama gejala dalam Gambar 3.26. Jika admin memiliki gambar yang terkait dengan gejala, dapat dimasukkan dengan mengakses tombol *select file*. Jika sudah selesai maka admin dapat menekan tombol simpan untuk menyimpan data gejala baru. Id gejala akan muncul secara otomatis pada form tambah gejala.



Gambar 3.26 Rancangan Halaman Tambah Gejala

j. Halaman Edit Gejala

Halaman edit gejala akan menampilkan data sebelumnya yang ingin kita rubah. Edit data gejala menampilkan id gejala, nama gejala, dan juga pertanyaan serta gambar yang terkait dengan gejala yang ada. Id gejala tidak dapat dirubah karena sudah menjadi milik permanen setiap gejala. Rancangan halaman edit gejala disajikan pada Gambar 3.27.

Gambar 3.27 Rancangan Halaman Edit Gejala

k. Halaman Aturan (*Rules*)

Halaman aturan akan menampilkan daftar aturan (*rules*) dari semua penyakit yang ada secara keseluruhan. Gambar 3.28 menampilkan id dan nama gejala atau aturan dari seluruh penyakit dalam sistem. Pakar dapat melakukan aksi untuk setiap penyakit yang ada dengan mengedit ataupun menghapusnya. Selain itu pakar dapat menambahkan aturan baru yang berkaitan dengan penyakit gigi di dalam sistem melalui tombol tambah yang terdapat di pojok kiri atas sistem.



Gambar 3.28 Rancangan halaman *rules* pakar

1. Halaman Tambah Aturan (*Rule*)

Gambar 3.29 menggambarkan rancangan halaman tambah aturan (*rule*) yang akan muncul apabila pakar mengakses tombol tambah untuk menambahkan gejala dan nama penyakit yang terdapat dalam sistem. Tampilan tambah aturan (*rule*) akan muncul secara *pop-out* pada layar. Untuk menambahkan *rule* baru, admin dapat mengisi kolom nama penyakit dan gejala yang sesuai.



Gambar 3.29 Rancangan halaman tambah aturan (*rule*) pakar

m. Halaman *Edit Aturan (Rule)*

Halaman edit aturan yang disajikan dalam Gambar 3.30 dapat diakses oleh admin ketika ingin merubah *rule* atau aturan yang sudah ada pada daftar aturan. Sistem akan menampilkan halaman edit secara *pop-out* dan admin dapat langsung merubah nama penyakit dan gejala yang sebelumnya telah dimasukkan.



Gambar 3.30 Rancangan halaman edit aturan (*rule*) pakar

n. Halaman *Level*

Halaman *level* yang disajikan dalam Gambar 3.31 merupakan perancangan untuk menambahkan tingkatan *level* pada setiap gejala dalam suatu penyakit. Pakar dapat memilih nama penyakit yang diinginkan dan ketika di klik cek maka sistem akan menampilkan seluruh gejala dari penyakit yang dipilih. Setelah itu pakar dapat melakukan pengurutan *level* dengan mengisi angka dalam kotak kosong untuk setiap gejala.

SP DIAGNOSA PENYAKIT GIGI Admin Site

Tambah Level

Pilih nama penyakit:

gejala yang sesuai

Level	Nama gejala
<input type="checkbox"/>	G001 Gusi Berdarah
<input type="checkbox"/>	G002 Darah memenuhi gusi
<input type="checkbox"/>	G003 Gusi terasa sakit

@Copyright Putri Dwi Pangesta.Computer Science Uinila 2017

Gambar 3.31 Rancangan halaman *level* pakar

2. Pengguna (*User*)

a. Halaman Beranda

Halaman beranda merupakan halaman pertama yang akan ditampilkan saat pengguna mengakses sistem sistem diagnosa penyakit gigi. Sistem akan menampilkan daftar menu *sidebar* diantaranya adalah menu penyakit, dan deteksi. Rancangan tampilan beranda untuk pengguna disajikan dalam Gambar 3.32.



Gambar 3.32 Rancangan halaman beranda pengguna

b. Halaman Penyakit

Halaman penyakit menampilkan daftar penyakit gigi yang terdapat dalam situs sebagai bahan pengetahuan bagi pengguna situs. Perancangan halaman penyakit gigi disajikan dalam Gambar 3.33. Halaman ini berisi tabel yang berisi kolom nomor, nama penyakit dan aksi lihat yang dapat diakses setiap pengguna situs untuk melihat secara detail penjelasan penyakit gigi yang ingin diketahuinya.



Gambar 3.33 Rancangan halaman daftar penyakit gigi pengguna

c. Halaman Lihat Data Penyakit Gigi

Pengguna situs diagnosa penyakit gigi dapat melihat secara detail data penyakit gigi yang diinginkannya dengan mengakses tombol lihat yang akan langsung mengantarkannya pada halaman lihat data penyakit yang akan tampil secara *pop-out*. Halaman ini berisi gambar, nama penyakit, definisi, solusi, serta gejala yang menyebabkan penyakit tersebut. Jika telah selesai pengguna dapat mengakses tombol ok untuk kembali ke halaman sebelumnya pada Gambar 3.33. Perancangan halaman lihat data detail penyakit gigi disajikan dalam Gambar 3.34.



Gambar 3.34 Rancangan halaman lihat data detail penyakit gigi pengguna

d. Halaman Deteksi

Pengguna dapat mengakses menu deteksi untuk memulai deteksinya terhadap penyakit gigi yang sedang dirasakannya. Pertama sistem akan menampilkan *form* identitas pada Gambar 3.35 yang harus diisi oleh pengguna. Selanjutnya sistem akan langsung menampilkan pertanyaan serta tombol keputusan untuk pengguna yang dirancang dalam tampilan halaman deteksi penyakit gigi pada Gambar 3.36. Pertanyaan akan dimulai dari yang paling umum menuju ke khusus. Kemudian setelah selesai bertanya dan menjawab maka situs akan menampilkan hasil deteksi penyakit gigi pengguna yang dirancang dalam Gambar 3.37 di bawah ini.

SP DIAGNOSA PENYAKIT GIGI

User

Logo

HOME

ABOUT

PENYAKIT

DIAGNOSA

Nama Pengguna :

Jenis Kelamin : ☐ Laki-laki ☐ Perempuan

Next Cancel

@Copyright Putri Dwi Pangestu.Computer Science Unila 2017

Gambar 3.35 Rancangan halaman deteksi penyakit gigi pengguna

SP DIAGNOSA PENYAKIT GIGI

User

Logo

HOME

ABOUT

PENYAKIT

DETEKSI

Pertanyaan

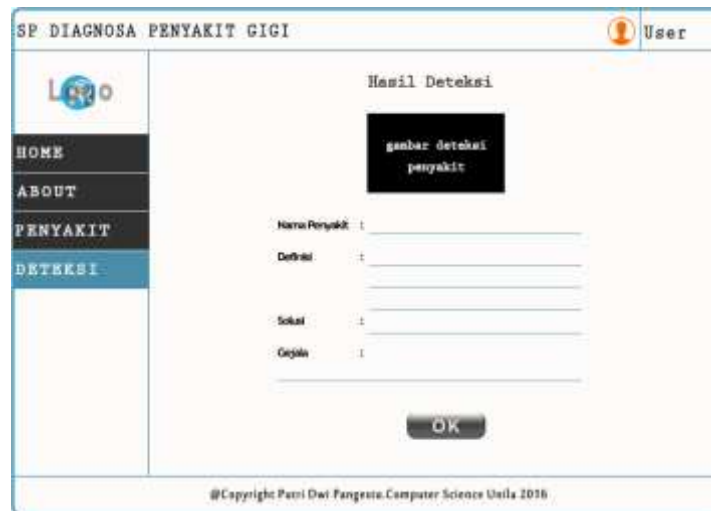
gambar gejala

Pertanyaan: Apakah ... ?

Ya Tidak

@Copyright Putri Dwi Pangestu.Computer Science Unila 2017

Gambar 3.36 Rancangan halaman deteksi penyakit gigi pengguna



Gambar 3.37 Rancangan halaman hasil deteksi penyakit gigi pengguna

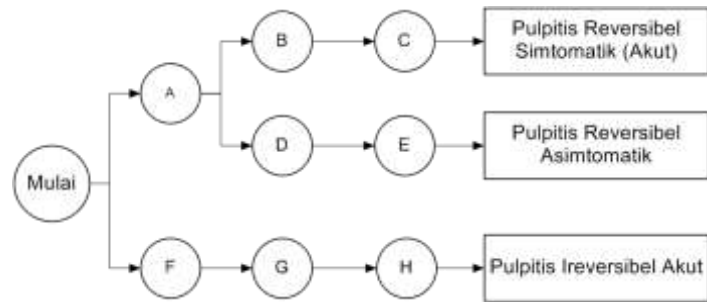
3.3.4.3 Coding

Setelah tahap perancangan sistem selesai, maka selanjutnya yang dilakukan adalah tahap *coding*. Dalam tahap ini sistem pakar diagnosa penyakit gigi akan dibangun dengan mengimplementasikan bahasa pemrograman PHP dan *Mysql*. Sedangkan metode penalaran atau inferensi yang digunakan dalam penelitian ini adalah metode *forward chaining*.

3.3.4.3.1 Metode *Forward Chaining*

Metode *forward chaining* merupakan teknik pencarian yang dimulai dengan fakta yang diketahui, kemudian mencocokkan fakta-fakta tersebut dengan bagian dari *rules IF-THEN*. Bila ada fakta **IF** yang cocok dengan bagian **IF**, maka *rule* tersebut dieksekusi. Pencocokan berhenti bila tidak ada lagi *rule* yang bisa dieksekusi.

Contoh studi kasus:



Inisialisasi gejala:

- A = Terasa sakit dengan rangsangan dingin, manis, asam
- B = Tidak ada rasa sakit spontan
- C = Rasa sakit hilang jika rangsangan dihilangkan
- D = Sakit spontan, sakit bila kavitas tertekan makanan
- E = Tetap sakit meskipun rangsang dihilangkan
- F = Tidak terasa sakit dengan rangsangan dingin, manis, asam
- G = Nyeri tajam menyengat
- H = Rasa sakit dapat hilang timbul secara spontan

Rule /aturan :

- **IF** A and B and C **THEN** Pulpitis Reversibel Simtomatik (Akut)
- **IF** A and D and E **THEN** Pulpitis Reversibel Asimtomatik
- **IF** F and G and H **THEN** Pulpitis Ireversibel Akut

Iterasi 1 : Apakah A bernilai benar (sesuai dengan fakta) ?

Jika A bernilai benar (sesuai dengan fakta) maka :

- Fakta : A
- *Rule* / aturan yang mungkin :

IF A and B and C **THEN** Pulpitis Reversibel Simtomatik (Akut)

IF A and D and E **THEN** Pulpitis Reversibel Asimtomatik

~~IF F and G and H THEN Pulpitis Ireversibel Akut~~

Iterasi 2 : Apakah B bernilai benar (sesuai dengan fakta) ?

Jika B benar (sesuai dengan fakta) maka :

- Fakta : A,B
- *Rule* / aturan yang mungkin :

IF A and B and C THEN Pulpitis Reversibel Simtomatik (Akut)

~~**IF A and D and E THEN Pulpitis Reversibel Asimtomatik**~~

Jika *rule*/aturan tinggal tersisa 1 maka, rule tidak dihapus dan akan di cek sampai

IF pada rule tersebut habis dan menghasilkan kesimpulan.

Iterasi 3 : Apakah C bernilai benar (sesuai dengan fakta)?

Jika C bernilai salah maka :

- Fakta : A, B
- *Rule* / aturan yang mungkin :

IF A and B and C THEN Pulpitis Reversibel Simtomatik (Akut)

Berdasarkan fakta yang diketahui yaitu A, dan B lalu fakta-fakta tersebut dicocokkan dengan *rule*/ aturan yang ada maka didapatlah kesimpulan yang paling mendekati yaitu Penyakit Pulpitis Reversibel Simtomatik (Akut).

3.3.4.4 Pengujian (*Testing*)

Tahapan *coding* yang sebelumnya dilakukan akan menghasilkan sistem pakar yang berguna untuk mendiagnosa suatu penyakit gigi. Untuk memastikan suatu sistem berjalan dengan semestinya maka dilakukan tahap pengujian sistem. Hal ini dilakukan agar tidak terjadi ketidaknyamanan saat orang banyak menggunakan

sistem. Pengujian pada sistem pakar diagnosa penyakit gigi akan dilakukan dalam pengujian internal dan eksternal.

3.3.4.4.1 Pengujian Internal

Peneliti akan melakukan pengujian internal dengan langsung memastikan fungsi sistem berjalan dengan baik. Selain itu peneliti juga akan menguji kepakaran sistem berdasarkan fakta-fakta yang telah disajikan.

a. Pengujian Fungsional

Dalam melakukan pengujian fungsional dari sistem pakar akan dilakukan metode *black box* dengan teknik *Equivalence Partitioning* (EP). EP adalah metode pengujian *black box* yang memecah atau membagi domain input dari program ke dalam kelas-kelas data sehingga *test case* dapat diperoleh. Perancangan *test case* EP berdasarkan evaluasi kelas *equivalence* untuk kondisi input yang menggambarkan kumpulan keadaan yang valid atau tidak. Kondisi input bisa berupa nilai *numeric*, *range* nilai, maupun kumpulan nilai yang berhubungan dengan kondisi Boolean. Rancangan daftar pengujian *black box* dapat dilihat pada tabel 3.1 dan 3.2 dibawah.

Tabel 3.1 Daftar pengujian *black box* untuk pakar

No	Kelas Uji	Daftar Pengujian	Kasus Uji	Hasil yang Diharapkan
1.	Login	Input data login	Username=admin, Password=admin, klik tombol login	Masuk ke halaman beranda sistem pakar
2.	Fungsi pada menu Beranda (Home)	Pengujian menampilkan beranda pada pakar	Pakar memilih menu Home	Tampil halaman beranda (Home)
3.	Fungsi pada menu Tentang (About)	Pengujian menampilkan halaman tentang pada pakar	Pakar memilih menu About	Tampil halaman tentang (About)
4.	Fungsi pada menu Penyakit	Pengujian menampilkan halaman penyakit pada pakar	Pakar memilih menu penyakit	Tampil halaman penyakit
		Tambah data penyakit	Pakar memasukkan data penyakit, gejala dengan lengkap dan menekan tombol simpan	Keluar <i>pop-up</i> “data berhasil disimpan” dan data tersimpan pada database
			Pakar memasukkan data penyakit, gejala dengan tidak lengkap dan menekan tombol simpan	Keluar <i>pop-up</i> “data masih ada yang kosong” dan kembali pada halaman pengisian tambah data penyakit baru
		Lihat data penyakit	Pakar menekan tombol lihat pada salah satu daftar penyakit	Tampil halaman data secara detail tentang penyakit
		Edit data penyakit	Pakar menekan tombol edit data pada salah satu daftar penyakit	Tampil halaman data penyakit yang dapat diedit
			Pakar mengganti isi data penyakit tetapi ada yang tidak diisi dan menekan tombol simpan	Keluar <i>pop-up</i> “data masih ada yang kosong” dan kembali pada halaman edit data penyakit

Tabel 3.1 Daftar pengujian *black box* untuk pakar lanjutan

No	Kelas Uji	Daftar Pengujian	Kasus Uji	Hasil yang Diharapkan
4.	Fungsi pada menu Penyakit	Hapus data penyakit	Pakar menekan tombol hapus data pada salah satu daftar penyakit	Tampil pemberitahuan “Data berhasil dihapus”
5.	Fungsi pada menu Gejala	Pengujian menampilkan halaman gejala pada pakar	Pakar memilih menu gejala	Tampil halaman gejala
		Tambah data gejala	Pakar memasukkan data gejala dengan lengkap dan menekan tombol simpan	Keluar <i>pop-up</i> “data berhasil disimpan” dan data tersimpan pada database
			Pakar memasukkan data gejala dengan tidak lengkap dan menekan tombol simpan	Keluar <i>pop-up</i> “data masih ada yang kosong” dan kembali pada halaman pengisian tambah data gejala baru
		Edit data gejala	Pakar menekan tombol edit data pada salah satu gejala	Tampil halaman data gejala yang dapat diedit
			Pakar mengganti isi gejala dan menekan tombol simpan	Keluar <i>pop-up</i> “data berhasil disimpan” dan data tersimpan pada database
			Pakar mengganti isi gejala tetapi ada yang tidak diisi dan menekan tombol simpan	Keluar <i>pop-up</i> “data masih ada yang kosong” dan kembali pada halaman edit gejala
		Hapus data gejala	Pakar menekan tombol hapus data pada salah satu gejala	Tampil pemberitahuan “Data berhasil dihapus”

Tabel 3.1 Daftar pengujian *black box* untuk pakar lanjutan

No	Kelas Uji	Daftar Pengujian	Kasus Uji	Hasil yang Diharapkan
6.	Fungsi pada menu <i>Rules</i>	Pengujian menampilkan halaman <i>rules</i> pada pakar	Pakar memilih menu <i>rules</i>	Tampil halaman <i>rules</i>
		Tambah data <i>rules</i>	Pakar memasukkan data <i>rules</i> dengan lengkap dan menekan tombol simpan	Keluar <i>pop-up</i> “data berhasil disimpan” dan data tersimpan pada database
			Pakar memasukkan data <i>rules</i> dengan tidak lengkap dan menekan tombol simpan	Keluar <i>pop-up</i> “data masih ada yang kosong” dan kembali pada halaman pengisian tambah data gejala baru
		<i>Edit data rules</i>	Pakar menekan tombol edit data pada salah satu <i>rules</i>	Tampil halaman data <i>rules</i> yang dapat diedit
			Pakar mengganti nama penyakit dan gejala, lalu menekan tombol simpan	Keluar <i>pop-up</i> “data berhasil disimpan” dan data tersimpan pada database
			Pakar mengganti isi <i>form rules</i> tetapi ada yang tidak diisi dan menekan tombol simpan	Keluar <i>pop-up</i> “data masih ada yang kosong” dan kembali pada halaman edit <i>rules</i>
		Hapus data gejala	Pakar menekan tombol hapus data pada salah satu gejala	Tampil pemberitahuan “Data berhasil dihapus”
7.	Fungsi pada menu <i>logout</i>	Pengujian menampilkan halaman <i>login</i>	Pakar memilih menu <i>logout</i>	Keluar pada tampilan halaman <i>login</i>

Tabel 3.2 Daftar pengujian black box untuk pengguna (*user*)

No.	Kelas Uji	Daftar Pengujian	Kasus Uji	Hasil yang Diharapkan
1.	Fungsi pada menu Beranda (<i>Home</i>)	Pengujian menampilkan beranda pada <i>user</i>	<i>User</i> memilih menu <i>Home</i>	Tampil halaman beranda (<i>Home</i>)
2.	Fungsi pada menu Tentang (<i>About</i>)	Pengujian menampilkan halaman tentang pada <i>user</i>	<i>User</i> memilih menu <i>About</i>	Tampil halaman tentang (<i>About</i>)
3.	Fungsi pada menu Penyakit	Pengujian menampilkan halaman daftar penyakit pada <i>user</i>	<i>User</i> memilih menu penyakit	Tampil halaman daftar penyakit
		Lihat data penyakit	<i>User</i> menekan tombol lihat pada salah satu daftar penyakit	Tampil halaman data secara detail tentang penyakit
4.	Fungsi pada menu deteksi	Pertanyaan gejala	<i>User</i> klik tombol ya atau tidak	Tampil pertanyaan berikutnya, apabila pertanyaan habis akan tampil hasil identifikasi

b. Pengujian Kepakaran Sistem

Pengujian kepakarn sistem dilakukan dengan menguji sistem pakar dalam mendeteksi penyakit gigi. Pengujian ini dilakukan dengan tujuan untuk menguji sistem dalam diagnosa penyakit gigi berdasarkan gejala atau fakta-fakta yang diberikan.

3.3.4.4.2 Pengujian Eksternal

Setelah selesai melakukan pengujian internal, maka dilakukan pengujian eksternal dengan menggunakan metode kuisioner. Tujuan dari pengujian ini adalah untuk

mengetahui penilaian dari kacamata seorang *user*/ pengguna/ responden terkait sistem yang telah dibangun.

3.3.5 Penyusunan Laporan

Tahap penyusunan laporan adalah tahapan terakhir dalam penelitian sistem pakar diagnosa penyakit gigi. Penyusunan laporan dilakukan apabila semua tahapan penelitian sudah terpenuhi dan tidak ada kesalahan lagi. Penulisan laporan bertujuan untuk mendokumentasikan seluruh kegiatan pembuatan sistem pakar dari awal hingga akhir. Penyusunan laporan dibuat sebagai bukti dokumentasi dari penelitian yang telah dilakukan.

V. PENUTUP

5.1 Kesimpulan

Berdasarkan penelitian yang telah dilakukan maka didapat beberapa kesimpulan sebagai berikut.

1. Telah berhasil dibangun Sistem Pakar Diagnosa Penyakit Gigi dan Jaringan Penyangga Gigi yang dapat membantu mempercepat diagnosa penyakit gigi pada masyarakat berdasarkan gejala-gejala morfologi (fakta) yang diberikan.
2. Sistem dapat memberikan informasi yang bermanfaat bagi pakar maupun bagi masyarakat mengenai penyakit gigi dan jaringan penyangga gigi yang ada.
3. Berdasarkan pengujian yang telah dilaksanakan, didapatkan rata-rata keakurasian sebesar 86,45% sehingga dapat disimpulkan bahwa sistem dapat mendiagnosa penyakit dengan cukup baik. Presentase yang didapatkan dipengaruhi oleh jumlah fakta atau gejala yang sesuai. Suatu penyakit memiliki akurasi tinggi bila semakin banyak fakta yang sesuai.
4. Berdasarkan hasil kuisioner yang dilakukan kepada pengguna, didapatkan hasil presentase dari para pakar yaitu dokter gigi yang paham mengenai diagnosa penyakit sebesar 86,28% yang tergolong sangat baik. Sedangkan presentase yang diperoleh dari pasien sebagai penderita penyakit adalah sebesar 88,85%

yang juga tergolong sangat baik. Presentase juga diperoleh dari penilaian orang awam sebesar 85,71% yang tergolong sangat baik.

5.2 Saran

Berdasarkan penelitian yang dilakukan maka diperoleh beberapa saran untuk pengembangan sistem lebih lanjut sebagai berikut.

1. Penambahan data- data terkait penyakit gigi dan jaringan penyangga gigi baik itu informasi mengenai penyakit, gejala-gejalanya, maupun solusinya.
2. Sistem dapat dikembangkan dalam bentuk aplikasi *mobile* untuk penelitian selanjutnya.

DAFTAR PUSTAKA

- Arhami, M. 2005. *Konsep Dasar Sistem Pakar*. ANDI. Yogyakarta.
- Azhary .2016. *Dentino: Jurnal Kedokteran Gigi Vol.I No.2*. Fakultas Kedokteran Universitas Lambung Mangkurat, Banjarmasin.
- Azwar, S. 2011. *Sikap dan Perilaku, Dalam: Sikap Manusia Teori dan Pengembangannya*. Pustaka Pelajar. Yogyakarta.
- Djarwanto, P.S. 1996. *Statistik Induktif*. BPFE-UGM. Yogyakarta.
- Fowler, Martin. 2004. *UML Distilled Panduan Singkat Bahasa pemodelan Objek Standar, Edisi 3*. Andi Publishing, Yogyakarta.
- Handojo, A. dan M. Isa Irawan. 2009. *Perancangan dan Pembuatan Aplikasi Sistem Pakar untuk Permasalahan Tindak Pidana terhadap Harta Kekayaan*. Universitas Kristen Petra.
- Hartati, Sri dan Sari Iswanti. 2008. *Sistem Pakar dan Pengembangannya*. Graha Ilmu: Yogyakarta.
- Hutapea, M. 2006. *Keajaiban-keajaiban Dalam Tubuh Manusia*. Jakarta: PT Gramedia Pustaka Utama.
- Julianti Riri, Mohan S.D., Erdaliza., Dini Anggia., Febry Fahmi., Laila Aidi., Marissa Alfian. 2008. *Gigi dan Mulut (Tutorial)*, Pekanbaru. Riau
- Kadir, Abdul. 2007. *From Zero to A Pro : Membuat Aplikasi Web dengan PHP dan Database MySQL*. Yogyakarta : Andi Offset.

Kemenkes RI. 2014. Infodatin. Pusat Data dan Informasi Kementerian Kesehatan RI. Jakarta Selatan.

Kristiani Anie, Nandang Koswara., Hetty Anggrawati., Ira Wijaya., Mukhlis Nafarin., Nurhayati., Suwarsono., Siti Salamah., Zaeni Dahlan., Nasri., Rahayu Budiarti., Vegaroosa Vione., Nurmini Mappahia., Nining Ningrum., Setyo U.A., Emma Krisyudhanti., Lies Elina., Arnetty. 2010. *Ilmu Penyakit Gigi dan Mulut*. Jurusan Kesehatan Gigi Politeknik Kesehatan Tasikmalaya.

Nidhra, Srinivas dan Jagruthi Dondeti.2012. *Black Box and White Box Testing Techniques- A Literature Review*. International Journal of Embedded Systems and Application (IJESA), Vol 2, No.2, PP. 29-50.

Oktarina, 2010. *Hubungan Pola Makan dan Kebiasaan Menyikat Gigi dengan Kesehatan Gigi dan Mulut (Karies) di Indonesia*. Pusat Penelitian dan Pengembangan Sistem dan Kebijakan Kesehatan, Surabaya.

Puspitasari, Heni. 2011. *Pemograman Web Database dengan PHP & MySQL*. Yogyakarta : Skripta.

Setyaningsih, Dwi .2007. *Menjaga Kesehatan Gigi dan Mulut*. Jakarta: CV. Sinar Cemerlang Abadi.

Silvia, Christine Leonita., Virginia., Yosusa J.C., Nina Selvani.2015.*Aplikasi Diagnosis Karies pada Gigi Manusia Berbasis Web*. E-Journal ISSN: 2085-4552. Universitas Kristen Wacana.

Siswanto. 2005. *Kecerdasan Tiruan*. Yogyakarta: Graha Ilmu.

Siswanto. 2010. *Kecerdasan Tiruan Edisi 2*. Yogyakarta: Graha Ilmu.

Sutojo, T., Edy, M., dan Vincent, S. 2011. *Kecerdasan Buatan*. Yogyakarta: Andi.

Wiwik, V.2015. *Jatisi: Penerapan Metode Forward Chaining untuk Mendeteksi Penyakit THT*. Teknik Informatika Universitas Potensi Utama. ISSN: 2407-4332. Vol. 1 No.2. PP. 123-138.

Zulkifli. 2013. *Model Prediksi Berbasis Neural Network untuk Pengujian Perangkat Lunak Metode Black Box*. Seminar Nasional Aplikasi Teknologi Informasi (SNATI). ISSN: 1907-5022, PP.33-37.