



UNIVERSITAS INDONESIA

**SISTEM PAKAR PENDIAGNOSA PENYAKIT BERBASIS
WEB**

SKRIPSI

Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana Teknik

**DEDI KURNIAWAN
0404030261**

**FAKULTAS TEKNIK
DEPARTEMEN ELEKTRO
DEPOK
JUNI 2009**

HALAMAN PERNYATAAN ORISINALITAS

Skripsi ini adalah hasil karya saya sendiri, dan semua sumber baik yang dikutip maupun dirujuk telah saya nyatakan dengan benar

Nama : Dedi Kurniawan

NPM : 0404030261

Tanda Tangan :

Tanggal : 17 Juni 2009

HALAMAN PENGESAHAN

Skripsi ini diajukan oleh :
Nama : Dedi Kurniawan
NPM : 0404030261
Program Studi : Elektro
Judul Skripsi : Sistem Pakar Pendiagnosa Penyakit Berbasis Web

Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Teknik pada Program Studi Elektro, Fakultas Teknik, Universitas Indonesia

DEWAN PENGUJI

Pembimbing : Dr. Ir. Anak Agung Putri Ratna, M.Eng (.....)
Penguji : Muhammad Salman ST, MIT (.....)
Penguji : Ir. A. Endang Sriningsih MT., Si (.....)

Ditetapkan di : Ruang Rapat Lantai 1 Departemen Teknik Elektro Depok

Tanggal : 29 Juni 2009

KATA PENGANTAR/UCAPAN TERIMA KASIH

Puji syukur saya panjatkan kepada Allah SWT, karena atas berkat dan rahmat-Nya, saya dapat menyelesaikan skripsi ini. Penulisan skripsi ini dilakukan dalam rangka memenuhi salah satu syarat untuk mencapai gelar Sarjana Teknik Program Studi Elektro pada Fakultas Teknik Universitas Indonesia. Saya menyadari bahwa, tanpa bantuan dan bimbingan dari berbagai pihak, dari masa perkuliahan sampai pada penyusunan skripsi ini, sangatlah sulit bagi saya untuk menyelesaikan skripsi ini. Oleh karena itu, saya mengucapkan terima kasih kepada:

- (1) Dr. Ir. Anak Agung Putri Ratna, M.Eng, selaku dosen pembimbing yang telah menyediakan waktu, tenaga, dan pikiran untuk mengarahkan saya dalam penyusunan skripsi ini;
- (2) Muhammad Salman ST, MIT, yang telah membantu saya dengan ide-idenya dan izinnnya untuk menggunakan fasilitas yang ada di ruang Mercator;
- (3) Sammy Yahya, S.Kes, yang telah membantu saya dalam memahami cara kerja dokter;
- (4) Ardy Satria Hasanuddin, yang telah membantu saya dalam menguasai pemrograman PHP;
- (5) Bramantya Gumilang, yang telah memberikan saya keleluasaan dalam menggunakan servernya;
- (6) orang tua dan keluarga saya yang telah memberikan dukungan moral dan material; dan
- (7) teman-teman saya di Teknik Elektro angkatan 2004 khususnya Bimo, Apul, Martius, Nanda Z, Rofan, Yunanto atas ide-ide serta saran dan juga bantuan berupa peralatan yang digunakan dalam skripsi ini.

Akhir kata, saya berharap Allah SWT berkenan membalas segala kebaikan semua pihak yang telah membantu. Semoga skripsi ini membawa manfaat bagi pengembangan ilmu.

Depok, 17 Juni 2009

Penulis

**HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI
TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS**

Sebagai sivitas akademik Universitas Indonesia, saya yang bertanda tangan di bawah ini:

Nama : Dedi Kurniawan
NPM : 0404030261
Program Studi : Elektro
Departemen : Elektro
Fakultas : Teknik
Jenis karya : Skripsi

demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Indonesia **Hak Bebas Royalti Noneksklusif (*Non-exclusive Royalty-Free Right*)** atas karya ilmiah saya yang berjudul:

SISTEM PAKAR PENDIAGNOSA PENYAKIT BERBASIS WEB

beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini Universitas Indonesia berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan memublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Depok

Pada tanggal : 17 Juni 2009

Yang menyatakan

()

ABSTRAK

Nama : Dedi Kurniawan
Program Studi : Teknik Elektro
Judul : Sistem Pakar Pendiagnosa Penyakit Berbasis Web

Sistem pakar pendiagnosa penyakit ini dapat diakses darimana pun sehingga sangat memudahkan orang dalam mendiagnosa penyakitnya. Sistem pakar ini dibuat dengan menggunakan bahasa pemrograman web PHP dan berjalan diatas server APACHE dengan basis data MySQL. Skripsi ini akan membahas tentang cara kerja dokter yang diadopsi pada sistem pakar pendiagnosa penyakit secara keseluruhan, dasar teori yang digunakan, perancangan algoritma, pemodelan, dan hasil pengujian beban server pada penggunaan sistem ini. Sistem ini dapat dijalankan dengan baik dan menurut tanggapan pengguna sistem ini diperlukan. Berdasarkan tanggapan pengguna, hasil diagnosa yang diberikan oleh sistem ini sudah cukup akurat.

Kata kunci:
Sistem pakar, dokter rekaan, berbasis web.

ABSTRACT

Name : Dedi Kurniawan
Study Program: Electrical Engineering
Title : Web based Disease Diagnostic Expert System

This expert system can be accessed from anywhere because it is web based, so it will much easier for people to diagnose their own disease. This system is built with PHP programming language and run under APACHE server with MYSQL as the database, so it can be easy to run anywhere. This final project explain about method of the doctor in diagnosing the patient whom adopted by this web based disease diagnostic expert system, basic theory, algorithm design, and result of measuring server performance in running this system. This system runs very well and based on user responds this system is really needed. Based on user responds, diagnostic result given by this system is pretty accurate.

Key words:
expert system, virtual doctor, web based.

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PERNYATAAN ORISINALITAS	ii
LEMBAR PENGESAHAN	iii
KATA PENGANTAR/UCAPAN TERIMA KASIH	iv
LEMBAR PERSETUJUAN PUBLIKASI KARYA ILMIAH	v
ABSTRAK	vi
ABSTRACT.....	vii
DAFTAR ISI.....	viii
DAFTAR GAMBAR.....	x
DAFTAR TABEL	xii
DAFTAR ISTILAH.....	xiii
1. PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Tujuan.....	2
1.3. Batasan Masalah	2
1.4. Sistematika Penulisan.....	2
2. LANDASAN TEORI SISTEM PAKAR PENDIAGNOSA PENYAKIT BERBASIS WEB	5
2.1. Cara Diagnosa Dokter	6
2.1.1. Wawancara.....	7
2.1.2. Pengecekan Fisik.....	7
2.1.3. Pengecekan Laboratorium	7
2.2. <i>In Absentia Care</i>	8
2.3. <i>Telemedicine</i>	8
2.4. <i>Virtual Doctor</i>	8
2.5. Contoh Aplikasi <i>Cyber Doctor/ Virtual Doctor</i>	9
2.6. Sistem Pakar	11
2.6.1. Struktur Dasar Sistem Pakar	13
2.6.2. Proses Pembuatan Sistem Pakar.....	14
2.6.3. Komponen dalam Keberadaan Sistem Pakar	14
2.6.4. Kategori Permasalahan Sistem Pakar	15
2.7. Aplikasi Pendukung	15
2.7.1. Sekilas Mengenai Apache Server.....	16
2.7.2. Sekilas Mengenai MySQL.....	16
2.7.3. Sekilas Mengenai PHP	17
3. SISTEM PAKAR UNTUK MENDIAGNOSA PENYAKIT BERBASIS WEB.....	19
3.1. Gambaran Umum	19
3.2. Bahasa Pemodelan.....	19
3.2.1. <i>Use case Diagram</i>	20
3.2.2. <i>Statechart Diagram</i>	21
3.2.3. <i>Activity Diagram</i>	22
3.2.4. <i>Sequence Diagram</i>	23

3.2.5. <i>Component Diagram</i>	24
3.2.6. <i>Deployment Diagram</i>	25
3.3. Algoritma Pemrograman	26
3.4. Normalisasi Tabel	28
3.4.1. Bentuk Normal ke Satu (1NF)	28
3.4.2. Bentuk Normal ke Dua (2NF)	29
3.4.3. Bentuk Normal ke Tiga (3NF)	29
4. IMPLEMENTASI DAN ANALISA SISTEM PAKAR PENDINGNOSA PENYAKIT BERBASIS WEB.....	32
4.1. Tampilan dari Sistem Pakar	32
4.2. Pengujian Waktu Eksekusi pada Server	34
4.2.1. Pengujian Pengaruh Jumlah Pertanyaan dengan Waktu Eksekusi.....	35
4.2.2. Pengujian Pengaruh Jawaban "Ya" dan "Tidak" dengan Waktu Eksekusi	36
4.2.3. Pengujian Jumlah Pengguna dengan Waktu Eksekusi	40
4.2. Tanggapan Pengguna terhadap Sistem Pakar Pendingnosa Penyakit ..	42
5. KESIMPULAN	45
DAFTAR ACUAN	46
DAFTAR PUSTAKA	47

DAFTAR GAMBAR

Gambar 2.1.	Halaman awal contoh aplikasi virtual doctor	9
Gambar 2.2.	List gejala penyakit dari aplikasi virtual doctor	10
Gambar 2.3.	Pertanyaan tambahan dari aplikasi virtual doctor.....	10
Gambar 2.4.	Analisa penyakit dari aplikasi virtual doctor.....	11
Gambar 2.5.	Cara kerja sistem pakar dalam memecahkan masalah.....	13
Gambar 3.1.	Gambaran umum sistem.....	19
Gambar 3.2.	Use case diagram dari Sistem.....	20
Gambar 3.3.	Statechart diagram dari Sistem.....	21
Gambar 3.4.	Activity diagram dari Sistem.....	22
Gambar 3.5.	Sequence diagram dari Sistem.....	24
Gambar 3.6.	Component diagram dari Sistem.....	25
Gambar 3.7.	Deployment diagram dari Sistem.....	25
Gambar 3.8.	Algoritma program.....	26
Gambar 4.1.	Halaman awal sistem pakar.....	32
Gambar 4.2.	Halaman awal memilih keluhan utama.....	33
Gambar 4.3.	Halaman pertanyaan gejala tambahan.....	33
Gambar 4.4.	Halaman hasil diagnosa.....	34
Gambar 4.5.	Grafik pengujian jumlah pertanyaan dengan waktu eksekusi.....	35
Gambar 4.6.	Grafik pengujian jumlah pertanyaan dengan waktu eksekusi di tiga titik.....	36
Gambar 4.7.	Grafik pengujian dengan delapan pertanyaan.....	37
Gambar 4.8.	Grafik pengujian dengan enam pertanyaan.....	38

Gambar 4.9. Grafik pengujian dengan dua pertanyaan.....	38
Gambar 4.10 Grafik pengujian perbedaan jumlah pengakses pada suatu waktu.....	41
Gambar 4.11 Grafik pengujian perbedaan jumlah pengakses pada suatu waktu dengan delapan pertanyaan.....	42
Gambar 4.12 Grafik tanggapan pengguna dengan sistem pakar pendiagnosa penyakit berbasis web.....	44

DAFTAR TABEL

Tabel 2.1. Tahapan diagnosa dokter dengan tingkat keakuratannya.....	6
Tabel 2.2. Perbandingan seorang ahli/pakar (Human Expert) dengan sebuah sistem pakar (Expert System).....	13
Tabel 3.1. Bentuk normal ke satu dari normalisasi tabel.....	28
Tabel 3.2. Bentuk normal ke dua tabel ket_penyakit.....	29
Tabel 3.3. Bentuk normal ke dua tabel gejala_penyakit.....	29
Tabel 3.4. Tabel gejala.....	29
Tabel 3.5. Bentuk normal ke tiga tabel gejala_penyaki.t.....	29
Tabel 4.1. Perbandingan waktu eksekusi jawaban “Ya” dan “Tidak”.....	39
Tabel 4.2. Persentase perbedaan waktu eksekusi jawaban “Ya” dan “Tidak”.....	40
Tabel 4.3. Tanggapan pengguna dengan sistem pakar pendiagnosa penyakit berbasis web.....	43

DAFTAR ISTILAH

FKUI	: Fakultas Kedokteran Universitas Indonesia
RAM	: Random Access Memory
CPU	: Central Processing Unit
GPRS	: General Packet Radio Service
3G	: Third-Generation Technology
PHP	: Hypertext Preprocessor
WAMP	: Windows, Apache, MySQL, PHP
IP	: Internet Protocol
CMS	: Content Management System
OS	: Operating System
Mbps	: Mega bit per Second
Cpanel	: Control Panel
NF	: Normal Form
DBMS	: Database Management System.
HTML	: Hypertext Markup Language.
IIS	: Internet Information Service.

BAB 1 PENDAHULUAN

1.1 Latar Belakang

Kemajuan teknologi yang demikian pesatnya mendorong masyarakat untuk terus menciptakan suatu terobosan baru di segala bidang. Penggunaan sistem baru yang lebih praktis dan cepat pelayanannya serta dapat memberikan kenyamanan bagi pengguna adalah suatu tren tersendiri di zaman yang demikian modern. Oleh karena itu, mulailah bermunculan pelayanan yang berbasis web atau yang sering disebut dengan nama *online service*. Layanan ini merambah di segala bidang, mulai dari jual-beli secara *online* hingga pendidikan *online* pun kini kerap dijumpai sebagai sarana yang mulai menjadi layanan yang digemari masyarakat dikarenakan kepraktisannya.

Sebagaimana diketahui bersama, bahwa semakin majunya peradaban maka himpitan akan kesibukan setiap manusia akan semakin bertambah sehingga diperlukan suatu sistem yang dapat memberikan kepraktisan bagi setiap orang, sehingga mereka yang ingin menggunakan suatu layanan dapat dengan mudah menggunakannya. Suatu sistem yang berbasis web dapat mengatasi kendala ini karena mereka menawarkan kepraktisan yang teramat sangat dapat memanjakan penggunaannya. Seluruh pelayanan yang ingin dinikmati pengguna ada di ujung jari mereka.

Pada zaman sekarang mungkin bukan merupakan suatu kendala lagi apabila seseorang ingin mengakses internet dikarenakan infrastruktur jaringan internet baik dengan maupun tanpa kabel sudah amat banyak dijumpai di berbagai tempat. Terlebih setiap operator seluler pun sekarang berlomba-lomba untuk menawarkan layanan data baik GPRS maupun 3G sebagai salah satu *application killer* mereka. Hal ini tentu saja akan makin mempermudah setiap individu untuk terus dapat mengakses Internet.

Salah satu pelayanan yang mungkin dapat ditawarkan kepada masyarakat adalah sebuah pelayanan kesehatan berbasis web. Hal ini tentu akan sangat

mempermudah masyarakat untuk melakukan konsultasi kesehatan, dan pastinya hal ini juga akan meringankan biaya yang perlu dikeluarkan apabila ingin melakukan konsultasi kesehatan. Terobosan ini juga dapat menjadi suatu pemecah masalah dalam menghadapi tantangan tentang wilayah Indonesia yang demikian luas dan dengan topografi yang sulit dijangkau. Sehingga dengan adanya layanan kesehatan *online* dapat menghadirkan suatu layanan kesehatan untuk masyarakat di seluruh Nusantara. Walaupun demikian dapat diketahui bahwa belum seluruh rakyat Indonesia yang sadar internet, tetapi kiranya terobosan ini dapat dimanfaatkan pada area-area terpencil yang masih mendapat jaringan informasi telekomunikasi.

Pelayanan kesehatan *online* ini akan mengadopsi cara kerja dari seorang dokter dalam mendiagnosa penyakit seorang pasien dan akan sedapat mungkin memberikan penjelasan yang detil akan penyakit tersebut beserta resep obat yang akan dianjurkan kepada seorang pasien untuk mengobati penyakitnya. Sistem ini akan membandingkan gejala atau symptom yang diderita oleh seorang pasien dengan sebuah pusat data tentang penyakit beserta gejalanya. Kemudian hasil dari perbandingan tersebut akan diambil yang memiliki tingkat ketepatan yang paling tinggi. Lalu begitu telah diketahui diagnosa tentang penyakit seorang pasien, maka kemudian dapat diketahui pula obat yang cocok untuk menyembuhkan penyakit pasien tersebut.

Namun demikian, tentunya sistem ini tidak akan serta merta menggantikan peran kerja seorang dokter secara keseluruhan. Sistem ini dapat berfungsi dengan optimal untuk kasus-kasus gangguan kesehatan ringan, dimana tidak memerlukan interaksi antara dokter dan pasiennya secara langsung.

1.2 Tujuan

Tujuan dari penyusunan skripsi ini adalah untuk mengaplikasikan rancangan sistem pakar pendiagnosa penyakit berbasis web yang dirancang dengan bahasa pemrograman PHP yang diintegrasikan pada basis data MySQL. Sistem ini mengadopsi cara kerja seorang dokter untuk mendiagnosa penyakit pasien dengan metode wawancara. Sistem ini memungkinkan pengguna

mengetahui penyakit yang dideritanya dengan menjawab pertanyaan yang diajukan oleh sistem, yang kemudian hasil jawaban dari pengguna tersebut akan diolah dan dibandingkan dengan pusat data tentang penyakit beserta gejalanya yang kemudian akan ditampilkan pula anjuran obat bagi pengguna dan penyebab dari penyakit tersebut. Setelah sistem ini selesai dibuat, maka selanjutnya akan diukur kinerja server untuk menjalankan sistem ini. Seberapa pengaruhnya perbandingan unjuk kerja server apabila sistem ini diakses oleh lima orang dan satu orang.

1.3 Batasan Masalah

Ada beberapa masalah yang akan dibatasi pada pembahasan skripsi ini, batasan masalahnya adalah sebagai berikut :

1. Metode pendiagnosaan yang akan digunakan pada skripsi ini merupakan metode wawancara yang digunakan oleh dokter pada umumnya.
2. Jumlah gejala dari suatu penyakit dibatasi hanya 5 gejala saja, dikarenakan pada umumnya gejala suatu penyakit hanya sekitar 5 saja.
3. Penyakit yang terdapat di basis data sangat terbatas, hal ini dimaksudkan hanya untuk mencoba apa sistem yang dibuat dapat berjalan dengan baik atau tidak.
4. Ketepatan pendiagnosaan yang dilakukan oleh sistem ini maksimal hanya sampai 80 persen, mengingat metode yang digunakan hanya metode wawancara saja.

1.4 Sistematika Penulisan

Sistematika penulisan pada skripsi ini adalah sebagai berikut :

BAB 1 PENDAHULUAN

Bab ini terdiri dari latar belakang, tujuan, batasan masalah, dan sistematika penulisan.

BAB 2 DEFINISI SISTEM PAKAR PENDIAGNOSA PENYAKIT

Bab ini menjelaskan tentang teori-teori dasar tentang cara pendiagnosaan dokter yang diadopsi oleh sistem ini, sistem pakar, dan teknologi yang digunakan pada sistem ini.

BAB 3 RANCANGAN SISTEM PAKAR PENDIAGNOSA PENYAKIT

Bab ini menjelaskan mengenai tahapan-tahapan yang dilakukan dalam merancang sistem pakar pendiagnosa penyakit berbasis web ini, seperti pemodelan, algoritma pemrograman, dan keterhubungan table-tabel pada basis data.

BAB 4 UJI COBA DAN ANALISA

Bab ini berisi pengujian dan analisa unjuk kerja server saat menjalankan sistem ini serta tanggapan pengguna terhadap sistem pakar pendiagnosa penyakit berbasis web.

BAB 5 KESIMPULAN

Bab ini berisi kesimpulan tentang keseluruhan skripsi

BAB II

SISTEM DIAGNOSA PENYAKIT BERBASIS WEB

Kesehatan merupakan hal yang begitu penting bagi manusia. Ironisnya banyak sekali penyakit-penyakit yang pada akhirnya terlambat didiagnosa sehingga mencapai tahap kronis yang membuatnya sulit untuk ditangani. Padahal setiap penyakit sebelum mencapai tahap kronis umumnya menunjukkan gejala-gejala penyakit yang telah diderita oleh pasien tetapi masih dalam tahap yang ringan misalnya seperti sakit kepala, batuk atau nyeri pada sendi. Sayangnya karena ketidaktahuan pada penyakit tersebut maka biasanya seorang penderita penyakit akan mengabaikan gejala-gejala ringan tersebut yang berpotensi untuk menjadi fatal jika tidak diberi tindakan pengobatan.

Kemajuan teknologi mendorong setiap orang untuk terus memperbaharui sistem pelayanan kepada seorang pengguna jasa. Sistem pelayanan yang dimaksud disini akan membahas tentang sistem pelayanan kesehatan. Sistem ini harus dapat memberikan kenyamanan kepada pasien dalam segala bidang. Hal yang perlu menjadi pertimbangan adalah bahwa pelayanan kesehatan konvensional akan sangat banyak menyita waktu seorang pasien, meskipun penyakit yang diderita tidak terlalu parah dan seorang pasien yang datang ke rumah sakit mungkin hanya akan ditanyai beberapa pertanyaan oleh seorang dokter mengenai gejala penyakitnya, tetapi pasien tersebut tetap saja diharuskan untuk meluangkan waktunya untuk datang ke rumah sakit, mengantri, dan bertemu dokter, yang sebenarnya hal seperti ini dapat dipermudah lagi demi kenyamanan pasien. Belum lagi seorang pasien diharuskan untuk membayar biaya yang cukup mahal untuk bertemu dokter.

Sistem pelayanan kesehatan seperti ini sebenarnya dapat diganti dengan sistem baru yang berbasiskan web asalkan diketahui pola kerja seorang dokter dalam mendiagnosa suatu penyakit. Sistem yang digunakan akan berbasiskan web sehingga sistem ini akan sangat mudah sekali dijangkau oleh setiap orang dalam pengaksesannya. Bahkan pada zaman sekarang setiap orang dapat mengakses

suatu situs dari telepon genggamnya masing-masing. Hal ini yang akan sangat memudahkan orang untuk menggunakan pelayanan ini.

2.1 Cara Diagnosa Dokter

Seorang dokter dalam mendiagnosa penyakit pasiennya akan melalui tiga tahapan yang akan dilakukannya :

1. Wawancara
2. Pengecekan Fisik
3. Pengecekan Laboratorium/Penunjang

Ketiga tahapan ini diurutkan berdasar seberapa tingkat keakuratan seorang dokter dalam mendiagnosa suatu penyakit. Perlu diingat bahwa seorang dokter dapat melakukan kesalahan pendiagnosaan dari suatu penyakit seorang pasiennya. Hal ini yang melatarbelakangi bahwa pendiagnosaan penyakit dari seorang dokter tingkat keakuratannya tidak mungkin mencapai angka absolute 100 %. Apabila ketiga tahapan tersebut dilakukan maka tingkat keakuratan dari diagnosa seorang dokter mencapai maksimal 98%. Tabel 2.1 menunjukkan tingkat keakuratan diagnosa seorang dokter berdasarkan tahapan-tahapannya.

Tabel 2.1 Tahapan diagnosa dokter dengan tingkat keakuratannya.

Tahapan	Tingkat Keakuratan
Wawancara	Maximum 80%
Pengecekan Fisik	Maximum 90%
Pengecekan Laboratorium/Penunjang	Maximum 98%

Dari data diatas dapat diketahui bahwa tahapan wawancara yang dilakukan oleh seorang dokter kepada pasiennya sudah cukup untuk mendapatkan gambaran tentang kemungkinan penyakit dari seorang pasien. Pada prakteknya, seorang dokter kemungkinan hanya melakukan tahapan wawancara untuk mendiagnosa

penyakit dari seorang pasien, mengingat kemungkinan keakuratan pada tahapan ini yang cukup besar, sehingga dua tahapan lainnya, hanya dilakukan untuk menambah keyakinan dari seorang dokter untuk memvonis seorang pasien tentang penyakit yang dideritanya.

2.1.1 Wawancara

Pada tahapan ini, seorang dokter akan menanyakan seorang pasien tentang gejala yang dideritanya. Dimulai dari pertanyaan pertama tentang keluhan utama dari seorang pasien. Keluhan utama ini merupakan gejala yang diderita oleh seorang pasien yang menyebabkan mengapa pasien tersebut datang ke rumah sakit. Keluhan utama setiap orang dapat berbeda-beda walaupun penyakit yang diderita sama, tergantung dari tingkat ketahanan fisik seseorang terhadap suatu penyakit. Setelah diketahui keluhan utamanya maka selanjutnya dokter akan menanyakan tentang gejala-gejala yang diderita oleh seorang pasien. Gejala-gejala inilah yang akan dibandingkan dengan kriteria suatu penyakit dari gejala yang ada.

2.1.2 Pengecekan Fisik

Tahapan ini merupakan tahapan dimana seorang dokter melihat tanda-tanda fisik yang ada dari tubuh seorang pasien, mulai dari ujung kepala hingga ujung kaki. Pengecekan ini dilakukan dikarenakan ada beberapa penyakit yang memiliki ciri-ciri yang dapat dilihat pada tanda-tanda fisik dari seorang pasien. Contohnya, bintik merah yang timbul apabila seseorang menderita penyakit demam berdarah. Beda dengan pengecekan laboratorium, pengecekan fisik tidak menggunakan bantuan dari laboratorium untuk mendapatkan hasilnya.

2.1.3 Pengecekan Laboratorium

Pengecekan laboratorium merupakan tahapan dimana seorang dokter memerlukan bantuan dari laboratorium untuk memeriksa seorang pasien tentang suatu penyakit yang dideritanya. Pengecekan ini akan menambah keakuratan seorang dokter dalam pendiagnosaannya. Pengecekan ini meliputi : pengambilan darah, tes urin, rontgen, dan lain-lain. Pengecekan laboratorium hanya dilakukan

pada jenis-jenis penyakit tertentu yang sangat rumit dalam penentuan jenis penyakitnya jika hanya mengandalkan tahapan wawancara dan pengecekan fisik.

2.2 *In Absentia Care*

Perawatan *In Absentia* merupakan cara pelayanan kesehatan dimana saat seorang pasien dan penyedia layanan kesehatan berada di suatu tempat yang berbeda.^[1] Kontak tatap muka secara langsung hanya dibutuhkan untuk mengawasi proses perawatan kesehatan agar dokter dapat lebih mengenal pasiennya secara personal. Pada kenyataannya, teknologi saat ini memungkinkan akan komunikasi yang tidak perlu berlangsung secara bersama-sama. Beberapa orang berpendapat bahwa jenis perawatan kesehatan ini dapat menyingkirkan tahapan pemeriksaan tradisional, diagnosa dan pengobatan. Pertanyaan berbasis pertanyaan kuisisioner yang dapat diakses secara online akan sangat memudahkan, terutama untuk seorang pengguna yang sibuk, hal ini dapat menjadi suatu solusi yang murah dan nyaman untuk melaksanakan diagnosa *online*.

2.3 Telemedicine

Telemedicine adalah layanan kesehatan yang dilakukan dari jarak jauh. Telemedicine merupakan suatu penerapan pengobatan medis yang berkembang pesat dimana informasi yang ditransfer melalui telepon, internet atau jaringan lain untuk tujuan konsultasi, dan kadang-kadang juga digunakan sebagai prosedur pemeriksaan medis jarak jauh.^[2] *Telemedicine* biasanya merujuk pada penggunaan teknologi informasi dan komunikasi untuk pemberian perawatan klinis.

Cakupan telemedicine cukup luas, meliputi penyediaan layanan kesehatan (termasuk klinis, pendidikan dan pelayanan administrasi) jarak jauh, melalui transfer informasi (audio, video, grafik), dengan menggunakan perangkat-perangkat telekomunikasi (audio-video interaktif dua arah, komputer) dengan melibatkan dokter, pasien dan pihak-pihak lain.

2.4 Virtual Doctor

Sebuah *virtual doctor* memiliki pengertian yang berdekatan dengan dunia *virtual*, dimana hal ini berarti sebuah hal yang tidak nyata namun diupayakan

untuk mendekati kenyataan. Aplikasi *virtual* yang baik merupakan aplikasi yang penerapannya baik fungsi maupun tingkah lakunya mirip dengan kenyataan. *Virtual doctor* berarti sebuah perangkat atau aplikasi yang mempunyai fungsi yang menyerupai fungsi seorang dokter pada dunia nyata. Kata *virtual* erat kaitannya dengan suatu hal yang didapat dari dunia maya, dikarenakan perkembangan dunia maya (internet) yang sangat pesat dan digunakan oleh banyak orang.

2.5 Contoh aplikasi *cyber doctor* / *virtual doctor*.

Aplikasi *virtual doctor* sebenarnya telah dikembangkan juga di Indonesia, sayangnya aplikasi ini memiliki antar muka yang kurang menggambarkan interaksi seorang pasien dengan seorang dokter. Terlebih aplikasi ini juga tidak menggunakan tahapan-tahapan yang sesuai dengan cara diagnosa seorang dokter pada tahapan wawancara. Gambar 2.1 menunjukkan halaman awal dari salah satu contoh aplikasi *virtual doctor*.

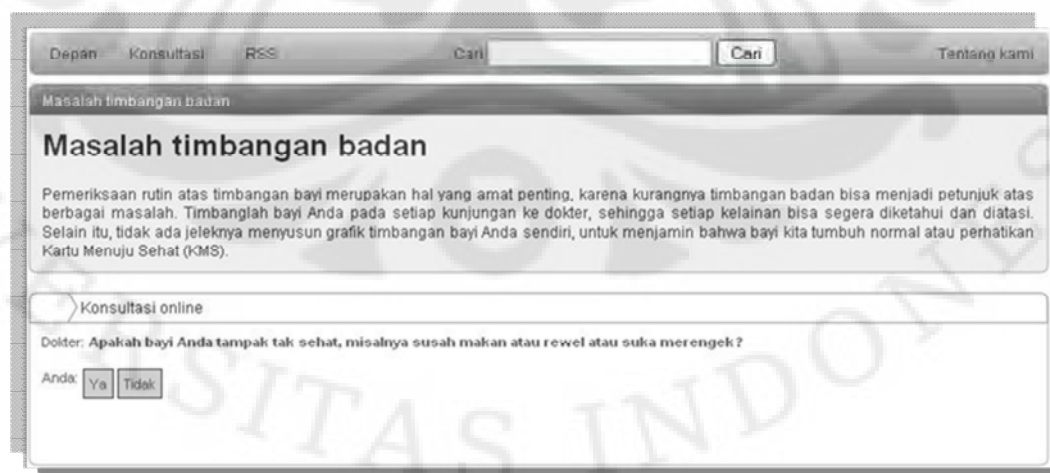


Gambar 2.1 Halaman awal dari salah satu contoh aplikasi *virtual doctor*



Gambar 2.2 List gejala penyakit dari aplikasi virtual doctor.

Pada gambar 2.2 dapat dilihat bahwa aplikasi ini langsung memberikan daftar masalah yang mungkin timbul dari seorang pasien, yang seharusnya pada tahapan pertama suatu aplikasi *virtual doctor* menanyakan keluhan utama dari seorang pasien, yang tentunya harus dibuat suatu fasilitas antar muka yang lebih dapat memberi kemudahan bagi seorang pengguna.



Gambar 2.3 Pertanyaan tambahan dari aplikasi *virtual doctor*.

Gambar 2.3 menunjukkan pertanyaan gejala tambahan dari aplikasi *virtual doctor*. Setelah berada pada suatu masalah, seorang pasien ditanya lagi akan suatu gejala tambahan, jawabannya “ya” dan “tidak”. Saat pasien menjawab “ya” maka selanjutnya hal yang terjadi adalah pengguna diberikan hasil diagnosa penyakitnya, seperti yang terlihat pada gambar 2.4 dibawah ini.



Gambar 2.4 Analisa penyakit dari aplikasi virtual doctor.

Hal ini tentu saja masih sangat jauh untuk mencerminkan mekanisme seorang dokter dalam mendiagnosa pasiennya. Seorang dokter dalam mendiagnosa pasiennya menanyakan segala keluhan yang mungkin diderita oleh seorang pasien, setelah itu baru dibandingkan dengan gejala penyakit yang mungkin diderita seorang pasien, lalu baru pasien tersebut dapat diketahui penyakit yang sedang dideritanya.

2.6 Sistem Pakar

Sistem pakar adalah suatu program komputer yang mengandung pengetahuan dari satu atau lebih pakar manusia mengenai suatu bidang spesifik.^[3] Jenis program ini pertama kali dikembangkan oleh periset kecerdasan buatan pada dasawarsa 1960-an dan 1970-an dan diterapkan secara komersial selama 1980-an. Bentuk umum sistem pakar adalah suatu perangkat lunak yang dibuat berdasarkan

suatu set aturan yang menganalisis informasi (biasanya diberikan oleh pengguna suatu sistem) mengenai suatu permasalahan yang spesifik serta analisis matematis dari masalah tersebut. Tergantung dari desainnya, sistem pakar juga mampu merekomendasikan suatu rangkaian tindakan pengguna untuk dapat menerapkan koreksi. Sistem ini memanfaatkan kemampuan penalaran untuk mencapai suatu kesimpulan. Tujuan dari sebuah sistem pakar adalah untuk mentransfer kepakaran/keahlian yang dimiliki seorang pakar/ahli kedalam komputer, dan kemudian kepada orang lain (*nonexpert*).

Pengertian dari kata pakar/ahli itu sendiri adalah seorang individu yang memiliki kemampuan atau pemahaman yang berlebih pada suatu masalah atau bidang ilmu. Misalnya: seorang dokter, penasehat keuangan, pakar mesin mobil, dll. Dengan bersandar pada definisi tersebut maka seorang pakar/ahli memiliki kemampuan yang meliputi kemampuan untuk mengenali dan merumuskan masalah, menyelesaikan masalah dengan cepat dan tepat, menjelaskan solusi, belajar dari pengalaman, restrukturisasi pengetahuan, menentukan relevansi/hubungan, dan memahami batas kemampuan. Kepakaran/keahlian dari seseorang dapat diperoleh dengan memiliki pemahaman yang luas dari tugas atau pengetahuan spesifik yang diperoleh dari pelatihan, membaca dan pengalaman.

Alasan mendasar mengapa sistem pakar (*expert system*) dikembangkan untuk menggantikan seorang pakar: ^[4]

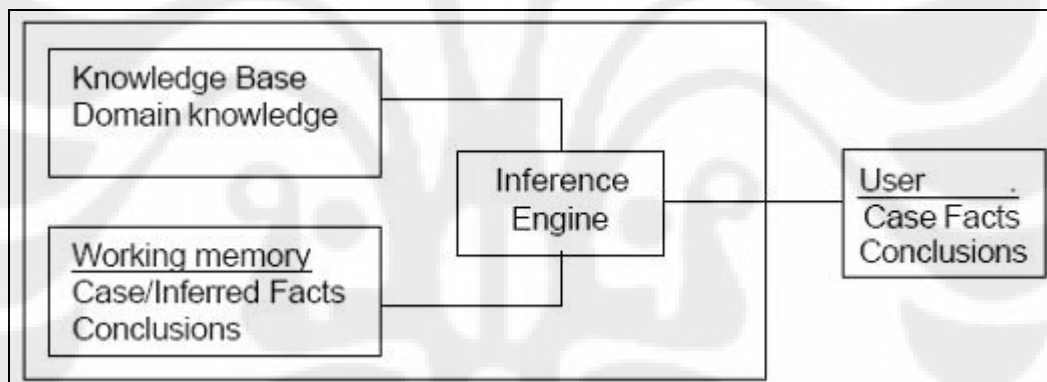
- Dapat menyediakan keahlian dari sebuah profesi setiap waktu dan di berbagai lokasi
- Secara otomatis mengerjakan tugas-tugas rutin yang membutuhkan seorang pakar.
- Seorang pakar dapat pensiun atau pergi.
- Biaya untuk menggunakan tenaga seorang pakar sangatlah mahal.
- Tenaga ahli dibutuhkan juga pada lingkungan yang tidak bersahabat

Tabel 2.2 Perbandingan seorang ahli/pakar (*Human Expert*) dengan sebuah sistem pakar (*Expert System*)^[4]

Parameter Perbandingan	Human Expert	Expert System
Waktu Kesiapan	Hari Kerja	Setiap Saat
Letak Pengaksesan	Di suatu tempat tertentu	Dimana Saja
Dapat Habis	Ya	Tidak
Performansi	Tidak Konsisten	Konsisten
Kecepatan	Tidak Konsisten	Konsisten
Biaya	Tinggi	Terjangkau

2.6.1 Struktur dasar sistem pakar

Berikut dapat dilihat pada gambar 2.5 bagian-bagian yang terdapat pada sistem pakar beserta kegunaannya dalam memecahkan masalah.



Gambar 2.5 Cara kerja sistem pakar dalam memecahkan masalah.^[2]

Knowledge base

Bagian ini merupakan bagian dari sistem pakar yang berisi domain pengetahuan. Bagian ini berisi tentang pengetahuan yang dibutuhkan untuk memahami, merumuskan dan menyelesaikan suatu masalah.

Working Memory

Bagian ini merupakan bagian dari sistem pakar yang berisi fakta-fakta masalah yang ditemukan dalam suatu sesi. Bagian ini berisi fakta-fakta tentang suatu masalah yang ditemukan dalam proses konsultasi.

Inference engine

Pada bagian ini terdapat suatu pemroses (*processor*) pada sistem pakar yang bertugas untuk mencocokkan fakta-fakta yang ada pada *working memori* dengan domain pengetahuan yang terdapat pada *knowledge base*, untuk menarik kesimpulan dari masalah yang dihadapi. Proses berpikir pada manusia dimodelkan dalam sistem pakar pada bagian ini.

2.6.2 Proses pembuatan sistem pakar

Terdapat dua proses dalam rangka pembuatan sistem pakar : ^[4]

1. *Knowledge Acquisition*

Pada proses ini terjadi kegiatan pengumpulan, pemindahan, dan perubahan dari kemampuan seorang pakar atau sumber pengetahuan ke dalam program komputer.

2. *Knowledge Engineering*

Pada proses ini terjadi kegiatan pengembangan suatu sistem pakar yang dilakukan oleh pemrogram yang dalam hal ini disebut *knowledge engineer*.

2.6.3 Komponen dalam Keberadaan Sistem Pakar

Ada tiga hal yang terlibat dalam keberadaan sistem pakar : ^[4]

1. *Domain Expert*

Merupakan bagian bidang ahli, meliputi keahlian dari seseorang pada suatu bidang.

2. *Knowledge Enginner*

Merupakan bagian dimana seorang pemrogram membangun suatu sistem pakar.

3. *End-user*

Merupakan bagian yang berisi pengguna awam yang akan menggunakan fasilitas dari sistem pakar tersebut.

2.6.4 Kategori Permasalahan Sistem Pakar

Terdapat beberapa permasalahan dalam sistem pakar secara umum :^[4]

1. Interpretasi – membuat kesimpulan atau deskripsi dari sekumpulan data mentah.
2. Prediksi – memproyeksikan akibat-akibat yang dimungkinkan dari situasi-situasi tertentu
3. Diagnosa – menentukan sebab malfungsi dalam situasi kompleks yang didasarkan pada gejala-gejala yang teramati
4. Desain – menentukan konfigurasi komponen-komponen sistem yang cocok dengan tujuan-tujuan kinerja tertentu yang memnuhi kendala -
5. Perencanaan – merencanakan serangkaian tindakan yang akan dapat mencapai sejumlah tujuan dengan kondisi awal tertentu
6. Debugging dan Perbaiki – menentukan dan menginterpretasikan cara-cara untuk mengatasi malfungsi.
7. Instruksi – mendeteksi dan mengoreksi defisiensi dalam pemahaman domain subyek
8. Pengendalian – mengatur tingkah laku suatu *environment* yang kompleks
9. Seleksi – mengidentifikasi pilihan terbaik dari sekumpulan (*list*) kemungkinan
10. Simulasi – pemodelan interaksi antara komponen-komponen sistem
11. Pengamatan – membandingkan hasil pengamatan dengan kondisi yang diharapkan

2.7 Aplikasi-aplikasi pendukung.

Untuk dapat menjalankan sistem diagnosa penyakit berbasis web dibutuhkan beberapa aplikasi pendukung seperti MySQL server dan APACHE server. Dalam mendapatkan kedua aplikasi pendukung tersebut, terdapat suatu software bernama WAMP server yang merupakan suatu paket yang mencakup kedua aplikasi tersebut. WAMP server merupakan singkatan dari Windows Apache MySQL PHP server. Untuk bahasa pemrograman yang dipakai adalah

bahasa pemrograman PHP dengan menggunakan basis data MySQL. Untuk lebih jelasnya, berikut ini adalah fungsi dari masing-masing komponen.

2.7.1 Sekilas Mengenai Apache Server

Apache merupakan salah satu dari sekian banyak aplikasi web server yang tersedia. Sebagai aplikasi *open source*, Apache memiliki fitur-fitur utama yang sangat banyak, sebagai contohnya adalah :^[5]

- Arsitektur modular
- Mendukung banyak sistem operasi, seperti Windows NT/2000/XP/Vista dan berbagai varian Unix
- Mendukung IP versi 6 (IPv6)
- Mendukung CGI (Common Gateway Interface) dan SSI (Server Side Include)
- Mendukung otentifikasi dan kontrol akses
- Mendukung SSL (Secure Socket Layer) untuk komunikasi terenkripsi
- Konfigurasi yang mudah dipahami
- Mendukung Virtual Host
- Pesan kesalahan multi bahasa dan bisa dimodifikasi

Apache berbasiskan kode dan ide-ide yang merupakan hasil pengembangan lebih lanjut dari *web server* berbayar NCSA yang dibuat oleh *National Center for Supercomputing Applications*. Pada sistem ini versi Apache yang digunakan adalah versi Apache 2.2.11.

Web server merupakan aplikasi yang berguna untuk memberikan layanan *web* dengan cara melayani *request* dari *client* ke *server* melalui *web browser*. *Web server* menggunakan HTTP (*Hypertext Transfer Protocol*) sebagai *protocol* komunikasinya. Apache memiliki beberapa fitur utama, antara lain :

2.7.2 Sekilas Mengenai MySQL

MySQL adalah sistem *Relational Database Management System* (RDBMS) berdasarkan SQL (*Structured Query Language*) sebuah bahasa pemrograman yang sudah standar di dalam dunia basis data.^[6] Pertama kali dirilis

pada Januari, 1998. MySQL merupakan salah satu aplikasi *Database Management System* (DBMS) yang bersifat *multi-thread* dan *multi-user*. MySQL dimiliki dan disponsori oleh perusahaan Swedia MySQL AB. Mereka memegang hak cipta untuk dari hampir semua kode sumbernya. Para pendiri dari MySQL AB adalah David Axmark, Allan Larsson, dan Michael "Monty" Widenius. MySQL berjalan pada hampir semua platform, termasuk Linux, Unix, dan Windows, serta menyediakan aplikasi program antarmuka (API) untuk berbagai bahasa pemrograman, termasuk C, C ++, Eiffel, Java, Perl, PHP, Python, dan tcl. Pada sistem ini versi MySQL yang digunakan adalah versi 5.1.32.

Kelebihan MySQL dibandingkan jenis basis data lainnya diantaranya :

1. Bersifat terbuka dalam penggunaannya (*open source*).
2. Menggunakan bahasa pemrograman yang umum dan mirip dengan yang digunakan oleh bahasa pemrograman basis data yang lain.
3. Terdapat banyak literatur untuk mempelajari MySQL sehingga membuat MySQL mudah untuk dipelajari
4. MySQL dapat melakukan pemrosesan basis data yang banyak dengan waktu yang sangat cepat
5. Dapat menyimpan *record* dalam jumlah yang sangat besar (lebih dari 50 juta *record*, 60 ribu tabel serta 5 milyar baris)
6. Memiliki sistem *user privilege* yang mudah dan efisien
7. Kompabilitas dengan berbagai sistem operasi dan *web server* yang ada.
8. Memiliki perintah-perintah yang memudahkan dalam pemanipulasian atau pengaksesan data.
9. Memiliki struktur tabel yang fleksibel.

2.7.3 Sekilas Mengenai PHP

PHP merupakan bahasa pemrograman yang paling umum dipakai sebagai bahasa pemrograman suatu aplikasi berbasis web.^[7] Contoh aplikasi web dinamis yang menggunakan PHP adalah CMS wordpress, suatu software management sistem yang sekarang sangat marak digunakan untuk pembuatan blog-blog pribadi.

PHP adalah singkatan dari Hypertext Preprocessor, yang merupakan sebuah bahasa pemrograman yang dipasang dan menyatu pada halaman HTML (Hypertext Markup Language). PHP pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1995 dan bersifat *open source* yang ditulis menggunakan sintaks bahasa C, Java, dan Perl. Pada waktu itu PHP masih bernama FI (Form Interpreted), yang wujudnya berupa sekumpulan script yang digunakan untuk mengolah data form dari web.

Pada tahun 1997, sebuah perusahaan bernama Zend menulis ulang interpreter PHP menjadi lebih bersih, lebih baik, dan lebih cepat. Kemudian pada Juni 1998, perusahaan tersebut merilis interpreter baru untuk PHP dan meresmikan rilis tersebut sebagai PHP 3.0.

Pada pertengahan tahun 1999, Zend merilis interpreter PHP baru dan rilis tersebut dikenal dengan PHP 4.0. PHP 4.0 adalah versi PHP yang paling banyak dipakai pada awal abad ke-21. Versi ini banyak dipakai disebabkan kemampuannya untuk membangun aplikasi web kompleks tetapi tetap memiliki kecepatan dan stabilitas yang tinggi.

Pada Juni 2004, Zend merilis PHP 5.0. Dalam versi ini, inti dari interpreter PHP mengalami perubahan besar. Versi ini juga memasukkan model pemrograman berorientasi objek ke dalam PHP untuk menjawab perkembangan bahasa pemrograman ke arah paradigma berorientasi objek.

PHP hampir dapat berjalan di semua sistem operasi seperti Windows, Unix, Linux dan variannya, Mac OS X, RISC OS dan sistem operasi lainnya. PHP juga kompatibel dengan *web server* yang banyak digunakan sekarang seperti Apache, IIS (*Internet Information Service*), Caudium, Xitami, Omni dan *web server* lainnya. PHP juga mampu berkomunikasi hampir dengan semua sistem *basis data* yang ada sekarang, seperti MySQL, PostgreSQL, Oracle dan lain-lain.

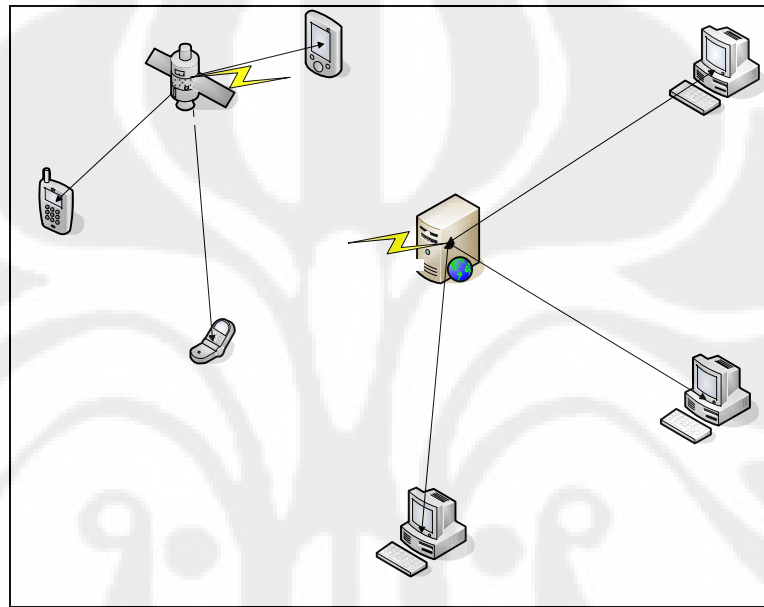
Script PHP dieksekusi di komputer *server* dimana *script* tersebut dijalankan, kemudian hasilnya dikirim ke *web browser client*. PHP membuat sebuah halaman web menjadi lebih dinamis, lebih interaktif dan halaman yang ditampilkan dibuat saat *client* melakukan *request* halaman tersebut sehingga informasi yang diterima oleh *client* adalah informasi yang baru. Pada sistem ini versi PHP yang digunakan adalah versi PHP 5.2.8.

BAB 3

SISTEM PAKAR UNTUK MENDIAGNOSA PENYAKIT BERBASIS WEB

3.1. Gambaran Umum

Gambaran umum mengenai sistem ini dapat digambarkan sebagai berikut :



Gambar 3.1 Gambaran Umum Sistem

Berdasarkan gambar 3.1 dapat dilihat bahwa server yang digunakan untuk menjalankan sistem ini dapat diakses dari manapun juga, termasuk dari telepon selular setiap orang. Server yang digunakan memiliki software apache server dan MySQL server agar dapat menjalankan program yang telah dibuat.

3.2 Bahasa pemodelan.

Bahasa pemodelan atau yang lebih sering disebut dengan UML (Unified Modeling Language) adalah sebuah "bahasa" yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem. UML dapat digunakan untuk membuat model semua jenis aplikasi piranti lunak,

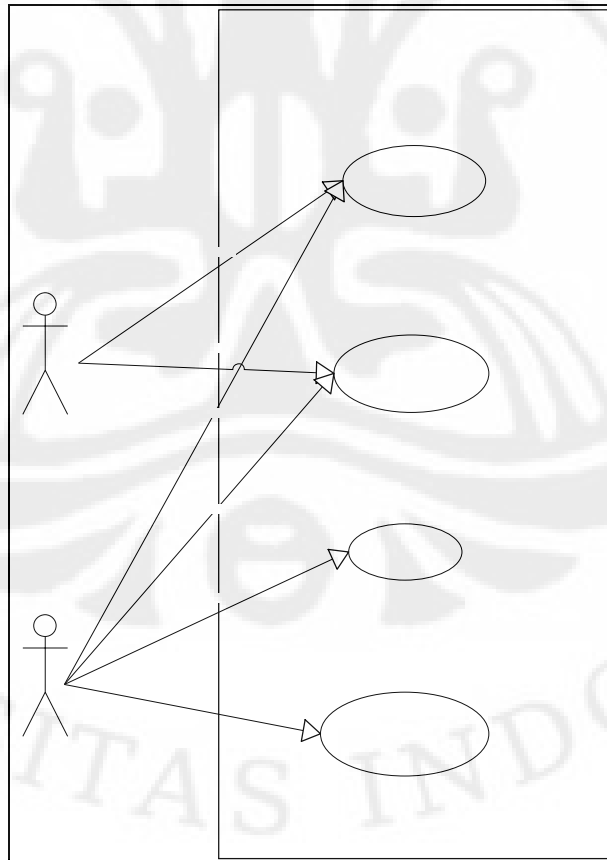
dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun.

Bahasa pemodelan yang akan digunakan adalah : [8]

1. *use case diagram*
2. *statechart diagram*
3. *activity diagram*
4. *sequence diagram*
5. *component diagram*
6. *deployment diagram*

3.2.1 Use Case Diagram.

Diagram ini menggambarkan fungsi dari sebuah sistem, dan interaksi yang dilakukan oleh aktor dengan sistem. Diagram ini menekankan tentang apa yang dapat diperbuat oleh sistem dan bukan bagaimana sistem itu melakukannya.

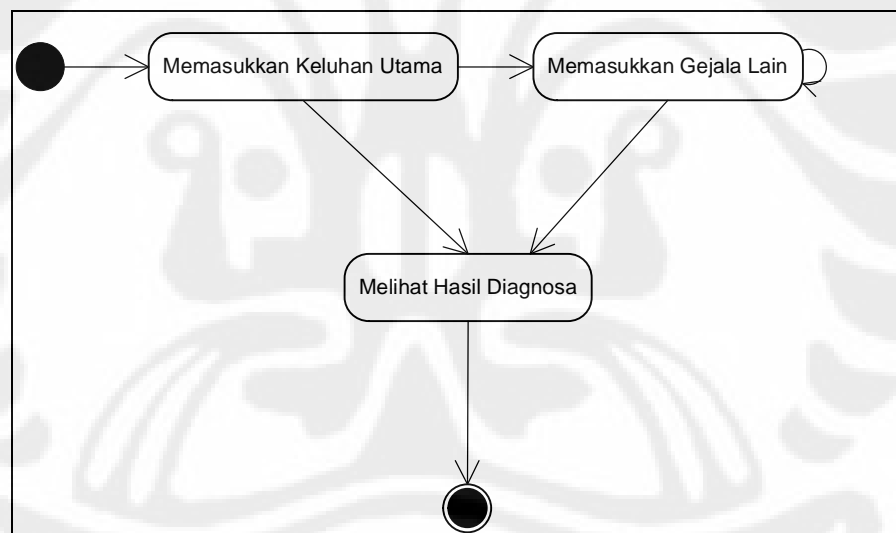


Gambar 3.2 Use Case Diagram dari Sistem

Gambar 3.2 tersebut merupakan diagram Use Case dari sistem ini. Dapat dilihat bahwa pengakses sistem ini dibagi menjadi dua: pengguna biasa dan administrator dari sistem. pengguna biasa dapat mengecek gejala yang diderita dan mendapatkan hasil diagnosa yang diberikan oleh sistem ini. Sedangkan Administrator dapat melakukan semua hal yang dilakukan oleh pengguna biasa sekaligus dapat login ke halaman admin dan memasukkan penyakit baru ke dalam basis data.

3.2.2 Statechart Diagram.

Diagram ini menggambarkan transisi dan perubahan keadaan (dari satu keadaan ke keadaan lainnya) suatu objek pada sistem dikarenakan input yang diterima. Gambar 3.3 menunjukkan perubahan keadaan ketika seorang pengguna sedang menggunakan sistem ini untuk mengecek gejalanya.



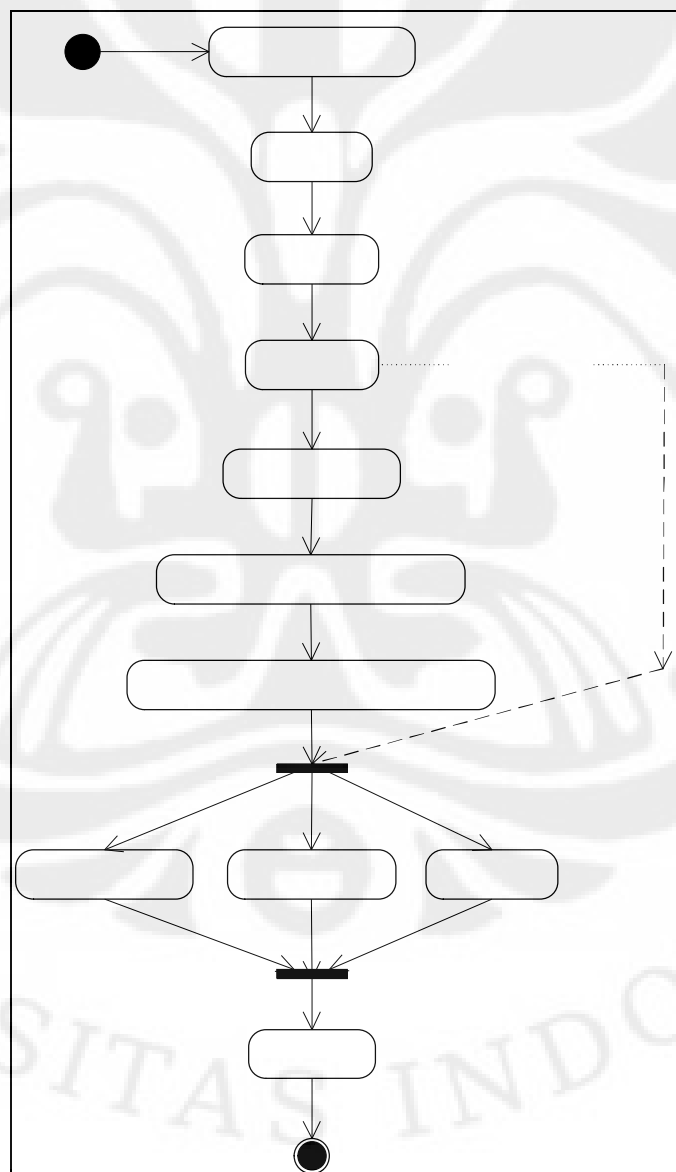
Gambar 3.3 Statechart Diagram dari Sistem

Pada tahap awal pengguna akan memasukkan keluhan utama dari gejala yang dideritanya, setelah itu sistem akan menanyakan gejala lain yang mungkin diderita oleh pengguna. Tanda panah berulang yang ada pada state “Memasukkan gejala lain” bermaksud untuk menunjukkan bahwa kemungkinan gejala lain yang diderita oleh pengguna tidak hanya satu, sehingga diberikan tanda panah berulang

untuk menunjukkannya. Setelah pengguna memasukkan seluruh gejala lain yang ditanyakan oleh sistem maka selanjutnya pengguna dapat melihat hasil diagnosa yang telah diproses oleh sistem berdasar gejala-gejala yang telah dimasukkan oleh pengguna.

3.2.3 Activity Diagram

Diagram ini menggambarkan berbagai alur aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alur berawal, hasil yang mungkin terjadi, dan bagaimana mereka berakhir.



Gambar 3.4 Activity Diagram dari Sistem

Berdasarkan gambar 3.4, maka pertama-tama sistem akan menampilkan semua keluhan utama kepada pengguna. Selanjutnya setelah pengguna memilih keluhan utama yang dideritanya maka sistem akan mengecek ke basis data tentang nama penyakit yang memiliki gejala yang sama dengan keluhan utama yang dipilih.

Penyakit yang memiliki gejala seperti keluhan utama yang dipilih pengguna akan ditandai. Lalu setelah itu, sistem akan menampilkan gejala lain yang dimiliki oleh penyakit-penyakit yang sudah ditandai tersebut. Apabila penyakit yang ditandai tidak memiliki gejala lain selain keluhan utama maka sistem akan menampilkan presentase kecocokan penyakit tersebut, penyebab dari penyakit tersebut, dan obat dari penyakit tersebut. Apabila penyakit yang ditandai memiliki gejala lain selain keluhan utama maka sistem akan menampilkan dan menanyakan gejala lain yang didapat dari gejala-gejala dari penyakit yang sudah ditandai yang kemungkinan diderita oleh pengguna.

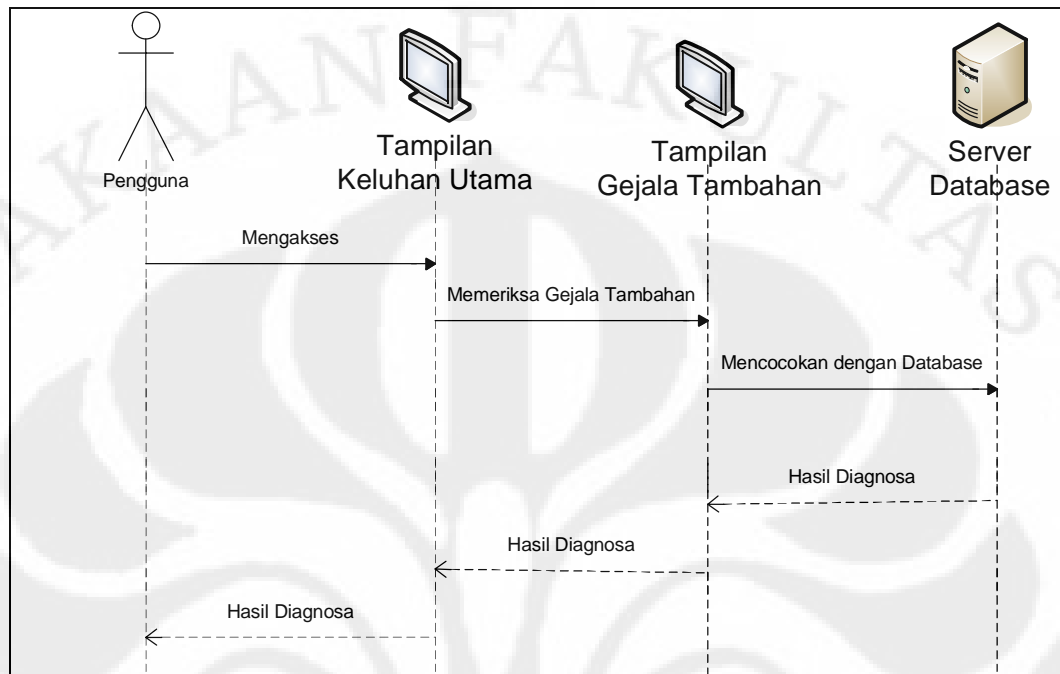
Setelah pengguna menjawab gejala lain yang mungkin dideritanya maka selanjutnya sistem akan membandingkan gejala yang diderita oleh pengguna dengan gejala-gejala yang dimiliki oleh penyakit yang sudah ditandai. Lalu setelah didapat tingkat persentase kecocokannya maka sistem akan menampilkan presentase kecocokan penyakit tersebut, penyebab dari penyakit tersebut, dan obat dari penyakit tersebut.

3.2.4 Sequence Diagram

Diagram ini menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa pesan yang digambarkan terhadap waktu. Diagram ini terdiri dari objek-objek yang terkait dengan sistem yang digambarkan pada posisi horizontal dan dimensi waktu yang digambarkan pada posisi vertikal.

Diagram ini biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah kejadian untuk menghasilkan keluaran hasil tertentu. Diawali dari apa yang memicu

aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan keluaran apa yang dihasilkan.

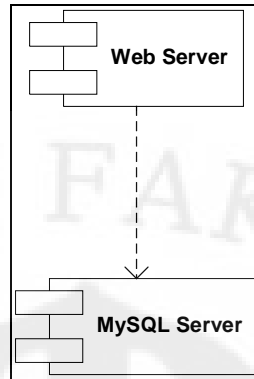


Gambar 3.5 Sequence Diagram dari Sistem

Gambar 3.5 menjelaskan tentang skenario yang terjadi saat seorang pengguna menggunakan sistem ini. Pertama-tama pengguna mengakses ke alamat sistem dari sistem ini. Selanjutnya sistem akan menampilkan list keluhan utama untuk dipilih oleh pengguna. Setelah pengguna memilih keluhan utama yang dideritanya, maka selanjutnya sistem akan menampilkan gejala tambahan yang mungkin diderita oleh pengguna. Kemudian setelah pengguna mengisi semuanya, maka sistem akan membandingkannya ke basis data yang kemudian hasil diagnosanya akan diberikan kepada pengguna.

3.2.5 Component Diagram.

Diagram ini menggambarkan struktur dan hubungan antar komponen piranti lunak, termasuk ketergantungan yang terjadi antara komponen tersebut.

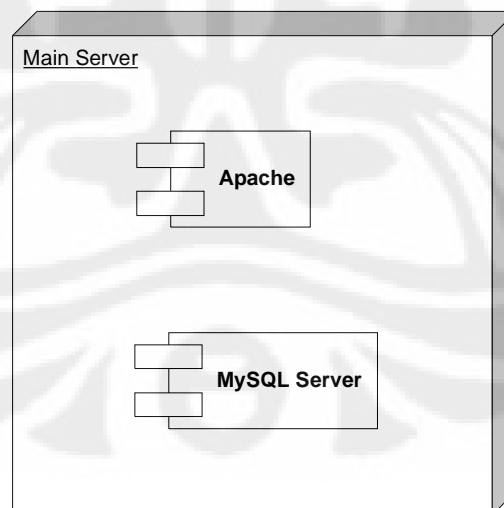


Gambar 3.6 Component Diagram dari Sistem

Gambar 3.6 menjelaskan hubungan ketergantungan dari aplikasi perangkat lunak yang digunakan pada sistem ini. Dapat dijelaskan pada gambar diatas bahwa web server yang dalam hal ini adalah apache memiliki ketergantungan pada MySQL server dikarenakan seluruh data yang ditampilkan oleh web server berasal dari basis data yang ada pada MySQL server.

3.2.6 Deployment Diagram.

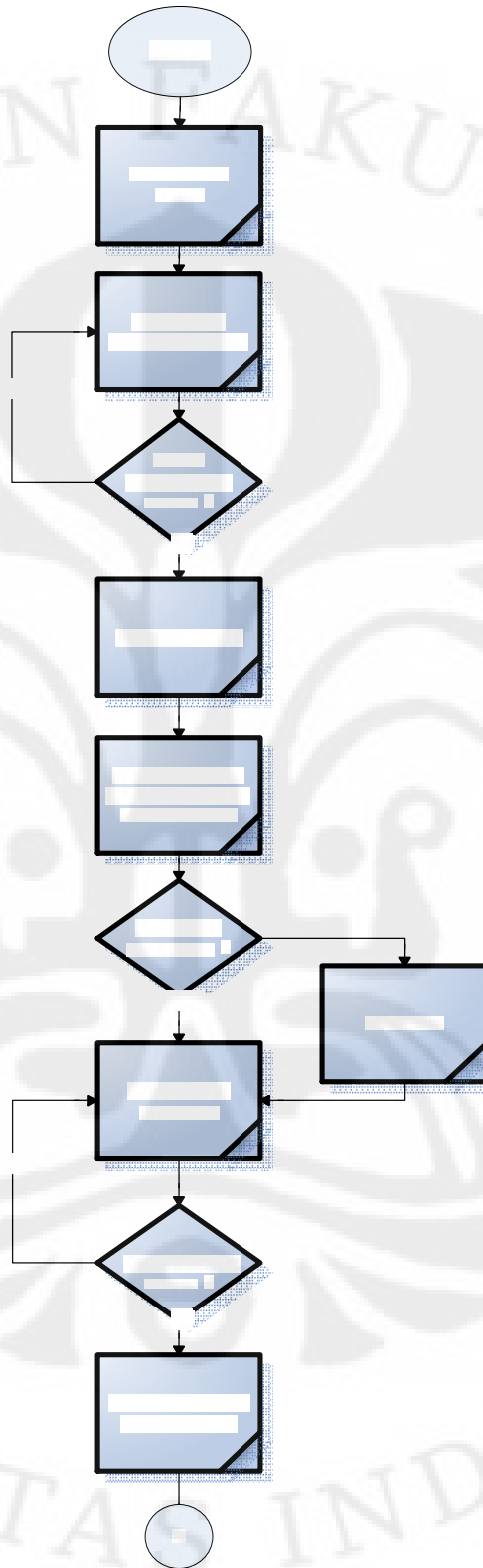
Diagram ini menggambarkan tentang penempatan dari komponen dan bagaimana komponen tersebut ditempatkan dalam infrastruktur sistem.



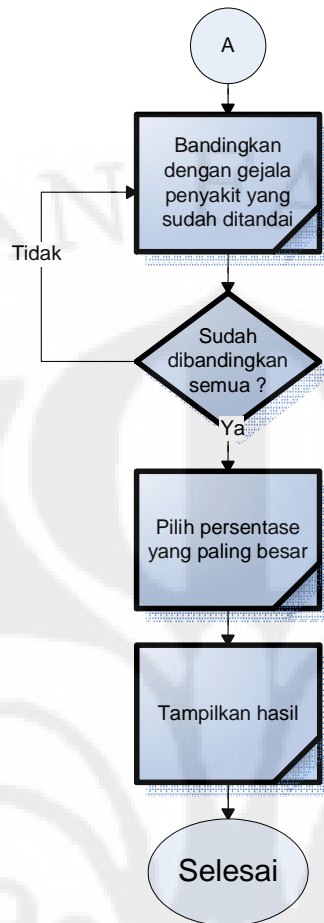
Gambar 3.7 Deployment Diagram dari Sistem.

Berdasarkan gambar 3.7, dapat dilihat bahwa web server apache dan MySQL server ditempatkan pada satu server utama yang sama.

3.3 Algoritma Pemrograman.



Gambar 3.8 Algoritma program.



Gambar 3.8 Algoritma program. (lanjutan)

Dari diagram alir yang ditunjukkan pada gambar 3.8, dapat dijelaskan seperti berikut :

1. Pertama-tama sistem akan menanyakan keluhan utama yang diderita oleh pengguna.
2. Setelah pengguna memasukkan keluhan utama yang dideritanya, maka selanjutnya sistem akan membandingkan dengan basis data yang ada.
3. Sistem akan membandingkan keluhan utama pengguna dengan seluruh gejala-gejala yang ada dari seluruh penyakit yang ada di basis data.
4. Setelah didapatkan gejala yang sama dengan keluhan utama yang diderita oleh pengguna maka sistem akan menandai penyakit yang memiliki gejala yang sama dengan keluhan utama si pengguna.

5. Lalu, sistem akan mengambil seluruh gejala yang dimiliki oleh penyakit-penyakit yang telah ditandai tadi.
6. Sistem akan mengecek apakah dari gejala-gejala yang sudah diambil tadi ada gejala yang sama, jika ada gejala yang sama maka sistem akan menghapus salah satu dari gejala yang sama tersebut (diunikkan).
7. Gejala-gejala yang sudah *unique* ini selanjutnya akan ditanyakan kepada pengguna. Apakah pengguna mengalami seperti gejala yang dimaksud.
8. Pertanyaan akan terus diulang hingga seluruh gejala-gejala tadi selesai ditanyakan.
9. Setelah pengguna menjawab seluruh pertanyaan sistem, maka sistem akan membandingkan gejala yang pengguna derita dengan gejala yang dimiliki oleh penyakit-penyakit yang sudah ditandai.
10. Sistem akan mengambil presentase terbesar dari kemungkinan penyakit yang diderita oleh pengguna dan kemudian ditampilkan kepada pengguna.

3.4 Normalisasi Tabel

Normalisasi merupakan teknik analisis data yang mengorganisasikan atribut-atribut data dengan cara mengelompokkan sehingga terbentuk entitas yang tidak-berulang, stabil, dan fleksible.^[9]

3.4.1 Bentuk Normal ke Satu (1NF)

Tabel 3.1 Bentuk normal ke satu dari normalisasi tabel

id_penyakit	nama_penyakit	gejala 1	gejala2	gejala 3	obat	penyebab
1	DBD	Demam	Nyeri	Bintik	Paracetamol	Virus
2	TBC	Batuk	Sesak	Demam	FDC	Bakteri

Pada bentuk ini yang menjadi *primary key* adalah “*id_penyakit*” dan bernilai unik antara satu dengan yang lainnya. Bentuk normal ke satu merupakan bentuk tabel yang sudah didapatkan *primary key* dari tabel tersebut.

3.4.2 Bentuk Normal ke Dua (2NF)

Tabel 3.2 bentuk normal ke dua tabel ket_penyakit.

id_penyakit	nama_penyakit	obat	penyebab
1	DBD	Paracetamol	Virus
2	TBC	FDC	Bakteri

Tabel 3.3 bentuk normal ke dua tabel gejala_penyakit.

id_penyakit	gejala 1	gejala2	gejala 3
1	Demam	Nyeri	Bintik
2	Batuk	Sesak	Demam

Pada bentuk ini yang menjadi *primary key* dari kedua tabel adalah id_penyakit, kedua tabel saling terhubung karena *primary key* masing-masing tabel. Setiap atribut bukan kunci harus memiliki ketergantungan kepada *primary key* masing-masing tabel.

3.4.3 Bentuk Normal ke Tiga (3NF)

Pada bentuk ini tabel 3.3 akan dipecah menjadi dua tabel sebagai berikut.

Tabel 3.4 tabel gejala

id_gejala	nama_gejala
1	Nyeri
2	Bintik
3	Sesak
4	Batuk
5	Demam

Tabel 3.5 bentuk normal ke tiga tabel gejala_penyakit

id_penyakit	gejala 1	gejala2	gejala 3
1	5	1	2
2	4	3	5

Pada tabel gejala yang menjadi *primary key* adalah id_gejala dan pada tabel gejala_penyakit yang menjadi *primary key* adalah id_penyakit. Bentuk normal ke

tiga memiliki syarat dimana atribut bukan kunci tidak boleh memiliki ketergantungan kepada atribut bukan kunci lainnya. Pemecahan tabel 3.3 menjadi tabel 3.4 dan 3.5 dimaksudkan agar nama gejala yang menyertai setiap penyakit menjadi lebih mudah terlihat perbedaannya antara gejala yang satu dengan gejala yang lain.

Setelah melalui tiga tahapan normalisasi tabel, maka basis data pada sistem akan menggunakan 3 tabel yang saling terhubung oleh *primary key* dari masing-masing tabel, ketiga tabel tersebut adalah :

1. Tabel ket_penyakit.

Tabel ini berisi nama penyakit, obat dari penyakit tersebut, dan penyebab dari penyakit tersebut. Pada tabel ini yang menjadi *primary key* adalah *id_penyakit*. Atribut *nama_penyakit*, *obat*, dan *penyebab* memiliki ketergantungan pada *primary key id_penyakit*.

Tabel 3.2 tabel ket_penyakit.

id_penyakit	nama_penyakit	obat	penyebab
1	DBD	Paracetamol	Virus
2	TBC	FDC	Bakteri

2. Tabel gejala_penyakit.

Tabel ini berisi penyakit yang berupa id penyakit dan gejalanya berupa id gejala. Pada tabel ini yang menjadi *primary key* adalah *id_penyakit*. *Primary key* pada tabel ini memiliki ketergantungan dengan *primary key* pada tabel ket_penyakit. Seperti terlihat pada tabel 3.6 dibawah ini, atribut *gejala1*, *gejala2*, dan *gejala3* berupa angka yang memiliki ketergantungan dengan *primary key* pada tabel 3.4.

Tabel 3.5 tabel gejala_penyakit

id_penyakit	gejala 1	gejala2	gejala 3
1	5	1	2
2	4	3	5

3. Tabel gejala.

Tabel ini berisi seluruh gejala dari seluruh penyakit. Id_gejala sebagai *primary key* dari tabel ini. id_gejala dan nama_gejala bersifat unik. Isi tabel ini yang akan menjadi data untuk ditampilkan pada halaman awal dari sistem dimana pengguna memasukkan keluhan utama yang dideritanya.

Tabel 3.4 tabel gejala

id_gejala	nama_gejala
1	Nyeri
2	Bintik
3	Sesak
4	Batuk
5	Demam

BAB 4

IMPLEMENTASI DAN ANALISA SISTEM PAKAR PENDIAGNOSA PENYAKIT BERBASIS WEB

4.1 Tampilan dari sistem pakar.

Berikut ini akan disajikan tampilan dari halaman - halaman yang terdapat pada sistem pakar ini.



Gambar 4.1 Halaman awal.

Gambar 4.1 menunjukkan halaman awal dari sistem ini. Pada halaman ini pengguna dipersilakan untuk memasukkan keluhan utama yang diderita untuk selanjutnya dapat diproses oleh sistem. Daftar keluhan utama telah disajikan dalam daftar menu. Pengguna dapat memilih daftar keluhan utama seperti terlihat pada gambar 4.2 yang menunjukkan cara memilih keluhan utama.

Welcome to Virtual Doctor



Silahkan Pilih Keluhan Utama Anda

batuk lebih dari 2 minggu	Submit
batuk mengeluarkan dahak (terutama waktu malam)	
berat badan menurun	
berkeringat pada waktu malam hari	
bintik-bintik merah	
bintil-bintil kemerahan dan berair	
buang air besar berdarah	
buang air besar lebih dari 3 kali sehari	
demam	
demam lebih dari 7 hari	
demam selama kurang lebih 2-7 hari	

Gambar 4.2 Halaman awal memilih keluhan utama.

Welcome to Virtual Doctor



Apa kamu menderita gejala muntah?

Ya Tidak

Gambar 4.3 Halaman pertanyaan gejala tambahan.

Dapat dilihat pada gambar 4.3, pengguna diajukan pertanyaan lagi. Pada halaman ini, pengguna akan ditanya mengenai gejala lain yang kemungkinan diderita, pengguna dapat menjawab dengan jawaban “Ya” dan “Tidak”.



Gambar 4.4 Halaman hasil diagnosa.

Selanjutnya setelah pengguna menjawab pertanyaan yang diajukan oleh sistem ini, maka pengguna dapat melihat hasil diagnosa sebagaimana ditampilkan pada gambar 4.4.

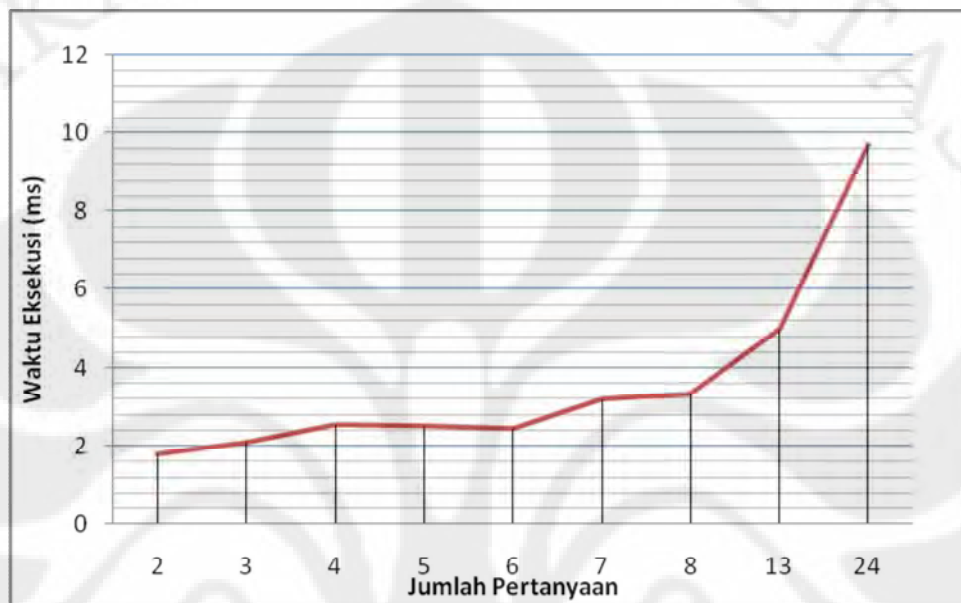
4.2 Pengujian waktu eksekusi pada server.

Untuk pengujian ini, digunakan server dengan spesifikasi sebagai berikut :

- CPU: Intel Xeon X3220
- RAM: 4 x Super Talent 2GB DDR2
- Drive Controller: Adaptec RAID 5405
- Hard Drive: 4 x Seagate Cheetah ST373455SS
- Location : Softlayer datacenter di Seattle, WA.
- Bandwidth : Uplink Port 1000Mbps
- PHP 5.2.8
- MySQL 5.0.67
- Apache 2.2.11

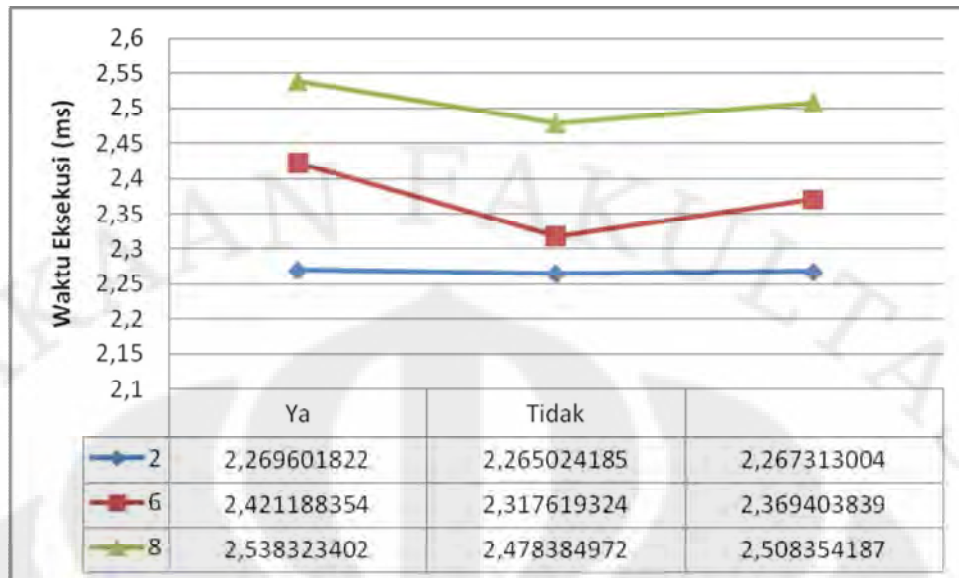
4.2.1 Pengujian pengaruh jumlah pertanyaan dengan waktu eksekusi.

Proses pengujian dilakukan dengan mengambil keluhan utama secara acak dan menghitung jumlah pertanyaan pada gejala tambahan.



Gambar 4.5 Grafik pengujian jumlah pertanyaan dengan waktu eksekusi

Berdasarkan grafik yang ditunjukkan pada gambar 4.5, dapat dilihat bahwa jumlah pertanyaan pada gejala tambahan mempengaruhi waktu eksekusi dari server. Secara garis besar dapat dilihat bahwa semakin banyak jumlah pertanyaan maka semakin lama waktu eksekusi yang dibutuhkan. Hal ini dikarenakan semakin banyaknya jumlah pertanyaan pada gejala tambahan membuat server harus lebih banyak lagi melakukan perintah-perintah untuk pengaksesan basis data, sehingga jelas semakin banyak jumlah pertanyaan yang ada pada pertanyaan gejala tambahan akan menambah beban server.



Gambar 4.6 Grafik pengujian jumlah pertanyaan dengan waktu eksekusi di tiga titik.

Gambar 4.6 menunjukkan grafik yang merupakan pengujian unjuk kerja server dengan mengukur pada tiga titik pengujian. Pengujian pertama dengan menggunakan dua pertanyaan yang dijawab dengan jawaban “Ya” untuk seluruh pertanyaan gejala tambahan. Kemudian pada titik pengujian yang sama dijawab dengan jawaban “Tidak” untuk seluruh pertanyaan pada gejala tambahan, setelah itu diambil rata-rata dari setiap pengukuran jawaban “Ya” dan “Tidak”. Hal yang sama persis dilakukan pada titik pengujian yang lain, yaitu dengan menggunakan enam dan delapan pertanyaan gejala tambahan. Hal ini memperkuat hasil pengujian pada pengukuran sebelumnya yang memperlihatkan bahwa semakin banyak pertanyaan pada gejala tambahan, maka akan semakin lama pula waktu eksekusi pada server.

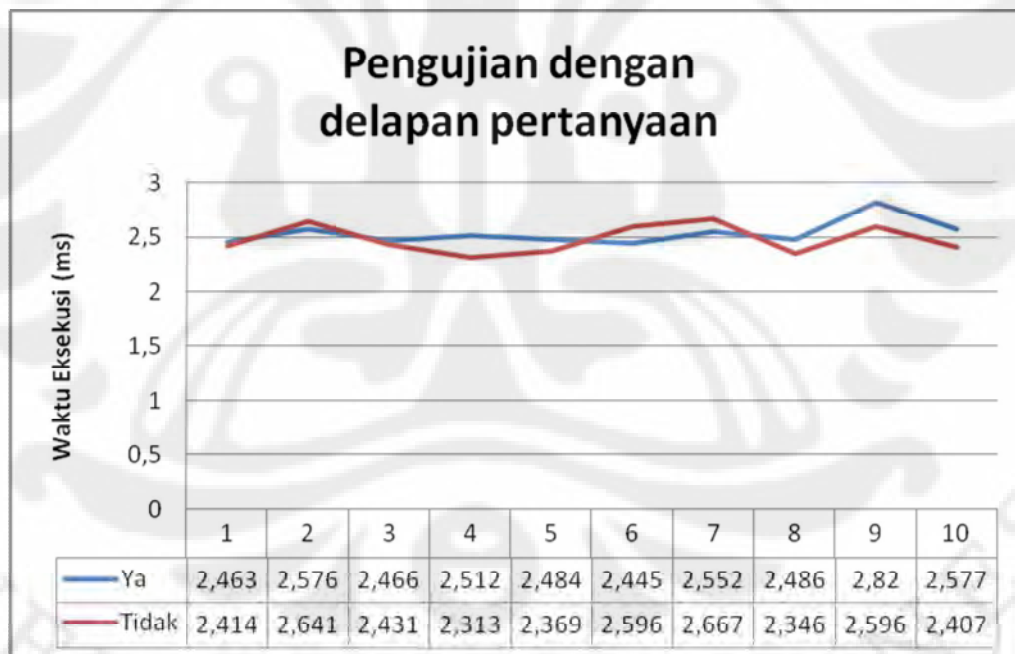
4.2.2 Pengujian pengaruh jawaban “Ya” dan “Tidak” dengan waktu eksekusi.

Proses pengujian dilakukan dengan mengambil keluhan utama yang sama hingga sebanyak sepuluh kali dengan mengisi pertanyaan pada gejala tambahan “Ya”, begitu juga dengan proses pengujian untuk jawaban “Tidak”. Pengujian

dilakukan dengan memilih keluhan utama yang memiliki delapan gejala tambahan lain, enam gejala tambahan lain, dan dua gejala tambahan lain.

Pada gambar – gambar dibawah ini yang menunjukkan pengujian pada titik-titik jawaban tertentu, jawaban “Ya” digambarkan dengan warna biru dan jawaban “Tidak” digambarkan dengan warna merah.

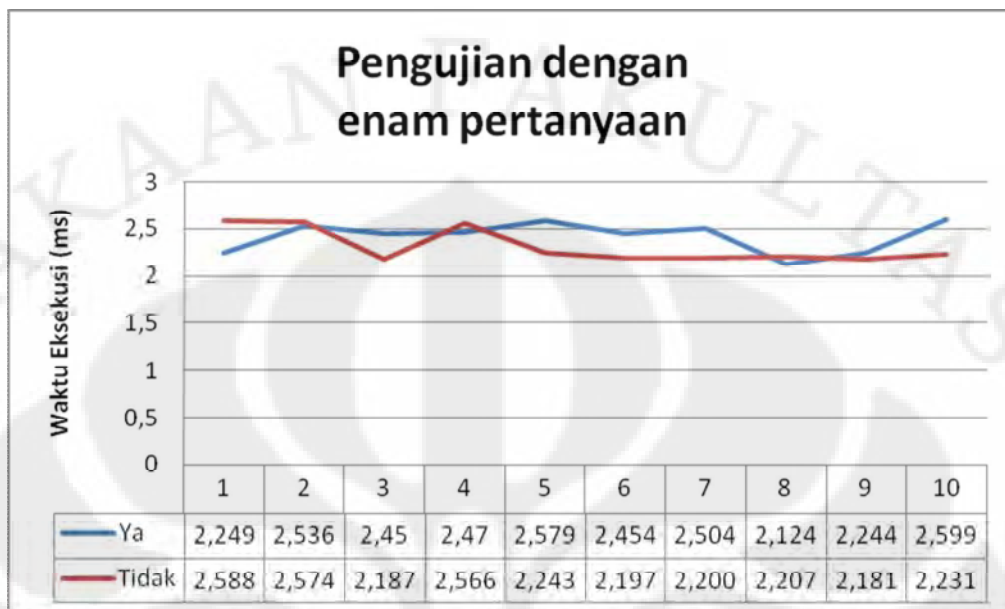
Gambar 4.7 menunjukkan grafik pengujian dengan delapan pertanyaan mengenai gejala tambahan yang mungkin diderita oleh pengguna. Dapat dilihat bahwa grafik yang digambarkan dengan garis biru untuk jawaban “Ya” dan garis merah untuk jawaban “Tidak” menunjukkan hasil yang berimbang. Sehingga dari grafik ini dapat ditarik kesimpulan bahwa pengisian jawaban “Ya” dan “Tidak” pada pertanyaan gejala tambahan dengan delapan pertanyaan tidak mempengaruhi unjuk kerja server secara signifikan.



Gambar 4.7 grafik pengujian dengan delapan pertanyaan.

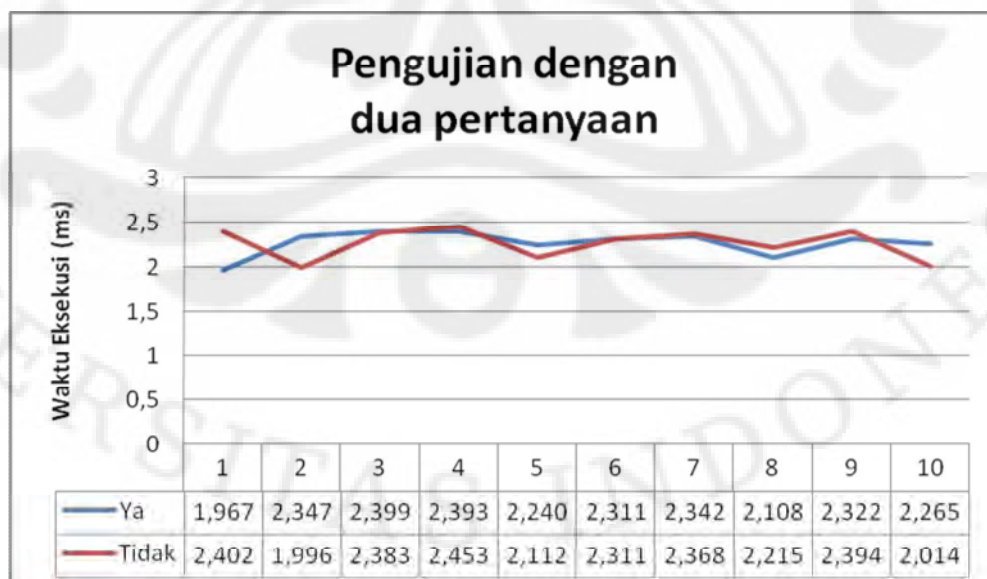
Hal yang sama juga dapat dilihat pada gambar 4.8. Untuk pengujian pada enam pertanyaan mengenai gejala tambahan dapat dilihat bahwa grafik yang ditunjukkan garis biru dan garis merah cukup berimbang. Hal ini menunjukkan

bahwa pemilihan jawaban “Ya” dan “Tidak” pada pertanyaan gejala tambahan dengan enam pertanyaan tidak mempengaruhi unjuk kerja sever.



Gambar 4.8 grafik pengujian dengan enam pertanyaan.

Hasil yang lebih jelas dapat dilihat pada gambar 4.9. Untuk pengujian dua pertanyaan mengenai gejala tambahan, grafik yang dihasilkan nyaris berhimpitan. Hal ini menunjukkan bahwa pemilihan jawaban “Ya” dan “Tidak” pada pertanyaan gejala tambahan dengan dua pertanyaan tidak mempengaruhi unjuk kerja server.



Gambar 4.9 grafik pengujian dengan dua pertanyaan.

Berdasarkan ketiga grafik yang disajikan pada gambar 4.7, gambar 4.8, dan gambar 4.9 di atas dapat dilihat bahwa pemilihan jawaban “Ya” dan “Tidak” pada pertanyaan mengenai gejala tambahan tidak mempengaruhi unjuk kerja server secara signifikan. Hal ini dikarenakan semua isian dari pertanyaan pada gejala tambahan akan dicocokkan dengan basis data, bedanya, jawaban “Tidak” tidak akan dimasukkan kedalam kelompok gejala yang diderita oleh pengguna.

Kelompok gejala yang diderita oleh pengguna ini selanjutnya akan dibandingkan dengan penyakit yang memiliki gejala seperti keluhan utama yang dipilih oleh pengguna, sehingga perbedaan jawaban “Ya” dan “Tidak” akan sedikit mengalami perbedaan perlakuan di akhir pencocokan. Pencocokan yang terjadi pun bukanlah pencocokan ke basis data, tetapi pencocokan antara *array* gejala penyakit yang memiliki gejala yang sama dengan keluhan utama dengan *array* gejala yang diderita oleh pengguna.

Tabel 4.1 Perbandingan waktu eksekusi jawaban “Ya” dan “Tidak”

Pertanyaan	Jawaban		Rata-rata
	Ya	Tidak	
2	2.269602	2.265024	2.267313
6	2.421188	2.317619	2.369404
8	2.538323	2.478385	2.508354

Tabel 4.1 menunjukkan perbandingan waktu eksekusi antara jawaban “Ya” dan “tidak” pada pertanyaan mengenai gejala tambahan yang kemungkinan diderita oleh pengguna. Hasil perhitungan waktu eksekusi pada titik – titik di dua pertanyaan, enam pertanyaan, dan delapan pertanyaan mengenai gejala tambahan diambil nilai rata – rata untuk perhitungan waktu eksekusi pada jawaban “Ya” dan jawaban “Tidak”, selanjutnya rata – rata dari dua jawaban ini diambil lagi rata – rata waktu eksekusinya.

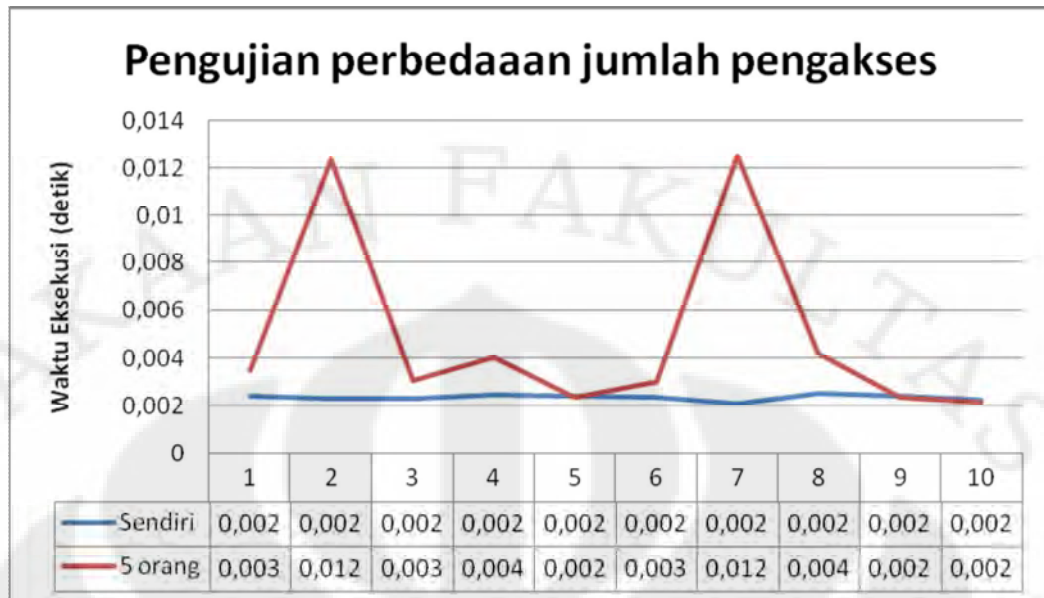
Tabel 4.2 Persentase perbedaan waktu eksekusi jawaban “Ya” dan “Tidak”

Pertanyaan	Jawaban		Persentase perbedaan dalam %
	Ya	Tidak	
2	2.269602	2.26502	0.201693384
6	2.421188	2.31762	4.27761147
8	2.538323	2.47838	2.361339407

Dapat dilihat pada tabel 4.2 bahwa perbedaan waktu eksekusi antara jawaban “Ya” dan “Tidak” tidak terlalu signifikan. Hal ini tampak dari persentase perbedaan yang menunjukkan nilai 0.2 % pada titik pengujian dengan dua pertanyaan gejala tambahan, 4.2 % pada titik pengujian dengan enam pertanyaan gejala tambahan, dan 2.3 % pada titik pengujian dengan delapan pertanyaan gejala tambahan.

4.2.3 Pengujian jumlah pengguna dengan waktu eksekusi

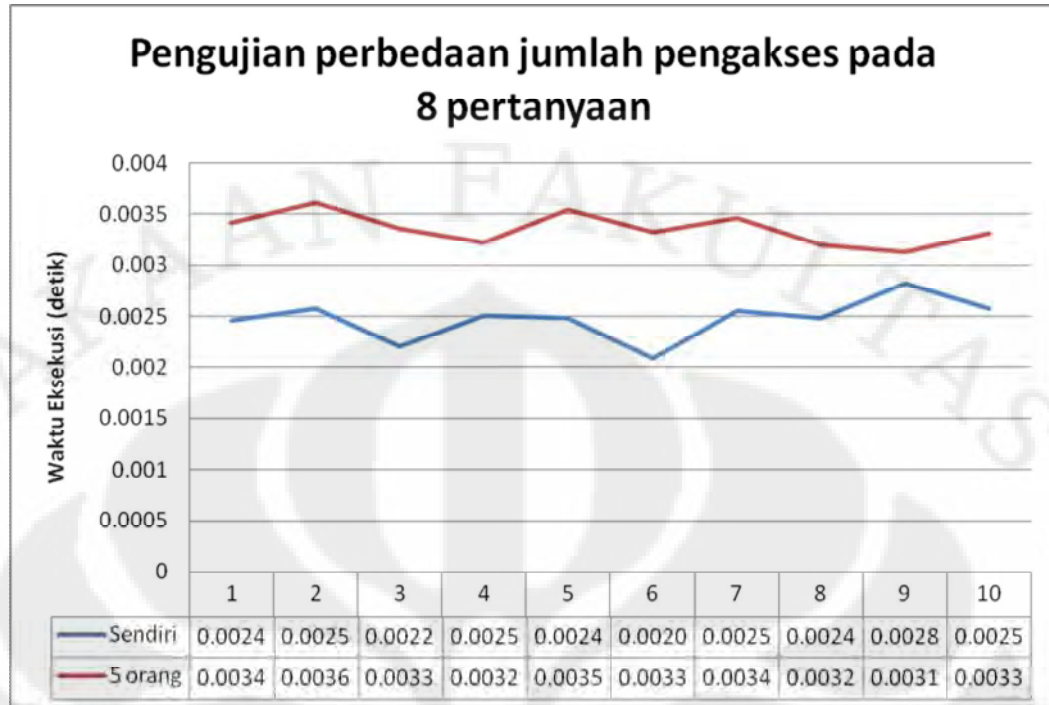
Pada pengujian ini, ingin diketahui tentang perbedaan jumlah pengakses dalam satu waktu dan pengaruhnya pada unjuk kerja server. Pengujian dilakukan dengan mengukur waktu eksekusi server untuk menampilkan hasil diagnosa dan membandingkan kinerja server apabila sistem diakses oleh satu orang pengguna dan lima orang pengguna secara bersama-sama. Pengujian dilakukan dengan mengambil keluhan utama yang sama pada pengujian sendiri dan pengujian lima orang secara bersama-sama agar hasil waktu eksekusi yang didapatkan murni dikarenakan jumlah pengakses yang berbeda dan bukan karena jumlah pertanyaan gejala tambahan yang berbeda.



Gambar 4.10 Grafik pengujian perbedaan jumlah pengakses pada suatu waktu.

Berdasarkan gambar 4.10, dapat dilihat bahwa waktu eksekusi yang ditunjukkan pada saat pengaksesan secara berbarengan oleh lima orang lebih tinggi dibandingkan apabila pengaksesan dilakukan secara sendirian. Hal ini menunjukkan bahwa jumlah pengakses pada suatu waktu mempengaruhi kinerja server dan memperlambat waktu eksekusi. Pada titik dua dan titik tujuh hasil yang didapatkan sangat besar dan jauh berbeda dengan waktu pengaksesan pada saat sendirian, hal ini dikarenakan mungkin pada titik-titik tersebut nilai dari pengaksesan tiap orang dari lima orang yang mengakses secara bersama-sama menghasilkan nilai yang besar, sehingga akan menghasilkan nilai yang besar pula untuk nilai rata-ratanya.

Waktu eksekusi rata-rata dari pengaksesan secara sendirian adalah sebesar 0.00235641 detik, sedangkan waktu eksekusi rata-rata dari pengaksesan lima orang yang dilakukan secara bersama-sama adalah 0.004922581 detik. Berdasarkan hal ini dapat dilihat bahwa meningkatnya jumlah pengakses dari satu orang menjadi lima orang secara bersama-sama membuat waktu eksekusi lebih lambat 109 persen.



Gambar 4.11 Grafik pengujian perbedaan jumlah pengakses pada suatu waktu dengan delapan pertanyaan.

Pada gambar 4.11 menunjukkan perbandingan jumlah pengakses pada titik delapan pertanyaan gejala tambahan. Berdasarkan gambar 4.11 dapat dilihat bahwa waktu eksekusi yang dibutuhkan oleh server saat diakses oleh lima orang secara bersamaan lebih lama dibandingkan bila server diakses oleh satu orang. Pengaksesan server secara bersamaan oleh lima orang secara bersamaan menyebabkan waktu eksekusi lebih lambat 26,25 persen dibandingkan apabila server hanya diakses oleh satu orang.

4.2. Tanggapan Pengguna Terhadap Sistem Pakar Pendiagnosa Penyakit.

Untuk mengetahui tanggapan pengguna tentang sistem pakar pendiagnosa penyakit, dilakukan kuisisioner dengan enam buah pertanyaan. Kuisisioner diajukan kepada dua belas orang secara acak yang berasal dari latar pendidikan yang berbeda. Berikut tabel yang memuat parameter pertanyaan dan hasilnya.

Tabel 4.3 Tanggapan pengguna dengan sistem pakar pendiagnosa penyakit berbasis web

No	Parameter	Jumlah Responder terhadap nilai					STDEV	Rata-rata	Optimis	Pesimis
		1	2	3	4	5				
1	tampilan sudah cukup baik	0	3	4	5	0	2.302173	3.17	4.47	1.87
2	baru pertama kali melihat sistem seperti ini	1	1	1	2	7	2.607681	4.08	5.56	2.6
3	sistem ini memudahkan anda	0	0	5	6	1	2.880972	3.7	5.33	2.07
4	pertanyaan yang diajukan relevance	0	2	4	4	1	1.788854	3.08	4.09	2.07
5	aplikasi ini diperlukan	0	0	3	7	2	2.880972	3.92	5.55	2.29
6	sistem ini perlu dikembangkan lebih lanjut	0	0	0	6	6	3.286335	4.5	6.36	2.64

Dari tabel 4.3, untuk pertanyaan pertama yaitu apakah tampilan pada sistem ini sudah cukup baik terdapat tiga responden yang menjawab kurang setuju, lima responden menjawab setuju, dan empat responden yang menjawab biasa saja.

Untuk pertanyaan kedua yaitu apakah sistem ini merupakan sistem pendiagnosa penyakit berbasis web yang pertama kali anda gunakan terdapat satu responden yang menjawab tidak setuju, satu responden menjawab biasa saja, satu responden menjawab kurang setuju, dua responden setuju, dan tujuh responden menyatakan sangat setuju.

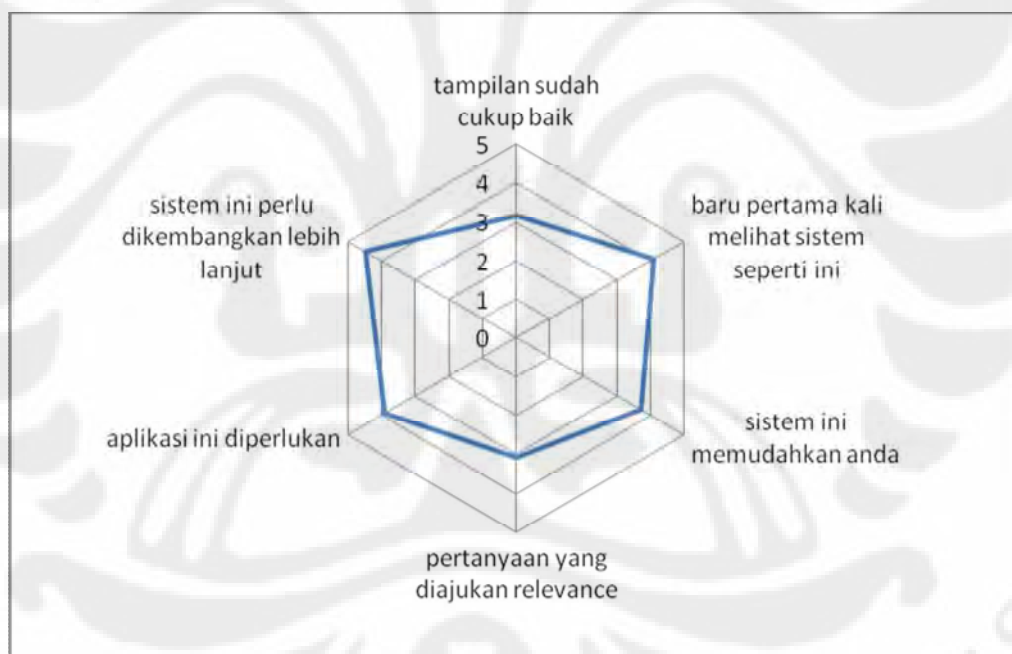
Mengenai pertanyaan apakah sistem pakar ini memudahkan anda untuk mengetahui penyakit yang anda derita, terdapat satu responden menjawab sangat setuju, enam responden menjawab setuju, lima responden menjawab biasa saja.

Pada pertanyaan apakah pertanyaan yang diajukan cukup relevance dengan penyakit yang anda derita terdapat dua responden menjawab tidak setuju, empat responden menjawab setuju, empat responden menjawab biasa saja, dan satu responden menjawab sangat setuju.

Tentang pertanyaan apakah aplikasi ini diperlukan terdapat tiga responden menjawab biasa saja, tujuh responden setuju, dan dua responden menjawab sangat setuju.

Pertanyaan terakhir yang mengenai apakah sistem ini perlu dikembangkan lebih lanjut enam orang responden menjawab setuju dan enam orang responden menjawab sangat setuju.

Dari keseluruhan pertanyaan yang diajukan kepada responden, dapat disimpulkan bahwa tampilan sistem masih perlu pembenahan agar nyaman dilihat, sistem seperti ini merupakan sistem pendiagnosa penyakit berbasis web yang pertama kali digunakan, sistem ini dapat membantu untuk memudahkan pengguna dalam mendiagnosa penyakit, pertanyaan yang diajukan oleh sistem kepada pengguna tentang gejala tambahan sudah cukup *relevance* dengan penyakit yang diderita, aplikasi seperti ini sangat diperlukan, dan sistem ini harus dikembangkan lebih lanjut. Gambar 4.11 menunjukkan diagram radar dari hasil tanggapan pengguna pada sistem ini.



Gambar 4.12 Grafik tanggapan pengguna dengan sistem pakar pendiagnosa penyakit berbasis web.

BAB 5

KESIMPULAN

1. Semakin spesifik keluhan utama yang dimasukkan pengguna pada sistem ini maka akan semakin sedikit pertanyaan pada gejala tambahan yang diajukan oleh sistem ini.
2. Pemilihan jawaban “Ya” dan “Tidak” pada pertanyaan gejala tambahan tidak mempengaruhi lama waktu eksekusi pada server.
3. Banyak pertanyaan yang ada pada gejala tambahan mempengaruhi waktu eksekusi pada server.
4. Jawaban “Ya” dan “Tidak” pada pertanyaan gejala tambahan tidak mempengaruhi banyaknya pertanyaan gejala tambahan.
5. Dari kuisioner yang diberikan kepada responden dapat diketahui bahwa sistem pakar pendiagnosa penyakit berbasis web masih sangat jarang ditemui, terlihat dari pengisian kuisioner yang menyebutkan bahwa responden baru pertama kali menggunakan sistem pendiagnosa penyakit berbasis seperti sistem ini.
6. Antara nilai 1-5 didapatkan rata-rata sekitar 4.17 bahwa responden yang menggunakan sistem ini menganggap hasil diagnosa yang diberikan oleh sistem ini sudah cukup akurat.
7. Sistem ini dapat berjalan dengan baik sebagaimana tujuan awal penyusunan skripsi ini dan dapat dilihat dari tanggapan dari para responder bahwa sistem ini diperlukan.

DAFTAR ACUAN

- [1] “In Absentia Health Care”. Wikipedia
http://en.wikipedia.org/wiki/In_absentia_health_care, terakhir diakses pada tanggal 13 april 2009.
- [2] “Telemedicine”. Wikipedia
<http://en.wikipedia.org/wiki/Telemedicine>, terakhir diakses pada tanggal 13 april 2009.
- [3] “Sistem Pakar”. Wikipedia
http://id.wikipedia.org/wiki/Sistem_pakar, terakhir diakses pada tanggal 4 mei 2009.
- [4] Tolle, Herman, *Pengantar sistem pakar*.
- [5] “Apache HTTP Server”, Wikipedia.
http://id.wikipedia.org/wiki/Apache_HTTP_Server, terakhir diakses tanggal 17 april 2009.
- [6] “Mysql”, Wikipedia.
<http://id.wikipedia.org/wiki/MySQL>, terakhir diakses pada tanggal 17 april 2009.
- [7] “PHP”, Wikipedia.
<http://en.wikipedia.org/wiki/PHP>, terakhir diakses pada tanggal 17 april 2009.
- [8] Dharwiyanti, Sri, *Pengantar unified modelling language*. 2003.
- [9] Takada, Nobuhiko, & Fukuda, Hirokatsu, *Dasar-dasar teori basis data*, JICA, 2004.

DAFTAR PUSTAKA

- Kadir, Abdul, *Dasar pemrograman web dinamis menggunakan PHP*, Andi, 2002.
- Subakti, Irfan, & Hidayatullah, Rahmat, *Aplikasi Sistem pakar untuk diagnosis awal gangguan kesehatan secara mandiri menggunakan variable-centered intelegent rule system*.
- Nafisah, Syifaun, & Effendy, Nazrul, *Implementasi pakar dalam bidang farmakologi dan terapi sebagai pendukung pengambilan keputusan berbasis web*.
- “Pengenalan Sistem Pakar”. Iputusandika <http://iputusandika.net/archives/12>, terakhir diakses pada tanggal 4 mei 2009.
- “Telemedis”. Wikipedia <http://id.wikipedia.org/wiki/Telemedis>, terakhir diakses pada tanggal 15 juni 2009.
- Mansjoer, Arif, *Kapita selekta kedokteran, jilid 1*, FKUI, 1999.