

PERANCANGAN SISTEM PAKAR DIAGNOSA AWAL
PENYAKIT SINUSITIS MENGGUNAKAN METODE *CERTAINTY*
FACTOR BERBASIS WEB

SKRIPSI



Oleh :

SANTI RAHMAN
2013020018

JURUSAN TEKNIK INFORMATIKA

SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN
KOMPUTER (STMIK) HANDAYANI

MAKASSAR

2017

PERANCANGAN SISTEM PAKAR DIAGNOSA AWAL
PENYAKIT SINUSITIS MENGGUNAKAN METODE *CERTAINTY*
FACTOR BERBASIS WEB

SKRIPSI

Diajukan sebagai salah satu syarat untuk mencapai gelar Sarjana Komputer
Jurusan Teknik Informatika

Disusun dan diajukan Oleh :

SANTI RAHMAN
2013020018

SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN
KOMPUTER (STMIK) HANDAYANI
MAKASSAR
2017

HALAMAN PERSETUJUAN UJIAN SKRIPSI

Judul : PERANCANGAN SISTEM PAKAR DIAGNOSA AWAL
PENYAKIT SINUSITIS MENGGUNAKAN METODE
CERTAINTY FACTOR BERBASIS WEB

SANTI RAHMAN

2013020018

TEKNIK INFORMATIKA

Telah diperiksa dan disetujui untuk diajukan sebagai salah satu syarat untuk
menempuh Ujian Skripsi guna memperoleh gelar Sarjana Komputer.

Makassar, 19 September 2017

Pembimbing 1

Pembimbing 2

Dr. H. Jamaluddin Sawaji, SE, MPM **Billy Eden William Asrul, S.Kom, MT**

Mengetahui

Ketua Jurusan Teknik Informatika

Billy Eden William Asrul, S.Kom, MT

HALAMAN PENGESAHAN
SISTEM PAKAR DIAGNOSA AWAL PENYAKIT SINUSITIS
MENGGUNAKAN METODE *CERTAINTY FACTOR* BERBASIS WEB

Oleh :

SANTI RAHMAN

2013020018

Telah dipertahankan di hadapan sidang penguji Skripsi STMIK Handayani

Pada 27 September 2017

Dan dinyatakan telah memenuhi syarat memperoleh gelar Sarjana Komputer

Tim Penguji :

Tanda Tangan

1. **Dr. H. Jamaluddin Sawaji, SE, MPM** 1.

Ketua Sidang

2. **Billy Eden William Asrul, S.Kom, MT** 2.

Sekretaris Sidang

3. **Supriadi Sahibu, S.Kom., MT . DIT** 3.

Anggota

4. **Dr. H. Mashur Razak, SE., MPM** 4.

Anggota

5. **Sitti Zuhriyah, S.Pd, M.Si** 5.

Anggota

Mengesahkan :

Wakil Ketua I,

Ketua Jurusan Teknik Informatika

Dr. Nasrullah, M.Si

Billy Eden William Asrul, S.Kom, MT

Mengetahui :

Ketua STMIK Handayani Makassar

Dr. Eng. Agussalim, MT

Perancangan Sistem Pakar Diagnosa Awal Penyakit Sinusitis menggunakan
Metode *Certainty Factor* Berbasis Web

Santi Rahman, 2017

Dibimbing oleh : Jamaluddin Sawaji dan Billy Eden William Asrul

ABSTRAK

Sinusitis adalah penyakit yang harus segera ditangani sejak dini namun terkadang kita tidak dapat berkonsultasi kepada dokter karena terhalang waktu dan tempat. Berdasarkan permasalahan ini, dibutuhkan sebuah aplikasi sistem pakar yang dapat membantu pasien selama 24 jam dimanapun pasien tersebut berada.

Desain penelitian yang digunakan adalah UML yang didesain secara terstruktur yang terdiri dari rancangan model *use case diagram*, *activity diagram*, *sequence diagram*, dan *class diagram*. *Text editor* yang digunakan dalam membangun sistem ini adalah sublime text, sedangkan bahasa pemrograman menggunakan PHP. *Javascript*, dan MySQL untuk pengolahan database. Dalam penelitian ini pengumpulan data diperoleh melalui wawancara.

Hasil dari penelitian ini adalah sebuah sistem berbasis web yang menyediakan informasi penyakit sinusitis dan sistem pakar diagnosa awal penyakit sinusitis menggunakan metode *Certainty Factor*. Sistem tersebut membantu pasien dalam mendiagnosa awal penyakit sinusitis.

Kata Kunci : *Certainty Factor* , *Sinusitis* ,Sistem Pakar

KATA PENGANTAR

Assalamu Alaikum Wr.Wb.

Dengan menyebut nama Allah SWT yang Maha Pengasih lagi Maha Penyayang, Kami panjatkan puja dan puji syukur atas kehadiran-Nya, yang telah melimpahkan rahmat, hidayah, dan inayah-Nya kepada kami, sehingga penulis dapat menyelesaikan penyusunan skripsi ini. Shalawat beserta salam semoga senantiasa terlimpah curahkan kepada Nabi Muhammad Shallallahu ‘alaihi wasallam.

Skripsi ini diajukan untuk memenuhi salah satu syarat memperoleh gelar sarjana (S1) pada STMIK Handayani Makassar dan juga merupakan laporan akhir yang wajib diselesaikan oleh setiap mahasiswa STMIK Handayani Makassar. Semoga dapat memberikan nilai tambah bagi semua pihak yang terkait didalamnya dan khususnya pada penulis.

Dalam penyusunan dan penulisan skripsi ini tidak lepas dari bantuan berupa sumbangan pemikiran, bimbingan serta dukungan dari berbagai pihak. Oleh karena itu dalam kesempatan ini penulis mengucapkan terima kasih yang setulus-tulusnya kepada yang terhormat :

1. Kedua orang tua saya yang saya cintai, keluarga yang senantiasa memberikan bantuan, nasihat, dukungan sehingga penulis dapat menyelesaikan studi dengan baik.
2. Dr. Eng. Agussalim, MT selaku ketua STMIK Handayani Makassar.

3. Billy Eden William Asrul, S.Kom, MT selaku ketua jurusan Teknik Informatika.
4. Dr. H. Jamaluddin Sawaji, SE, MPM selaku pembimbing I dalam penyusunan skripsi ini, sehingga dapat selesai dengan baik.
5. Billy Eden William Asrul, S.Kom, MT selaku pembimbing II dalam penyusunan skripsi ini, sehingga dapat selesai dengan baik.
6. dr. Syahrijuita, M.Kes, Sp. THT-KL selaku Dokter Spesialis THT-KL dan pakar dari penyakit sinusitis.
7. Fuad Fauzi Alfian Sofyan yang bersedia membantu, mengajar dan memberikan semangat, terimakasih atas bantuan dan dukungannya dalam penyelesaian skripsi ini.
8. Akbar Abustang yang bersedia membantu, mengajar dan memberikan semangat, terimakasih atas bantuan dan dukungannya dalam penyelesaian skripsi ini
9. Muhammad Lutfi yang bersedia membantu dan memberikan semangat, terimakasih atas bantuan dan dukungannya dalam penyelesaian skripsi ini
10. Sahabat-sahabat saya : Nur.Mauliana.S, Dian Ardika A.M, St.Hajar, Fuad Fauzi Alfian Sofyan, Aswin, Yusri.B, Sukram Prawijaya, Mohammad Hosni Mubaroq, Aditya Puspa, Muh. Risal Razak, Husyadi Saputra, Ari Bayu, terimakasih atas kebersamaannya dalam merasakan susah senang dalam penyusunan skripsi ini, serta yang selalu memberikan dukungan, semangat, solusi dan keceriaannya sehingga memberikan pengaruh positif terhadap penulis.

Akhirnya, penulis menyadari bahwa penulisan skripsi ini masih banyak kekurangan. Oleh karena itu, kritik dan saran yang sifatnya mendidik, dan membangun, senantiasa penulis terima dengan lapang dada.

Akhir kata semoga skripsi ini dapat bermanfaat bagi pembaca, maupun kepada masyarakat umum. Semoga Allah Subhanahu Wata'ala memberikan rahmat dan karunia-Nya bagi kita semua, Aamiin.

Wassalamu Alaikum Warahmatullahi Wabarakatu

Makassar, 25 Agustus 2017

Penulis

DAFTAR ISI

HALAMAN JUDUL.....	i
HALAMAN PERSETUJUAN UJIAN SKRIPSIiii
ABSTRAK	v
KATA PENGANTARvi
DAFTAR ISI.....	.ix
DAFTAR TABEL.....	xii
DAFTAR GAMBARxiii
BAB I PENDAHULUAN	1
A. Latar Belakang	1
B. Batasan Masalah.....	4
C. Rumusan Masalah	5
D. Tujuan Penelitian	5
E. Penegasan Konsep.....	5
BAB II TINJAUAN PUSTAKA.....	6
A. Konsep Dasar Sistem	6
B. Konsep Dasar Perancangan Sistem.....	7
C. Sistem Pakar.....	9
D. Sinusitis	19

E. Pengobatan pada Sinusitis.....	28
F. <i>Forward Chaining</i> (Runut Maju).....	31
G. <i>Certainty Factor</i> (Faktor Kepastian).....	33
H. <i>Unified Modelling Language</i> (UML).....	36
I. <i>Website</i>	45
J. <i>Internet</i>	46
K. Bahasa Pemrograman.....	46
L. Aplikasi Pendukung	51
M. Metode Pengujian <i>White Box</i>	52
N. Metode Pengujian <i>Black Box</i>	60
 BAB III METODE PENELITIAN.....	 64
A. Penggambaran Sistem	64
B. Analisis Kebutuhan	67
C. Teknik Pengumpulan Data.....	70
D. Lokasi dan Waktu Penelitian	70
E. Tahapan Kegiatan.....	70
 BAB IV PERANCANGAN DAN IMPLEMENTASI SISTEM	 72
A. Pemodelan Sistem	72
B. Perancangan Database Tabel.....	81
C. Perancangan Antar Muka.....	83
D. Implementasi	89
E. Pengujian Sistem.....	104

BAB V KESIMPULAN	118
A. Kesimpulan	118
B. Saran.....	118
DAFTAR PUSTAKA	120
LAMPIRAN	

DAFTAR TABEL

Tabel 2.1 Tabel Daftar Penyakit beserta Gejala Sinusitis.....	27
Tabel 2.2 Tabel Use Case Diagram.....	37
Tabel 2.3 Tabel Activity Diagram	40
Tabel 2.4 Tabel Sequence Diagram	43
Tabel 2.5 Tabel Class Diagram.....	44
Tabel 4.1 Tabel Penyakit.....	81
Tabel 4.2 Tabel Gejala	81
Tabel 4.3 Tabel Gejala Penyakit	82
Tabel 4.4 Tabel Kelompok Gejala	82
Tabel 4.5 Tabel User	82
Tabel 4.6 Tabel Pengguna.....	83
Tabel 4.7 Gejala Penyakit Sinusitis	91
Tabel 4.8 Tabel Penyakit Sinusitis.....	92
Tabel 4.9 Nilai CF yang diisi oleh Pakar	93
Tabel 4.10 Perhitungan Manual Metode <i>Certainty Factor</i>	96
Tabel 4.11 Pengujian <i>Black Box Admin Expert Sistem</i>	106
Tabel 4.12 Pengujian <i>Black Box User</i>	109

DAFTAR GAMBAR

Gambar 2.1 Struktur Dasar Sistem Pakar	12
Gambar 2.2 Sistem Penanganan Masalah	12
Gambar 2.3 Struktur Sistem Pakar	13
Gambar 2.4 Rhinosinusitis.....	20
Gambar 2.5 Diagram Forward Chaining.....	32
Gambar 2.6 Flowgraph.....	55
Gambar 2.7 Flowchart dan Flowgraph.....	56
Gambar 2.8 Translating.....	57
Gambar 3.1 Use Case Diagram Sistem yang sedang Berjalan.....	64
Gambar 3.2 Use Case Diagram Sistem yang Diusulkan.....	66
Gambar 4.1 Use Case Diagram	73
Gambar 4.2 Class Diagram	76
Gambar 4.3 Activity Diagram User/Pasien.....	77
Gambar 4.4 Activity Diagram Admin.....	78
Gambar 4.5 Sequence Diagram User/Pasien	79
Gambar 4.6 Sequence Diagram Admin	80

Gambar 4.7 Rancangan Output Beranda.....	84
Gambar4.8 Rancangan Output Tentang Pakar.....	84
Gambar 4.9 Rancangan Output Diagnosa.....	85
Gambar 4.10 Rancangan Output Gejala Penyakit.....	85
Gambar 4.11 Rancangan Output Hasil Diagnosa	86
Gambar 4,12 Rancangan Output Daftar Pengunjung.....	86
Gambar 4.13 Rancangan Input Login	86
Gambar 4.14 Rancangan Input Data Gejala.....	87
Gambar 4.15 Rancangan Input Data Penyakit	87
Gambar 4.16 Rancangan Input Data Gejala Penyakit.....	88
Gambar 4.17 Rancangan Input Data Kelompok Gejala.....	88
Gambar 4.18 Tampilan Login	96
Gambar 4.19 Tampilan Data Gejala.....	97
Gambar 4.20 Tampilan Data Kelompok Gejala.....	97
Gambar 4.21 Tampilan Data Penyakit	98
Gambar 4.22 Tampilan Data Gejala Penyakit.....	98
Gambar 4.23 Tampilan Data Daftar Pengunjung.....	99

Gambar 4.24 Tampilan Login User/Pasien	99
Gambar 4.25 Tampilan Menu Beranda	100
Gambar 4.26 Tampilan Menu Tentang Pakar	101
Gambar 4.27 Tampilan Menu Diagnosa	101
Gambar 4.28 Tampilan Gejala Penyakit	102
Gambar 4.29 Tampilan Menu Panduan	102
Gambar 4.30 Tampilan Menu Hasil Analisis Diagnosa.....	103
Gambar 4.31 Tampilan Laporan Hasil Diagnosa.....	104
Gambar 4.32 Flowchart Sistem.....	113
Gambar 4.33 Flowgraph.....	116

BAB I

PENDAHULUAN

A. Latar belakang

Sinusitis merupakan peradangan yang terjadi pada organ sinus. Sinus sendiri adalah rongga udara yang terdapat di daerah wajah yang langsung terhubung dengan hidung. Peradangan pada sinus ini dapat menyebabkan penimbunan lendir pada rongga sinus dan menjadi media bagi pertumbuhan bakteri. Hal ini umumnya merupakan akibat dari meluasnya infeksi hidung ringan seperti *selesma* biasa. Membuang lendir lewat hidung secara paksa dapat mendorong kuman masuk pada sinus-sinus tersebut. Kejadian sinusitis umumnya disertai atau dipicu oleh *rhinitis* sehingga sinusitis sering juga disebut dengan *Rhinosinusitis*. *Rhinosinusitis* adalah penyakit inflamasi yang sering ditemukan dan mungkin akan terus meningkat *prevalensinya*.

Saat ini di Indonesia penyakit sinusitis merupakan penyakit dengan jumlah penderita cukup banyak antara 1,3 dan 3,5 per 100 kasus orang dewasa per tahun. Dari data DEPKES RI (2003) menyebutkan bahwa penyakit hidung dan sinus berada pada urutan ke-25 dari 50 pola penyakit peringkat utama atau selkitar 102.817 penderita rawat jalan di rumah sakit. Selain itu masyarakat Indonesia sedikit mengetahui gejala penyakit sinusitis karena hampir sama dengan gejala penyakit flu biasa. Hal ini bisa menyebabkan kesalahan dalam menindaklanjuti penyakit sinusitis dan bisa menyebabkan dampak yang lebih besar lagi.

Dalam era globalisasi, komputer merupakan suatu kebutuhan utama dalam menunjang kerja-kerja manusia. Peran komputer kini menjadi lebih meluas, tidak hanya menjadi alat bantu penyelesaian masalah-masalah yang dihadapi manusia. Semakin cerdas sistem yang dibuat dan semakin ditingkatkan level penanganan informasi masukannya maka, semakin aktif peranan yang dimainkan komputer. Salah satu cabang ilmu komputer yang banyak dimanfaatkan oleh manusia untuk membantu kerjanya adalah pembentukan sistem pakar yang merupakan salah satu sub bidang ilmu kecerdasan buatan (Arhammi, 2005:2). Kecerdasan buatan (*Artificial Intellegence*) adalah kegiatan menyediakan mesin seperti komputer dengan kemampuan untuk menampilkan perilaku yang dianggap cerdas jika diamati oleh manusia.

Sistem Pakar adalah sistem yang menirukan penalaran seorang pakar agar komputer dapat menyelesaikan masalah seperti yang biasa dilakukan oleh para ahli. Pengetahuan yang disimpan didalam sistem pakar umumnya diambil dari seorang manusia yang pakar dalam masalah tersebut. Peran penting seorang pakar dapat digantikan oleh program komputer yang pada prinsip kerjanya untuk memberikan solusi yang pasti seperti yang biasa dilakukan oleh pakar. Sistem pakar biasanya digunakan untuk konsultasi, analisis, diagnosis dan membantu mengambil keputusan.

Website (Web) merupakan halaman situs sistem informasi yang dapat diakses secara cepat. Website ini didasari dari adanya perkembangan teknologi informasi dan komunikasi. Melalui perkembangan teknologi informasi,

tercipta suatu jaringan antar komputer yang saling berkaitan. Jaringan yang dikenal dengan istilah internet. Kehadiran internet ini memunculkan media baru yang dapat menjangkau semua kalangan secara cepat, tepat dan bersamaan dengan biaya yang murah. Internet suatu media yang sangat besar manfaatnya yang menjadikan komunikasi baru bagi masyarakat dunia, yang dapat menjadi media pertukaran dan penyampaian informasi. Penggunaan *website* pada perancangan sistem pakar diagnosa awal penyakit sinusitis dapat memberikan informasi tentang penyakit sinusitis serta dapat mendiagnosa awal penyakit sinusitis.

Dari hasil pemaparan di atas dapat disimpulkan beberapa penjelasan sebagai berikut. Pertama "tujuan perancangan sistem pakar berbasis Web", yaitu untuk menyelesaikan suatu permasalahan tertentu dengan meniru kerja dari para pakar/ahli dengan menggunakan metode *Certainty Factor* (CF) serta penyediaan informasi seputar penyakit sinusitis yang dapat diakses dimana saja dan kapan saja. Kedua "tentang penyakit Sinusitis", yaitu mempunyai jenis-jenis dan juga gejala-gejala. Untuk itu mengambil judul **PERANCANGAN SISTEM PAKAR DIAGNOSA AWAL PENYAKIT SINUSITIS MENGGUNAKAN METODE *CERTANTY FACTOR* BERBASIS WEB.**

B. Batasan Masalah

Dari pemaparan di bagian latar belakang dapat dijadikan landasan untuk menentukan batasan masalah dalam penelitian yang akan dilakukan. Adapun batasan tersebut sebagai berikut :

1. Sistem pakar yang peneliti rancang ini hanya untuk mendiagnosa awal penyakit Sinusitis.
2. Parameter penilaian pada sistem pakar diagnosa penyakit sinusitis ini yaitu gejala Mayor : nyeri sinus, hidung buntu, ingus *purulen*, *post nasal drip*, dan gangguan penghirup. Sedangkan gejala Minor : nyeri kepala, nyeri geraham, nyeri telinga, batuk, demam dan *halitosis*.
3. Jenis-jenis sinusitis dibagi menjadi 4 yaitu : *sinusitis maksila*, *sinusitis etmoid*, *sinusitis frontal* dan *sinusitis sfenoid*.
4. Interaksi yang digunakan antar pemakai dan sistem berupa tampilan beberapa gejala-gejala dari penyakit sinusitis dan berakhir pada suatu kesimpulan yang akan di tampilkan sistem.
5. Informasi untuk menyusun gejala-gejala merupakan fakta yang diperoleh dari pakar / dokter Spesialis THT-KL
6. Keluaran (*Output*) dari sistem pakar ini nanti berupa penyakit *Sinusitis Maksila*, *Sinusitis Etmoid*, *Sinusistis Frontalis* dan *Sinusitis Sfenoid*.
7. Sistem pakar ini menggunakan metode *Certaninty factor* (CF).

C. Rumusan Masalah

Bagaimana merancang suatu sistem untuk mendiagnosa awal penyakit Sinusitis melalui sistem pakar dengan menggunakan metode *Certainty Factor*?

D. Tujuan Penelitian

Tujuan dari penelitian ini adalah merancang sistem yang dapat mendiagnosa awal penyakit sinusitis menggunakan metode *Certainty Factor*.

E. Penegasan Konsep

1. Sistem pakar adalah suatu program komputer yang dirancang untuk memodelkan kemampuan penyelesaian masalah yang dilakukan seorang pakar. Sistem Pakar secara umum, Sistem pakar (*expert system*) adalah sistem yang berusaha mengadopsi pengetahuan manusia ke komputer, agar komputer dapat menyelesaikan masalah seperti yang biasa dilakukan oleh para ahli. (Sri Kusumadewi, 2003:109)
2. Menurut Kamus Besar Bahasa Indonesia, diagnosa dan diagnosis adalah penentuan jenis penyakit dengan cara meneliti (memeriksa gejala-gejalanya), sedangkan pengertian mendiagnosa adalah penentuan jenis penyakit dengan cara meneliti atau memeriksa gejalanya.
3. Menurut Abi Sambuda dalam William (1992), Sinusitis adalah peradangan *sinus paranasalis*, dapat disebabkan oleh *infeksi* maupun bukan *infeksi*, dari *bakteri*, jamur, *virus*, *alergi* maupun sebab *autoimun*.

BAB II

TINJAUAN PUSTAKA

A. Konsep Dasar Sistem

Sistem berasal dari bahasa Latin *systema* atau bahasa Yunani *systema* yang berarti suatu kesatuan yang terdiri dari komponen atau elemen yang dihubungkan bersama untuk memudahkan aliran informasi, materi, atau energi. Sistem juga merupakan kesatuan bagian-bagian yang saling berhubungan yang berada dalam suatu wilayah serta memiliki item-item penggerak.

Definisi sistem menurut Kristanto (2008:1), “ Sistem merupakan kumpulan elemen-elemen yang saling terkait dan bekerja sama untuk memproses masukan (*input*) yang ditujukan kepada sistem tersebut dan mengolah masukan tersebut sampai menghasilkan (*output*) yang diinginkan”.

Menurut Dadi Rosadi dalam Jogiyanto (2005) mendefinisikan sistem bahwa : “ Sistem dapat berupa abstrak atau fisik, *system* yang abstrak adalah susunan yang teratur dari gagasan-gagasan tentang tuhan, manusia dan sebagainya sedangkan *System* yang bersifat fisik adalah serangkaian unsur yang bekerja sama untuk mencapai tujuan ”.

Jadi dapat disimpulkan bahwa sistem merupakan sekumpulan elemen-elemen yang saling terkait untuk mencapai suatu tujuan. Dan bekerja sama dalam memproses masukan (*Input*) yang ditujukan kepada suatu sistem untuk menghasilkan keluaran (*Output*) yang diinginkan.

B. Konsep Dasar Perancangan Sistem

1. Pengertian Perancangan Sistem

Untuk menghasilkan sebuah sistem yang handal, maka perlu dilakukan perancangan sistem. Menurut Adi Nugroho (2004), "Perancangan sistem adalah tahap awal dimana pendekatan awal untuk menyelesaikan masalah yang dipilih selama perancangan sistem, struktur keseluruhan diputuskan". Sedangkan menurut Susanto (2004:332) "perancangan sistem adalah proses menyusun atau mengembangkan sistem informasi yang baru.

Dengan melihat pendapat para ahli di atas, maka dapat disimpulkan bahwa dapat diartikan sebagai berikut :

- a. Tahap setelah analisis pengembangan sistem
- b. Pendefinisian dari kebutuhan fungsional
- c. Persiapan untuk rancang bangun implementasi
- d. Menggambarkan bagian sistem yang dibentuk
- e. Perancangan sistem dapat berupa penggambaran dan perancangan.

Perancangan sistem adalah penggambaran, perencanaan dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah ke dalam satu kesatuan yang utuh dan berfungsi sebagai sistem yang baru untuk menggantikan sistem yang lama.

2. Tahap Dalam Perancangan Sistem

Tahap dalam perancangan sistem mempunyai dua maksud atau tujuan utama yaitu memenuhi kebutuhan pemakai sistem dan

memberikan gambaran yang jelas dan rancang bagan yang lengkap kepada programmer dan ahli-ahli teknik lainnya yang terlibat. Oleh karena itu, untuk mencapai tujuan ini, analisis sistem harus dapat mencapai sasaran-sasaran dari sistem yang dirancang. Tahap-tahap dalam perancangan sistem informasi adalah sebagai berikut :

a. Analisis Sistem

Tahapan analisis sistem dimulai karena adanya permintaan terhadap sistem yang baru. Permintaan bisa datang dari seorang *maneger* diluar departemen sistem informasi atau dari pihak eksekutif yang melihat adanya masalah atau menemukan adanya peluang baru berasal dari bagian yang bertanggung jawab terhadap perancangan sistem informasi, yang bermaksud dari perancangan sistem ini adalah mengatasi masalah-masalah yang belum tertangani. Tujuan utama analisis sistem adalah untuk menentukan hal-hal detail tentang yang akan di kerjakan oleh sistem yang diusulkan.

b. Desain Sistem

Tahap berikutnya setelah tahapan analisis sistem adalah tahap perancangan sistem (System Design), tahap perancangan sistem atau desain sistem mempunyai tujuan yaitu : Memberikan gambaran secara umum tentang kebutuhan informasi kepada pemakai sistem secara logika. Perancangan sistem secara umum atau perancangan konseptual merupakan persiapan dari perancangan sistem secara terinci. Konsep fisik yaitu memberikan gambaran yang jelas dan

rancang bangun lengkap kepada programmer komputer dan ahli-ahli teknik lainnya.

c. Implementasi Sistem

Tahap berikutnya setelah sistem selesai dirancang dan dibangun adalah tahap meletakkan sistem supaya siap dioperasikan, pada saat peralihan dari sistem lama ke sistem baru perlu dilakukan pelatihan dan panduan seperlunya.

d. Operasi dan Pemeliharaan

Setelah masa sistem berjalan sepenuhnya menggantikan sistem yang lama, sistem memasuki pada tahapan operasi dan pemeliharaan. Selama sistem beroperasi, pemeliharaan sistem tetap di perlukan karena beberapa alasan. Pertama mungkin sistem mengandung kesalahan yang dulunya belum terdeteksi selama masa pengujian sistem, sehingga kesalahan-kesalahan sistem perlu diperbaiki. Kedua, pemeliharaan diperlukan karena perubahan lingkungan, atau adanya permintaan kebutuhan baru oleh pemakai. Ketiga, pemeliharaan juga biasa dipicu karena kinerja sistem yang menurun.

C. Sistem Pakar

1. Pengertian Sistem Pakar

Menurut Siswanto Sistem pakar (Kecerdasan tiruan:2010)

merupakan program, yaitu:

- a. Program komputer yang menangani masalah dunia nyata masalah yang kompleks yang membutuhkan *interpretasi* pakar.

- b. Program komputer untuk menyelesaikan masalah dengan menggunakan komputer dengan model penalaran manusia dan mencapai kesimpulan yang sama dengan yang dicapai oleh seorang jika berhadapan dengan masalah.

Jadi dapat kita simpulkan bahwa sistem pakar merupakan salah satu cabang kecerdasan buatan yang menggunakan pengetahuan yang dimiliki seorang ahli untuk mengambil kesimpulan dari suatu masalah yang berbasis komputer.

2. Konsep Sistem Pakar

Menurut Repository dalam Efraim Turban (sistem pakar:2011), konsep dasar sistem pakar mengandung keahlian, ahli, pengalihan, aturan dan kemampuan menjelaskan. Keahlian adalah suatu kelebihan penguasaan di bidang tertentu yang diperoleh dari pelatihan membaca dan pengalaman. Contoh bentuk pengetahuan yang merupakan pengalaman :

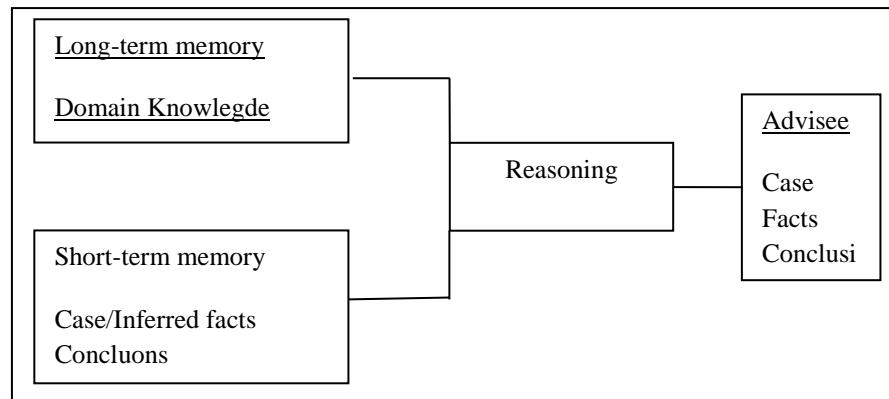
- a. Fakta-fakta pada lingkup permasalahan tertentu.
- b. Teori-teori lingkup permasalahan tertentu.
- c. Prosedur-prosedur dan aturan yang berkenaan dengan lingkup permasalahan tertentu.
- d. Strategi-strategi global untuk menyelesaikan masalah.
- e. *Meta-Knowledge* (Pengetahuan tentang pengetahuan)

Bentuk-bentuk ini memungkinkan para ahli mengambil keputusan yang lebih cepat dan lebih baik daripada seorang yang bukan ahli. Seorang ahli adalah seorang yang mampu menjelaskan suatu tanggapan,

mempelajari hal-hal seputar topik permasalahan, menyusun kembali pengetahuan, memecah aturan-aturan, dan menentukan *relevan* tidaknya keahlian mereka. Pengalihan keahlian dari para ahli ke komputer untuk kemudian dialihkan lagi ke orang yang bukan ahli merupakan tujuan utama dari sistem pakar. Proses ini membutuhkan beberapa aktivitas seperti tambahan pengetahuan (dari para ahli atau sumber lainnya), *representasi* pengetahuan (ke komputer), *inferensi* pengetahuan, dan pengalihan pengetahuan ke *user*.

Pengetahuan yang disimpan di komputer disebut dengan nama basis pengetahuan (*knowledge base*). Ada 2 tipe pengetahuan, yaitu : fakta dan prosedur (biasanya berupa aturan). Salah satu fitur yang harus dimiliki oleh sistem pakar adalah kemampuan untuk menalar. Jika keahlian-keahlian sudah tersimpan sebagai basis pengetahuan dan sudah tersedia program yang mampu mengakses basis data, maka komputer harus dapat diprogram untuk membuat inferensi. Proses inferensi ini dikemas dalam bentuk motor inferensi (*inferensi engine*). Sebagian besar sistem pakar *komersial* dibuat dalam bentuk *rule-based system*, yang mana pengetahuannya disimpan dalam bentuk aturan-aturan. Aturan tersebut biasanya berbentuk *IF-THEN*. Fitur lainnya dari sistem pakar adalah kemampuan untuk merekomendasi. Kemampuan inilah yang membedakan sistem pakar dengan sistem konvensional.

3. Struktur Dasar Sistem Pakar

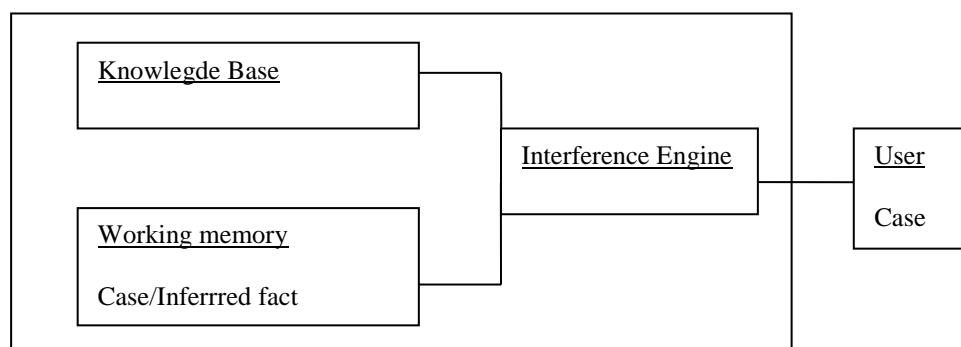


Gambar 2.1 Struktur Dasar Sistem Pakar

(sumber:Balza Achmad, 2006)

Gambar di atas menjelaskan sistem penyelesaian masalah yang dilakukan oleh manusia dimana fakta dan kasus yang ditangkap pada saat *konsultasi* akan disimpan pada *short-term memory*. Kemudian manusia akan berpikir (*reasoning*) untuk menyesuaikan dengan pengetahuan yang dimiliki dan merupakan *domain knowledge*nya hingga akhirnya sang ahli memberikan suatu kesimpulan, saran, dan solusi.

Expert System



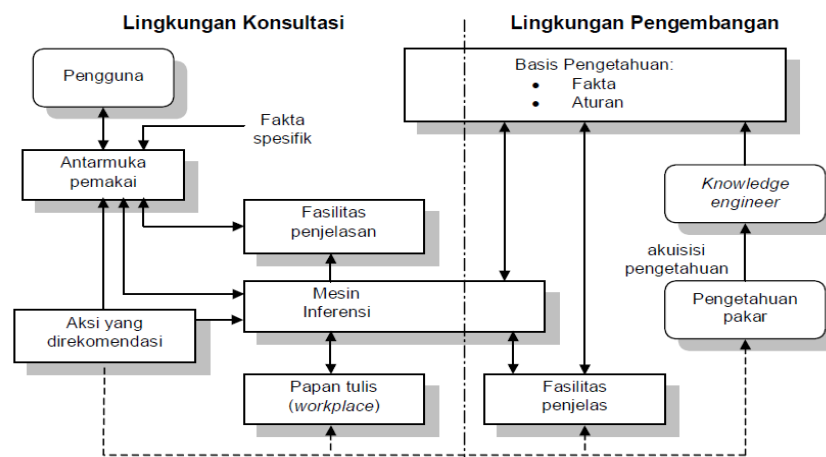
Gambar 2.2 Sistem Penanganan Masalah

(asumber:Balza Achmad, 2006)

Working memory merupakan bagian dari sistem pakar yang berisi fakta-fakta masalah yang ditemukan dalam suatu sesi. Berisi fakta-fakta tentang suatu masalah yang ditemukan dalam proses konsultasi. Sedangkan *interfence engine* merupakan *prosesor* pada sistem pakar yang menyesuaikan fakta-fakta yang ada pada *working memory* dengan *domain* pengetahuan yang terdapat pada *knowledge base* untuk menarik kesimpulan dari masalah yang dihadapi.

4. Arsitektur, Komponen dan Fasilitas Sistem Pakar

Menurut Repository (sistem pakar:2011). Sistem pakar terdiri dari dua bagian pokok, yaitu lingkungan pengembangan (*Development Enviroment*). Lingkungan pengembangan digunakan sebagai pembangun sistem pakar baik dari segi pembangun komponen maupun basis pengetahuan. Lingkungan *konsultasi* digunakan oleh seorang yang bukan ahli untuk *konsultasi*.



Gambar 2.3 Struktur Sistem Pakar

(sumber:Balza Achmad, 2006)

Berikut ini adalah komponen-komponen *arsitektur* sistem pakar :

- a. Antarmuka pengguna (*user interface*) adalah perangkat lunak yang menyediakan media komunikasi antara pengguna dengan sistem. Antarmuka menerima informasi dari pengguna dan mengubahnya ke dalam bentuk yang dapat diterima oleh sistem. Selain itu antarmuka menerima informasi dari sistem dan menyajikannya ke dalam bentuk yang dapat dimengerti oleh pengguna.
- b. Basis pengetahuan (*knowledge base*). Pada bagian ini berisi pengetahuan-pengetahuan yang dibutuhkan untuk memahami, memformulasikan dan menyelesaikan masalah. Pengetahuan ini diperoleh dari akumulasi pengetahuan pakar dan sumber-sumber pengetahuan lainnya seperti buku-buku, jurnal ilmiah, maupun dokumentasi yang tercetak lainnya.
- c. Akuisisi Pengetahuan, adalah akumulasi, transfer dan transformasi keahlian dalam menyelesaikan masalah dari sumber pengetahuan ke dalam program komputer. Pada tahap ini, *akuisisi* pengetahuan berusaha menyerap pengetahuan untuk selanjutnya ditransfer ke dalam basis pengetahuan.
- d. Mesin Inferensi (*Inference engine*), merupakan program yang berisi *metodologi* yang digunakan untuk melakukan penalaran terhadap informasi-informasi dalam basis pengetahuan dan *blackboard*, serta digunakan untuk memformulasikan *konklusi*.

e. *Workspace*, merupakan area dari sekumpulan memori kerja. *Workspace* digunakan untuk merekam hasil-hasil antara dan yang dapat direkam, yaitu:

- 1) Rencana : bagaimana menghadapi masalah
- 2) Agenda : aksi-aksi yang potensial yang sedang menunggu untuk dieksekusi.
- 3) Solusi : calon aksi yang akan dibangkitkan.

Sedangkan untuk menjadikan sistem pakar menjadi lebih menyerupai seorang pakar yang berinteraksi dengan pemakai, maka dilengkapi dengan fasilitas berikut :

- 1) Fasilitas penjelasan (*explanation facility*) merupakan proses menentukan keputusan yang dilakukan oleh mesin inferensi selama sesi konsultasi mencerminkan proses penalaran seorang pakar. Karena pemakai kadangkala bukanlah ahli dalam bidang tersebut, maka dibutuhkan fasilitas penjelasan. Fasilitas penjelasan inilah yang dapat memberikan informasi kepada pemakai mengenai jalannya penalaran sehingga dihasilkan suatu keputusan. Bentuk penjelasannya dapat berupa keterangan yang diberikan setelah suatu pertanyaan diajukan, yaitu penjelasan atas pertanyaan mengapa, atau penjelasan atas pertanyaan bagaimana sistem mencapai *konklusi*.
- 2) Fasilitas akuisisi pengetahuan (*knowledge acquisition facility*) merupakan perangkat lunak yang menyediakan fasilitas dialog

antara pakar dengan sistem. Fasilitas akuisisi ini digunakan untuk memasukkan fakta-fakta dan kaidah-kaidah sesuai dengan perkembangan ilmu. Meliputi proses pengumpulan, pembedaan, dan perubahan dari kemampuan pemecahan masalah seorang pakar atau sumber pengetahuan terdokumentasi (buku,dll) ke program komputer, yang bertujuan untuk memperbaiki dan atau mengembangkan *basis* pengetahuan (*knowledge base*).

5. Karakteristik Sistem Pakar

Menurut Repository (sistem pakar:2011) ada berbagai ciri dan karakteristik yang membedakan Sistem Pakar dengan sistem yang lain. Ciri dan karakteristik ini menjadi pedoman utama dalam pengembangan Sistem Pakar. Ciri dan karakteristik yang dimaksud antara lain:

- a. Pengetahuan Sistem Pakar merupakan suatu konsep, bukan berbentuk *numeris*. Hal ini dikarenakan komputer melakukan proses pengolahan data secara *numerik* sedangkan keahlian dari seorang pakar adalah fakta dan aturan-aturan, bukan *numerik*.
- b. Informasi dalam Sistem Pakar tidak selalu lengkap, subyektif, tidak konsisten, subyek terus berubah dan tergantung pada kondisi lingkungan sehingga keputusan yang diambil bersifat tidak pasti dan tidak mutlak “ya” atau “tidak” akan tetapi menurut ukuran kebenaran tertentu. Oleh karena itu dibutuhkan kemampuan sistem untuk belajar secara mandiri dalam menyelesaikan masalah-masalah dengan pertimbangan-pertimbangan khusus.

- c. Kemungkinan solusi Sistem Pakar terhadap suatu permasalahan adalah bervariasi dan mempunyai banyak pilihan jawaban yang dapat diterima, semua faktor yang ditelusuri memiliki ruang masalah yang luas dan tidak pasti. Oleh karena itu diperlukan sistem yang fleksibel dalam menangani kemungkinan solusi dari berbagai permasalahan.
- d. Perubahan atau pengembangan pengetahuan dalam Sistem Pakar dapat terjadi setiap saat bahkan sepanjang waktu sehingga diperlukan kemudahan dalam modifikasi sistem untuk menampung jumlah pengetahuan yang semakin besar dan semakin bervariasi.
- e. Pandangan dan pendapat setiap pakar tidaklah selalu sama, yang oleh karena itu tidak ada jaminan bahwa solusi Sistem Pakar merupakan jawaban yang pasti benar. Setiap pakar akan memberikan pertimbangan-pertimbangan berdasarkan faktor subyektif.
- f. Keputusan merupakan bagian terpenting dari Sistem Pakar. Sistem Pakar harus memberikan solusi yang akurat berdasarkan masukan pengetahuan meskipun solusinya sulit sehingga fasilitas informasi sistem harus selalu diperlukan.

6. Kelebihan Sistem Pakar

Menurut Muhammad Dahria (Jurnal SAINTIKOM, 2011), Kelebihan yang diperoleh dari sistem pakar yaitu :

- a. Memungkinkan orang awam bisa melakukan pekerjaan seorang pakar.
- b. Meningkatkan produktivitas kerja dengan jalan meningkatkan efisiensi pekerjaan.

- c. Menghemat waktu dalam menyelesaikan pekerjaan atau masalah yang kompleks.
- d. Menyederhanakan beberapa operasi.
- e. Pengolahan berulang-ulang secara otomatis.
- f. Tersedianya pengetahuan pakar bagi masyarakat luas.

7. Kekurangan Sistem Pakar

Menurut Entin Martiana (2012) kelemahan sistem pakar yaitu:

- a. Pengembangan sistem pakar sangat sulit, seorang pakar yang baik sulit diperoleh. Mendapatkan pengetahuan seorang pakar dan mengalihkannya menjadi sebuah program merupakan pekerjaan yang melelahkan dan memerlukan biaya yang besar.
- b. Sistem pakar sangat mahal untuk mengembangkan, mencoba dan mengirimkannya ke pemakai terakhir memerlukan biaya tinggi.
- c. Hampir semua sistem pakar (*expert system*) masih harus dapat diimplementasikan dalam komputer besar, sistem pakar yang dijalankan pada komputer pribadi tergolong sistem pakar kecil dan kurang canggih.
- d. Sistem pakar tidak 100 % menguntungkan karena produk seseorang tidak ada yang sempurna dan tidak selalu benar, oleh karena itu perlu dikaji ulang secara teliti sebelum digunakan.

8. Tahapan Pengembangan Sistem Pakar

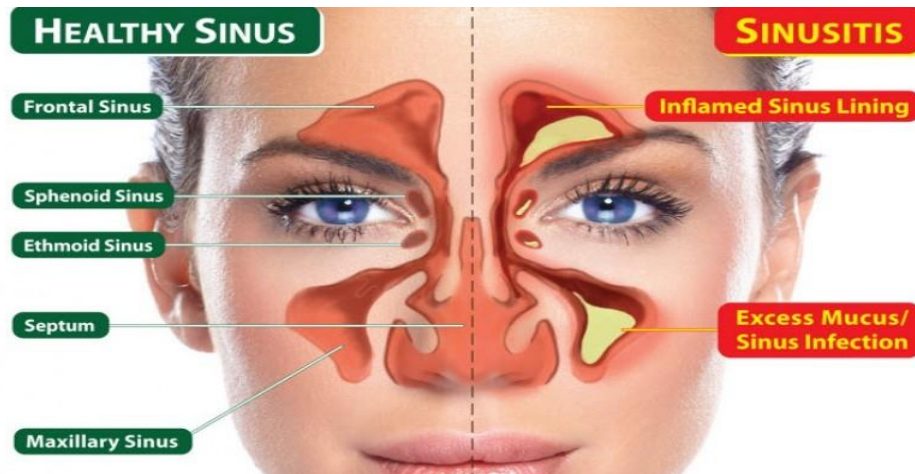
Menurut Repository (2011), Tahapan-tahapan pengembangan sistem pakar antara lain:

- a. *Identifikasi*, Merupakan tahap penentuan hal-hal yang penting sebagai dasar dari permasalahan yang akan dianalisis.
- b. *Konseptualitas*, Hasil *identifikasi* masalah, *dikonseptualisasikan* dalam bentuk *relasi* antar data, hubungan antar pengetahuan dan konsep-konsep penting dan ideal yang akan diterapkan.
- c. *Formalisasi*, Konsep-konsep dari *konseptualisasi* diimplementasikan secara *formal* dalam tahap *formalisasi*.
- d. *Implementasi*, Apabila pengetahuan sudah *diformalisasikan* secara lengkap maka tahap *implementasi* dapat dimulai dengan membuat garis besar masalah kemudian memecahkan masalah kedalam modul-modul.
- e. *Evaluasi*, Tahap ini merupakan tahap pengujian terhadap sistem pakar yang telah dibangun dan untuk menemukan kesalahan-kesalahan yang masih ada.
- f. Pengembangan sistem, Fungsi dari pengembangan sistem adalah agar sistem yang dibangun tidak menjadi usang dan *investasi* tidak sia-sia. Hal pengembangan sistem yang paling berguna adalah proses *dokumentasi* sistem dimana didalamnya tersimpan semua hal penting yang menjadi tolak ukur pengembangan sistem di masa mendatang.

D. SINUSITIS

Menurut Abi Sambuda dalam William (1992), sinusitis adalah peradangan *sinus paranasalis*, dimana dapat disebabkan oleh infeksi maupun bukan infeksi, dari bakteri, jamur, *virus*, alergi maupun sebab *autoimun*.

Sinusitis merupakan peradangan yang terjadi pada *mukosa sinus paranasal*. Umumnya disertai dan dipicu oleh *rinitis* sehingga sering disebut juga *rhinosinusitis*. Bila mengenai beberapa sinus disebut *multisinus* sedangkan bila mengenai semua *sinus paranasal*, disebut *pansinusiti* (Endang,1990).



Gambar 2.4. *Rhinosinusitis*

(sumber : perbedaan antara sinusitis oleh Galena, 2016)

Pansinusiti atau *Sinus Paranasal* merupakan rongga berisi udara yang berbatasan langsung dengan rongga hidung dan berhubungan dengannya. Sinus sebelah *lateral* ialah sinus *maksila (Antrum)* dan sel-sel dari sinus *etmoid*, sebelah *kranial* adalah sinus *frontal*, dan sebelah *dorsal* adalah sinus *sphenoid*. (Buku Saku Ilmu Kesehatan Telinga, Hidung, dan Tenggorokan). Adapun penjelasan dari *Pansinusiti* atau *Sinus paranasal*, yaitu :

1. Sinus *Frontalis* (*Frontal Sinus*)

Bentuk dan ukuran sinus *frontal* sangat bervariasi, dan seringkali juga sangat berbeda bentuk dan ukurannya dari sinus pasangannya. Ukuran rata-rata sinus *frontal* yaitu tinggi 3 cm, lebar 2-2,5 cm, dalam 1,5-2 cm dan isi rata-rata 6-7 ml. Dinding depan sinus *frontal* hampir selalu *diploik*, terutama pada bagian luar atau sudut *infero-lateral* dan pada *sulkus superior* tempat pertemuan dinding *anterior* dan *posterior*.(Repository:Kecerdasan Buatan,2011).

2. Sinus *Etmoidalis* (*Ethmoid Sinus*)

Sinus Etmoid pada orang dewasa berbentuk seperti piramid dengan dasarnya pada bagian *posterior*. Ukurannya dari *anterior* ke *posterior* adalah 4-5 cm, tinggi 2,4 cm dan lebarnya 0,5 cm di *anterior* sedangkan di bagian *posterior* 1,5 cm. (Soetjipto, 2007). Sinus *etmoid* berongga-rongga yang terdiri dari sel-sel seperti sarang tawon, terdapat di dalam *massa* bagian *lateral os etmoid* dan terletak di antara *konka media* dan dinding *medial orbita*. (Repository:Kecerdasan Buatan,2011)

3. Sinus *Maksilaris* (*Maxillary Sinus*)

Sinus *maksila* berbentuk piramid. Sinus *maksila* disebut juga *antrum Highmore* merupakan *sinus paranasal* yang terbesar. Saat lahir sinus *maksila* bervolume 6-8 ml, sinus kemudian berkembang dengan cepat dan akhirnya mencapai ukuran maksimal, yaitu 15 ml saat dewasa dan merupakan sinus yang sering *terinfeksi*.(Repository:Kecerdasan Buatan,2011).

4. Sinus *Sfenoid* (*Sfenoid Sinus*)

Sinus *sfenoid* terletak dalam *os sfenoid* di belakang sinus *etmoid posterior*.

Sinus *sfenoid* dibagi dua oleh sekat yang disebut *septum intersfenoid*.

Ukurannya adalah 2 cm tingginya, dalamnya 2,3 cm dan lebarnya 1,7 cm.

Volume bervariasi dari 5-5,7 ml. Saat sinus berkembang, pembuluh darah dan *nervus* di bagian *lateral os sfenoid* akan menjadi sangat berdekatan dengan rongga sinus .(Repository:Kecerdasan Buatan,2011). Adapun beberapa *klasifikasi* pada sinusitis yaitu :

a. *Klasifikasi* Sinusitis

1) Berdasarkan Lokasinya :

Menurut Abi sumbada dalam Adams (2008) Ditemukan beberapa pasang *sinus paranasalis*, yaitu : *frontalis*, *ethmoidalis*, *maksilaris* dan *sphenoidalis*.

- a) *Sinusitis maksilaris* : menyebabkan nyeri daerah *maksila* seperti sakit gigi dan kepala.
- b) *Sinusitis frontalis* : menyebabkan nyeri pada daerah belakang dan atas mata.
- c) *Sinusitis ethmoidalis* : menyebabkan nyeri pada daerah belakang mata, maupun sakit kepala.
- d) *Sinusitis sphenoidalis* : menyebabkan nyeri pada daerah belakang mata, tetapi lebih sering pada *vertex* kepala.

2) Berdasarkan Durasinya :

Menurut Adams dalam Abi sumbada (2008), Sinusitis berdasarkan durasinya dibagi menjadi:

a) Sinusitis akut

Sinusitis akut bila gejalanya berlangsung dari beberapa hari sampai 4 minggu, Sinusitis akut memiliki gejala *subjektif* dan gejala *objektif*. Gejala *subjektif* bersifat *sistemik* dan *lokal*. Gejala *sistemik* berupa demam dan rasa lesu. Gejala *lokal* dapat kita temukan pada hidung, *sinus paranasal* dan tempat lainnya sebagai nyeri alih (*referred pain*). Gejala pada hidung dapat terasa adanya ingus yang kental dan berbau mengalir ke *nasofaring*. Selain itu, hidung terasa tersumbat. Gejala pada *sinus paranasal* berupa rasa nyeri dan nyeri alih (*referred pain*).

Gejala *subjektif* yang bersifat *lokal* pada sinusitis berupa :

- 1) Gejala *sinusitis maksila* berupa : rasa nyeri dibawah kelopak mata dan kadang tersebar ke *alveolus* sehingga terasa nyeri di gigi. Nyeri alih (*referred pain*) dapat terjadi di dahi dan depan telinga.
- 2) Gejala sinusitis *etmoid* berupa : rasa nyeri pada pangkal hidung, kantung medius, kadang-kadang pada bola mata atau belakang bola mata. Akan terasa makin sakit bila

pasien mengerakkan bola matanya. Nyeri alih (*referred pain*) dapat terasa pada *pelipis (parietal)*.

- 3) Gejala *sinusitis frontal* berupa : rasa nyeri yang terlokalisir pada dahi atau seluruh kepala.
- 4) Gejala *sinusitis sfenoid* berupa : rasa nyeri pada *verteks, oksipital*, belakang bola mata atau daerah *mostaid*.

Gejala objektif sinusitis akut yaitu tampak bengkak pada muka pasien :

- 1) Gejala *sinusitis maksila* berupa pembengkakan pada pipi dan kelopak mata bawah.
 - 2) Gejala *sinusitis frontal* berupa pembengkakan pada dahi dan kelopak mata atas.
 - 3) Pembekakan jarang terjadi pada *sinusitis etmoid* dan *sfenoid* kecuali *kompiliasi*.
- b) Sinusitis subakut

Sinusitis subakut bila gejala berlangsung 4 sampai 8 minggu. Dikatakan sinusitis subakut bila tanda akut sudah reda dan perubahan *histologik mukosa* sinus masih *reversible*, misalnya sudah berubah menjadi jaringan *granulasi* atau *polipoid*. Gejala klinisnya sama dengan sinusitis akut, hanya tanda-tanda radang akutnya (demam, sakit kepala hebat, nyeri tekan) sudah reda.

c) Sinusitis kronis

Sinusitis kronis adalah komplikasi dari berbagai penyakit radang sinus pada umumnya, yang dapat ditegakkan berdasarkan riwayat gejala yang diderita sudah lebih dari 12 minggu, dan sesuai dengan 2 kriteria mayor atau 1 kriteria mayor ditambah 2 kriteria minor. Penyebab multi faktorial dan juga termasuk alergi, faktor lingkungan seperti debu, infeksi bakteri, atau jamur. Faktor non alergi seperti rinitis asomotor dapat juga menyebabkan masalah sinus kronis. Menurut Screiber dalam Abi Sumbada (2005).

Etiologi sinusitis kronis. Infeksi kronis pada sinusitis kronis dapat disebabkan :

- a) Gangguan *drainase* : Gangguan *drainase* dapat disebabkan *obstruksi mekanik* dan kerusakan *silia*.
- b) Perubahan *mukosa* : Perubahan *mukosa* dapat disebabkan alergi, *defisiensi imunologik*, dan kerusakan *silia*.
- c) Pengobatan : Pengobatan infeksi akut yang tdiak sempurna. Sebaiknya, kerusakan *silia* dapat disebabkan oleh gangguan *drainase*, perubahan *mukosa* dan *polusi* bahan kimia.

Gejala sinusitis kronis secara *subjektif*, sinusitis kronis memberikan gejala yaitu :

- a) Hidung : Terasa ada *sekret* dalam hidung.
- b) *Nasofaring* : Terasa ada *sekret pasca nasal (post nasal drip)*. *sekret* ini memicu terjadinya batuk kronis.
- c) *Faring* : Rasa gatal dan tidak nyaman di tenggorokan.
- d) Telinga : Gangguan pendengaran karen sumbatan *tuba Eustachius*.Kepala : Nyeri kepala / sakit kepala yang biasanya terasa pasa pagi hari dan berkurang atau menghilang setelah siang hari. Penyebabnya belum diketahui pasti. Mungkin karena malam terjadi penimbungan ingus dalam *sinus paranasal* dan rongga hidung serta terjasi *statis vena*.
- e) Mata : Terjadi infeksi mata melalui penjalaran duktus nasolakrimalis.
- f) Saliran Napas : Terjadi batuk dan kadang-kadang terjadi komplikasi pada paru-paru seperti bronkitis, bronkiesktasis, dan asma bronkial.
- g) Saluran cerna : Terjadi gastroenteritis akibat tertelannya mukopus. Sering terjadi pada anak-anak.

Secara *objektif*, gejala sinusitis kronis tidak seberat sinusitis akut. Tidak terjadi pembengkakan pada wajah pada sinusitis kronis. Pada pemeriksaan *rinoskopii anterior*

ditemukan *sekret* kental *purulen* di *meatus nasi medius* dan *meatus nasi superior*. *Sekret purulen* juga ditemukan di *nasofaring* dan dapat turun ke tenggorokan pada pemeriksaan *rinoskopi posterior*. Pemeriksaan *mikrobiologik* sinusistis kronis. Biasanya sinusitis kronis terinfeksi oleh kuman campuran, bakteri *aerob* (*S. Aureus*, *S. Viridans* & *H. influenzae*) dan bakteri *anaerob* (*Peptostreptookus* & *Fusobakterium*) (Muhammad,2007).

3) Berdasarkan penyebabnya :

- a) *Rhinogenik* (penyebab kelainan atau masalah di hidung), Segala sesuatu yang menyebabkan sumbatan pada hidung dapat menyebabkan sinusitis.
- b) *Dentogenik/Odontogenik* (penyebab kelainan gigi), yang sering menyebabkan sinusitis infeksi pada gigi geraham atas (*pre molar dan molar*) .(Abi Sumbada, 2008).

Tabel.2.1 Daftar Penyakit beserta Gejala Sinusitis

No	Gejala	Penyakit				
		Frontal	Sfenoid	Etmoid	Maksila	Sinusitis
1.	Demam	-	-	-	-	X
2.	Nyeri tekan pada pipi	-	-	-	X	-
3.	Hidung Tersumbat	-	-	-	-	X

4.	Ingus terasa jatuh kebelakang		-	-	-	X
5.	Nyeri tekan pada Dahi	X	-	-	-	-
6.	Nyeridiatas Kedua Mata	X	-	-	-	-
7.	Nyeri Gigi	-	-	-	X	-
8.	Nyeri di pangkal Hidung	-	-	X	-	-
9.	Nyeri di Kantus Medius	-	-	X	-	-
10.	Nyeri pada Sudut Mata	-	-	X	-	-
11.	Nyeri di bawah Kelopak Mata	-	-	-	X	-
12.	Nyeri di Verteks / puncak kepala	-	X	-	-	-
13.	Nyeri di belakang Bola Mata	-	X	-	-	-
14.	Nyeri pada rahang atas	-	-	-	X	-
15.	Nyeri tekan daerah sinus pada wajah	X	X	X	X	-
16.	Lesu	-	-	-	-	X
17.	Sakit kepala	-	-	-	-	X

(Sumber: dr, Hj,Syahrijuita,M.Kes,Sp. THT-KL)

E. Pengobatan pada Sinusitis

Sinusitis akut biasanya membutuhkan waktu dua minggu untuk membaik. Banyak orang yang berhasil menangani gejala sinusitis sendiri dirumah tanpa perlu bantuan dari medis. Sinusitis biasanya disebabkan oleh virus, seperti *influenza*, sehingga seseorang tidak perlu diresepkan *antibiotik*.

(Medkes:2014). Adapun beberapa langkah dalam penanganan pada sinusitis yaitu sebagai berikut :

1. Perawatan dirumah

Untuk meringankan gejala sinusitis, ada beberapa hal yang bisa dilakukan dirumah, antara lain:

- a. Kompres hangat pada wajah, hal ini akan meringankan rasa nyeri dan memberikan kenyamanan.
- b. Semprot atau tetes hidung akan membantu meringankan hidung yang tersumbat. Menghirup udara hangat dan lembab juga dapat membantu.
- c. Pastikan minum cukup cairan dan cukup beristirahat.

2. Obat-obatan

Jika dibutuhkan obat penghilang rasa sakit, obat *over the counter* penghilang rasa sakit seperti *parasetamol* dan *ibuprofen* bisa digunakan. Baca keterangan dan cara penggunaan obat pada kemasannya, jika memang tidak mengerti, tanyakan pada dokter.

Semprotan dan tetes hidung *dekongestan* hidung dapat mengurangi pembengkakan dan peradangan yang disebabkan sinusitis. Semprot atau tetes hidung *dekongestan* tidak boleh digunakan lebih dari lima hari berturut-turut. Jika digunakan lebih dari ini, akan menyebabkan *kongesti* sinus.

Obat-obat yang bersifat sebagai *dekongestan* terhadap *mukosa* seperti topikal oxymetazoline atau obat dekongestan yang lain, dapat membantu untuk melapangkan mukosa. Baik sinusitis akut dan kronis, dokter mungkin akan meresepkan obat semprot hidung yang mengandung steroid.

Meskipun umumnya tidak digunakan untuk mengobati sinusitis, dokter mungkin juga akan meresepkan *antibiotik* jika terjadi infeksi yang disebabkan bakteri. Penderita mungkin diberikan *antibiotik* bila :

- a. Memiliki masalah kesehatan lain, seperti penyakit jantung, yang akan menambah resiko infeksi.
- b. Gejala parah dan terjadi lebih dari tujuh hari.

Antibiotik yang dipilih adalah golongan *penisilin* seperti *amoksisilin*, jika diperkirakan kumannya telah *resisten* atau memproduksi *beta-laktamase*, maka dapat diberikan *amoksisilin-klavunat* atau jenis *sefalosporin* generasi kedua. Pada sinusitis *antibiotik* diberikan selama 10-14. Jika pasien menderita sinusitis kronis dan menderita alergi atau demam, dokter mungkin akan meresepkan *antihistamin*. Hal ini akan meringankan sinusitis.

3. Operasi

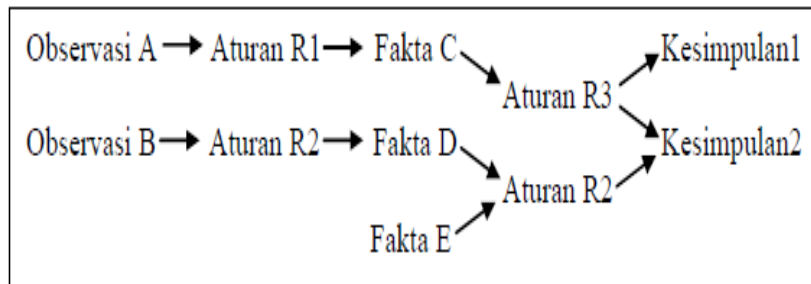
Jika pasien menderita sinusitis kronis dan tidak mendapatkan manfaat dari pengobatan dengan obat-obatan, spesialis THT mungkin akan menyarankan menjalani pembedahan. Pembedahan atau operasi

akan membantu menemukan sinus yang terblokir dan meningkatkan *drainase lendir*. Ada sejumlah prosedur pembedahan yang bisa dilakukan seperti :

- a. *Functional endoscopic sinus sugery (FESS)*, dokter bedah kan membersihkan sinus dan memperluas lubang *drainase* dengan menggunakan *endoskop*. Bisa dilakukan dengan *anestesi* lokal atau umum.
- b. *Balloon sinuplasty* adalah operasi jenis lain yang digunakan untuk mengatasi sinusitis kronis. Dokter bedah akan menempatkan tabung fleksibel kecil ke dalam sinus yang terblokir dan kemudian ujung balon akan mengembalikan fungsi sinus. Tindakan ini memerlukan *anestesi* umum.

F. *Forward Chaining (Runut Maju)*

Forward Chaining merupakan proses perunutan yang dimulai dengan menampilkan kumpulan data atau fakta yang meyakinkan menuju konklusi akhir. Runut maju biasa juga disebut sebagai penalaran *forward (forward reasoning)* atau pencarian yang dimotori data (*data driven search*). Dengan kata lain penalaran dimulai dari fakta terlebih dahulu untuk mendapatkan kebenaran *hipotesa*. (Indra Kurniawan,2012). Gambar berikut menunjukkan proses *forward chaining*.



Gambar 2.5 Diagram *Forward Chaining*

(Sumber :Indra Kurniawan,2012)

Jadi dimulai dari *premis-premis* atau informasi masukan (*if*) dahulu kemudian menuju *konklusi* atau *drived information (then)* atau dapat dimodelkan sebagai berikut :

IF (informasi masukan)

THEN (konklusi)

Informasi masukan dapat berupa data, bukti, temuan, atau pengamatan. Sedangkan *konklusi* dapat berupa tujuan, *hipotesa*, penjelasan atau *diagnosis*. Sehingga jalannya penalaran runut maju dapat dimulai dari data menuju tujuan, dari bukti menuju *hipotesa*, dari temuan menuju penjelasan, atau dari pengamatan menuju diagnosa. (Indra Kurniawan,2012). Dan berikut adalah contoh dari *forward chaining*.

IF nyeri gigi

AND nyeri pada rahang atas

AND demam

AND hidung tersumbat

AND ingust terasa jatuh kebelakang

AND nyeri dibawah kelopak mata

AND nyeri tekan pada pipi

THEN sinusitis maksila

Secara sederhana runut maju atau *forward caining* sederhana untuk kaidah diatas, agar sistem pakar mencapai *konklusi* harus *disuplay* dahulu fakta nyeri pada pipi, lesu, nyeri gigi, nyeri dibawah kelopak mata, di dahi dan depan telinga. Barulah sistem mengeluarkan *konklusi* bahwa *user* menderita penyakit *sinusitis maksila*.

G. Certainty Factor (faktor Kepastian)

Menurut Wahyu Hardianto dalam (Turban,2015) Faktor kepastian (*Certainty factor*) menyatakan kepercayaan dalam sebuah kejadian (atau fakta atau *hipotesa*) berdasarkan bukti atau penilaian pakar. *Certainty Factor* menggunakan suatu nilai untuk mengasumsikan derajat keyakinan seorang pakar terhadap suatu data. Sedangkan menurut David McAllister dalam Widiyanto (2015) *Certainty factor* adalah suatu metode untuk membuktikan apakah suatu fakta itu pasti atau tidak pasti yang berbentuk *metric* yang biasanya digunakan dalam sistem pakar. Metode ini sangat cocok untuk sistem pakar yang mendiagnosa sesuatu yang belum pasti. *Certainty Factor* merupakan konsep keyakinan dan ketidakyakinan yang kemudian diformulakan dalam rumus dasar sebagai berikut :

$$CF [P,E] = MB [P,E] - MD [P,E]$$

Keterangan :

CF = *Certainty Factor*

MB = *Measure of Belief* (Tingkat Kepercayaan)

MD = *Measure of Disbelief* (Tingkat Ketidakpercayaan)

P = *Probability*

E = *Evidence* (Peristiwa/Fakta)

Berikut ini adalah deskripsi beberapa kombinasi *Certainty Factor* terhadap berbagai kondisi :

1. *Certainty Factor* untuk kaidah dengan premis tunggal (*Single premis*):

$$CF (H,E) = CF(E) * CF(rule) = CF(user) * CF(Pakar)$$

2. *Certainty Factor* untuk kaidah dengan premis majemuk (*multiple premis*):

$$CF (A AND B) = \text{Minimum} (CF(a),CF(b)) * CF(rule)$$

$$CF (A OR B) = \text{Maximum} (CF(a),CF(b)) * CF(rule)$$

3. *Certainty Factor* untuk kaidah dengan kesimpulan yang serupa (*similarly concluded*) :

$$CF_{COMBINE} (CF1,CF2) = CF1 + CF2 (1-CF1);$$

Jika CF1 dan CF2 Positif

$$\text{CFCOMBINE}(\text{CF1}, \text{CF2}) = \text{CF1} + \text{CF2} (1 + \text{CF1});$$

Jika CF1 dan CF2 Negatif

$$\text{CFCOMBINE}(\text{CF1}, \text{CF2}) = \text{CF1} + \text{CF2} / (1 - \min\{|\text{CF1}|, |\text{CF2}|\});$$

Jika CF1 atau CF2 Negatif

Adapun kelebihan dan kekurangan dari Metode *Certainty Factors* sebagai berikut :

1. Kelebihan dan kekurangan Metode *Certainty Factors*
 - a. Kelebihan metode *Certainty Factor* adalah :
 - 1) Metode ini cocok dipakai dalam sistem pakar untuk mengukur sesuatu apakah pasti atau tidak pasti dalam mendiagnosa penyakit.
 - 2) Perhitungan dengan menggunakan metode ini dalam sekali hitung hanya dapat mengolah dua data saja sehingga keakuratan data dapat terjaga.
 - b. Kekurangan metode *Certainty Factor* adalah :
 - 1) Ide umum dari pemodelan ketidakpastiaan manusia dengan menggunakan *numerik* metode *certainty factor* biasanya diperdebatkan. Sebagian orang akan membantah pendapat bahwa formula untuk metode metode *Certainty Factor* diatas memiliki sedikit kebenaran.

- 2) Metode ini hanya dapat mengolah ketidakpastian /kepastian hanya 2 data saja. Perlu dilakukan beberapa kali pengolahan data untuk yang lebih dari 2 buah.

H. UML (*Unified Modeling Language*)

UML singkatan dari *Unified Modeling Language* yang berarti bahasa pemodelan standar. Menurut Prabowo Pudjo Widodo dan Herawati (2011 : 6) Ketika kita membuat model menggunakan konsep UML ada aturan-aturan yang harus diikuti. Bagaimana elemen pada model-model yang kita buat berhubungan satu dengan yang lainnya harus mengikuti standar yang ada. UML bukan hanya sekedar diagram tetapi juga mencertiakan konteksnya . UML diaplikaikan untuk maksud tertentu biasanya antara lain untuk :

1. Merancang Perangkat Lunak
2. Sarana Komunikasi antara perangkat lunak dengan proses bisnis
3. Menjabarkan sistem secara rinci untuk analisa dan mencari apa yang diperlukan sistem.
4. Mendokumentasi sistem yang ada, proses-proses dan organisansinya.



Menurut Nugroho (2010:6), UML (*Unified Modeling Language*) adalah ‘bahasa’ pemodelan untuk sistem atau perangkat lunak yang berparadigma ‘*berorientasi objek*’. Pemodelan (*modeling*) sesungguhnya digunakan untuk penyederhanaan permasalahan-permasalahan yang kompleks sedemikian rupa sehingga lebih mudah dipelajari dan dipahami.

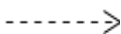



Menurut Widodo (2011:10), “Beberapa literature menyebutkan bahwa UML menyediakan sembilan jenis diagram, yang lain menyebutkan delapan karena ada beberapa diagram yang digabung, misalnya diagram komunikasi, Diagram urutan dan diagram pewaktuan digabung menjadi diagram interaksi”. Namun demikian model model itu dapat dikelompokkan berdasarkan sifatnya yaitu statis atau dinamis. Jenis diagram itu antara lain:




1. *Use Case Diagram*

Bersifat Statis, Diagram ini memperlihatkan himpunan *use case* dan *actor* aktor (suatu jenis khusus dari kelas) diagram ini terutama sangat penting dalam untuk mengorganisasi dan memodelkan perilaku sebuah sistem yang dibutuhkan serta diharapkan pengguna. Menurut Prabowo Pudjo, Widodo dan Herlawati (2011 : 21) Use Case menggambarkan fungsi tertentu dalam suatu sistem berupa komponen, kejadian atau kelas.

Tabel 2.2 *Use Case Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
2		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan

			mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (<i>independent</i>).
3		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
4		<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara <i>eksplisit</i> .
5		<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
6		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
7		<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.

8		<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor
9		<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
10		<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi



(sumber: Prabowo Pudjo,Widodo dan Herlawati:2011)

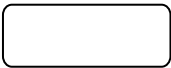

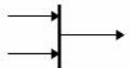
2. *Activity Diagram*




Menurut Prabowo Pudjo, Widodo dan Herlawati (2011 : 143) Diagram aktifitas lebih memfokuskan diri pada eksekusi dan alur sistem dari pada bagaimana sistem itu dirakit. Diagram ini tidak hanya memodelkan *software* melainkan memodelkan model bisnis juga. Diagram aktifitas menunjukkan aktifitas sistem dalam bentuk aksi-aksi. Ketika digunakan dalam pemoelan software, diagram

aktifitas merepresentasikan pemanggilan suatu fungsi tertentu misalnya call. Sedangkan bila dimodelkan dalam pemodelan bisnis, diagram ini menggambarkan aktifitas yang dipicu oleh kejadian-kejadian di luar seperti pemesanan atau kejadian-kejadian internal.

Tabel 2.3 *Activity Diagram*

Simbol	Nama	Keterangan
	Start Point	Start state dengan tegas menunjukkan dimulainya suatu workflow pada sebuah activity diagram. Hanya ada satu start state dalam sebuah workflow. Pada UML, start state digambarkan dengan simbol lingkaran yang solid
	End Point	End state menggambarkan akhir atau terminal dari pada sebuah activity diagram Bisa terdapat lebih dari satu end state pada
		sebuah activity diagram. Pada UML, end state digambarkan dengan simbol sebuah bull's eye (mata sapi)

	<p>Activities</p>	<p>Activity menggambarkan sebuah pekerjaan/tugas dalam workflow.</p> <p>Pada UML, activity digambarkan dengan simbol belah ketupat='lozenge' (horizontal top and bottom with convex sides).</p>
	<p>Fork (Pencabangan)</p>	<p>Diagram Activity dapat dibagi menjadi beberapa jalur kelompok yang menunjukkan obyek yang mana yang bertanggung jawab untuk suatu aktifitas.</p> <p>Peralihan tunggal (<i>single transition</i>) timbul dari setiap adanya <i>activity</i> (aktifitas), yang saling menghubungkan pada aktifitas berikutnya.</p>
	<p>Joint (Penggabungan / Merge)</p>	<p>Sebuah <i>transition</i> (transisi) dapat membuat cabang ke dua atau lebih percabangan <i>exclusive transition</i> (transisi eksklusif).</p> <p>Label <i>Guard Expression</i> (ada di dalam []) yang menerangkan output (keluaran) dari percabangan.</p> <p>Percabangan akan menghasilkan bentuk menyerupai bentuk intan. <i>Transition</i> bisa bercabang menjadi beberapa aktifitas paralel yang disebut Fork. <i>Fork</i> beserta <i>join</i></p>

		(gabungan dari hasil output <i>fork</i>) dalam diagram berbentuk <i>solid bar</i> (batang penuh).
	Decision (Kondisi)	Decision adalah suatu titik/point pada activity diagram yang mengindikasikan suatu kondisi dimana ada kemungkinan perbedaan transisi Pada UML, decision digambarkan dengan sebuah simbol diamond
	State Transition (Alur)	State transition menunjukkan kegiatan apa berikutnya setelah suatu kegiatan sebelumnya. Pada UML, state transition digambarkan oleh sebuah <i>solid line</i> dengan panah
	Swimlane (Kelompok)	<i>Object swimlane</i> untuk menggambarkan objek mana yang bertanggung jawab untuk aktivitas tertentu.

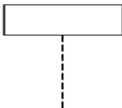


(Sumber : Prabowo Pudjo, Widodo dan Herlawati, 2011)

3. *Sequence Diagram.*

Menurut Prabowo Pudjo, Widodo dan Herlawati (2011 : 174).

Diagram sequence dimaksudkan untuk mengembangkan komunikasi antara objek, bukan memanipulasi data saat berkomunikasi.

Table 2.4 *Sequence Diagram.*

NO	GAMBAR	NAMA	KETERANGAN
1		<i>LifeLine</i>	Objek <i>entity</i> , antarmuka yang saling berinteraksi.
2		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi
3		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi



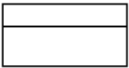


(Sumber : Prabowo Pudjo, Widodo dan Herlawati (2011))



4. *Class Diagram*

Class diagram menurut Munawar (2005 : 28) merupakan himpunan dari objek-objek yang sejenis. Sebuah objek memiliki keadaan sesaat (state) dan perilaku (behavior). State sebuah objek adalah kondisi objek tersebut yang dinyatakan dalam

attribute/properties. Sedangkan perilaku suatu objek mendefinisikan bagaimana sebuah objek bertindak/beraksi dan memberikan reaksi.

Table 2.5 *Class Diagram.*

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
2		<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
3		<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
4		<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor
5		<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.

6		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempegaruhi elemen yang bergantung padanya elemen yang tidak mandiri
7		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya

(Sumber : Prabowo Pudjo, Widodo dan Herlawati (2011)

I. Website

Menurut Sidik Betha (2012:1) *World Wide Web* atau WWW atau juga dikenal dengan WEB adalah salah satu layanan yang didapat diakses oleh pemakai computer yang terhubung ke internet dengan fasilitas *hypertext* untuk menampilkan data berbentuk *text*, gambar, suara, animasi dan data multimedia lainnya. dengan menggunakan teknologi *hypertext*, pemakai dituntun menemukan informasi dengan mengikuti *link* yang disediakan dalam dokumen web yang ditampilkan dalam web *browser*. Sehingga dapat disimpulkan bahwa kelebihan dari aplikasi web adalah dapat diakses kapan pun dan dimana pun selama ada jaringan internet.

Web merupakan bagian dari *Internet* yang paling cepat berkembang dan paling populer. Web bekerja berdasarkan pada tiga mekanisme berikut (Aji Supriyanto, 2007: 3):

1. Protokol standard aturan yang digunakan untuk berkomunikasi pada *computer networking*, *Hypertext Transfer Protocol* (HTTP) adalah protokol untuk WWW.
2. *Address* WWW memiliki aturan penamaan alamat Web yaitu URL (*Uniform Resource Locator*) yang digunakan sebagai standar alamat Internet.
3. HTML digunakan untuk membuat dokumen yang bisa diakses melalui Web.

J. Internet

Pengertian *internet* secara umum adalah kumpulan jaringan komputer yang terhubung baik secara langsung ataupun secara tidak langsung (*Nirkabel*) antara satu perangkat ke perangkat lainnya yang bertujuan untuk pertukaran data menggunakan protokol TCP/IP. Ada banyak sebenarnya arti *internet* ini menurut pendapat ahli di seluruh dunia, namun pergeseran makna *internet* yang perlu disadari adalah internet tidak hanya komunikasi antar perangkat komputer, melainkan juga perangkat lainnya seperti *mobile device* dan bahkan berbagai bentuk peralatan lainnya.

K. Bahasa pemrograman

1. HTML

HTML adalah file teks murni yang dapat dibuat dengan editor teks sembarang. Dokumen ini dikenal sebagai *web page*. Dokumen HTML merupakan dokumen yang disajikan dalam browser *web surfer*. Dokumen ini umumnya berisi informasi atau interface aplikasi di dalam Internet.

Ada dua cara untuk membuat sebuah *web page*: dengan HTML editor atau dengan editor teks biasa. Dokumen HTML disimpan dengan ekstensi *.htm atau *.html (Betha Sidik dan Husni I. Pohan, 2012: 9).

Tag-tag Dasar HTML

Suatu dokumen HTML memiliki komponen-komponen atau tag-tag dasar sebagai berikut (Betha Sidik dan Husni I. Pohan, 2012: 9):

a. HTML

Merupakan tag dasar yang mendefinisikan bahwa dokumen ini adalah dokumen HTML. Tag ini merupakan satu keharusan bagi pemrograman Web untuk menuliskannya sebagai tag pertama dalam dokumen HTML.

`<html>` di awal dokumen dan `</html>` di akhir dokumen

1) Head

Merupakan tag berikutnya setelah `<html>` untuk menuliskan keterangan tentang dokumen Web yang akan ditampilkan.

`<head>` di awal setelah `<html>`, dan `</head>` di akhir *section* head

2) Title

Merupakan tag di dalam head yang harus dituliskan untuk memberikan judul/informasi pada caption browser Web tentang topik atau judul dari dokumen Web yang ditampilkan dalam browser.

`<title>Judul Dokumen</title>`

3) Body

Merupakan section utama dalam dokumen Web. Pada section ini semua isi dokumen yang akan ditampilkan di dalam browser harus dituliskan.

<body> di awal, segera setelah tag </head>

</body> di akhir, diletakkan sebelum tag </html>

2. MySQL

MySQL adalah perangkat lunak yang tergolong sebagai *Database Management System* (DBMS). Perangkat lunak ini bermanfaat untuk mengelola data dengan cara yang sangat fleksibel dan cepat. Berikut adalah sejumlah aktivitas yang terkait dengan data yang didukung oleh MySQL (Abdul Kadir, 2010: 10).

- a) Menyimpan data ke dalam tabel;
- b) Menghapus data dalam tabel;
- c) Mengubah data dalam tabel;
- d) Mengambil data yang tersimpan dalam tabel;
- e) Memungkinkan untuk memilih data tertentu yang diambil;
- f) Memungkinkan untuk melakukan pengaturan hak akses terhadap data.

3. PHP (*Hypertext Preprocessor*)

PHP merupakan salah satu bahasa pemrograman website yang paling banyak digunakan saat ini. Dimana PHP merupakan salah satu

bahasa (*script*) pemrograman yang sering digunakan pada sisi server sebuah Web (Wahana Komputer, 2010: 1).

Aturan Penulisan PHP

Pendefinisian pemrograman PHP dapat dilakukan dengan menggunakan beberapa cara. Terdapat empat cara mendefinisikan *script* PHP, yaitu:

- a. *short style*, merupakan cara yang paling praktis : `<??>`
- b. *standart style*, gaya penulisan ini mirip dengan penulisan program XML : `<?php ?>`
- c. *javascript style* yang mirip dengan penulisan program javascript :
- d. `<script language="PHP"> </script>`
- e. *style ASP* yang sudah dikenal oleh pengguna ASP : `<% %>`

Dalam penulisan *script* PHP, ada dua cara yang sering digunakan, yaitu:

1) *Embedded Script*

Embedded Script yaitu *script* PHP yang menyatu dengan tag-tag HTML dalam suatu file. File yang berisi tag HTML dan *script* PHP ini disimpan dengan ekstensi *.php.

Contoh dari *Embedded Script* :

```
<html>
<head>
<title>Embedded Script</title>
</head>
<body>
<?php
```

```
Echo "Hello World";  
  
?>  
  
</body>  
  
</html>
```

2) *Non-Embedded Script*

Non-Embedded Script adalah *script* program PHP murni.

Termasuk tag HTML yang disisipkan dalam *script* PHP.

Contoh dari *Non-Embedded Script* :

```
<?php  
echo "<html>";  
echo "<head>";  
echo "<title>Menenal PHP</title>";  
echo "</head>";  
echo "<body>";  
echo "<p>Hello World</p>";  
echo "</body>";  
echo "</html>";  
  
?>
```

4. Java script

Javascript adalah bahasa scripting yang handal yang berjalan pada sisi client. JavaScript merupakan sebuah bahasa scripting yang dikembangkan oleh Netscape. Untuk menjalankan script yang ditulis dengan JavaScript kita membutuhkan JavaScript-enabled browser yaitu browser yang mampu menjalankan JavaScript.

5. CSS

Cascading Style Sheets (CSS) adalah suatu bahasa stylesheet yang digunakan untuk mengatur tampilan suatu dokumen yang ditulis dalam bahasa markup. Penggunaan yang paling umum dari CSS adalah untuk memformat halaman web yang ditulis dengan HTML dan XHTML. Walaupun demikian, bahasanya sendiri dapat dipergunakan untuk semua jenis dokumen XML termasuk SVG dan XUL. Spesifikasi CSS diatur oleh World Wide Web Consortium (W3C).

L. Aplikasi Pendukung

1. XAMPP

Menurut Yogi wicaksono (2008:7) “XAMPP adalah sebuah *software* yang berfungsi untuk menjalankan *website* berbasis PHP dan menggunakan pengolah data *MySQL* dikomputer local”. XAMPP berperan sebagai *server web* pada komputer anda. XAMPP juga dapat disebut sebuah *CPanel server virtual*, yang dapat membantu anda melakukan *preview* sehingga dapat memodifikasi *website* tanpa harus *online* atau terakses dengan internet.

2. Sublime Text 3

Sublime Text adalah teks editor berbasis Python, sebuah teks editor yang elegan, kaya akan fitur, cross-platform, mudah dan simpel yang cukup terkenal di kalangan pengembang, penulis, dan desainer. Para programmer biasanya menggunakan sublime Text untuk menyunting source code yang sedang dikerjakan.

3. Microsoft Visio 2007 Pro

Microsoft Visio (atau sering disebut Visio) adalah sebuah program aplikasi komputer yang sering digunakan untuk membuat diagram, diagram alir (flowchart), UML, brainstorm, dan skema jaringan yang dirilis oleh Microsoft Corporation. Aplikasi ini menggunakan grafik vektor untuk membuat diagram-diagramnya.

Visio aslinya bukanlah buatan Microsoft Corporation, melainkan buatan Visio Corporation, yang diakuisisi oleh Microsoft pada tahun 2000. Versi yang telah menggunakan nama Microsoft Visio adalah Visio 2002, Visio 2003, Visio 2007, Visio 2010 dan Visio 2013 yang merupakan versi terbaru. Visio 2013 Standard dan Professional menawarkan antarmuka pengguna yang sama, tapi seri Professional menawarkan lebih banyak pilihan template untuk pembuatan diagram yang lebih lanjut dan juga penataan letak (layout). Selain itu, edisi Professional juga memudahkan pengguna untuk mengoneksikan diagram-diagram buatan mereka terhadap beberapa sumber data dan juga menampilkan informasi secara visual dengan menggunakan grafik.

M. Metode Pengujian *White Box*

1. *White Box Testing*

Ujicoba *Whitebox* merupakan metode desain uji kasus yang menggunakan struktur kontrol dari desain prosedural untuk menghasilkan kasus-kasus uji. Dengan menggunakan metode ujicoba whitebox, para pengembang software dapat menghasilkan kasus-kasus uji yang :

- a. Menjamin bahwa seluruh independent paths dalam modul telah dilakukan sedikitnya satu kali,
- b. Melakukan seluruh keputusan logikal baik dari sisi benar maupun salah.
- c. Melakukan seluruh perulangan sesuai batasannya dan dalam batasan operasionalnya.
- d. Menguji struktur data internal untuk memastikan validitasnya

Mengapa menghabiskan banyak waktu dan usaha dengan menguji *logikal software*?. Hal ini dikarenakan sifat kerusakan alami dari *software* itu sendiri, yaitu :

- a. Kesalahan logika dan kesalahan asumsi secara *proposional* terbalik dengan kemungkinan bahwa alur program akan dieksekusi. Kesalahan akan selalu ada ketika mendesain dan implementasi fungsi, kondisi atau kontrol yang keluar dari alur utama. Setiap harinya pemrosesan selalu berjalan dengan baik dan dimengerti sampai bertemu "kasus spesial" yang akan mengarahkannya kepada kehancuran.
- b. Sering percaya bahwa alur logikal tidak akan dieksekusi ketika diketemukannya, mungkin akan dieksekusi dengan basis regular. Alur logika program biasanya berkebalikan dari intuisi, yaitu tanpa disadari asumsi mengenai alur kontrol dan data dapat mengarahkan

pada kesalahan desain yang tidak dapat terlihat hanya dengan satu kali ujicoba.

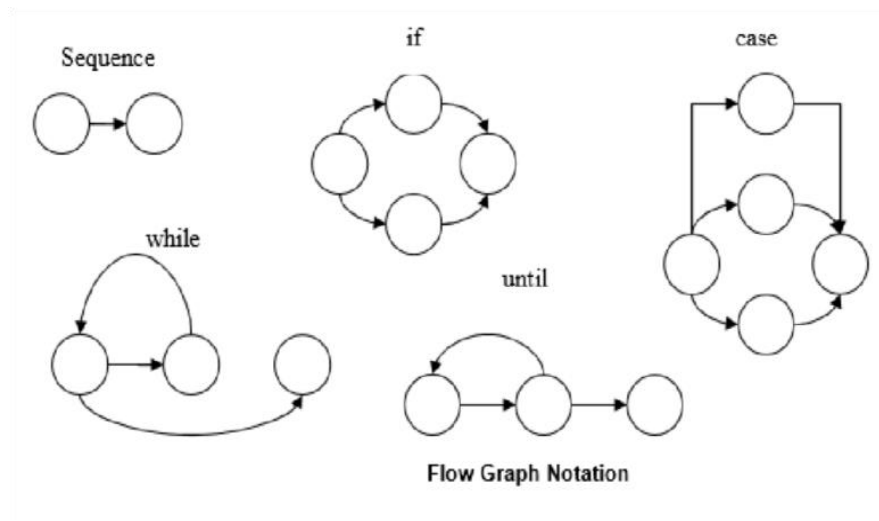
- c. Kesalahan typographical (cetakan) bersifat random. Ketika program diterjemahkan kedalam kode sumber bahasa pemrograman, maka akan terjadi kesalahan penyetikan. Banyak yang terdeteksi dengan mekanisme pemeriksaan sintaks, tetapi banyak juga yang tidak terdeteksi sampai dengan dimulainya ujicoba. Karena alasan tersebut diatas, maka ujicoba whitebox testing diperlukan selain blackbox testing.

2. Ujian Berbasis Alur (*Basis Path Testing*)

Ujicoba berbasis alur merupakan teknik ujicoba *whitebox* pertama yang diusulkan oleh Tom McCabe. Metode berbasis alur memungkinkan perancang kasus uji untuk menghasilkan ukuran kompleksitas logikal dari desain prosedural dan menggunakan ukuran ini untuk mendefinisikan himpunan basis dari alur eksekusi. Kasus uji dihasilkan untuk melakukan sekumpulan basis yang dijamin untuk mengeksekusi setiap perintah dalam program, sedikitnya satu kali selama ujicoba.

3. Notasi graf Alur (*Path Graph Notation*)

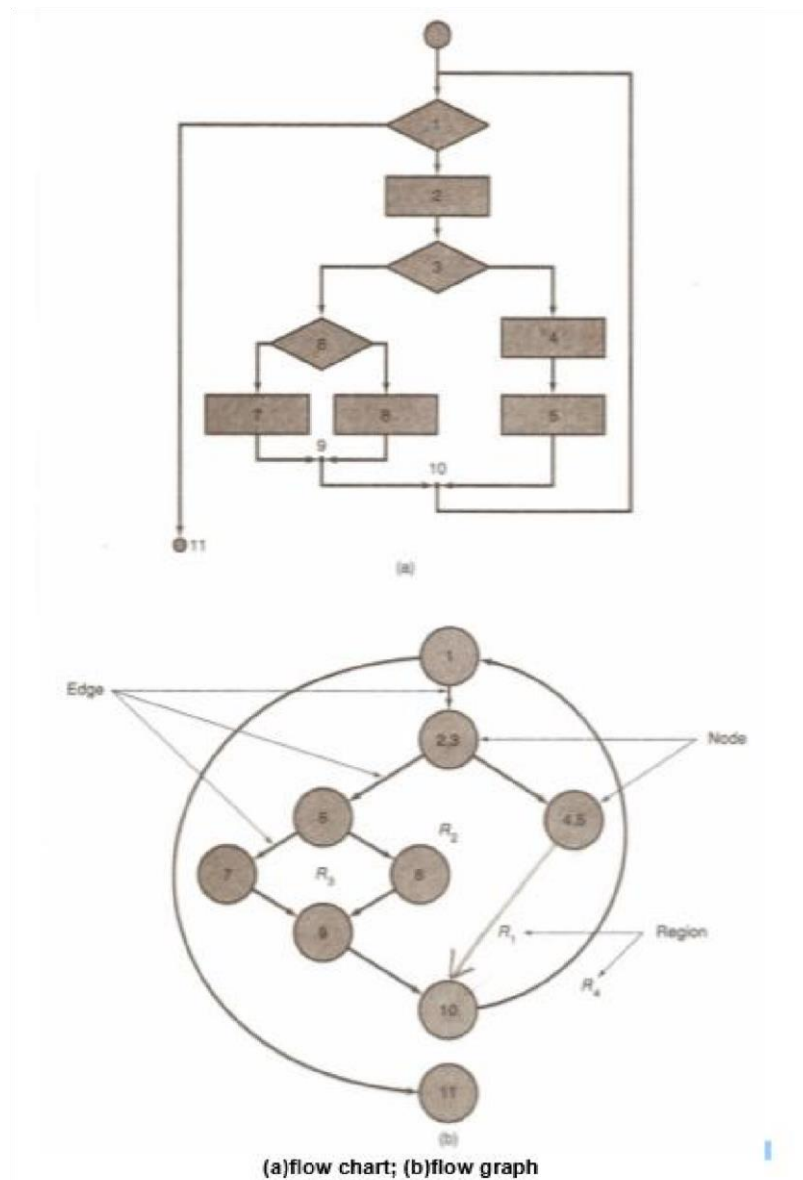
Notasi sederhana untuk merepresentasikan alur kontrol disebut graf alur (*flow graph*), seperti gambar dibawah ini :



Gambar 2.6. *flowgraph*

(Sumber :Ayuliana,Testing dan Implementasi 2009)

Untuk mengilustrasikan kegunaan dari diagram alir dapat dilihat pada gambar dibawah ini. Gambar bagian (a) digunakan untuk menggambarkan struktur kontrol program, sedangkan gambar bagian (b) setiap lingkaran disebut dengan flow graph node, merepresentasikan satu atau lebih perintah prosedural. Urutan dari simbol proses dan simbol keputusan dapat digambarkan menjadi sebuah node, sedangkan anak panah disebut edges, menggambarkan aliran dari kontrol sesuai dengan diagram alir. Sebuah edge harus berakhir pada sebuah node walaupun tidak semua node merepresentasikan perintah prosedural. Area yang dibatasi oleh edge dan node disebut region, area diluar graph juga dihitung sebagai region.

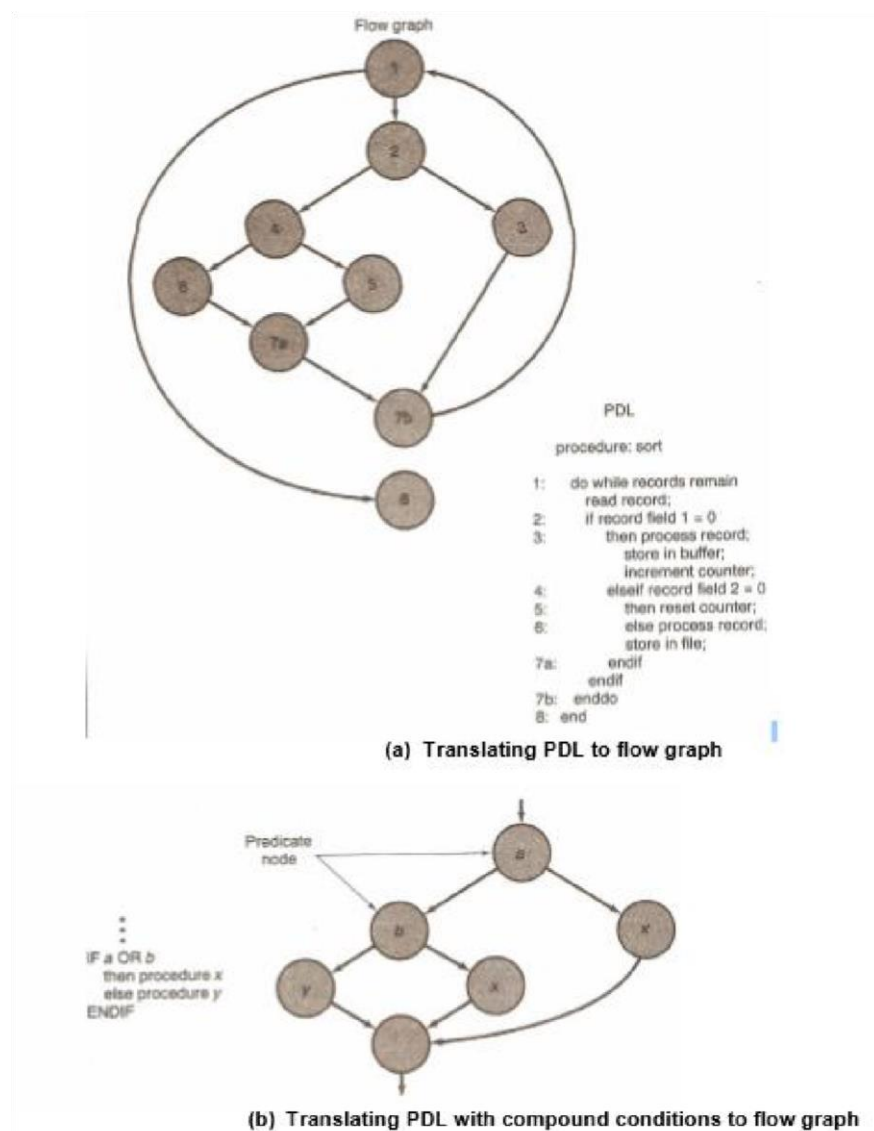


Gambar 2.7 *Flowchart* dan *Flowgraph*

(Sumber :Ayuliana,Testing dan Implementasi 2009)

Setiap representasi rancangan prosedural dapat diterjemahkan kedalam flow graph. Gambar (a) di bawah ini merupakan bagian dari PDL (Program Design Language) dan flow graph-nya (perhatikan nomor untuk setiap perintahnya) Ketika kondisi gabungan ditemukan, maka

penggambaran flow graph akan menjadi lebih rumit. Kondisi gabungan biasanya muncul jika satu atau lebih operator Boolean (OR, AND, NAND, NOR) ditemukan dalam perintah, seperti terlihat pada gambar (b) dibawah ini :



Gambar 2.8 *Translating*

(Sumber :Ayuliana,Testing dan Implementasi 2009)

4. Cyclomatic Complexity

Cyclomatic Complexity merupakan software metric yang menyediakan ukuran kuantitatif dari kompleksitas logikal suatu program. Ketika digunakan dalam konteks metode ujicoba berbasis alur, nilai yang dikomputasi untuk kompleksitas cyclomatic mendefinisikan jumlah independent path dalam himpunan basis suatu program dan menyediakan batas atas untuk sejumlah ujicoba yang harus dilakukan untuk memastikan bahwa seluruh perintah telah dieksekusi sedikitnya satu kali.

Independent path adalah alur manapun dalam program yang memperkenalkan sedikitnya satu kumpulan perintah pemrosesan atau kondisi baru. Contoh *independent path* dari gambar flow graph diatas :

Path 1 : 1 – 11

Path 2 : 1 – 2 – 3 – 4 – 5 – 10 – 1 – 11

Path 3 : 1 – 2 – 3 – 6 – 8 – 9 – 10 – 1 – 11

Path 4 : 1 – 2 – 3 – 6 – 7 – 9 – 10 – 1 – 11

Misalkan setiap path yang baru memunculkan edge yang baru, dengan

path : 1 - 2 – 3 – 4 – 5 – 10 - 1 - 2 – 3 – 6 – 8 – 9 – 10 – 1 – 11

Path diatas tidak dianggap sebagai independent path karena kombinasi path diatas telah didefinisikan sebelumnya Ketika ditetapkan dalam graf alur, maka independent path harus bergerak sedikitnya 1 edge

yang belum pernah dilewati sebelumnya. Kompleksitas cyclomatic dapat dicari dengan salah satu dari 3 cara berikut :

- a. Jumlah region dari graf alur mengacu kepada kompleksitas cyclomatic
- b. Kompleksitas cyclomatic untuk graf alur G didefinisikan :

$$V(G) = E - N + 2$$

Dimana E = jumlah edge, dan N = jumlah node

- c. Kompleksitas cyclomatic untuk graf alur G didefinisikan :

$$V(G) = P + 1$$

Dimana P = jumlah predicates nodes

Berdasarkan flow graph gambar (b) diatas, maka kompleksitas cyclomatic-nya dapat di hitung sebagai berikut :

- a. Flow graph diatas mempunyai 4 region
- b. $V(G) = 11 \text{ edges} - 9 \text{ nodes} + 2 = 4$
- c. $V(G) = 3 \text{ predicates nodes} + 1 = 4$

Hasil kompleksitas *cyclomatic* menggambarkan banyaknya path dan batas atas sejumlah ujicoba yang harus dirancang dan dieksekusi untuk seluruh perintah dalam program.

N. Metode Pengujian *Black Box*

1. Black Box Testing

Menurut Black (2007,45), *behavioral* atau *black box tests* merupakan pengujian mengenai apa yang dilakukan oleh system, terutama perilakunya (*behavior*) dan masalah-masalah bisnis. *Black box test* dilakukan untuk mengidentifikasi bug yang terdapat pada hasil-hasil, pemrosesan dan perilaku dari *sistem Black box test* biasanya dilakukan oleh *tester*.

Sementara itu Tian (2005, 74) menyebutkan bahwa *Black box Testing* atau disebut juga *Functional Testing* berfokus pada perilaku eksternal dari suatu *software* atau berbagai komponennya sambil memandang obyek yang diuji sebagai sebuah kotak hitam (*black box*) sehingga mencegah tester untuk melihat isi-isi di dalamnya, *Black box testing* menverifikasi penanganan yang benar dari fungsi-fungsi eksternal yang disediakan oleh *software* atau apakah perilaku yang diamati tersebut memenuhi harapan - harapan user atau spesifikasi produk (2005, p53).

Tian (2005, 75) berpendapat bahwa bentuk yang paling sederhana dari *Black Box Test* (BBT) adalah dengan mulai menjalankan *software* dan melakukan pengamatan dengan harapan mudah untuk membedakan mana hasil yang diharapkan dan mana tidak. Bentuk ini disebut juga "*ad hoc testing*". Setelah dilakukan pengujian berulang kali dan ditentukan bahwa masalah-masalah terjadi karena *software* dan bukan karena *hardware*, maka informasi tersebut disampaikan kepada pihak yang bertanggung

jawab untuk memperbaiki masalah-masalah tersebut. Bentuk lain dari BBT adalah penggunaan *checklist* yang spesifik yang berisikan daftar fungsi-fungsi eksternal apa yang seharusnya ada serta beberapa informasi mengenai kinerja yang diharapkan atau pasangan input-input.

2. Ciri-ciri *Black Box Testing*

- a. *Black box testing* berfokus pada kebutuhan *fungsi* pada *software*, berdasarkan pada spesifikasi kebutuhan dari *software*.
- b. *Black box testing* melakukan pengujian tanpa pengetahuan detail struktur internal dari sistem atau komponen yang dites. juga disebut sebagai *behavioral testing*, *specification-based testing*, *input/output testing* atau *functional testing*.
- c. *Black box testing* bukan teknik alternatif daripada *white box testing*. Lebih daripada itu, ia merupakan pendekatan pelengkap dalam mencakup *error* dengan kelas yang berbeda dari metode *white box testing*.

3. Kelebihan *Black box testing*

- a. Dapat memilih *subnet test* secara efektif dan efisien.
- b. Dapat menemukan cacat
- c. Memaksimalkan *testing investmen*

4. Kelemahan *Black box testing*

Tester tidak pernah yakin apakah *PI* tersebut benar-benar lulus uji.

5. Metode *Black box testing*

- a. Dilakukan oleh penguji *independent*.

b. Melakukan pengujian berdasarkan apa yang dilihat, hanya fokus terhadap *fungsionalitas* dan *uotput*. Pengujian lebih ditujukan pada desain *software* sesuai standar dan reaksi apabila terdapat celah-celah bug/vulnerabilitas pada program aplikasi tersebut setelah dilakukan *white box testing*.

6. Pegujian Validasi

Agus (2012) Setelah semua kesalahan diperbaiki maka langkah selanjutnya adalah validasi testing. Pengujian validasi dikatakan berhasil bila fungsi yang ada pada perangkat lunak sesuai dengan yang diharapkan pemakai. Validasi perangkat lunak merupakan kumpulan seri uji coba black box yang menunjukkan sesuai dengan yang diperlukan.

Kemungkinan kondisi setelah pengujian:

- a. Karakteristik performansi fungsi sesuai dengan spesifikasi dan dapat diterima.
- b. Penyimpangan dari spesifikasi ditemukan dan dibuatkan daftar penyimpangan.

7. Pengujian *Alpha* dan *Beta*

Agus (2012) Apabila perangkat lunak dibuat untuk pelanggan maka dapat dilakukan *acceptance test* sehingga memungkinkan pelanggan untuk memvalidasi seluruh keperluan. Test ini dilakukan karena memungkinkan pelanggan menemukan kesalahan yang lebih rinci dan membiasakan pelanggan memahami perangkat lunak yang telah dibuat.

- a. Pengujian *Alpha*

Dilakukan pada sisi pengembang oleh seorang pelanggan. Perangkat Lunak digunakan pada setting yg natural dengan pengembang “yang memandang” melalui bahu pemakai dan merekam semua kesalahan dan masalah pemakaian.

b. Pengujian Beta

Dilakukan pada satu atau lebih pelanggan oleh pemakai akhir perangkat lunak dalam lingkungan yang sebenarnya. Pengembang biasanya tidak ada pada pengujian ini. Pelanggan merekam semua masalah di hasil (real atau imajiner) yang ditemui selama pengujian dan melaporkan pada pengembang pada interval waktu tertentu.

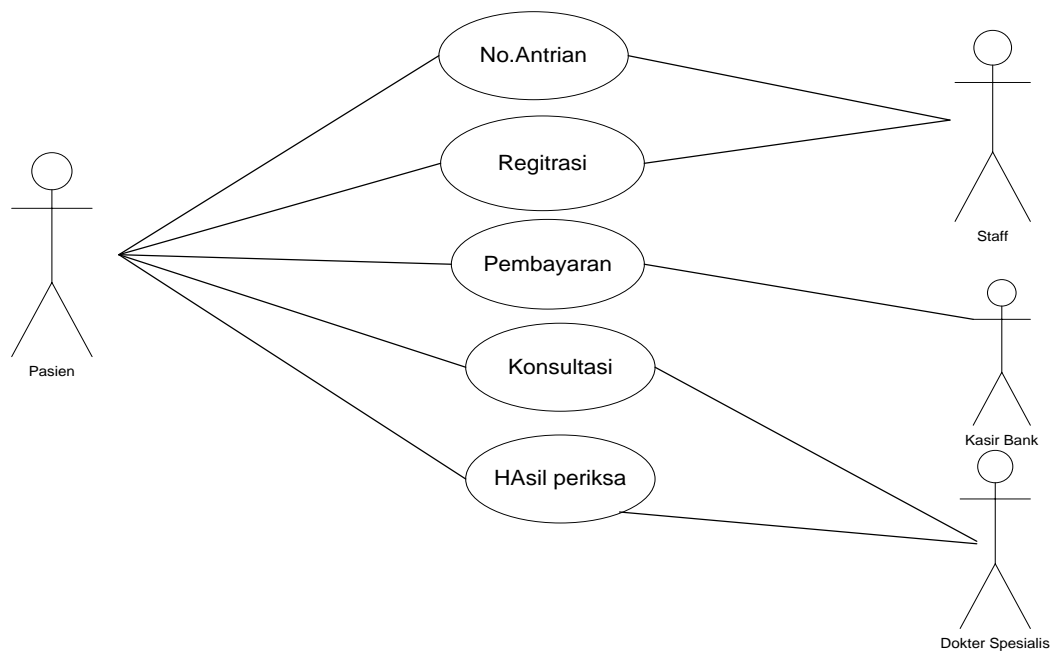
BAB III

METODE PENELITIAN

A. Penggambaran Sistem

1. Sistem Yang sedang Berjalan

Sistem yang berjalan dapat dituangkan dalam bentuk diagram-diagram yang disesuaikan dengan prinsip OOAD (*Object Oriented Analysis and Design*) menggunakan UML Diagram.



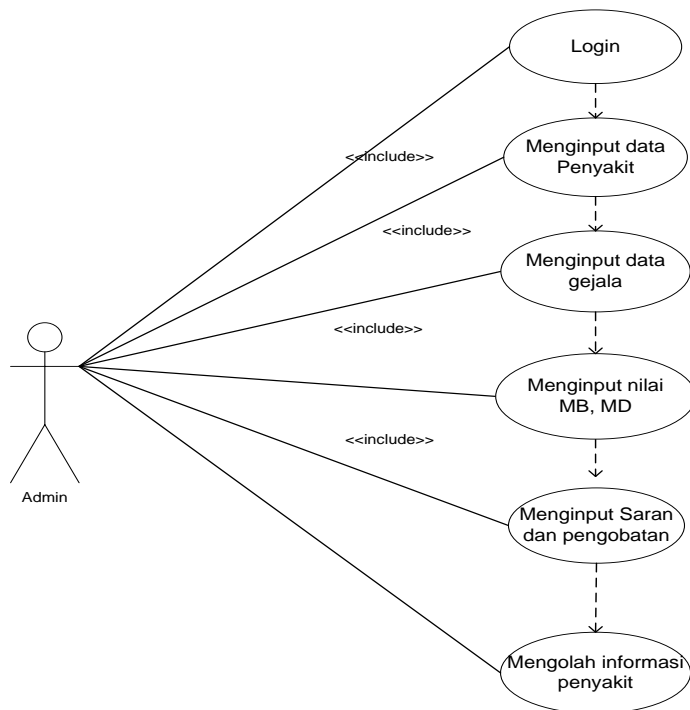
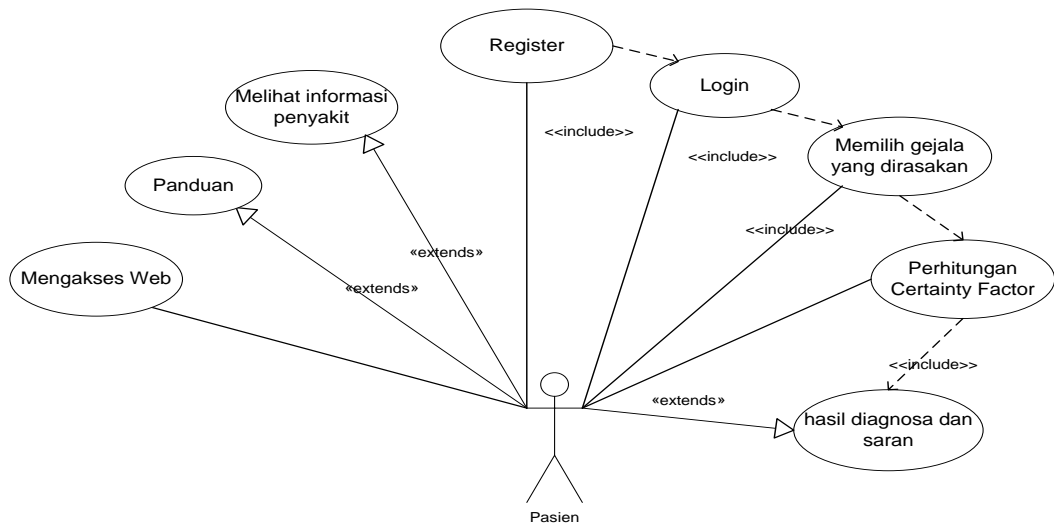
Gambar 3.1 Use Case Diagram sistem yang sedang berjalan

Penjelasan dari Use Case Diagram diatas yaitu :

- a. Pasien mengambil nomor antrian
- b. Melakukan registrasi, dimana pasien menyetorkan berkas ke bagian staff.
- c. Setelah pasien menyetorkan berkas, maka bagian staff akan memberikan kartu periksa dan memberikan kwitansi pembayaran yang ditujukan ke bagian Bank.
- d. Pasien melakukan pembayaran di bagian Bank yang tersedia, kemudian pihak kasir Bank memberikan kwitansi pembayaran yang telah tervalidasi sebagai bukti pembayaran.
- e. Pasien masuk ke dalam ruang Dokter Spesialis THT-KL dan melakukan konsultasi.
- f. Dokter Spesialis THT-KL akan memberikan hasil periksa (diagnosa) kepada pasien.
- g. Pasien diberikan resep obat oleh dokter dan menebusnya dibagian apotek.

2. Sistem Yang Diusulkan

Setelah analisa sistem yang berjalan dilaksanakan maka kemudian dilakukan desain sistem yang baru. Tujuan utama dari sistem yang baru ini adalah memberikan kemudahan bagi masyarakat umum dalam mendiagnosa penyakit Sinusitis yang di derita.



Gambar 3.2 Use Case Diagram sistem yang diusulkan

Penjelasan dari Use Case Diagram yang diusulkan diatas yaitu :

- a. Use case terdiri dari 2 aktor yaitu : Admin dan Pasien
- b. Semua aktor melakukan login, tapi untuk pasien harus melakukan registrasi terlebih dahulu kemudian login ke sistem.
- c. Admin menginput data penyakit, data gejala, menginput nilai Mb dan Md dan mengolah informasi tentang sinusitis.
- d. Admin dapat menambah, mengubah, menghapus semua data yang telah diinputkan.
- e. Pasien dapat memilih dengan cara mencentang gejala-gejala sesuai yang dialami pasien.
- f. Pasien dapat melihat informasi tentang penyakit sinusitis dan dapat melihat panduan penggunaan sistem pakar diagnosa sinusitis.
- g. Pasien juga dapat mengetahui informasi informasi tentang pakar, baik kontak klinik dan data dari dokter Spesialis THT-KL.
- h. Kemudian pasien melihat hasil diagnosa penyakit sinusitis yang di derita.

B. Analisis Kebutuhan

Hal pertama yang perlu dilakukan dalam analisis kebutuhan sistem adalah menentukan kebutuhan-kebutuhan sistem. Kebutuhan sistem terbagi menjadi dua yaitu kebutuhan fungsional dan kebutuhan sistem non-fungsional.

1. Kebutuhan Fungsional

Kebutuhan fungsional adalah kebutuhan-kebutuhan yang memiliki keterkaitan langsung dengan sistem, kebutuhan fungsional dari sistem ini meliputi :

a. Kebutuhan user

- 1) Sistem menyediakan informasi mengenai penyakit sinusitis
- 2) Sistem menyediakan informasi tentang gejala-gejala penyakit Sinusitis.
- 3) Sistem dapat menampilkan hasil dagnosa dari konsultasi penyakit yang dilakukan yang terdiri Sinusitis Maksila, Sinusitis Frontalis, Sinusitis etmoid dan Sinusitis Sfenoid.
- 4) Sistem dapat memberikan saran kepada pasien sebagai hasil analisis

b. Pengujian

- 1) Validasi.
- 2) White Box Testing

2. Kebutuhan Non-fungsional

Kebutuhan non fungsional adalah usulan yang direkomendasikan kepada pengguna agar sistem pakar yang dibangun dapat *user friendly* dan perangkat keras yang digunakan dapat mendukung secara maksimal terhadap kinerja sistem pakar.

a. Kebutuhan Antarmuka (*Interface*)

1) Kebutuhan Antarmuka Pengguna

Sistem pakar ini di bangun menggunakan pemrograman PHP untuk mengolah user *interface*. Sistem ini disediakan menu utama dengan sub menu yang dapat diakses sesuai dengan fungsinya masing-masing.

2) Kebutuhan Antarmuka Perangkat Keras

Untuk kebutuhan spesifikasi perangkat keras yang digunakan dalam sistem ini antara lain :

- a) Laptop Acer intel inside e1
- b) Memory Ram 32 MB atau yang lebih tinggi
- c) Harddisk 500 Gb

3) Kebutuhan Antarmuka Perangkat Lunak

Untuk kebutuhan spesifikasi perangkat lunak yang digunakan dalam sistem ini antara lain :

- a) OS : Microsoft Windows 7
- b) XAMPP
- c) MySQL
- d) Sublime Text 3
- e) Apache
- f) Web Browser

C. Teknik Pengumpulan Data

1. Wawancara

Pengumpulan data dengan cara melakukan komunikasi dan wawancara secara langsung dengan Dokter Spesialis THT-KL.

2. Studi Pustaka

Pengumpulan data dengan menggunakan atau mengumpulkan sumber-sumber tertulis, dengan cara membaca, mempelajari dan mencatat hal-hal penting yang berhubungan dengan masalah yang sedang dibahas guna memperoleh gambaran secara teoritis yang dapat menunjang pada penyusunan proposal.

D. Lokasi dan waktu Penelitian

Penelitian di lakukan pada klinik THT-KL dr. Hj Syahrijuita. Klinik Spesials THT-KL yang berlokasi di jalan batua raya IV no 17, Panakkukang, Kota Makassar. Sebagai sumber untuk mendapatkan data dan informasi dalam merancang dan membangun sistem pakar diagnosa penyakit Sinusitis dengan menggunakan metode *Cetainty Factor*.

E. Tahapan Kegiatan

Tahap-tahap yang dilakukan dalam perancangan sistem ini adalah sebagai berikut :

1. Pengumpulan data : mengumpulkan data—data yang berkaitan dengan penelitian.

2. Analisis sistem : penguraian dari suatu aplikasi yang utuh ke dalam bagian-bagian komponennya dengan maksud untuk mengidentifikasi dan mengevaluasi permasalahan, kesempatan, keselamatan, hambatan yang terjadi dan kebutuhan yang diharapkan sehingga dapat diusulkan perbaikannya.
3. Desain sistem : mengidentifikasi komponen-komponen sistem informasi yang akan dibuat secara rinci, diantaranya : desain model, desain output, desain input dan desain database.
4. Pembuatan sistem : merupakan tahap menstransformasikan hasil analisis ke dalam bentuk aplikasi *website*.
5. Pengujian sistem : mengetahui cara kerja dari aplikasi yang dirancang secara terperinci sesuai spesifikasi dan menilai apakah setiap fungsi atau prosedur yang dirancang sudah bebas dari kesalahan logika.

BAB IV

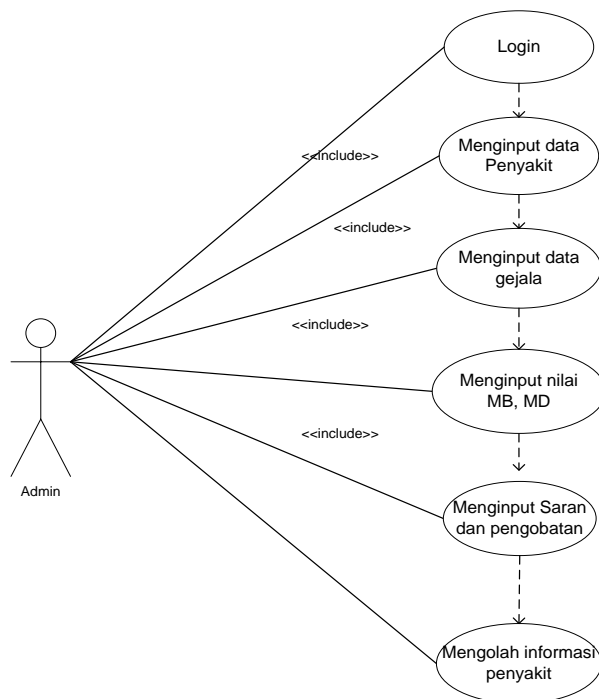
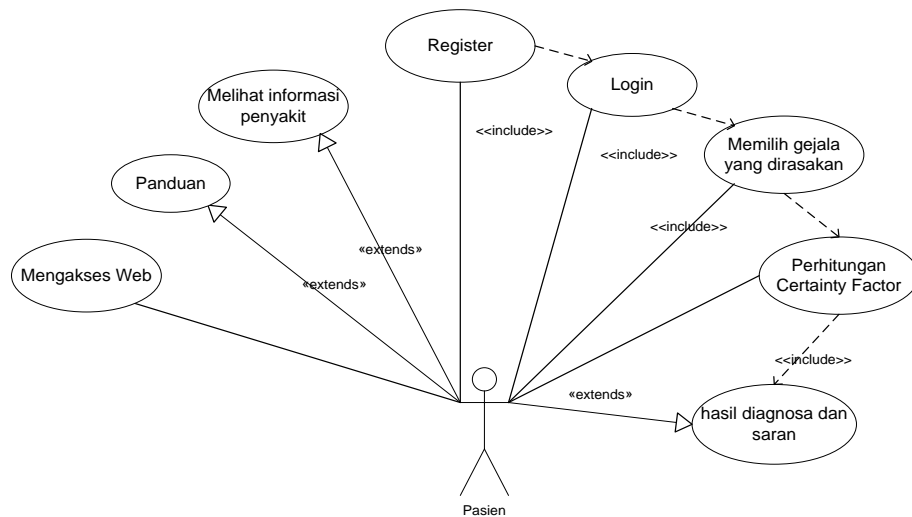
PERANCANGAN DAN IMPLEMENTASI SISTEM

A. Pemodelan Sistem

Pemodelan sistem adalah suatu kegiatan membuat desain teknis berdasarkan evaluasi yang telah dilakukan pada kegiatan analisis.

Pada pemodelan / perancangan desain sistem ini, akan menguraikan tentang pemodelan sistem yang akan dirancang yang bertujuan untuk menggambarkan kondisi dan bagian-bagian yang berperan dalam sistem yang dirancang. Pemodelan sistem yang dilakukan adalah dengan membuat rancangan *flowchart*, *use-case diagram*, *activity diagram*, *sequence diagram* dan *class diagram*

1. *Use Case Diagram*



Gambar 4.1 Use Case Diagram

Pada *Use Case Diagram* terdapat dua aktor yaitu : Admin dan Pasien

a. Admin

Actor : Admin

Brief Description : *Use Case login*, Menginput Data Penyakit, Menginput Data Gejala, Menginput Nilai CF, Mengolah Informasi Penyakit serta panduan penggunaan dan logout.

Main Flow : super admin pertama kali melakukan login untuk masuk ke sistem, setelah itu admin akan melakukan pengolahan informasi tentang penyakit sinusitis, menginput data penyakit sinusitis, menginput data gejala sinusitis, dan menginput nilai CF yaitu nilai MB (*Measure of Belief*) dan nilai MD (*Measure of Disbelief*) serta melakukan *logout*.

b. Pasien

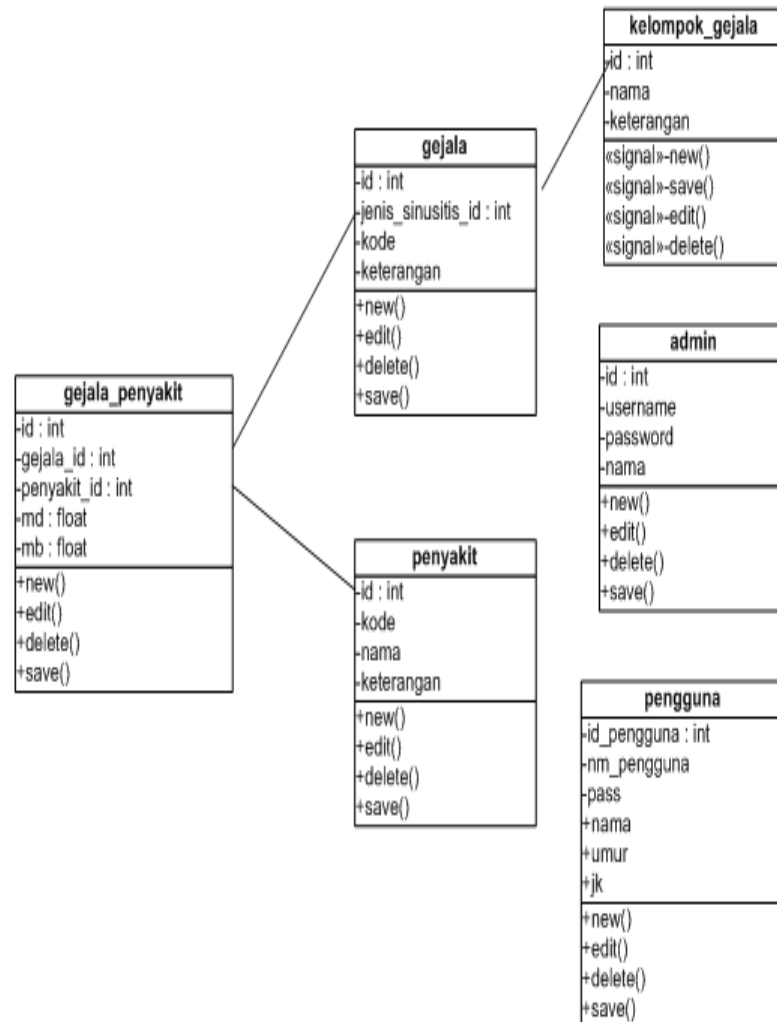
Actor : *User / Pasien*

Brief Description : Mengakses Web.

Main Flow : Untuk mendapatkan informasi tentang penyakit sinusitis *user / pasien* melakukan akses ke halaman front web informasi diagnosa penyakit sinusitis yang nantinya akan menampilkan beberapa halaman page dari web sistem pakar diagnosa penyakit

sinusitis, seperti halaman beranda yang berisi informasi seputar sinusitis, halaman panduan yang berisi tentang petunjuk penggunaan sistem pakar, Informasi pakar Spesialis sinusitis, pada halaman beranda juga terdapat pilihan untuk melakukan registrasi untuk *user / pasien* yang ingin melakukan konsultasi, halaman diagnosa yang berisi : halaman gejala penyakit dimana *user / pasien* harus memilih dan mencentang gejala yang sesuai dengan gejala yang dirasakan setelah selesai maka *User / Pasien* akan mendapat hasil dari diagnosa serta saran pengobatan terhadap penyakit sinusitis yang dialami.

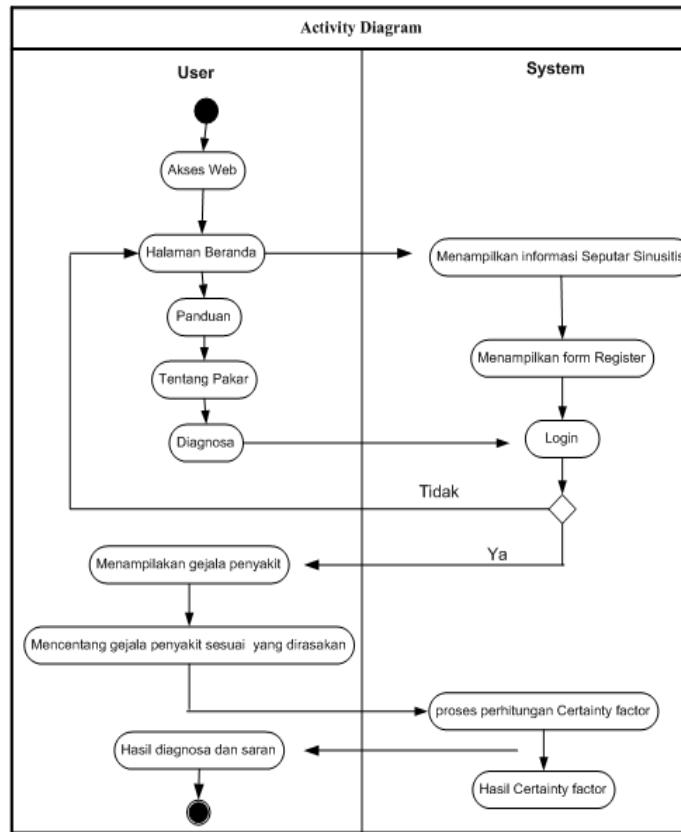
2. Class Diagram



Gambar 4. 2 Class Diagram

Class Diagram digunakan untuk menampilkan kelas-kelas dan paket-paket di dalam sistem. . *Class Diagram* memberikan gambaran sistem secara statis dan relasi antar mereka.

3. Activity Diagram User / Pasien

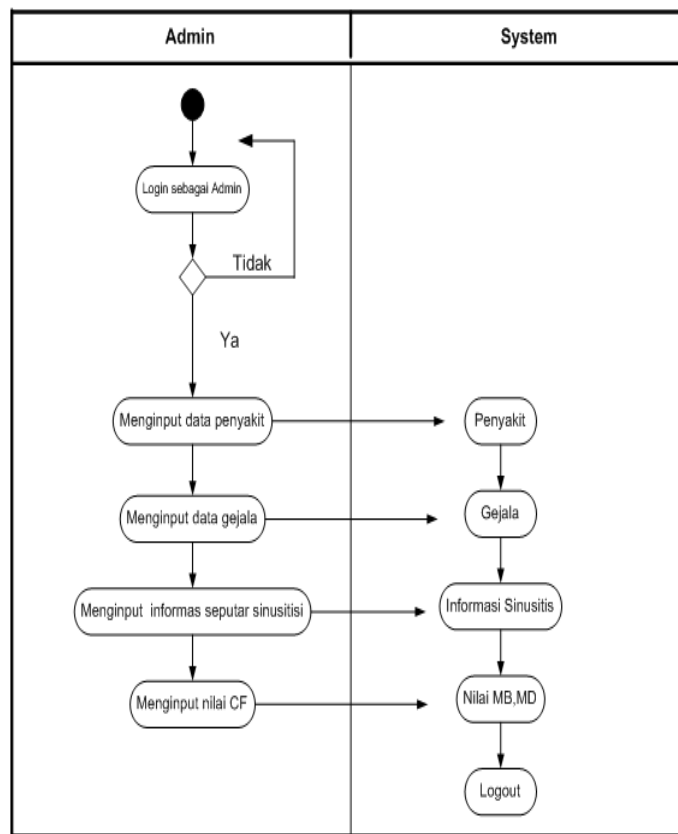


Gambar.4.3. Activity Diagram user / pasien

Gambar 4.3 Activity Diagram User / Pasien di atas adalah gambaran proses yang dilakukan user. Sistem mulai dari user melakukan akses web sistem pakar diagnosa penyakit sinusitis, pada halaman beranda uer akan langsung melihat informasi seputar penyakit sinusitis secara umum,lalu pada halaman beranda ini juga user akan diarahkan untuk mengklik button Register untuk melakukan pendaftaran, setelah itu user dapat memilih halaman diagnosa akan menampilkan halaman gejala penyakit sebagai halaman berkonsultasi dengan mencentang beberapa gejala sesuai yang dirasakan,

setelah itu user memilih gejala penyakit tersebut maka proses perhitungan CF dimulai dan akan menghasilkan persentase penyakit sinusitis yang dialami dan mendapatkan saran pengobatannya.

4. Activity Diagram Admin / Pakar

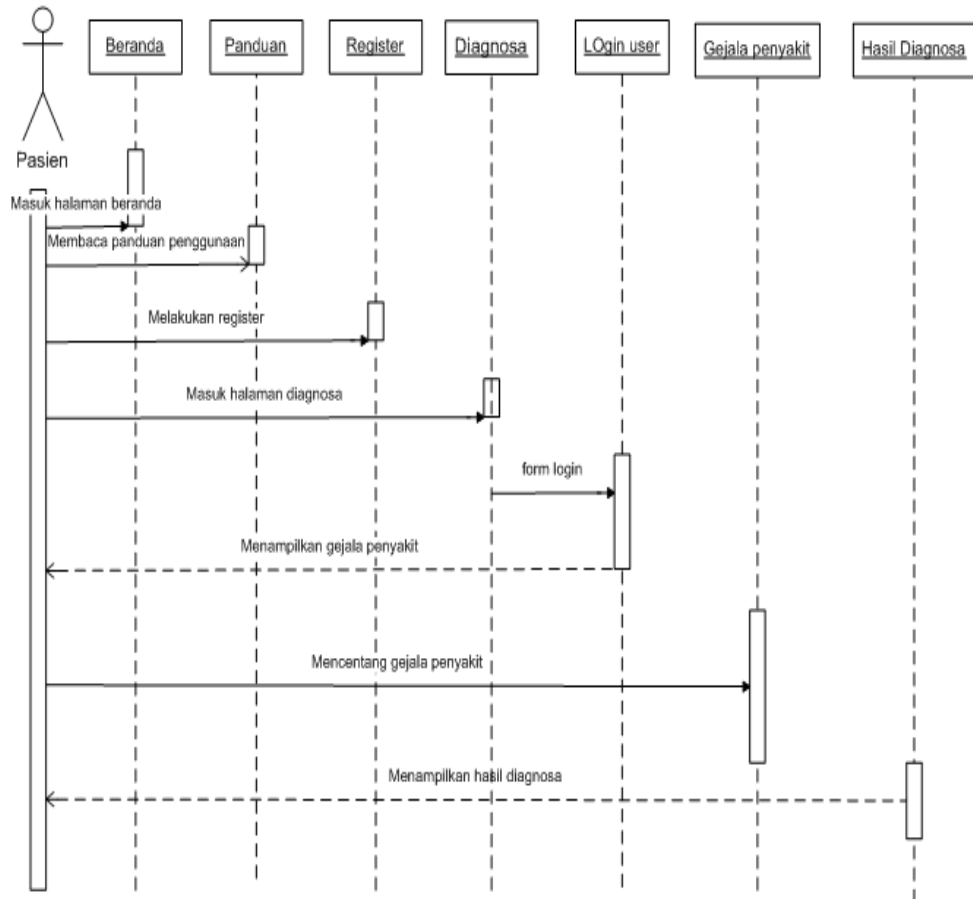


Gambar 4.4 Activity Diagram Admin

Gambar 4.4 Activity Diagram Admin di atas adalah gambaran proses yang dilakukan admin. Sistem mulai dari admin melakukan login, jika gagal sistem akan kembali ke halaman login dan jika berhasil admin dapat menginput data penyakit, data gejala, informasi penyakit dan

menginput nilai CF dimana admin memasukka nilai MB serta MD setelah admin logout.

5. Sequence Diagram User / Pasien

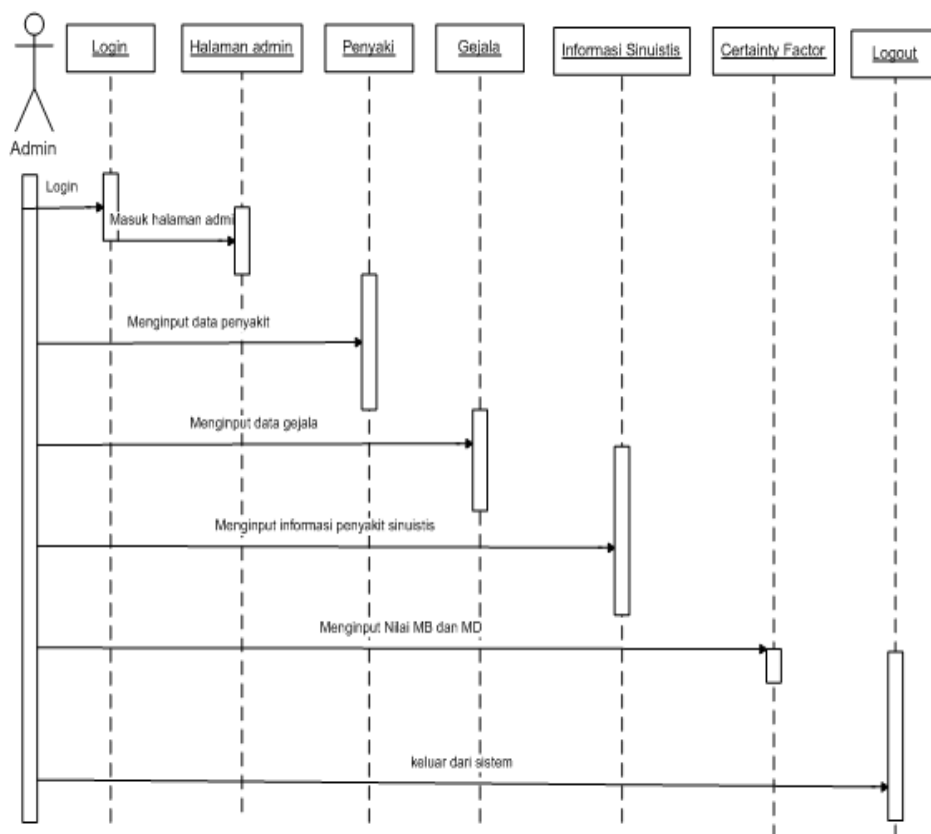


Gambar 4.5 Sequence diagram user / pasien

Gambar 4.5 Sequence diagram user / pasien. Pertama-tama pasien membuka web dan masuk ke halaman beranda dan melakukan register untuk terdaftar sebagai pasien. Setelah melakukan registrasi pasien dapat ke halaman diagnosa, halaman diagnosa ini terdapat halaman gejala penyakit dan pasien akan mencentang gejala panyakit yang mereka

rasakan, setelah pasien selesai memilih maka proses perhitungan Certainty factor dimulai selanjutnya akan menghasilkan hasil diagnosa berupa berapa besar persentase penyakit sinusitis yang diderita beserta saran pengobatannya. Pasien juga dapat melihat panduan cara penggunaan sistem pakar ini.

6. *Sequence diagram Admin*



Gambar 4.6 *Sequence diagram Admin*

Gambar 4.6 diatas adalah sequence diagram admin. Pertama-tama admin melakukan login. Maka sistem akan memvalidasi data admin. Didalam halaman utama admin ada menu master tabel, kemudian admin

menginput data penyakit, data gejala, data informasi sinusitis. Admin juga menginput nilai CF berupa nilai MB, MD untuk dapat direlasikan ke gejala penyakit sinusitis maka semua penginputan data tersimpan ke *database*, sehingga akan menghasilkan besar persentase penyakit sinusitis yang dialami pasien. Dan terakhir admin melakukan logout untuk keluar dari sistem.

B. Perancangan Database Tabel

4.1 Tabel Penyakit

No	Nama Field	Type	Ukuran	Keterangan
1	Id	Int	11	AUTO_INCREMENT
2	Kode	Varchar	5	-
3	Nama	Varchar	50	-
4	Keterangan	Vvarchar	255	-

4.2 Tabel Gejala

No	Nama Field	Type	Ukuran	Keterangan
1	Id	Int	11	AUTO_INCREMENT
2	Jenis_sinusitis-id	Int	11	-
3	Kode	Varchar	5	-
4	Keterangan	Varchar	255	-

4.3 Tabel gejala_penyakit

No	Nama field	Type	Ukuran	Keterangan
1	Id	Int	11	AUTO_INCREMENT
2	Gejala_id	Int	11	-
3	Penyakit_id	Int	11	-
4	Mb	Float	-	-
5	Md	Float	-	-

4.4 Tabel kelompok_gejala

No	Nama field	Type	Ukuran	Keterangan
1	Id	Int	11	AUTO_INCREMENT
2	Nama	Varchar	20	-
3	Keterangan	Varchar	2555	-

4.5 Tabel User

No	Nama field	Type	Ukuran	Keterangan
1	Id	Int	11	AUTO_INCREMENT
2	Username	Varchar	255	-
3	Password	Varchar	255	-
4	Nama	Varchar	2555	-

Tabel 4.6 Tabel Pengguna

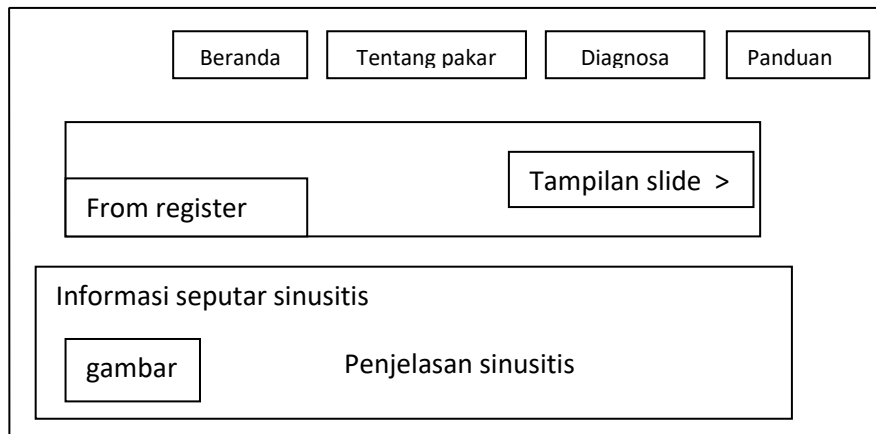
No	Nama field	Type	Ukuran	Keterangan
1	Id_pengguna	Int	11	AUTO_INCREMENT
2	Nm_pengguna	Varchar	100	-
3	Pass	Varchar	100	-
4	Nama	Varchar	120	-
5	Umur	Varchar	50	-
6	Jk	Enum	-	-

C. Perancangna Antar Muka

1. Rancangan *output*

a. Rancangan *output* beranda

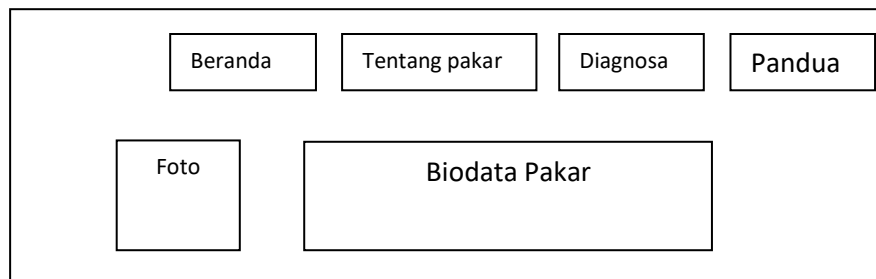
Halaman beranda adalah halaman awal yang pertama ditampilkan saat aplikasi dijalankan di *web browser*. Halaman beranda ini terdiri dari beberapa menu yaitu menu beranda, menu tentang pakar dan menu diagnosa. Pada halaman beranda ini juga terdapat form daftar untuk diagnosa bagi *user / pasien* yang belum terdaftar dan mendapatkan *username* dan *password*.



Gambar 4.7 rancangan *output* beranda

b. Rancangan *output* Tentang Pakar

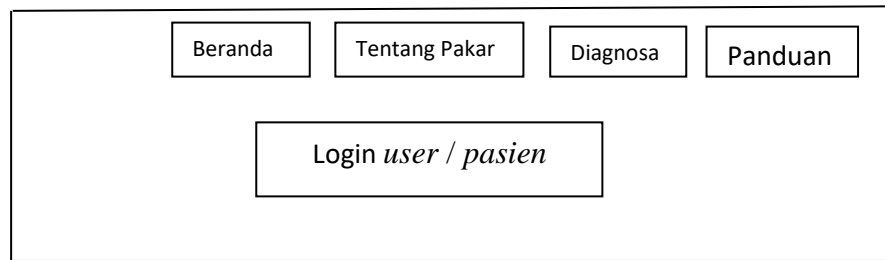
Halaman tentang pakar adalah halaman yang menjelaskan tentang biodata seorang pakar spesialis THT-KL, di halaman ini user juga dapat melihat kontak dari klinik yang dimiliki oleh dokter Spesialis THT-KL.



Gambar 4.8 rancangan *output* tentang pakar

c. Rancangan output Diagnosa

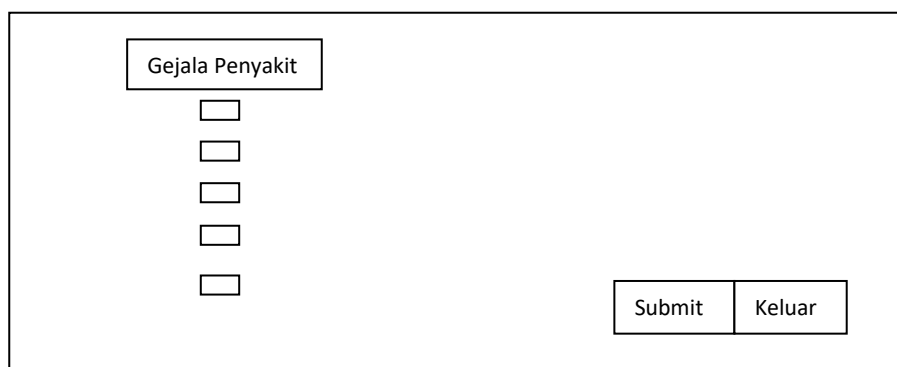
Halaman diagnosa adalah halaman yang digunakan untuk mendiagnosa penyakit sinusitis yang diderita. Pada halaman ini user / pasien melakukan login dengan memasukkan username dan password. Setelah itu user / pasien dapat melakukan diagnosanya dengan mencentang gejala-gejala yang dirasakan.



Gambar 4.9 rancangan *output* diagnosa

d. Rancang *output* Gejala Penyakit

Halaman gejala penyakit adalah halaman yang berisi tentang gejala-gejala penyakit sinusitis, dimana pada halamn ini user / pasien akan mencentang beberapa gejala penyakit yang dirasakan oleh user /



Gambar 4.10 Rancangan *output* gejala penyakit

e. Rancangan *output* Hasil Diagnosa

Rancangan halaman hasil diagnosa adalah laporan hasil analisis yang telah dilakukan oleh seorang user/pasien. Pada halaman ini user/pasien dapat melihat hasil analisis, yang berupa: gejala yang dipilih, hasil diagnosa, dan kesimpulan.

Hasil Analisi
Gajala yang dipilih
Hasil Diagnosa
Kesimpulan
<input type="button" value="Deteksi Ulang"/> <input type="button" value="Cetak"/>

Gambar 4.11 rancangan *output* hasil diagnosa

f. Rancangan *output* Daftar Pengunjung

Rancangan halaman Daftar Pengunjung adalah laporan hasil data registrasi yang telah dilakukan oleh seorang user/pasien. Pada halaman ini admin dapat melihat data daftar pengunjung dan dapat menghapus data pengunjung.

No	Nama	Umur	Jenis Kelamin	Alamat	Aksi
					Delete

Gambar 4.12 Rancangan *output* daftar pengunjung

2. Rancangan *Input Login*

a. Rancangan *Input Login*

Username
password
masuk

Gambar 4.13 Rancangan *input Login*

Halaman rancangan *Input Login* adalah halaman login untuk admin. Pada halaman ini admin melakukan login dengan memasukkan *username* dan *password*. Setelah itu admin dapat menginputkan, mengubah dan menyimpan data.

b. Rancangan *Input* data gejala

New

No	Nama	Keterangan	Aksi
			Edit Delete

Gambar 4.14 Rancangan *Input* data gejala

Halaman rancangan *Input* data gejala adalah halaman khusus untuk admin. Pada halaman ini admin menginput, mengubah dan menyimpan data gejala yang di dapatkan dari seorang pakar.

c. Rancangan *Input* data penyakit

New

No	Nama	Keterangan	Aksi
			Edit delete

Gambar 4.15 Rancangan *input* data penyakit

Halaman rancangan *Input* data penyakit adalah halaman khusus untuk admin. Pada halaman ini admin menginput, menambah, mengubah dan menyimpan data gejala yang di dapatkan dari seorang pakar.

d. Rancangan *input* data gejala penyakit

New					
No	Gejala	Penyakit	Mb	Md	Aksi
					Edit delete

Gambar 4.16 Rancangan *input* data gejala penyakit

e. Rancangan Input data kelompok gejala

New				
No	Kode	Keterangan	Kelompok penyakit	aksi
				Edit delete

Gambar 4. 17 Rancangan *input* data kelompok gejala

Halaman rancangan *Input* data jenis sinusitis adalah halaman khusus untuk admin. Pada halaman ini admin menginput, mengubah dan menyimpan data jenis sinusitis serta pengobatan dan penangana terhadap penyakit sinusitis, yang di dapatkan dari seorang pakar.

D. Implementasi

1. Implementasi *Forward Chaining*

Langkah-langkah yang dilakukan dalam implementasi sistem dengan proses runut maju (mesin inferensi) *Forward Chaining* adalah sebagai berikut :

- a. Menentukan premis-premis atau informasi masukan (*if*)
- b. Kemudian menuju konklusi atau driven information (*then*)

untuk keempat jenis penyakit sinusitis tersebut dapat dimodelkan sebagai berikut :

1) Sinusitis *Frontalis*

IF demam

AND hidung tersumbat

AND ingus terasa jatuh kebelakang

AND nyeri tekan pada dahi

AND nyeri diatas kedua mata

THEN Sinusitis Frontalis

2) Sinusitis *Etmoid*

IF demam

AND hidung tersumbat

AND ingus terasa jatuh kebelakang

AND nyeri di kantus medius

AND nyeri pada bagian sudut mata

AND nyeri di pangkal hidung

THEN sinusitis Etmoid

3) Sinusitis Sfenoid

IF demam

AND hidung tersumbat

AND ingus terasa jatuh kebelakang

AND nyeri di verteks / puncak kepala

AND nyeri dibelakang bola mata

THEN Sinusitis Sfenoid

4) Sinusitis Maksila

IF demam

AND hidung tersumbat

AND ingus terasa jatuh kebelakang

AND nyeri dibaawah kelopak mata

AND nyeri di gigi

AND nyeri pada rahang atas

AND nyeri tekan pada pipi

THEN sinusitis Maksila

2. Penyusunan Basis Pengetahuan

Tabel 4.7 Gejala penyakit Sinusitis

Kode	Keterangan
GS001	Nyeri dibawah kelopak mata
GS002	Terasa Nyeri di Gigi
GS003	Nyeri pada rahang atas
GS004	Nyeri tekan pada pipi
GS005	Nyeri tekan pada Dahi
GS006	Nyeri di atas kedua mata
GS007	Nyeri di Kantus Medius
GS008	Nyeri tekan pada Sudut mata
GS009	Nyeri di Pangkal Hidung
GS010	Nyeri di verteks / puncak kepala
GS011	Nyeri di belakang bola mata

GS012	Demam
GS013	Hidung tersumbat
GS014	Ingus terasa jatuh kebelakang
GS015	Nyeri tekan daerah sinus pada wajah

(Sumber: dr, Hj,Syahrijuita,M.Kes,Sp. THT-KL)

Tabel 4.8 Penyakit Sinusitis

Kode	Nama
PS001	Sinusitis Frontalis
PS002	Sinusitis Ethmoid
PS003	Sinusitis Sfenoid
PS004	Sinusitis Maksila
PS005	Sinusitis

(Sumber: dr, Hj,Syahrijuita,M.Kes,Sp. THT-KL)

3. Memberikan nilai CF untuk setiap gejala yang ada. Seorang pakar, dr.Syahrijuita,M.Kes,Sp.THT-KL memberikan nilai CF (Certainty Factor) Nilai kepastian pada setiap gejala penyakit sinusitis seperti pada Berikut ini :

Tabel 4. 9 Nilai CF yang telah diisi oleh pakar.

dr.Syahrijuita,M.Kes,Sp.THT-KL berdasarkan gejala yang sering di derita

No	Penyakit	Maksila	Frontalis	Ethmoid	Sfenoid	Sinusitis
	Gejala					
1	Nyeri tekan pada dahi	-	0.9	-	-	
2	Nyeri diatas kedua mata	-	0.8	-	-	
3	Nyeri pangkal hidung	-		0.7	-	
4	Nyeri dikantus medius	-	-	0.6	-	
5	Nyeri dibawah kelopak mata	0.7	-	-	-	
6	Nyeri gigi	0.6	-	-	-	
7	Nyeri rahang atas	0.5	-	-	-	
8	Nyeri tekan pada pipi	0.5	-	-	-	
9	Nyeri pada sudut mata		-	0.8	-	
10	Nyeri diverteks (puncak kepala)	-	-	-	0.9	
11	Nyeri di belakang bola mata	-	-	-	0.7	
12	Demam	0.6	0.5	0.6	0.7	0.6
13	Hidung tersumbat	0.7	0.7	0.7	0.7	0.7
14	Ingus jatuh ke belakang	0.6	0.5	0.4	0.6	0.6
15	Nyeri tekan daerah pada wajah	-	-	-	-	0.7
16	Lesu	0.6	0.5	0.6	0.6	0.5
17	Sakit kepala	-	-	-	-	0.5

4. Implementasi Algoritma Certainty Factor

Algoritma adalah prosedur langkah-langkah untuk perhitungan. Algoritma digunakan untuk perhitungan, pemrosesan data dan penalaran otomatis. Algoritma yang digunakan dalam sistem ini adalah *Certainty Factor (CF)*. Metode ini adalah metode yang sangat cocok digunakan pada sistem pakar, dimana metode ini digunakan untuk menyelesaikan masalah ketidakpastian data.

- a. Berikut implementasi metode *Certainty Factor* dalam bahasa pemrograman PHP

$$CF_{COMBINE}(CF1,CF2) = CF1 + CF2(1-CF1)$$

Implementasi :

```
// proses hitung
$listPenyakit=$this->gejala_penyakit_model->get_by_gejala($gejala);

$penyakit = [];

$i=0;

foreach($listPenyakit->result() as $value){

$listGejala=$this->gejala_penyakit_model-
>get_gejala_by_penyakit($value->penyakit_id,$gejala);

$combineCFmb=0;
$CFBeforemd=0;
$j=0;

foreach($listGejala->result() as $value2){

$j++;
if($j==1)
$combineCFmb=$value2->mb;
$combineCFmd=$value2->md;
else
```

```

$combineCFmb = $combineCFmb + ($value2->mb * (1 -
$combineCFmb));

$combineCFmd = $combineCFmd + ($value2->mb * (1 -
$combineCFmd));

$combinehasil = $combineCFmb - $combineCFmd;

}

if($combineCF >= 0.5)

{
$penyakit[$i] = array('kode' => $value->kode,
                    'nama' => $value->nama,
                    'kepercayaan' => $combineCF * 100,
                    'keterangan' => $value->keterangan);
    $i++;
}
}
function cmp($a, $b)
{
return ($a["kepercayaan"] > $b["kepercayaan"]) ? -1 : 1;
}
usort($penyakit, "cmp");
$data["listPenyakit"] = $penyakit;
$this->load->view("home/diagnosa/index", $data);
}
}

```

Implementasi lebih lengkap bisa dilihat pada lampiran berikut.

Berikut adalah hasil diagnosa penyakit sinusitis berdasarkan data gejala dengan nilai tingkat kepercayaan dan ketidakpercayaan dari seorang pakar.

Perhitungan manual metode *certainty factor* untuk penyakit sinusitis.

Tabel 4.10 Perhitungan manual metode Certainty Factor penyakit Sinusitis

Tabel gejala penyakit

No	Gejala	Penyakit	Md	Mb	CFmb	CFmd	CF= CFmb - CFmd	CF*100
1	Nyeri tekan pada pipi	Sinusitis Maksila	0,25	0,75			0,5	50
2	Nyeri diatas kedua mata	Sinusitis Frontalis	0,3	0,8				
3	Sakit kepala	Sinusitis	0,3	0,5				
4	Hidung tersumbat	Sinusitis	0,2	0,7	0,88	0,36	0,52	52
5	Nyeri tekan pada dahi	Sinusitis Frontalis	0,15	0,85				
6	Ingus tersa jatuh kebelakang (post nasal drip)	Sinusitis	0,3	0,6				
7	Demam	Sinusitis	0,2	0,6				
8	Lesu	Sinusitis	0,3	0,5				
9	Nyeri di pangkal hidung	Sinusitis Etmoid	0,15	0,75			0,6	60
10	Nyeri di bawah kelopak mata	Sinusitis Maksila	0,05	0,75				
11	Nyeri di gigi	Sinusitis Maksila	0,2	0,6				
12	Nyeri di bagian sudut mata	Sinusitis Etmoid	0,1	0,6				
13	Nyeri di kantung medius	Sinusitis Etmoid	0,15	0,85				
14	Nyeri di puncak kepala (verteks)	Sinusitis Sfenoid	0,05	0,95			0,9	90
15	Nyeri di belakang bola mata	Sinusitis Sfenoid	0,15	0,75				
16	Nyeri pada rahang atas	Sinusitis Maksila	0,3	0,5				
17	Nyeri dan rasa tekan pada wajah	Sinusitis	0,2	0,7				

5. Pembahasan Program Utama

a. Tampilan *Login*

Gambar 4.18 Tampilan *Login*

Gambar 4.17 adalah *form login* untuk admin. Proses pada *form login* dilakukan dengan menginputkan *username* dan *password* yang telah dimiliki seorang admin untuk mendapatkan hak akses sebagai admin.

b. Menu Data Gejala

No	Kode	Keterangan	Kelompok Penyakit	Aksi
1	05001	Nyeri telan pada pipi	Gejala Mayor	Edit Delete
2	05002	Nyeri diatas kedua mata	Gejala Mayor	Edit Delete
3	05003	Sakit kepala	Gejala Minor	Edit Delete
4	05004	Salutir memuntah	Gejala Mayor	Edit Delete
5	05006	Nyeri telan pada jidat	Gejala Mayor	Edit Delete
6	05007	Ingus jatuh ke belakang (post nasal drip)	Gejala Minor	Edit Delete
7	05008	Dansem	Gejala Mayor	Edit Delete
8	05009	Lemas	Gejala Minor	Edit Delete
9	05010	Nyeri di pangkal hidung	Gejala Mayor	Edit Delete
10	05011	Nyeri dibawah kelopak mata	Gejala Mayor	Edit Delete

Gambar 4.19 Tampil data gejala

Tampilan menu data gejala yang merupakan salah satu menu ditampilkan untuk admin. Menu ini hanya di akses atau digunakan oleh admin saja dimana admin menginputkan data gejala yang seorang pakar.

c. Menu Data Kelompok Gejala

No	Nama	Keterangan	Aksi
1	Gejala Mayor	-	Edit Delete
2	Gejala Minor	-	Edit Delete

Gambar 4.20 Tampilan data kelompok gejala

Tampilan menu data kelompok gejala yang merupakan salah satu menu ditampilkan utama untuk admin. Menu ini hanya di akses atau digunakan oleh admin saja dimana admin menginputkan data kelompok gejala yang seorang pakar.

d. Menu Data Penyakit

No	Kode	Nama	Keterangan	Aksi
1	FS001	Sinusitis Frontalis	Peranganan yang dapat anda lakukan : Konsumsi obat antibiotik golongan penisilin seperti amoksisilin selama 10-14 hari, konsumsi obat antiagregasi seperti parasetamol, pengompasan obat tetes hidung dan kompres hangat pada bagian nyeri serta istirahat yang cukup.	Edit Delete
2	FS002	Sinusitis Ethmoid	Peranganan yang dapat anda lakukan : Konsumsi obat antibiotik golongan penisilin seperti amoksisilin selama 10-14 hari, konsumsi obat antiagregasi seperti parasetamol, pengompasan obat tetes hidung dan kompres hangat pada bagian nyeri serta istirahat yang cukup.	Edit Delete
3	FS003	Sinusitis Sfenoid	Peranganan yang dapat anda lakukan : Konsumsi obat antibiotik golongan penisilin seperti amoksisilin selama 10-14 hari, konsumsi obat antiagregasi seperti parasetamol, pengompasan obat tetes hidung dan kompres hangat pada bagian nyeri serta istirahat yang cukup.	Edit Delete
4	FS004	Sinusitis Maksila	Peranganan yang dapat anda lakukan : Konsumsi obat antibiotik golongan penisilin seperti amoksisilin selama 10-14 hari, konsumsi obat antiagregasi seperti parasetamol, pengompasan obat tetes hidung dan kompres hangat pada bagian nyeri serta istirahat yang cukup.	Edit Delete
5	FS005	Sinusitis	Peranganan yang dapat anda lakukan adalah : Kompres hangat yang nyerit, Minum obat pereda nyeri seperti parasetamol, Menghirup uap panas dan beristirahat cukup.	Edit Delete

Gambar 4.21 Tampilan Data Penyakit

Tampilan menu data penyakit yang merupakan salah satu menu ditampilkan untuk admin. Menu ini hanya di akses atau digunakan oleh admin saja dimana admin menginputkan data penyakit yang seorang pakar.

e. Menu Data Gejala Penyakit

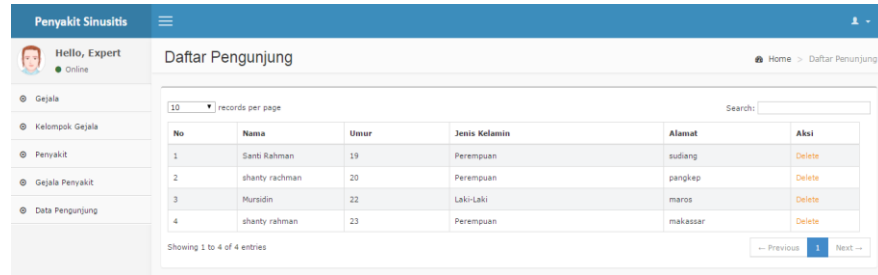
No	Gejala	Penyakit	MD	Mb	Aksi
1	Nyeri tekan pada gigi	Sinusitis Maksila	0.25	0.75	Edit Delete
2	Nyeri diatas kedua mata	Sinusitis Frontalis	0.3	0.6	Edit Delete
3	Sakit kepala	Sinusitis	0.3	0.5	Edit Delete
4	Hidung tersumbat	Sinusitis	0.2	0.7	Edit Delete
5	Nyeri tekan pada dahi	Sinusitis Frontalis	0.15	0.85	Edit Delete
6	Ingus jatuh ke belakang (post nasal drip)	Sinusitis	0.3	0.6	Edit Delete
7	Demam	Sinusitis	0.2	0.6	Edit Delete
8	Lemas	Sinusitis	0.3	0.5	Edit Delete
9	nyeri di pangkal hidung	Sinusitis Ethmoid	0.15	0.75	Edit Delete
10	Nyeri dibawah kelopak mata	Sinusitis Maksila	0.05	0.75	Edit Delete

Gambar 4.22 Tampilan data gejala penyakit

Tampilan menu data gejala penyakit yang merupakan salah satu menu ditampilkan untuk admin. Menu ini hanya di akses atau digunakan oleh admin saja, dan pada menu ini admin menginputkan nilai Mb dan Md yang telah diberikan oleh seorang pakar. Nilai Mb

dan Md adalah nilai tingkat kepercayaan dan ketidakpercayaan seorang pakar terhadap suatu gejala.

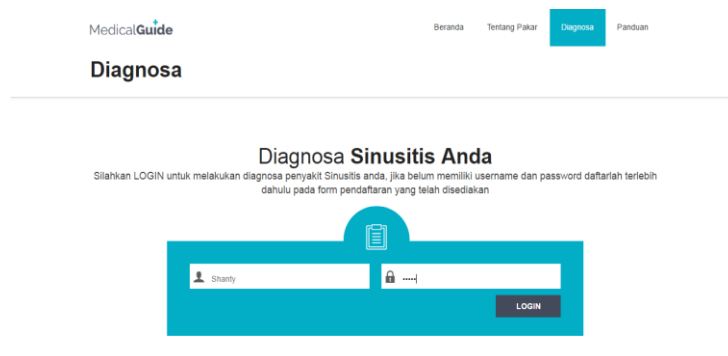
f. Menu Data Daftar Pengunjung



Gambar 4.23 Tampilan data daftar pengunjung

Tampilan menu data daftar pengunjung yang merupakan salah satu menu ditampilkan utama untuk admin. Menu ini hanya dapat diakses oleh admin saja dimana admin melihat dan menghapus data daftar pengunjung.

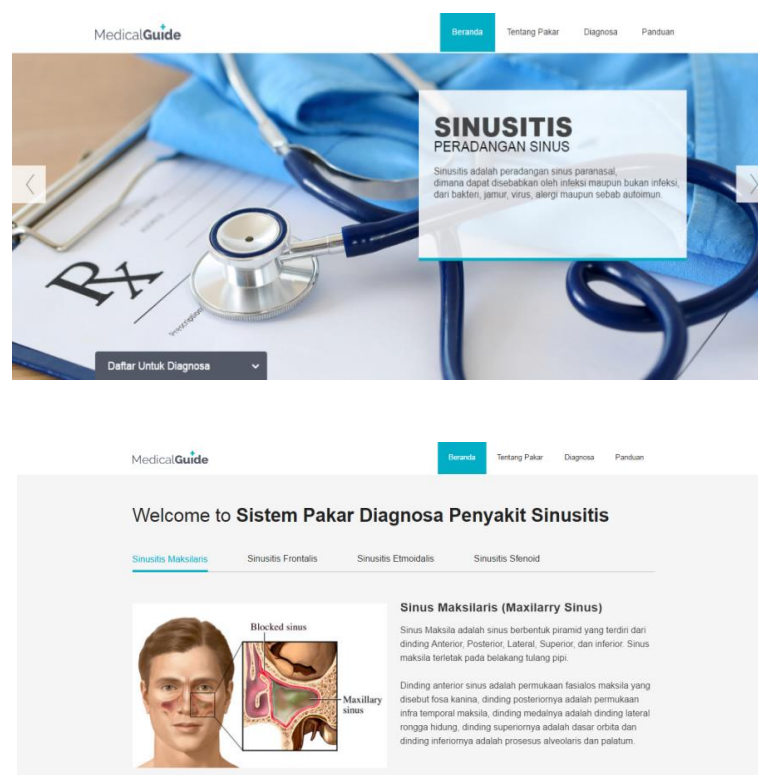
g. Tampilan Menu Login User



Gambar 4.24 Tampilan login user/pasien

Tampilan Menu login diatas untuk user/ pasien. Proses pada form login ini dilakukan dengan memasukkan username dan password .Setelah login untuk user/ pasien akan masuk ke halaman gejala penyakit.

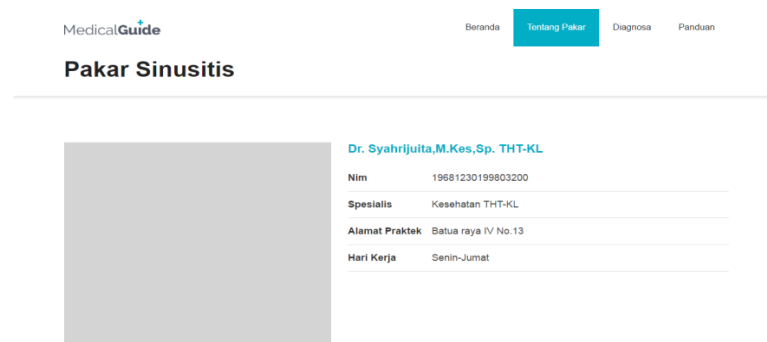
h. Tampilan Menu Beranda



Gambar 4.25 Tampilan menu beranda

Tampilan menu beranda adalah tampilan awal yang pertama ditampilkan saat aplikasi di jalankan di web browser. Halaman beranda terdiri dari beberapa menu yaitu : beranda, tentang pakar, diagnosa dan halamn panduan. Dan halaman ini berisikan tentang informasi seputar penyakit sinusitis.

i. Tampilan Menu Tentang Pakar



Gambar 4.26 Tampilan menu tentang pakar

Halaman menu tentang pakar adalah halaman yang berisi tentang informasi data diri pakar atau dokter spesialis THT-KL

j. Tampilan Menu Diagnosa



Gambar 4.27 Tampilan menu diagnosa

Halaman menu diagnosa adalah halaman yang berisi form login user / pasien sebelum masuk ke halaman gejala penyakit.

k. Tampilan Menu Diagnosa setelah Login



Gambar 4.28 Tampilan gejala penyakit

Halaman menu diagnosa yang diatas adalah menu diagnosa setelah user/pasien melakukan login, pada halaman ini user/pasien diminta untuk melakukan diagnosa dengan memilih atau memberi centang/ceklis pada gejala yang diasakan.

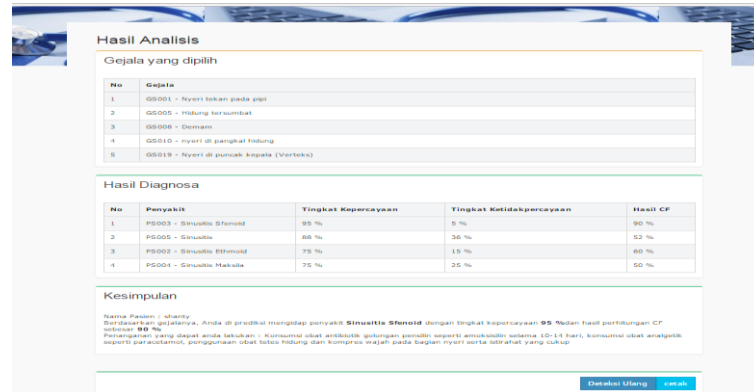
1. Tampilan Menu Panduan



Gambar 4.29 Tampilan menu panduan

Halaman menu panduan adalah halaman yang menjelaskan bagaimana cara penggunaan aplikasi, di halaman ini user/pasien diberitahukan bagaimana cara untuk menggunakan aplikasi mulai dari daftar sampai nantinya mendapatkan hasil diagnosa.

m. Tampilan Hasil Analisis Diagnosa



Gambar 4.30 Tampilan menu hasil analisis diagnosa

Halaman menu hasil analisis adalah halaman yang berupa tampilan atau hasil diagnosa dari pemilihan gejala. Pada hasil analisis ini berisi : gejala yang dipilih, hasil diagnosa yang berupa persentase besar penyakit sinusitis yang diderita dan jenis sinusitis apa yang diderita serta kesimpulan yang berupa penjelasan singkat tentang penyakit yang diderita dan saran serta pengobatan.

n. Tampilan Laporan hasil analisis diagnosa

8/24/2017

Hasil Analisis

Gejala yang dipilih

No	Gejala
1	GS001 - Nyeri tekan pada pipi
2	GS005 - Hidung tersumbat
3	GS008 - Demam
4	GS010 - nyeri di pangkal hidung
5	GS019 - Nyeri di puncak kepala (Verteks)

Hasil Diagnosa

No	Penyakit	Tingkat Kepercayaan	Tingkat Ketidakpercayaan	Hasil CF
1	PS003 - Sinusitis Sfenoid	95 %	5 %	90 %
2	PS005 - Sinusitis	88 %	36 %	52 %
3	PS002 - Sinusitis Ethmoid	75 %	15 %	60 %
4	PS004 - Sinusitis Maksila	75 %	25 %	50 %

Kesimpulan

Nama Pasien : shanty
 Berdasarkan gejalanya, Anda di prediksi mengidap penyakit **Sinusitis Sfenoid** dengan tingkat kepercayaan **95 %** dan hasil perhitungan CF sebesar **90 %**
 Penanganan yang dapat anda lakukan : Konsumsi obat antibiotik golongan pensilin seperti amoksisilin selama 10-14 hari, konsumsi obat analgetik seperti paracetamol, penggunaan obat tetes hidung dan kompres wajah pada bagian nyeri serta istirahat yang cukup

Gambar 4.31 Tampilan menu alporan hasil analisis diagnosa

Halaman menu laporan hasil analisis adalah halaman yang berupa tampilan atau hasil diagnosa dari pemilihan gejala.

E. Pengujian Sistem

Pengujian sistem dilakukan untuk memeriksa kekompakan atau kinerja antara komponen sistem yang diimplementasikan. Tujuan utama dari pengujian *system* adalah untuk memastikan bahwa elemen-elemen atau komponen-komponen dari *system* yang telah dirancang dapat berfungsi sesuai dengan yang diharapkan.

Metode pengujian yang digunakan adalah metode pengujian *Black Box* dan *White Box*.


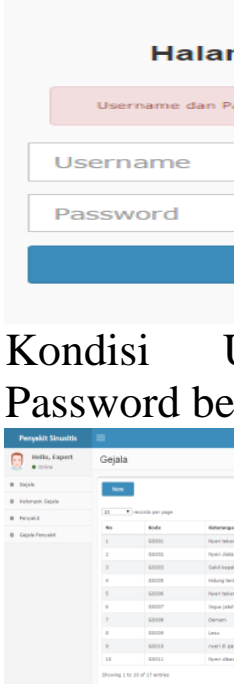

1. Pengujian *Black Box*

Pengujian *Black Box* merupakan metode perancangan data uji yang didasarkan pada spesifikasi perangkat lunak, Data yang diuji dibangkitkan, dieksekusi pada perangkat lunak dan kemudian keluaran dari perangkat lunak dicek apakah sudah sesuai dengan yang diharapkan atau tidak.

Pengujian yang dilakukan terhadap aplikasi dengan cara mengecek setiap fungsi bagian dari program yang ada dengan menggunakan tabel pengujian, apakah program berjalan sesuai dengan yang diharapkan atau tidak.

a. Pengujian *black box* Admin *expert* sistem

Tabel 4.11 Pengujian *black box* Admin *expert* sistem

No	Skenario Pengujian	Test case	Hasil yang diharapkan	Hasil
1.	Input username admin expert yang kondisi salah dan kondisi benar di <i>from login</i>		Sistem akan mengembalikan Admin expert sistem ke From login jika Username dan Password Admin salah dan jika benar sistem mengarahkan ke menu admin	
2.	Tambah data gejala		Sistem akan	From tamb

			<p>di form telah diisi dengan benar,selanjutka n klik button save dan data gejala akan tersimpan. Tapi jika data belum terisi semua maka sistem akan menampilkan pesan error.</p>
3.	Edit data gejala	Edit	<p>Sistem akan menampilkan form edit gejala jika admin klik Edit pada tabel gejala.</p>
4.	Delete data gejala	Delete	<p>Sistem akan menampilkan form delete</p>

Pesan error a from tamba diisi semua.

Gejala Form

▲ Kolom keterangan Wajib Disi.

Kode
GS000

Keterangan
Masukkan keterangan

Kelompok Penyakit
Gejala Mayor

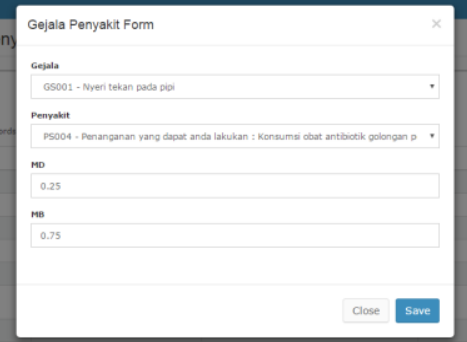
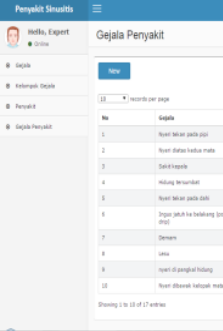
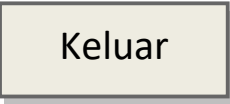

Gejala Form

Kode
GS001

Keterangan
Nyeri tekan pada pipi



Kelompok Penyakit
Gejala Mayor

Apakah data ini akan dihapus?

5.	Input gejala, penyakit dan nilai MB,MD dari pakar.		Sistem akan menampilkan form gejala penyakit yang berisi gejala,penyakit, Md dan Mb	
6.	Log Out Admin		Sistem akan menampilkan halaman login jika admin klik Keluar.	

b. Pengujian *black box* user / pasien

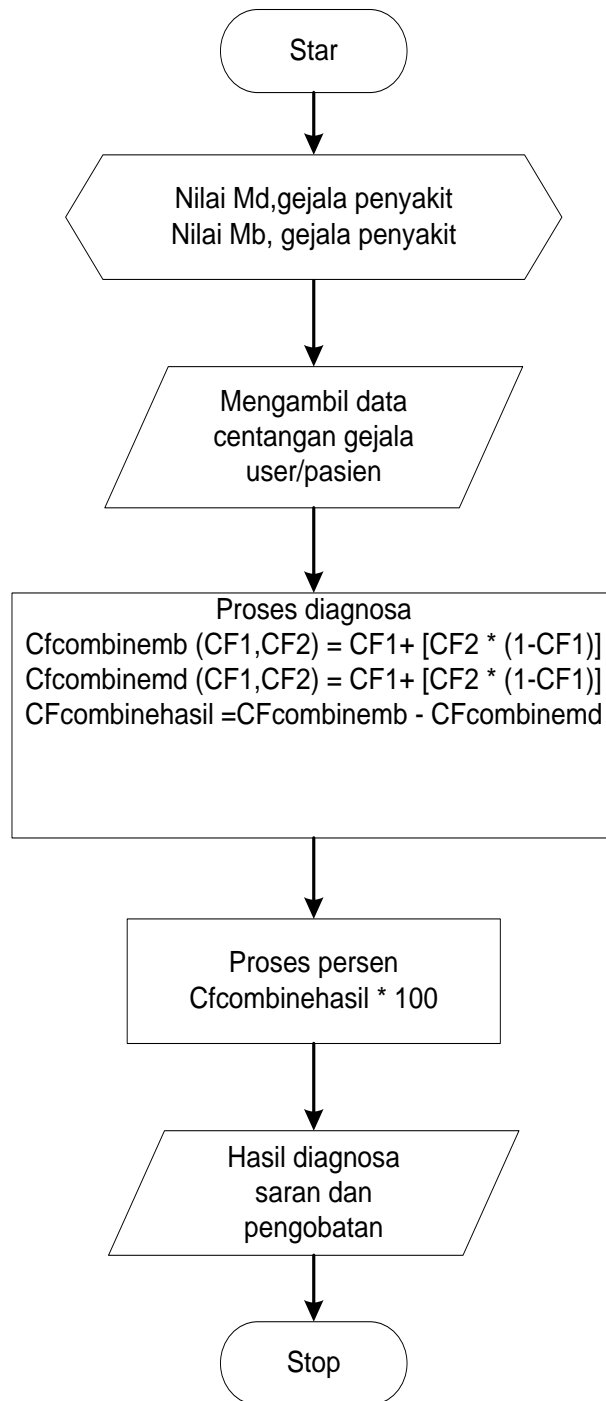
Tabel 4.12 Pengujian *black box* User / pasien

No	Skenario pengujian	Test case	Hasil yang diharapkan	Hasil pengujian
1.	Input username user Input password user Input umur user Input jenis kelamin user Input nama lengkap user		Sistem akan menampilkan form daftarnya pada beranda. Klik tanda "v" untuk melihat form daftarnya dan mengisi semua data. user yang sudah	

Berdasarkan tabel-tabel hasil pengujian *black box* diatas maka dapat diambil kesimpulan bahwa semua form yang digunakan dinyatakan *valid* atau diterima dan sesuai dengan yang diinginkan.

2. Pengujian *White box*

Pengujian *White box* merupakan pengujian yang didasarkan pada pengecekan terhadap detail perancangan, menggunakan struktur kontrol dari desain program secara prosedur. Untuk memebagi pengujian ke dalam beberapa kasus sekilas dapat diambil kesimpulan *White box testing* merupakan petunjuk untuk mendapatkan program yang benar.



Gambar 4.32 Flowchart diagram pengujian *White box Certainty factor*

```

$listPenyakit = $this->gejala_penyakit_model->get_by_gejala($gejala);

$penyakit = [];

$i=0;

foreach($listPenyakit->result() as $value){

$listGejala = $this->gejala_penyakit_model->get_gejala_by_penyakit($value->penyakit_id,$gejala);

$combineCFmb=0;

$combineCFmd=0;

$CFBefore=0;

$j=0;

foreach($listGejala->result() as $value2){

$j++;

if($j==3){

$combineCFmb=$value2->mb;

$combineCFmd=$value2->md;

}

else

$combineCFmb =$combineCFmb + ($value2->mb * (1 - $combineCFmb));

$combineCFmd =$combineCFmd + ($value2->md * (1 - $combineCFmd));

$combinehasil=$combineCFmb-$combineCFmd;

}

if($combinehasil);

{

$penyakit[$i]=array('kode'=>$value->kode,

'nama'=>$value->nama,

```

```
'kepercayaan'=>$combineCFmb*100,
```

```
'hasilcf'=>$combinehasil*100,
```

```
'tidakpercaya'=>$combineCFmd*100,
```

```
'keterangan'=>$value->keterangan);
```

```
$i++;
```

```
}
```

```
}
```

```
function cmp($a, $b)
```

```
{
```

```
return ($a["kepercayaan"] > $b["kepercayaan"]) ? -1 : 1;
```

```
}
```

```
usort($penyakit, "cmp");
```

```
$data["listPenyakit"] = $penyakit;
```

```
$this->load->view("home/diagnosa/index", $data);
```

```
}
```

```
}
```

```
<?php
```

```
if(sizeof($listPenyakit)>0) { ?> <p>
```

```
Nama Pasien : <?php echo $this->session->userdata('nm_pengguna');  
?><br>
```

```
Berdasarkan gejalanya, Anda di prediksi mengidap penyakit <b><?php  
echo $listPenyakit[0]['nama'];?></b> dengan tingkat kepercayaan  
<b><?php echo $listPenyakit[0]['kepercayaan'];?> %</b> dan hasil  
perhitungan CF sebesar <b><?php echo $listPenyakit[0]['hasilcf'];?>  
%</b><br/>
```

```
<?php echo $listPenyakit[0]['keterangan'];?>
```

</p>

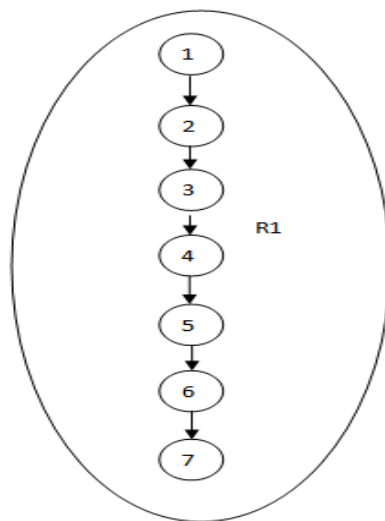
<?php }else{?>

<p>

Penyakit tidak dapat diprediksi karena tingkat kepercayaan gejala terlalu rendah

</p>

<?php }?>



Gambar 4.33 Flowgraph

Menentukan kompleksitas *cyclomatic* dari grafik aliran.

$$P = 0$$

$$E = 6$$

$$N = 7$$

$$\text{Region} = 1$$

$$V(G) = E - N + 2$$

$$= 6 - 7 + 2$$

$$= 1$$

$$V(G) = P + 1$$

$$= 0 + 1$$

$$= 1$$

Keterangan :

E : Jumlah busur atau link

N : Jumlah Simpul

1. Step 4 : Menentukan dasa jalur independen

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

BAB V

KESIMPULAN

A. Kesimpulan

Kesimpulan dari penelitian ini adalah :

1. Sistem pakar yang dibuat dapat memberikan kemudahan kepada *user / pasien* dalam mendiagnosa awal penyakit sinusitis.
2. Aplikasi sistem pakar diagnosa penyakit sinusitis menggunakan metode *certainty factor* berbasis web ini, dapat bekerja sesuai dengan yang diharapkan. Sistem dapat mendiagnosa awal dan memberikan informasi tentang saran-saran pengobatan dari penyakit sinusitis.
3. Aplikasi sistem pakar diagnosa awal penyakit sinusitis ini di buat berbasis web dengan tampilan yang sederhana, sehingga *user / pasien* tidak mengalami kesulitan dalam menggunakan aplikasi.

B. Saran

Saran yang dapat diberikan setelah penelitian ini dilakukan adalah :

1. Sistem dapat dikembangkan ke bahasa pemrograman yang berbasis mobile.
2. Dalam pengembangannya penyakit yang dibahas tidak hanya penyakit sinusitis.
3. Sistem yang dibuat kiranya memiliki *user interface* yang jauh lebih baik.
4. Pengembangan sistem selanjutnya, disarankan dalam pemberian pembobotan sebaiknya dilakukan lebih dari satu pakar.

5. Kedepannya aplikasi sistem pakar ini dapat ditambahkan fitur yang menyediakan informasi rumas sakit atau tempat-tempat praktek ataupun klinik yang memiliki poli THT-KL terdekat, sehingga penggunaan diberikan kemudahan dalam pencarian dokter THT-KL yang menjadi rujukan aplikasi.

DAFTAR PUSTAKA

- Achmad Balza. (2006). "Diktat mata kuliah Kecerdasan buatan". Jurnal Diktat kecerdasan buatan Yogyakarta.
- Andri,Kristanto (2008:1), Perancangan Sistem Informasi dan Aplikasinya, Gava Media, Yogyakarta
- Ayuliana (2009). Testing dan Implementasi (online). Tersedia: rifiana.staff.gunadarma.ac.id/.../Teknik+Pengujian+perangkat+Lunak+-+White+Box.p. [13 Maret 2017]
- Dahria,Muhammad. (2011). "Pengembangan Sistem Pakar dalam membangun suatu Aplikasi" Jurnal SAINTIKOM vol.10/No.3/2011 : 199-200
- Dianagita. (2016).Konsep sistem pakar, (online) tersedia :
<https://dianagita.wordpress.com/2016/11/06/konsep-sistem-pakar/>
(diakses 07 januari 2017)
- Gelena. (2011). Perbedaan antara sinusitis dan deviated septum dan bagaimana cara mengobatinya deviated septum selain dengan melakukan operasi, (online) tersedia :
<https://www.galena.co.id/q/apa-perbedaan-antara-sinusitis-dan-deviated-septum-bagaimana-cara-mengobati-deviated-septum-selain-dengan-melakukan-operasi>, (diakses 11 februari 2017).
- Hartono Arief, & Iskandar Nurbaiti. (2010). Sinus Paranasal, dalam Buku Ajar Ilmu Kesehatan Tenggorokan,Hidung dan Telinga, edisi 12. Jakarta. Buku Kedokteran EGC.
- Hardianto,wahyu. Jusak.Sulistiowati. (2016). Rancang Bangun Aplikasi Sistem Pakar Diagnosa Penyakit THT Berbasis Web dengan menggunakan Metode *Certainty factor*. Jurnal JSIKA vol.5, No. 4.tahun 2016,ISSN 2338-137X.
- Hermawan. 2004. Unified Modelling Language (UML). Andi. Yogyakarta.
- Jovienana. (2013).Sistem pakar, (online) tersedia :
<http://www.slideshare.net/jovienana/sistem-pakar-12645520> (diakses 24 januari 2017)
- Jogiyanto, H., M., 2005, Analisis dan Desain Sistem Informasi, Andi Offset,Yogyakarta.
- Kadir,A.(2010).Mudah Mempelajari Database MySQL. Yogyakarta:Andi.

- KBBI . (2016). Pengertian Diagnosis dan mendiagnosais menurut Kamus Besar Bahasa Indonesia (KBBI). (online). Tersedia : <http://kbbi.web.id/diagnosis>. [27 Januari 2016].
- Kurniawan, Indra. (2012).” Rancang Bangun Sistem Pakar dalam Diagnosa Penyakit Telinga, Hidung dan Tenggorokan dengan menggunakan metode *Cretainty Factor*”.skripsi,Fakultas Sains dan Teknologi,Universitas Airlangga. Surabaya.
- Kusumadewi, Sri. (2003). *Artifical Intelligence* (Teknik dan aplikasinya). Yogyakarta:Graha ilmu
- Martiana, Entin. (2012). “ *Expert System* (Baru)”. Modul Sistem Pakar . Surabaya
- Mulyanto E., Sutojo T. & Suhartono. 2011. Kecerdasan Buatan. Andi. Yogyakarta.
- Munawar.(2005). “Pemodelan Visual dan UML.Yogjakarta:Graha Ilmu
- MedKes. (2014).Gejala, penyebab dan pengobatan sinusitis, (online) tersedia : <https://www.medkes.com/2014/06/gejala-penyebab-dan-pengobatan-sinusitis.html?m=1>. (diakses 07 Agustus 2017)
- Novita,Rahmat,danBeta. (2012). “Penerapan Model Certainty factor untuk mendeteksi gejala kanker Mulut Rahim”. Jurnal Sistem Informasi Bisnis. Semarang
- Nugroho. (2010). Rekayasa Perangkat Lunak Menggunakan UML dan Java. Andi Yogyakarta.
- Nugroho, Adi.(2004). Konsep Pengembangan Sistem Basis Data. Penerbit Informatika. Bandung.
- Nurul,Yustina,dan Didik. (2012). “Penerapan Metode Certainty factor dalam mendiagnosa dini penyakit Telinga Hidung dan Tenggorokan”. Jurnal Ilmiah SINUS. Semarang
- Prabowo Pudjo, Widodo dan Herlawati. (2011). Menggunakan UML. Bandung:InformatikaBandung
- Repository. (2011). “ Bab 2 Tinjauan Pustaka Anatomi Sinus Paranasal”. (online) tersedia : <http://repository.usu.ac.id/bitstream/123456789/47108/5/ChapterII/20I.pdf> (24 januari 2017)
- Repository. (2011). “ Bab 2 Sinusitis”. (online) tersedia :

- <http://repository.usu.ac.id/bitstream/123456789/47108/5/Chapter%20I.pdf>
(24 januari 2017)
- Repository. (2011). “ Bab 2 Sistem Pakar”. (online) tersedia :
<http://repository.usu.ac.id/bitstream/123456789/27985/4/Chapter%20I.pdf>
(24 januari 2017)
- Repository. (2016). “ BAB 11 LANDASAN TEORI”. (online) tersedia :
http://repository.potensi_utama.ac.id/bitstream/123456789/27985/4/Chapter%20II.pdf (24 januari 2017)
- Sambuda,Abi. (2008).” Korelasi antara Rhinitis dengan sinusitis pada pemeriksaan sinus paranasalis di instansi Radiologi RSUD DR.Moewardi Surakarta”.skripsi,fakultas kedokteran,Universitas sebelas Maret Surakarta.
- Sidik, B. dan Husni, I.P. (2012). *Pemrograman Web dengan HTML*. Bandung: Informatika.
- Siswanto. (2010). *Kercadanan Tiruan*. Yogyakarta:Graha Ilmu
- Supriyanto,A.(2007). *WEB dengan HTML dan XML*. Yogyakarta:Grahallmu
- Susanto, Azhar.(2004). *Sistem Informasi Manajemen Konsep dan pengembangannya*. Bandung:Lingga Jaya
- Turnip Mardi. (2015). “sistem pakar diagnosa penyakit THT menggunakan metode backward Chining” *Riau journal of computer Science* vol.1/No.1/2015 : 1-8| 1
- Verina Wiwi. (2015). “Penerapan metode forward chaining untuk medeteksi penyakit THT” *Jurnal Jatini*,vol 1 No.2 Universitas Potensi utama Yogyakarta.
- Widiyanto, A.B. 2015, *Sistem Pakar untuk Mengidentifikasi Jenis Kayu dengan Metode Certainty Factor dan Forward Chaining*. Skripsi: Universitas SebelasMaret.Surakarta.

L

A

M

P

I

R

A

N

LAMPIRAN SOURCE CODE

PERANCANGAN SISTEM PAKAR DIAGNOSA AWAL PENYAKIT SINUISTIS MENGGUNAKAN METODE *CERTAINTY FACTOR* BERBASIS WEB

1. Source Code Registrasi Pasien / User

```
<div class="make-appointment">
  <div class="container">
    <ul id="accordion" class="accordion">
      <li>
        <div class="link"><i class="fa fa-database"></i><span class="appointment-
title">Daftar Untuk Diagnosa</span> <i class="icon-chevron-down"></i> </div>
        <section class="bgcolor-3">
          <p class="error" id="error" style="display:none;"></p>
          <p class="success" id="success" style="display:none;"></p>
          <!-- <form name="appointment_form" id="appointment_form"
method="post" action="#" onSubmit="return false">
-->          <?php
                echo validation_errors();
                echo form_open("dashboard/add");?>
                <span class="input input--kohana">
                  <input class="input__field input__field--kohana"
type="text" id="input-29" name="nm_pengguna" required="" />
                  <label class="input__label input__label--kohana"
for="input-29">
                    <i class="icon-add-user icon icon--kohana"></i>
                    <span class="input__label-content input__label-content--
kohana">Username</span>
                </label>
                </span>
```

```

        <span class="input input--kohana">
            <input class="input__field input__field--
kohana" type="password" id="input-29" name="pass" required=""/>
            <label class="input__label input__label--kohana"
for="input-30">
                <i class="icon-locked icon icon--kohana"></i>
                <span class="input__label-content input__label-content--
kohana">Password</span>
            </label>
        </span>

        <span class="input input--kohana last">
            <input class="input__field input__field--kohana" type="number"
id="input-31" name="umur" required=""/>
            <label class="input__label input__label--kohana" for="input-31">
                <i class="icon-ecg-line icon icon--kohana"></i>
                <span class="input__label-content input__label-content--
kohana">Umur</span>
            </label>
        </span>

        <span class="input input--kohana">
            <select class="input__field input__field--kohana" type="text"
id="input-31" name="jk" style="padding-left: 110px;" required="">
                <option style="color: black;"></option>
                <option style="color: black;">Laki-Laki</option>
                <option style="color: black;">Perempuan</option>
            </select>
            <label class="input__label input__label--kohana"
style="padding-left:0px;">
                <span class="input__label-content ">Jenis Kelamin :</span>
            </label>

```

```
        </span>
        <span class="input input--kohana">
            <input class="input__field input__field--kohana" type="text"
id="input-44" name="nama" required=""/>
            <label class="input__label input__label--kohana" for="input-30">
                <i class="icon-locked icon icon--kohana"></i>
                <span class="input__label-content input__label-content--
kohana">Nama Lengkap</span>
            </label>
        </span>
        <span class="input input--kohana last">
            <input class="input__field input__field--kohana" type="text"
id="input-45" name="alamat" required=""/>
            <label class="input__label input__label--kohana" for="input-31">
                <i class="icon-ecg-line icon icon--kohana"></i>
                <span class="input__label-content input__label-content--
kohana">Alamat</span>
            </label>
        </span>
        <input name="submit" type="submit" value="KIRIM" >
        <?php echo form_close();?>
    </section>
</li>
</ul>
</div>
</div>
```

2. Source Code *Login Pasien / User*

```
<?php
    echo form_open('diagnosa/login');
?>
    <?php if($this->session->flashdata('success')){ ?>
        <div class="alert alert-success">
            <button type="button" class="close" data-dismiss="alert">x</button>
        <?php echo $this->session->flashdata('success'); ?>
        </div>
    <?php }else if($this->session->flashdata('error')){ ?>
        <div class="alert alert-danger">
            <button type="button" class="close" data-dismiss="alert">x</button>
        <?php echo $this->session->flashdata('error'); ?>
        </div>
    <?php }?>
        <span class="input input--kohana">
            <input class="input__field input__field--kohana" type="text"
id="input-29" name="nm_pengguna" />
            <label class="input__label input__label--kohana" for="input-29">
                <i class="icon-user6 icon icon--kohana"></i>
                <span class="input__label-content input__label-content--
kohana">Username</span>
            </label>
        </span>
        <span class="input input--kohana last">
            <input style="color: black;" class="input__field input__field--
kohana" type="password" id="input-31" name="pass"/>
            <label class="input__label input__label--kohana" for="input-31">
```

```

        <i class="icon-locked icon icon--kohana"></i>
        <span class="input__label-content input__label-content--kohana">Password</span>
    </label>
</span>
    <input name="submit" type="submit" value="Login">
<?php echo form_close();?>

```

3.Source Code *Login Admin*

```

<div class="content" style="padding:10px 40px;">
    <form method="post" action="<?php echo base_url('login/submit')?>">
        <div style="padding: 0 30%; margin-top: 100px">
            <div class="text-center">
                <h3><strong>Halaman Login</strong></h3>
            </div>
            <br />
            <?php if($this->session->flashdata('success')){ ?>
                <div class="alert alert-success">
                    <button type="button" class="close" data-dismiss="alert">x</button>
                    <?php echo $this->session->flashdata('success'); ?>
                </div>
            <?php }else if($this->session->flashdata('error')){ ?>
                <div class="alert alert-danger">
                    <button type="button" class="close" data-dismiss="alert">x</button>
                    <?php echo $this->session->flashdata('error'); ?>
                </div>
            <?php }?>

```

```

    <div class="form-group">
        <input type="text" name="username" id="username_id" class="form-
control input-lg" placeholder="Username" required>
    </div> <!-- / Username -->

    <div class="form-group">
        <input type="password" name="password" id="password_id"
class="form-control input-lg" placeholder="Password" required>
    </div> <!-- / Password -->

    <div class="form-actions">
        <input type="submit" value="Masuk" class="btn btn-primary btn-block
btn-lg">
    </div> <!-- / .form-actions -->
</div>
</form>
</div>

```

4. Sourec Code Edit Gejala

```

<div class="modal-dialog" >
    <div class="modal-content">
        <div class="modal-header">
            <button type="button" class="close" data-
dismiss="modal" aria-hidden="true">&times;</button>
            <h4 class="modal-title">Gejala Form</h4>
        </div>
        <div class="modal-body">
            <div class="box-body table-responsive">
                <span id="form-msg">

```

```

        </span>

        <?php echo
form_open(base_url()."gejala/".$post,'id="formedit"?>
        <div class="box-body">
            <div class="row">
                <div class="col-md-12">
                    <input type="hidden" id="id" name="id" value="<?php
echo($row['id']); ?>" />
                    <div class="form-group">
                        <label>Kode</label>
                        <input type="text" class="form-control" id="kode"
name="kode" value="<?php echo($row['kode']); ?>" placeholder="Masukkan
kode" />
                    </div>
                    <div class="form-group">
                        <label>Keterangan</label>
                        <input type="text" class="form-control"
id="keterangan" name="keterangan" value="<?php echo($row['keterangan']); ?>"
placeholder="Masukkan keterangan" />
                    </div>
                    <div class="form-group">
                        <label>Kelompok Penyakit</label>
                        <select id="jenis_sinusitis_id" class="form-control"
name="jenis_sinusitis_id" required>
                            <option></option>
                            <?php foreach($listJenisSinusitis->result() as $value){?>
                                <option value="<?php echo $value->id ?>" <?php echo ($value-
>id == $row['jenis_sinusitis_id']) ? 'selected:':"; ?>><?php echo $value->nama
?></option>

```

```

<?php } ?>

</select>

        </div>

    </div>

    <div class="col-md-6">

    </div>

</div>

</div>

<?php echo form_close(); ?>

        </div><!-- /.box-body -->

</div>

<div class="modal-footer">

        <button type="button" class="btn btn-default" data-
dismiss="modal">Close</button>

        <button id="btn-save" type="button" class="btn btn-
primary">Save</button>

    </div>

</div>

</div>

```

5. Source Code Delete Gejala

```

function delete_gejala(id){
    $(document).ready(function() {
        bootbox.confirm("Apakah data ini akan dihapus?", function(result)
{
            if(result) {
                $.ajax({
                    type: "POST",

```



```

        <th>Gejala</th>
    </tr>
    <tr>
        <?php $i = 1; foreach($listGejala->result() as $value){ ?>
            <tr>
                <td width="30px"><?php echo $i++?></td>
                <td><?php echo $value->kode." - ".$value->keterangan?></td>
            </tr>
        <?php }?>
    </tr>
</table>
</div><!--box body-->
</div><!--box-->
<div class="box box-success">
    <div class="box-header with-border">
        <h6 class="box-title">Hasil Diagnosa</h6>
    </div><!-- /.box-header -->
    <div class="box-body">
        <table id="tbl-list" class="table table-bordered table-striped">
            <tr>
                <th width="50px">No</th>
                <th>Penyakit</th>
                <th>Tingkat Kepercayaan</th>
                <th>Tingkat Ketidakpercayaan</th>
            </tr>
            <tr>
                <?php $i = 1; foreach($listPenyakit as $value){ ?>

```

```

        <tr>
            <td width="30px"><?php echo $i++?></td>
            <td><?php echo $value['kode']." - ".$value['nama']?></td>
            <td><?php echo $value['kepercayaan']?> %</td>
            <td><?php echo $value['jjj']?> %</td>
        </tr>
    <?php }?>
</tr>
</table>
</div><!--box body-->
</div><!--box-->

```

```

<div class="box box-success">
    <div class="box-header with-border">
        <h6 class="box-title">Kesimpulan</h6>
    </div><!-- /.box-header -->
    <div class="box-body">
        <?php if(sizeof($listPenyakit)>0) { ?>
            <p>
                Berdasarkan gejalanya, unggas tersebut di prediksi mengidap
                penyakit <b><?php echo $listPenyakit[0]['nama'];?></b> dengan tingkat
                kepercayaan <b><?php echo $listPenyakit[0]['kepercayaan'];?> %</b><br/>
                <?php echo $listPenyakit[0]['keterangan'];?>
            </p>
            <?php }else{ ?>
                <p>
                    Penyakit tidak dapat diprediksi karena tingkat kepercayaan gejala
                    terlalu rendah
                </p>
            </div>

```

```
</p>
<?php }?>
</div><!--box body-->
<div class="box-footer clearfix">
    <a class="btn btn-sm btn-primary btn-flat pull-right" href="<?php echo
base_url()?>">Deteksi Ulang</a>
</div>
</div><!--box-->
</div>
```