

SKRIPSI

SISTEM PAKAR UNTUK MENDIAGNOSA PENYAKIT YANG DISEBABKAN NYAMUK MENGGUNAKAN *METODE CERTAINTY*



Disusun Oleh :

CUPLIK WAHYU RATRI TRI PAMUNGKAS

NIM 04.12.673

**JURUSAN TEKNIK ELEKTRO S-1
KONSENTRASI TEKNIK KOMPUTER DAN INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
2010**



LEMBAR PERSETUJUAN

**SISTEM PAKAR UNTUK MENDIAGNOSA PENYAKIT YANG
DISEBABKAN NYAMUK MENGGUNAKAN METODE CERTAINTY**

SKRIPSI

*Disusun dan Diajukan sebagai Salah Satu Syarat Untuk Memperoleh
Gelar Sarjana Teknik Komputer dan Informatika Strata Satu (S-1)*

Disusun Oleh :

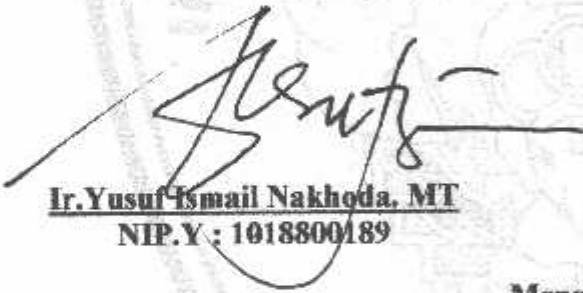
CUPLIK WAHYU RATRI TRI PAMUNGKAS

NIM : 04.12.673

Diperiksa dan Disetujui

Dosen Pembimbing I

Dosen Pembimbing II


Ir. Yusuf Ismail Nakhoda, MT
NIP.Y : 1018800189


Dr. Eng. Aryuanto Soetejo, ST, MT
NIP.Y : 1030800417

Mengetahui

Ketua Jurusan Teknik Elektro S-1




Ir. F. Yudi Limpraptono, MT
NIP Y. 1039500274

**JURUSAN TEKNIK ELEKTRO S-1
KONSENTRASI TEKNIK KOMPUTER DAN INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
2010**

ABSTRAK

SISTEM PAKAR UNTUK MENDIAGNOSA PENYAKIT YANG DISEBABKAN NYAMUK MENGGUNAKAN METODE CERTAINTY.

Cuplik Wahyu Ratri Tri Pamungkas
Jurusan Teknik Elektro S-1, Konsentrasi Teknik Komputer dan Informatika
Fakultas Teknologi Industri
Institut Teknologi Nasional Malang.

Pembimbing : (1) Ir. Yusuf Ismail Nakhoda, MT (2) Dr. Eng. Aryuanto Soetejo, ST, MT

Bertambahnya jumlah penduduk mempercepat terjadinya penularan penyakit dari satu orang ke orang yang lain. Faktor pertumbuhan dan *mobilitas* penduduk ini juga dapat mempengaruhi perubahan gambaran *epidemiologis* dari penyakit bersumber nyamuk. *Epidemiologis* merupakan studi *sistematis* yang dilakukan untuk mempelajari fakta – fakta yang berperan atau mempengaruhi kejadian dan perjalanan suatu penyakit atau kondisi tertentu yang menimpa masyarakat.

Sistem pakar Untuk Mendiagnosa Penyakit Yang Disebabkan Nyamuk merupakan suatu aplikasi sistem pakar yang digunakan untuk mendiagnosa penyakit yang disebabkan nyamuk. Pemangunan aplikasi ini menggunakan Borland Delphi 7 dan MySQL sebagai *database*. Aplikasi ini menggunakan *metode certainty* atau faktor kepastian dalam pengambilan keputusan.

Diharapkan aplikasi ini dapat mempermudah *user* dalam mendiagnosa penyakit yang disebabkan nyamuk, dan dapat memperoleh hasil diagnosa yang benar dan akurat.

Kata kunci : Nyamuk, CF, Sistem Pakar.

Penulis menyadari bahwa skripsi ini masih banyak yang perlu disempurnakan. Oleh sebab itu kritik dan saran yang membangun sangat diharapkan.

Akhir kata, Penulis mohon maaf kepada semua pihak apabila selama penyusunan skripsi ini Penulis secara tidak sengaja telah membuat kesalahan. Semoga skripsi ini dapat memberikan manfaat bagi kita semua.

Malang, Februari 2010

Penulis

DAFTAR ISI

LEMBAR PERSETUJUAN.....	i
ABSTRAK	ii
KATA PENGANTAR.....	iii
DAFTAR ISI.....	v
DAFTAR GAMBAR.....	ix
DAFTAR TABEL	xii

BAB I

PENDAHULUAN	1
1.1. Latar Belakang.....	1
1.2. Rumusan Masalah.....	2
1.3. Batasan Masalah	2
1.4. Tujuan	2
1.5. Metodologi.....	3
1.5.1. Metode pengumpulan data	3
1.5.2. Metode pengembangan sistem pakar.....	4
1.5.3. Pengujian sistem.....	4
1.6. Sistematika Penulisan	5

BAB II

DASAR TEORI.....	6
2.1. Kecerdasan Buatan.....	6

2.2. Sistem Pakar.....	7
2.2.1. Konsep sistem pakar.....	9
2.2.2. Struktur sistem pakar.....	10
2.3. Faktor Kepastian.....	12
2.4. Gangguan Kesehatan.....	14
2.5. Borland Delphi 7.....	15
2.6. MySQL.....	16
BAB III	
ANALISA DAN PERANCANGAN SISTEM	20
3.1. Analisa Sistem.....	20
3.1.1. Analisa Kebutuhan.....	20
3.1.2. Spesifikasi Kebutuhan.....	22
3.1.3. Data Flow Diagram.....	24
3.1.3.1. Diagram Konteks.....	24
3.1.3.2. DFD Level 1.....	26
3.2. Perancangan Perangkat Lunak.....	28
3.2.1. Perancangan Sistem.....	28
3.2.1.1. Diagram Blok Sistem.....	28
3.2.2. Perancangan Basis Data.....	29
3.2.2.1. Hubungan antar Entitas pembentukan sistem dalam Diagram-ER.....	29
3.2.2.2. Relasi Antar Tabel.....	30
3.2.2.3. Struktur Tabel Database.....	31

3.2.3. Desain Modul.....	33
3.2.4. <i>Flowchart</i>	37

BAB IV

IMPLEMENTASI DAN PENGUJIAN SISTEM.....	38
4.1. Implementasi Sistem.....	38
4.1.1. Menu Utama	38
4.1.2. Menu <i>user</i>	39
4.1.3. Menu <i>Administrator Area</i>	40
4.1.4 Menu Laporan.....	40
4.2. Pengujian Sistem.....	41
4.2.1. Pengujian <i>Login</i>	41
4.2.2. Pengujian Konsultasi penyakit.....	42
4.2.3. Pengujian Informasi Penyakit.....	44
4.2.4. Pengujian <i>Logout</i>	45
4.3. Pengujian <i>menu Administrator Area</i>	45
4.3.1. Pengujian <i>menu User Account</i>	45
4.3.2. Pengujian Data Penyakit.....	47
4.3.3. Pengujian Data Gejala	48
4.3.4. Pengujian Data Diagnosa.....	49
4.4. Pengujian Menu Laporam.....	51
4.5. Pengujian Sistem Pakar dengan Pakar dan Sumber Lain	52

BAB V

PENUTUP..... 55

5.1. Kesimpulan 55

5.2. Saran 56

Daftar Pustaka

Lampiran

DAFTAR GAMBAR

BAB II DASAR TEORI

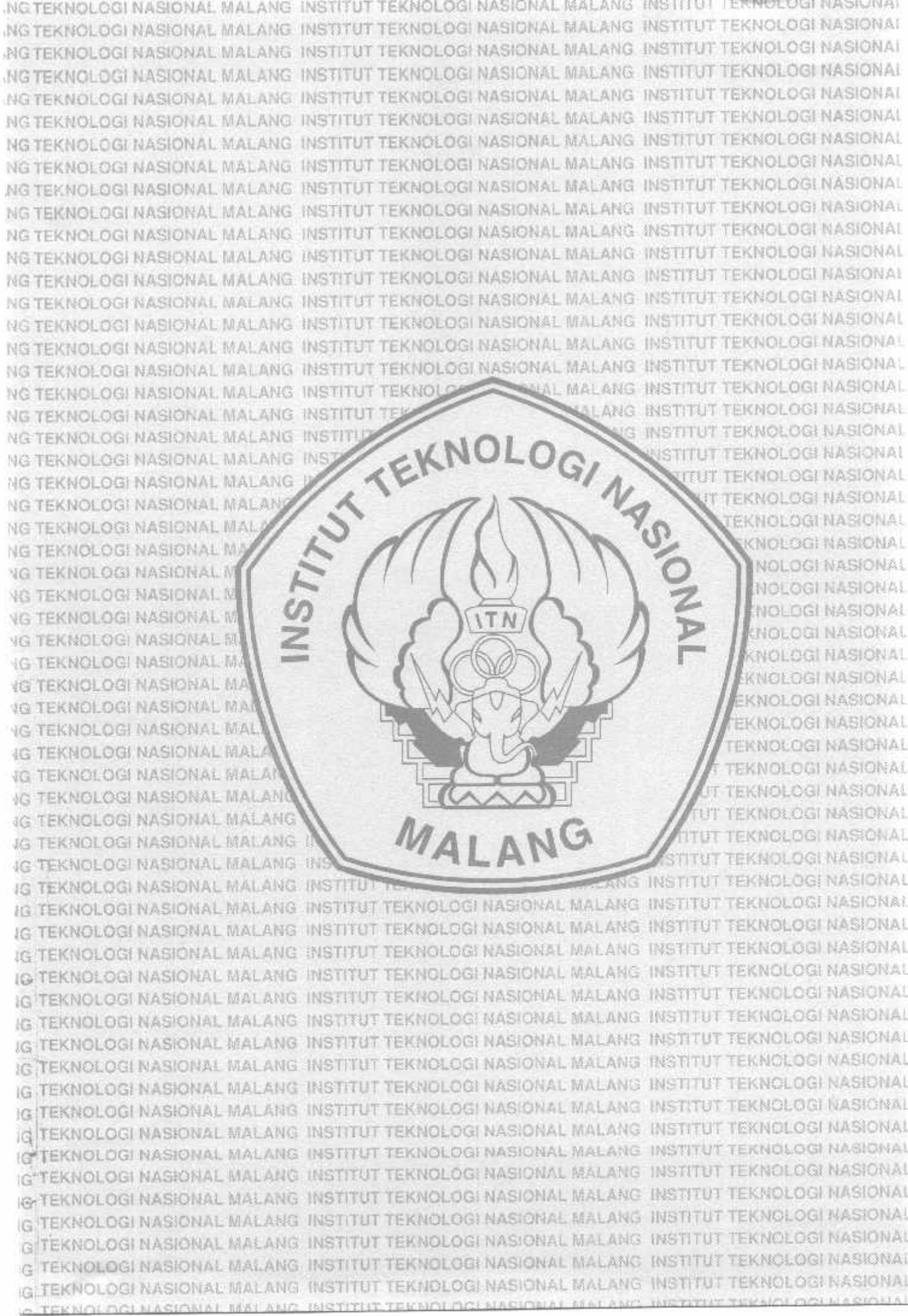
Gambar 2.1. Konsep Dasar Fungsi Sistem Pakar	10
Gambar 2.2. Stuktur Sistem Pakar.....	11
Gambar 2.3. Proses <i>Forward Chaining</i>	12
Gambar 2.4. Proses <i>Backward Chaining</i>	12

BAB III ANALISA DAN PERANCANGAN SISTEM

Gambar 3.1. Diagram pohon perancangan	20
Gambar 3.2. Diagram konteks sistem pakar untuk diagnosa penyakit yang disebabkan nyamuk.....	24
Gambar 3.3. DFD Level 1 Sistem Pakar untuk Mendiagnosa penyakit Yang disebabkan nyamuk.....	26
Gambar 3.4. Diagram Blok Sistem Pakar untuk Mendiagnosa penyakit Yang disebabkan nyamuk.....	29
Gambar 3.5. ERD Sistem Pakar untuk Mendiagnosa penyakit yang disebabkan nyamuk.....	30
Gambar 3.6. Conceptual Data Model (CDM).....	31
Gambar 3.7. Physical Data Model (PDM).....	31
Gambar 3.8. Bagan sistem pakar untuk <i>administrator</i>	35
Gambar 3.9. Bagan sistem pakar untuk Pakar.....	36
Gambar 3.10. Bagan sistem pakar untuk Pasien.....	36
Gambar 3.11. Algoritma proses diagnosa.....	37

BAB IV IMPLEMENTASI DAN PENGUJIAN SISTEM

Gambar 4.1. Menu Utama.....	39
Gambar 4.2. Gambar Menu <i>User</i>	39
Gambar 4.3. Gambar Menu <i>Administrator</i>	40
Gambar 4.4. Gambar Menu Laporan	40
Gambar 4.5. <i>Form Login</i>	41
Gambar 4.6. Menu <i>Adminidtrator</i> & laporan dapat diakses.....	42
Gambar 4.7. <i>Form</i> Data Gejala untuk <i>User</i>	42
Gambar 4.8. <i>Form</i> Konsultasi.....	43
Gambar 4.9. <i>Form</i> Konsulatasi & hasil Diagnosa	44
Gambar 4.10. <i>Form Informasi Penyakit</i>	44
Gambar 4.11. Konfirmasi <i>Logout</i>	45
Gambar 4.12. <i>Form User Account</i>	46
Gambar 4.13. Penyimpanan Data <i>User</i>	46
Gambar 4.14. <i>Form</i> Data Penyakit	47
Gambar 4.15. Penyimpanan Data Penyakit	48
Gambar 4.16. <i>Form</i> Data Gejala.....	48
Gambar 4.17. Penyimpanan Data Gejala.....	49
Gambar 4.18. <i>Form</i> Data Diagnosa	50
Gambar 4.19. Penyimpanan Data Diagnosa	50
Gambar 4.20 Laporan Data <i>User</i>	51
Gambar 4.21 Laporan Data Penyakit.....	51
Gambar 4.22. Laporan Data Gejala	52
Gambar 4.23. Laporan Data Diagnosa.....	52



BAB 1

PENDAHULUAN

1.1 Latar Belakang

Bertambahnya jumlah penduduk mempercepat terjadinya penularan penyakit dari satu orang ke orang yang lain. Faktor pertumbuhan dan *mobilitas* penduduk ini juga dapat mempengaruhi perubahan gambaran *epidemiologis* dari penyakit bersumber nyamuk. *Epidemiologis* merupakan studi *sistematis* yang dilakukan untuk mempelajari fakta – fakta yang berperan atau mempengaruhi kejadian dan perjalanan suatu penyakit atau kondisi tertentu yang menimpa masyarakat.

Dalam hal ini, masyarakat pasti mengakui bahwa nyamuk merupakan serangga pengganggu. Namun, kebanyakan orang tidak menyadari seberapa besar *species – species* nyamuk tertentu di sekitarnya yang menimbulkan resiko penularan penyakit yang berbahaya. Padahal, kenyataan menunjukkan bahwa gangguan kesehatan yang diakibatkan oleh nyamuk muncul karena kontak manusia dengan nyamuk *vektor yang infeksiosa*, maka upaya – upaya pengendalian penyakit yang disebabkan nyamuk harus dilakukan dengan cara pengelolaan lingkungan atau dengan cara yang telah dikenal masyarakat seperti gerakan 3M (menutup, menguras, menutup/ menimbun semua tempat yang memungkinkan menjadi tempat perkembang biakan nyamuk). Hal ini juga memerlukan keterlibatan seorang pakar untuk mengenali segala penyakit yang disebabkan oleh nyamuk.

Salah satu bidang ilmu yang termasuk dalam kecerdasan buatan adalah sistem pakar (*Expert System*). Sistem pakar merupakan perangkat lunak yang mendukung pemecahan masalah dengan rumen yang menirukan pemikiran seorang pakar. Ide dasarnya adalah kepakaran ditransfer dari seorang pakar ke komputer yang khusus ditujukan dalam

perancangan otomatis tingkah laku cerdas dalam sistem kecerdasan komputer yang disimpan dalam *database* komputer untuk mencari solusi.

Berdasarkan uraian permasalahan diatas, penulis tertarik membangun suatu sistem pakar untuk mendiagnosa penyakit yang disebabkan nyamuk, serta melakukan penelitian dan menuangkannya dalam penulisan ilmiah yang berjudul "Sistem Pakar Untuk Mendiagnosa Penyakit Yang Disebabkan Nyamuk Menggunakan *Metode Certainty*".

1.2 Rumusan Masalah

Dari permasalahan diatas, dapat dirumuskan suatu permasalahan yaitu bagaimana menerapkan dengan *intelegeni* para ahli atau pakar dalam menganalisa, mengidentifikasi dan menerapkan *system* pakar melalui metode *system* pakar untuk menganalisa dan mengidentifikasi penyakit yang disebabkan oleh nyamuk dengan menggunakan *metode certainty*.

1.3 Tujuan

Tujuan dari skripsi ini adalah mengembangkan sistem pakar untuk mendiagnosa penyakit yang disebabkan nyamuk agar dapat mempermudah kinerja pakar dalam mendiagnosa penyakit sehingga dapat dihasilkan data yang terstruktur dan *akurat*.

1.4 Batasan Masalah

Batasan masalah yang digunakan diharapkan mampu membatasi pembahasan agar sesuai dengan tujuan pembuatan sistem itu sendiri. Adapun batasan masalah yang diajukan adalah sebagai berikut :

1. Jenis penyakit ada 5 dengan beberapa gejala yang ada.
2. Penelitian ini tidak membahas tentang jenis – jenis nyamuk yang menyebabkan penyakit.
3. Penelitian ini menggunakan teks pada proses *input* maupun *output* untuk mendeteksi penyakit yang ditimbulkan oleh gigitan nyamuk.
4. Penelitian ini hanya membahas identifikasi penyakit yang ditimbulkan oleh gigitan nyamuk.
5. Dalam merancang program ini menggunakan *metode certainty*.
6. Aplikasi program menggunakan Borland Delphi 7.0.
7. Pengguna aplikasi ini hanya kalangan tertentu yang memiliki hak akses terhadap program.

1.5 Metodologi

1.5.1 Metode pengumpulan data

Data merupakan sumber atau bahan mentah yang sangat berharga bagi proses menghasilkan informasi. Oleh sebab itu dalam pengambilan data perlu dilakukan penanganan secara cermat dan hati-hati, sehingga data yang diperoleh dapat bermanfaat dan berkualitas.

Dalam pengumpulan data penyusun menggunakan metode sebagai berikut :

1. Studi Lapangan / *Survey*

Teknik pengumpulan data dengan cara terjun secara langsung dan melihat secara sistematis terhadap obyek masalah.

2. Studi Pustaka / *Literatur*

Pengumpulan data ini dilakukan dengan cara mencari bahan-bahan kepustakaan sebagai landasan teori yang ada hubungannya dengan permasalahan yang dijadikan obyek penelitian.

3. Wawancara / *Interview*

Teknik pengumpulan data dengan mengadakan komunikasi atau tanya jawab secara langsung dengan dokter.

1.5.2. Metode pengembangan sistem pakar

Menentukan proses kerja sistem pakar untuk mendiagnosa penyakit yang disebabkan oleh gigitan nyamuk dengan menggunakan faktor kepastian.

Adapun proses kerja dari sistem pakar ini terbagi atas :

1. Proses *Login*

Proses ini merupakan proses untuk mendapatkan identifikasi yang telah diberikan oleh *user* (pakar). Identifikasi yang diberikan adalah *username* dan *password*.

2. Proses Konsultasi

Proses ini merupakan proses untuk memasukkan data gejala dari *user* dan sistem akan memberikan data gangguan dan penyakit dari data gejala yang dimasukkan tersebut serta pengobatan.

3. Proses *Maintenance Data*

Proses ini merupakan proses untuk memasukan data dan memperbarui data.

1.5.3. Pengujian sistem

Pengujian sistem adalah metode desain tes kasus yang menggunakan struktur kontrol desain prosedural untuk menjalankan sistem. Dengan menggunakan metode pengujian, sistem pakar dapat dilakukan rekayasa tes kasus yang diantaranya:

1. Dioperasikan asisten dokter atau orang yang mengerti sedikit banyak tentang penyakit yang disebabkan nyamuk.
2. Memberikan jaminan bahwa semua jalur independen pada suatu modul telah digunakan paling tidak satu kali.

1.6 Sistematika Penulisan

Pada penulisan skripsi ini terdiri atas lima pembahasan, yaitu Pendahuluan, Tinjauan Pustaka, Perancangan dan Pembuatan Sistem, Hasil dan Analisa serta Penutup. Secara singkat, kelima bagian tersebut adalah sebagai berikut :

1. BAB I : PENDAHULUAN

Bab ini berisi tentang latar belakang, tujuan, permasalahan, batasan masalah, dan sistematika pembahasan dari skripsi ini.

2. BAB II : TINJAUAN PUSTAKA

Bab ini berisi penjelasan tentang tinjauan umum Sistem Pakar Diagnosa Penyakit dan teori-teori yang mendukung dalam perancangan dan pembuatan sistem yang meliputi *Borland Delphi 7.0, Basis Data dan MySQL*.

3. BAB III : PERANCANGAN DAN PEMBUATAN SISTEM

Bab ini berisi tentang perencanaan dan pembuatan Sistem Pakar Diagnosa Penyakit dan menjelaskan metode yang dilakukan mulai dari perancangan database sampai *design user interface*.

4. BAB IV : HASIL DAN ANALISA

Bab ini berisi pembahasan hasil pengujian dan analisa mengenai cara kerja dari sistem.

5. BAB V : PENUTUP

Bab ini berisi kesimpulan dan saran dari hasil pembahasan pada skripsi ini.

BAB II

DASAR TEORI

2.1 Kecerdasan Buatan

Kecerdasan buatan merupakan salah satu bagian ilmu komputer yang mempelajari bagaimana cara membuat mesin (komputer) agar dapat melakukan pekerjaan seperti yang dilakukan oleh manusia.

Kecerdasan Buatan (*Artificial Intelligence* atau *AI*) didefinisikan sebagai kecerdasan yang ditunjukkan oleh suatu entitas buatan. Sistem seperti ini umumnya dianggap komputer. Kecerdasan diciptakan dan dimasukkan ke dalam suatu mesin (komputer) agar dapat melakukan pekerjaan seperti yang dapat dilakukan manusia. Beberapa macam bidang yang menggunakan kecerdasan buatan antara lain sistem pakar, permainan komputer, logika fuzzy, jaringan syaraf tiruan dan robotika.

Banyak hal yang kelihatannya sulit untuk kecerdasan manusia, tetapi untuk Informatika relatif tidak bermasalah. Seperti contoh: mentransformasikan persamaan, menyelesaikan persamaan integral, membuat permainan catur atau Backgammon. Di sisi lain, hal yang bagi manusia kelihatannya menuntut sedikit kecerdasan, sampai sekarang masih sulit untuk direalisasikan dalam informatika. Seperti contoh: Pengenalan Obyek/Muka, bermain Sepakbola.

Walaupun *AI* memiliki konotasi fiksi ilmiah yang kuat, *AI* membentuk cabang yang sangat penting pada ilmu komputer, berhubungan dengan perilaku, pembelajaran dan adaptasi yang cerdas dalam sebuah mesin. Penelitian dalam *AI* menyangkut pembuatan mesin untuk mengotomatisasikan tugas-tugas yang membutuhkan perilaku

cerdas. Termasuk contohnya adalah pengendalian, perencanaan dan penjadwalan, kemampuan untuk menjawab diagnosa dan pertanyaan pelanggan, serta pengenalan tulisan tangan, suara dan wajah. Hal-hal seperti itu telah menjadi disiplin ilmu tersendiri, yang memusatkan perhatian pada penyediaan solusi masalah kehidupan yang nyata. Sistem *AI* sekarang ini sering digunakan dalam bidang ekonomi, obat-obatan, teknik dan militer, seperti yang telah dibangun dalam beberapa aplikasi perangkat lunak komputer rumah dan video game ^[5].

2.2 Sistem Pakar

Sistem pakar adalah salah satu cabang dari kecerdasan buatan yang membuat penggunaan secara luas *knowledge* yang khusus untuk penyelesaian masalah tingkat manusia yang pakar. Seorang pakar adalah orang yang mempunyai keahlian dalam bidang tertentu, yaitu pakar yang mempunyai *knowledge* atau kemampuan khusus yang orang lain tidak mengetahui atau mampu dalam bidang yang dimilikinya tersebut.

Pengetahuan dalam sistem pakar mungkin saja seorang ahli, atau pengetahuan yang umumnya terdapat dalam buku, majalah dan orang yang mempunyai pengetahuan tentang suatu bidang ^[4].

Seorang pakar dengan sistem pakar mempunyai banyak perbedaan. Perbandingan kemampuan antara kemampuan seorang pakar dengan sebuah sistem pakar dapat dilihat dalam Tabel 2.1 berikut ini:

Tabel 2.1. Perbandingan Kemampuan Seorang Pakar dengan Sistem Pakar^[6].

Faktor	Seorang Pakar	Sistem Pakar
Time availability	Hari kerja	Setiap saat
Geografis	Lokal/tertentu	Dimana saja
Keamanan	Tidak tergantung	Dapat digantikan
<i>Perishable</i> /dapat habis	Ya	Tidak
Perfomansi	Berubah-ubah	Tetap
Kecepatan	Berubah-ubah	Tetap
Biaya	Tinggi	Terjangkau

Manfaat dari sistem pakar yaitu :

1. Bisa melakukan proses secara berulang secara otomatis
2. Menyimpan pengetahuan dan keahlian para pakar
3. Mampu mengambil dan melestarikan keahlian para pakar (terutama yang termasuk keahlian langka)
4. Mampu beroperasi dalam lingkungan yang berbahaya
5. Tidak memerlukan biaya saat tidak digunakan, sedangkan pada pakar manusia memerlukan biaya sehari-hari.
6. Dapat digandakan (diperbanyak) sesuai kebutuhan dengan syarat meminta ijin pada para medis yang punya aplikasi ini.
7. Dapat memecahkan masalah lebih cepat daripada kemampuan manusia dengan catatan menggunakan data yang sama.
8. Menghemat waktu dalam pengambilan keputusan

9. Meningkatkan kualitas dan produktivitas

Adapun kelemahan sistem pakar yaitu :

1. Biaya yang diperlukan untuk membuat, memelihara, dan mengembangkannya sangat mahal
2. Sulit dikembangkan, hal ini erat kaitannya dengan ketersediaan pakar di bidangnya
3. Sistem pakar tidak 100% benar karena seseorang yang terlibat dalam pembuatan sistem pakar tidak selalu benar. Oleh karena itu perlu diuji ulang secara teliti sebelum digunakan

2.2.1 Konsep Sistem Pakar

Konsep dasar sistem pakar terdiri dari beberapa unsur/elemen antara lain :

1. Keahlian

Keahlian adalah suatu kelebihan penguasaan pengetahuan di bidang tertentu yang diperoleh dari pelatihan, membaca atau pengalaman.

2. Ahli

Seorang ahli adalah seseorang yang mampu menjelaskan suatu tanggapan, mempelajari hal-hal baru seputar system permasalahan (*domain*), menyusun kembali pengetahuan jika dipandang perlu, memecahkan aturan-aturan jika dibutuhkan dan menentukan relevan tidaknya keahlian mereka.

3. Pengalihan keahlian

Pengalihan pengetahuan dari para ahli ke sistem untuk kemudian dialihkan lagi ke orang lain yang bukan ahli, merupakan tujuan utama dari sistem pakar.

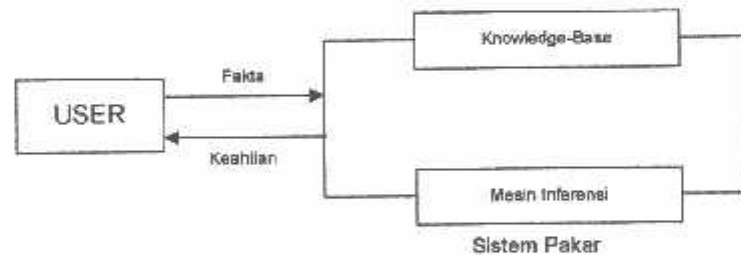
4. Aturan

Sebagian besar sistem pakar komersil dibuat dalam bentuk *rule-based systems* yang mana pengetahuan disimpan dalam aturan-aturan *IF-THEN*

5. Kemampuan menjelaskan

Setiap sistem pakar harus memiliki kemampuan untuk menjelaskan Individu yang terlibat dalam lingkungan pengembangan sistem pakar ada tiga macam yaitu : Pakar, Perakayasa system, Pemakai.

Gambar 2.1 menggambarkan konsep umum suatu sistem pakar. Pengguna menyampaikan fakta atau informasi untuk sistem pakar dan kemudian menerima saran dari pakar atau jawaban. Bagian dalam sistem pakar terdiri dari dua komponen utama, yaitu *knowledge base* yang berisi pengetahuan dan mesin inferensi yang menggambarkan kesimpulan. Kesimpulan tersebut merupakan respons dari sistem pakar atas permintaan user.

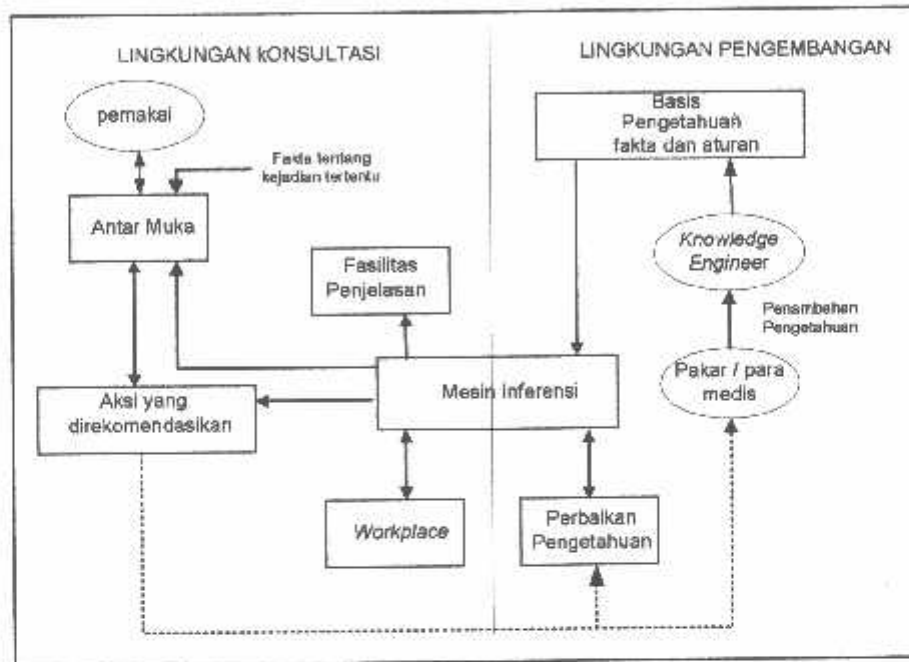


Gambar 2.1. Konsep Dasar Fungsi Sistem Pakar^[4].

2.2.2 Konsep Struktur Pakar

Sistem pakar disusun oleh dua bagian utama, yaitu lingkungan pengembang dan lingkungan konsultasi. Lingkungan pengembangan sistem pakar digunakan untuk memasukkan pengetahuan pakar ke dalam lingkungan

sistem pakar, sedangkan lingkungan konsultasi digunakan oleh pengguna yang bukan pakar untuk memperoleh pengetahuan pakar. Komponen-komponen sistem pakar dalam dua agian tersebut dapat dilihat dalam Gambar 2.2 berikut ini:



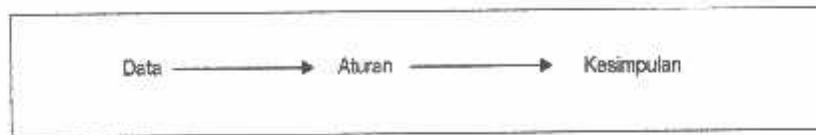
Gambar 2.2. Struktur Sistem Pakar [4]

Komponen-komponen utama yang terdapat dalam sistem pakar seperti dalam Gambar yaitu antarmuka pengguna, basis pengetahuan, akuisisi pengetahuan, mesin inferensi, *workplace*, fasilitas penjelasan, perbaikan pengetahuan.

Ada dua cara yang dapat dikerjakan dalam melakukan inferensi yaitu:

1. *Forward Chaining* (Pelacakan ke depan)

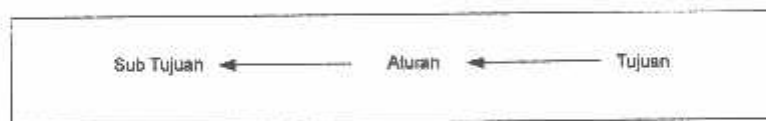
Pada metode ini cocok digunakan untuk menangani masalah pengendalian (*controlling*) dan peramalan (*prognosis*.) Gambar 2.3 menggambarkan proses *forward chaining*.



Gambar 2.3. Proses *Forward Chaining*^[4].

2. Backward Chaining (Pelacakan ke belakang)

Pada metode ini cocok digunakan untuk memecahkan masalah diagnosa. Metode ini merupakan cara efisien untuk memecahkan masalah terstruktur. Tujuan inferensi ini adalah mengambil pilihan terbaik dari banyak kemungkinan. Gambar 2.4 menunjukkan proses *backward chaining*.



Gambar 2.4. Proses *Backward Chaining*^[4].

2.3 Faktor Kepastian

Dalam menghadapi suatu masalah sering ditemukan jawaban yang tidak memiliki kepastian penuh. Ketidakpastian ini dapat berupa probabilitas atau keboleh jadian yang tergantung dari hasil suatu kejadian. Hasil yang tidak pasti disebabkan oleh dua faktor yaitu aturan yang tidak pasti dan jawaban pengguna yang tidak pasti atas suatu pertanyaan yang diajukan oleh sistem. Hal ini sangat mudah dilihat pada sistem diagnosis penyakit dimana pakar tidak dapat mendefinisikan hubungan antara gejala dan

penyebabnya secara pasti dan pasien tidak dapat merasakan suatu gejala dengan pasti pula, pada akhirnya akan ditemukan banyak kemungkinan diagnosa.

Ketidakpastian dapat dianggap sebagai suatu kekurangan informasi yang memadai untuk membuat suatu keputusan. Ketidakpastian merupakan suatu permasalahan karena mungkin menghalangi kita dalam membuat suatu keputusan yang terbaik bahkan mungkin dapat menghasilkan suatu keputusan yang buruk. Dalam dunia medis, ketidakpastian mungkin menghalangi pemeriksaan yang terbaik untuk para pasien dan berperan untuk suatu terapi yang keliru. Dalam bisnis, ketidakpastian dapat berarti kerugian uang.

Faktor kepastian diperkenalkan oleh Shortliffe Buchanan dalam pembuatan MYCIN (Sistem Pakar Medis yang mendiagnosa/memperbaiki infeksi bakteri). Faktor kepastian merupakan nilai parameter klinis yang diberikan MYCIN untuk menunjukkan besarnya kepercayaan. Faktor kepastian didefinisikan sebagai berikut⁽³⁾:

$$CF[H,E] = MB[H,E] - MD[H,E] \quad \dots(1)$$

$CF[H,E]$ = *Certainty Factor* dari hipotesis H yang dipengaruhi oleh gejala (*evidence*) E. besarnya CF berkisar antara -1 sampai dengan 1. Nilai -1 menunjukkan ketidakpercayaan mutlak sedangkan nilai 1 menunjukkan kepercayaan mutlak.

$MB[H,E]$ = ukuran kepercayaan terhadap hipotesis H yang dipengaruhi oleh gejala E.

$MD[H,E]$ = ukuran ketidakpercayaan terhadap hipotesis H yang dipengaruhi oleh gejala E.

Penggabungan kepercayaan dan ketidakpercayaan dalam bilangan yang tunggal memiliki dua kegunaan, yaitu pertama faktor kepastian digunakan untuk tingkatan hipotesis di dalam urutan kepentingan dan yang kedua ada dalam urutan pengujian.

2.4 Gangguan Kesehatan

Pada sub bab ini akan dijelaskan mengenai macam-macam gangguan kesehatan yang dapat didiagnosa oleh sistem pakar ini.

SistemPakar Untuk Mendignosa Penyakit Yang disebabkan Nyamuk Menggunakan Metode Certainty dapat mendiagnosa 5 macam penyakit, yaitu :

a. Demam Berdarah (DBD)

Demam Berdarah Dengue (DBD) atau Hemorrhagic Fever adalah suatu penyakit yang disebabkan oleh virus Dengue Famili Flaviviridae dengan genusnya adalah Flavivirus. Virus ini mempunyai empat serotype yang dikenal dengan DEN – 1, DEN – 2, DEN – 3, DEN – 4. Selama ini secara klinik mempunyai tingkat manifestasi yang berbeda tergantung dari serotype virus Dengue tersebut.

b. Malaria

Malaria adalah penyakit yang disebabkan oleh parasit protozoa yang paling biasa di dunia, yaitu *plasmodium* yang menyumbang kepada lebih kurang 3 juta kes dan 1.5 hingga 2.7 juta kematian setiap tahun. Ia disebarkan oleh nyamuk betina genus *Anopheles* (nyamuk tiruk), terutamanya *Anopheles sudaicus* di Asia dan *An. gambiae* di Afrika. Ramai orang mendapat malaria semasa mengembara ke negara-negara tropika atau subtropika.

c. Japanese Encephalitis

Japanese Encephalitis (JE) adalah penyakit yang disebabkan oleh flavivirus dan disebarkan oleh nyamuk. Penyakit ini menyerang susunan saraf pusat otak (otak, spinalis, dan meninges) yang disebabkan oleh Japanese Encephalitis virus (JEV) yang ditularkan dari binatang melalui gigitan nyamuk culex.

d. Chikungunya

Chikungunya adalah sejenis demam virus yang disebabkan alphavirus yang disebarkan oleh gigitan nyamuk dari spesies *Aedes aegypti*. Namanya berasal dari sebuah kata dalam bahasa Makonde yang berarti "yang melengkung ke atas", merujuk kepada tubuh yang membungkuk akibat gejala-gejala *arthritis* penyakit ini

e. Filariasis

Penyakit Kaki Gajah (*Filariasis* atau *Elephantiasis*) adalah golongan penyakit menular yang disebabkan oleh cacing *Filaria* yang ditularkan melalui berbagai jenis nyamuk. Setelah tergigit nyamuk, parasit (*larva*) akan menjalar dan ketika sampai pada jaringan sistem limpa maka berkembanglah menjadi penyakit tersebut. Penyakit ini bersifat menahun (kronis) dan bila tidak mendapatkan pengobatan, dapat menimbulkan cacat menetap berupa pembesaran kaki, lengan dan alat kelamin baik perempuan maupun laki-laki. Penyakit Kaki Gajah bukanlah penyakit yang mematikan, namun demikian bagi penderita mungkin menjadi sesuatu yang dirasakan memalukan bahkan dapat mengganggu aktifitas sehari-hari.

2.5 Borland Delphi

Delphi adalah kompiler / penterjemah bahasa Delphi (awalnya dari Pascal) yang merupakan bahasa tingkat tinggi sekelas dengan Basic, C. Bahasa Pemrograman di Delphi disebut bahasa prosedural artinya bahasa/sintaknya mengikuti urutan tertentu / prosedur. Ada jenis pemrograman non-prosedural seperti pemrograman untuk kecerdasan

buatan seperti bahasa Prolog. Delphi termasuk Keluarga Visual sekelas Visual Basic, Visual C, artinya perintah-perintah untuk membuat objek dapat dilakukan secara visual. Pemrogram tinggal memilih objek apa yang ingin dimasukkan kedalam Form/Window, lalu tingkah laku objek tersebut saat menerima event/aksi tinggal dibuat programnya. Delphi merupakan bahasa berorientasi objek, artinya nama objek, properti dan metode/procedure dikemas menjadi satu kemasan (*encapsulate*).

Borland Delphi merupakan suatu bahasa pemrograman yang memberikan berbagai fasilitas pembuat aplikasi visual. Keunggulan bahasa pemrograman ini terletak pada produktivitas, kualitas, pengembangan perangkat lunak, kecepatan kompilasi, pola desain yang menarik serta diperkuat dengan pemrogramannya yang terstruktur. Keunggulan lain dari Delphi adalah dapat digunakan untuk merancang program aplikasi yang memiliki tampilan seperti program aplikasi lain yang berbasis windows.

Khusus untuk pemrograman database, Borland Delphi menyediakan fasilitas objek yang kuat dan lengkap yang memudahkan programmer dalam membuat program. Format database yang dimiliki Delphi adalah format database Paradox, dBase, MS. Access, ODBC, SyBASE, Oracle dan lain-lain^[1].

2.6 MySQL

MySQL merupakan *Relational Database Management System* (RDBMS) yang didistribusikan secara gratis di bawah lisensi GPL (*General Public License*). Di mana setiap orang bebas untuk menggunakan *MySQL*, namun tidak boleh dijadikan produk turunan yang bersifat *closed source* atau komersial. *MySQL* sebenarnya merupakan

turunan salah satu konsep utama dalam database sejak lama, yaitu SQL (*Structure Query Language*)^[7].

SQL adalah sebuah konsep pengoperasian *database*, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis. Keandalan suatu *system database* (DBMS) dapat diketahui dari cara kerja *optimizer*-nya dalam melakukan proses perintah-perintah SQL, yang dibuat oleh user maupun program - program aplikasinya. Sebagai *database server*, *MySQL* dapat dikatakan lebih unggul dibandingkan dengan *database server* yang lainnya dalam *query* data^[7].

Sebagai *database* yang memiliki konsep *database* modern, *MySQL* memiliki banyak sekali keistimewaan. Berikut ini beberapa keistimewaan yang dimiliki oleh *MySQL* :

- *Portability*. *MySQL* dapat berjalan stabil pada berbagai sistem operasi di antaranya adalah seperti *Windows*, *Linux*, *FreeBSD*, *Mac OS X server*, *Solaris*, *Amiga*, *HP-UX* dan masih banyak lagi.
- *Open Source*. *MySQL* didistribusikan secara *open source* (gratis), di bawah lisensi GPL.
- *Multuser*. *MySQL* dapat digunakan oleh beberapa *user* dalam waktu yang bersamaan tanpa mengalami masalah atau konflik. Hal ini memungkinkan sebuah *database server MySQL* dapat diakses *client* secara bersamaan.
- *Performance Tuning*. *MySQL* memiliki kecepatan yang menakjubkan dalam menangani *query* sederhana, dengan kata lain dapat memproses lebih banyak SQL per satuan waktu.

- *Column Types.* MySQL memiliki tipe kolom yang sangat kompleks, seperti *signed/unsigned integer, float, double, char, varchar, text, blob, date, time, datetime, year, set* serta *enum*.
- *Command dan Function.* MySQL memiliki operator dan fungsi secara penuh yang mendukung perintah *SELECT* dan *WHERE* dalam *query*.
- *Security.* MySQL memiliki beberapa lapisan sekuritas seperti level *subnetmask*, nama *host*, dan *user* dengan sistem perizinan yang mendetail serta *password* terenkripsi.
- *Stability dan Limits.* MySQL mampu menangani *database* dalam skala besar, dengan jumlah *records* lebih dari 50 juta dan 60 ribu table serta 5 miliar baris. Selain itu, batas *indeks* yang dapat di tampung mencapai 32 *indeks* pada tiap tabelnya.
- *Connectivity.* MySQL dapat melakukan koneksi dengan client menggunakan *protocol* TCP/IP, *Unix socket (Unix)*, atau *Named Pipes (NT)*. 10. *Localisation* MySQL dapat mendeteksi pesan kesalahan (*error code*) pada *client* dengan menggunakan lebih dari dua puluh bahasa. Meski demikian, bahasa Indonesia belum termasuk di dalamnya.
- *Localisation.* MySQL dapat mendeteksi pesan kesalahan (*error code*) pada *client* dengan menggunakan lebih dari dua puluh bahasa. Meski demikian, bahasa Indonesia belum termasuk di dalamnya.
- *Interface.* MySQL memiliki *interface* (antar muka) terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API (*Application Programming Interface*).
- *Client dan Tools.* MySQL dilengkapi dengan berbagai tool yang dapat digunakan untuk administrasi *database*, dan pada setiap tool yang ada disertai petunjuk *online*.

- Struktur Tabel. *MySQL* memiliki struktur tabel yang lebih fleksibel dalam menangani *ALTER TABLE*, dibandingkan *database* lainnya semacam *PostgreSQL* ataupun *Oracle*^[7].



BAB III

ANALISA DAN PERANCANGAN SISTEM

3.1 Analisa Sistem

Analisa sistem dibuat untuk menggambarkan cara kerja program yang dibutuhkan oleh pengguna, untuk membangun dasar bagi pembuatan desain perangkat lunak dan untuk membatasi serangkaian persyaratan yang dapat menunjang begitu perangkat lunak dibuat. Proses analisa menjelaskan mengenai perancangan sistem meliputi perancangan sistem, perancangan desain perangkat lunak dan perancangan desain modul.

Perancangan yang dilakukan dapat digambarkan dengan diagram pohon seperti dalam gambar 3.1 berikut:



Gambar 3.1 Diagram pohon perancangan

3.1.1 Analisa Kebutuhan Perangkat Lunak

Sistem yang dibuat adalah sistem pakar. Sistem pakar berusaha mengadopsi pengetahuan manusia ke komputer agar komputer dapat menyelesaikan masalah seperti yang biasa dilakukan oleh para medis. Pengetahuan dan kemampuan para medis dapat dialihkan ke dalam sistem pakar sehingga sistem pakar tersebut memiliki keahlian menyerupai para medis dalam melakukan diagnosa.

Untuk mendapatkan informasi mengenai gejala gangguan kesehatan yang dialami pasien terdapat sesi konsultasi yang merupakan sesi tanya-jawab antara pengguna dan sistem pakar. Dari informasi tersebut dapat ditentukan jenis gangguan kesehatan yang diderita oleh pasien dan tindakan apa yang perlu dilakukan untuk mengatasi gangguan kesehatan tersebut.

Informasi yang diberikan pada sistem pakar adalah informasi mengenai gejala fisik yang dialami oleh pasien. Terkadang pasien sulit untuk menggambarkan apa yang dirasakannya sehingga sangat memungkinkan kehilangan beberapa informasi. Untuk itulah diberikan faktor kepastian (*certainly factor*) untuk mengatasi ketidakpastian data.

Kebutuhan sistem pakar tersebut antara lain :

1. Pengguna dapat melakukan konsultasi (sesi tanya jawab) dengan memasukkan gejala-gejala fisik yang dialami pasien untuk mendapatkan informasi gangguan kesehatan pada pasien dan cara mengatasinya dengan faktor kepastiannya.
2. Pakar dapat menentukan jenis gangguan kesehatan, gejala-gejalanya, aturan-aturan yang digunakan untuk mendiagnosa setiap gangguan kesehatan berdasarkan gejala-gejalanya serta memberi solusi dari setiap gangguan kesehatan.
3. *User* pakar harus melakukan *login* untuk dapat mengakses sistem secara keseluruhan sedangkan *user* bukan pakar dapat langsung mengakses tanpa perlu *login* terlebih dahulu.

4. Sistem diimplementasikan dengan bahasa pemrograman Borland Delphi 7 dan basis data MySQL.
5. Sistem operasi yang digunakan adalah sistem operasi Windows.

3.1.2 Spesifikasi Kebutuhan

Spesifikasi kebutuhan dibutuhkan untuk menjelaskan kebutuhan perangkat lunak yang telah didefinisikan sebelumnya secara lebih detail dan tepat yang akan menjadi dasar bagi perancangan dan implementasi. Definisi dan spesifikasi perangkat lunak sebagai berikut :

1. Pengguna dapat melakukan konsultasi (sesi tanya jawab) dengan memasukkan gejala-gejala fisik yang dialami pasien untuk mendapatkan informasi gangguan kesehatan pada alat kelamin dan cara mengatasinya serta faktor kepastiannya.
 - 1.1 Pengguna harus dapat melakukan konsultasi dengan sistem pakar yaitu dengan diberikan pertanyaan oleh sistem dan harus dijawabnya.
 - 1.2 Pengguna harus mendapatkan informasi tentang gangguan kesehatan yang dialami pasien.
 - 1.3 Pengguna harus mendapatkan informasi faktor kepastian gangguan kesehatan yang dialami pasien.
 - 1.4 Pengguna harus mendapat informasi bagaimana mengatasi gangguan kesehatan tersebut.
2. Pakar dapat menentukan jenis gangguan kesehatan, gejala-gejalanya, aturan-aturan yang digunakan untuk mendiagnosa setiap gangguan kesehatan

berdasarkan gejala-gejalanya serta memberi solusi dari setiap gangguan kesehatan.

2.1 Pakar harus dapat menentukan gejala-gejala dan gangguan kesehatan.

2.2 Pakar harus dapat menentukan aturan-aturan yang digunakan untuk mendiagnosa gangguan kesehatan berdasarkan gejala-gejala yang ada.

2.3 Pakar dapat menentukan solusi dari setiap gangguan kesehatan.

3. *User* pakar harus melakukan *login* untuk dapat mengakses sistem secara keseluruhan sedangkan *user* bukan pakar dapat langsung mengakses tanpa perlu *login* terlebih dahulu.

3.1 *User* pakar harus memberikan masukan *username* dan *password* yang terdaftar pada sistem.

3.2 *User* pakar dapat melanjutkan jika *login* yang diberikan adalah valid.

3.3 Sistem akan memberikan pesan kesalahan saat *login* yang diberikan tidak valid dan *user* pakar tidak dapat melanjutkan.

3.4 *User* bukan pakar dapat langsung mengakses tanpa perlu login terlebih dahulu dan tidak bisa mengakses *data master*.

4. Sistem diimplementasikan dengan bahasa pemrograman Borland Delphi 7 dan basis data MySQL.

4.1 Bahasa pemrograman yang digunakan dalam implementasi sistem pakar ini adalah Borland Delphi 7.

4.2 Database yang digunakan dalam pembuatan sistem ini adalah MySQL.

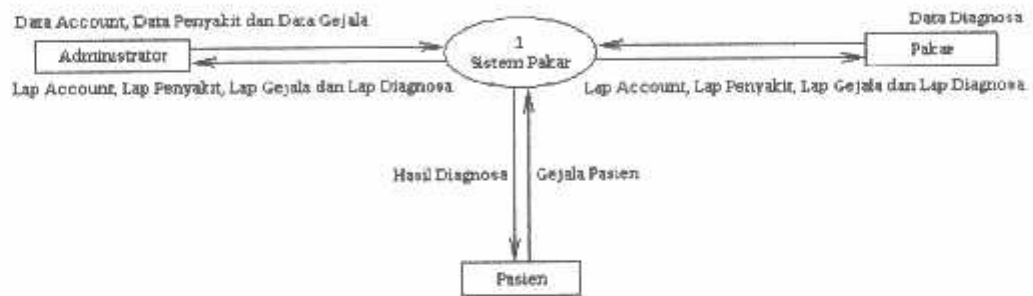
5. Sistem operasi yang digunakan adalah sistem operasi Windows.

5.1 Sistem yang dibuat akan menggunakan Microsoft Windows XP.

3.1.3 Data Flow Diagram

Data Flow Diagram (DFD) merupakan salah satu alat yang digunakan untuk menggambarkan keseluruhan sistem Pakar yang akan dibuat. DFD yang pertama kali dibuat adalah DFD level 0 atau *Context Diagram*.

3.1.3.1 Diagram Konteks



Gambar 3.2 Diagram Konteks Sistem Pakar Untuk Mendiagnosa Penyakit Yang Disebabkan Nyamuk.

Berdasarkan Gambar 3.2 proses “Sistem Pakar untuk Mendiagnosa Penyakit Yang Disebabkan Nyamuk Menggunakan *Metode Certainty*” mempunyai data masukan berupa:

1. Data Diagnosa

Data diagnosa diberikan *user* pakar untuk dilakukan proses validasi *login*, parameter yang digunakan adalah :

a. *Username*

Berisi masukan *username* yang diberikan oleh pakar.

b. *Password*

Berisi masukan *password* yang diberikan oleh pakar.

2. Data account, data penyakit dan data gejala

Data account, data penyakit dan data gejala diberikan administrator untuk memasukkan semua data – data yang diperlukan oleh system pakar.

3. Gejala pasien

Gejala pasien diberikan oleh pasien kedalam sistem pakar untuk mendapatkan hasil diagnosa penyakit yang diderita pasien.

Berdasarkan Gambar 3.1 proses “Sistem Pakar untuk mendiagnosa penyakit yang disebabkan nyamuk menggunakan *metode certainty*” mempunyai data keluaran berupa:

1. Laporan *account*, laporan penyakit, laporan gejala dan laporan diagnosa

Berisi laporan – laporan *account*, penyakit, gejala dan diagnosa yang diberikan oleh sistem pakar kepada pakar dan *administrator*.

2. Hasil diagnosa

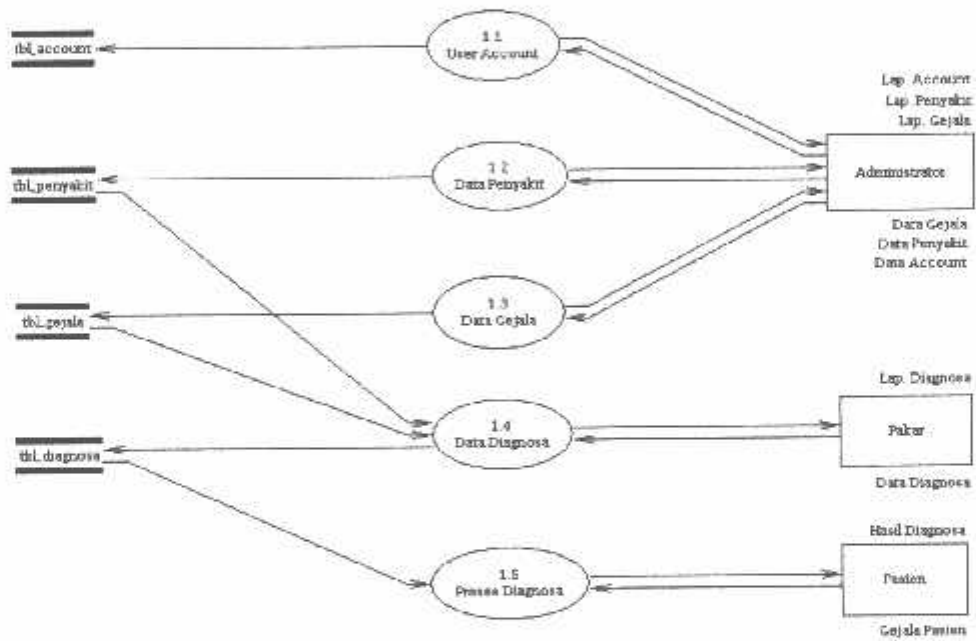
Berisi hasil diagnosa penyakit kepada pasien berdasarkan gejala yang diberikan oleh pasien ke dalam sistem pakar.

Untuk masuk ke dalam sistem tersebut *user* pakar harus memasukkan *login* terlebih dahulu. Jika *data-login-pakar* yang dimasukan sudah benar maka *user* pakar dapat memasukkan data yang diperlukan oleh sistem. Jika *user* pakar salah dalam memasukan *login* maka akan muncul pesan kesalahan dan proses akan tetap berada pada *proses login*. Hal ini dikarenakan akuisisi pengetahuan harus benar-benar dilakukan oleh orang yang yang berhak. *User* bukan pakar dapat langsung masuk kedalam sistem tanpa harus melakukan proses login terlebih

dahulu. Pilihan-pilihan proses yang akan didapatkan pengguna bukan pakar yaitu: konsultasi dan informasi penyakit.

3.1.3.2 DFD Level 1

DFD level 1 menggambarkan proses-proses yang lebih detail dari diagram konteks untuk Sistem Pakar untuk Mendiagnosa penyakit yang disebabkan nyamuk menggunakan *metode certainty*. DFD level 1 digambarkan dalam Gambar 3.3.



Gambar 3.3 DFD Level 1 Sistem Pakar Untuk Mendiagnosa Penyakit Yang Disebabkan Nyamuk Menggunakan *Metode Certainty*

Gambar 3.3 merupakan DFD level 1 sistem pakar untuk mendiagnosa penyakit yang disebabkan nyamuk menggunakan *metode certainty*. Dari diagram tersebut, maka dapat dijabarkan lebih detail proses-proses tersebut sebagai berikut:

1.1 *User Account*

User account mendapatkan *input* berupa data gejala, data penyakit dan data *account* dan memasukkan inputan kedalam *database* *tbl_account*. *User account* memberikan *output* kepada *administrator* berupa laporan *account*, laporan penyakit, dan laporan gejala.

1.2 *Data Penyakit*

Dalam Form Data Penyakit, data penyakit mendapatkan *input* berupa data gejala, data penyakit dan memasukkan inputan kedalam *database* *tbl_penyakit*. Data penyakit memberikan *output* kepada *administrator* berupa laporan penyakit dan laporan gejala.

1.3 *Data Gejala*

Dalam *form* data gejala, data gejala mendapat *input* dari *administrator* berupa data penyakit dan data gejala, dan memasukkan inputan kedalam *database* *tbl_gejala*. Data gejala memberikan *output* kepada *administrator* berupa laporan gejala dan laporan penyakit.

1.4 *Data Diagnosa*

Data diagnosa mendapatkan *input* dari pakar berupa data diagnosa, dan memasukkan inputan kedalam *database* *tbl_diagnosa*. Data diagnosa juga mendapatkan inputan dari *database* *tbl_gejala* dan *tbl_penyakit*. Data diagnosa memberikan *output* kepada pakar berupa laporan diagnosa.

1.5 Proses Diagnosa

Pada proses diagnosa menerima inputan dari pasien berupa gejala pasien dan database *tbl_diagnosa*, kemudian proses diagnosa mengeluarkan *output* kepada pasien berupa hasil diagnosa.

3.2 Perancangan Perangkat Lunak

Sistem Pakar untuk Mendiagnosa penyakit yang disebabkan oleh gigitan nyamuk dibuat dengan menggunakan bahasa pemrograman Borland Delphi 7 dan menggunakan basis data MySQL. Sistem Operasi yang digunakan dalam perancangan aplikasi ini adalah Microsoft Windows XP.

Sistem Pakar untuk Mendiagnosa penyakit yang disebabkan oleh gigitan nyamuk dirancang agar mampu :

1. Melakukan konsultasi dengan *user*
2. Memberikan informasi gangguan kesehatan dari hasil konsultasi dengan *user* disertai faktor kepastian
3. Memberikan informasi cara mengatasi gangguan kesehatan tersebut.

3.2.1 Perancangan Sistem

Perancangan sistem merupakan tahap awal dari perancangan perangkat lunak. Perancangan ini dilakukan untuk mengetahui kondisi sistem secara umum. Perancangan sistem ini meliputi diagram blok sistem.

3.2.1.1 Diagram Blok Sistem

Diagram blok sistem menggambarkan setiap blok atau bagian dari dari sistem aplikasi. Sistem Pakar untuk Mendiagnosa penyakit yang disebabkan oleh gigitan nyamuk dirancang untuk dapat memberikan informasi kepada pengguna,

gangguan kesehatan yang diderita disertai dengan faktor kepastiannya. Pengguna memberi masukkan berupa gejala-gejala gangguan kesehatan. Dari gejala-gejala tersebut sistem pakar akan memberikan informasi mengenai gangguan kesehatan, faktor kepastian serta solusi untuk mengatasi gangguan kesehatan tersebut. Sistem pakar tersebut dapat digambarkan dalam diagram blok seperti yang diperlihatkan dalam Gambar 3.3.



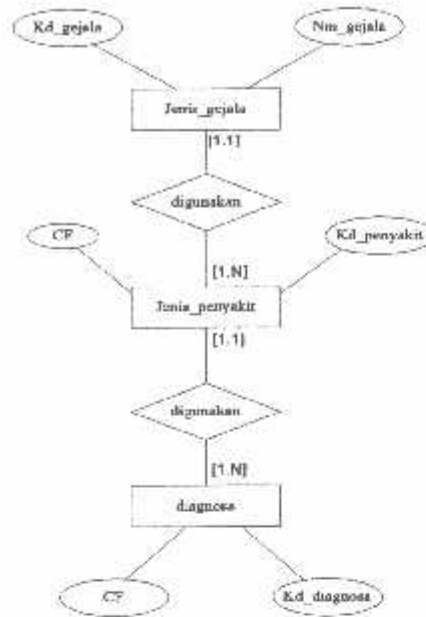
Gambar 3.4 Diagram Blok Sistem Pakar untuk Mendiagnosa penyakit yang disebabkan oleh gigitan nyamuk

3.2.2 Perancangan Basis Data

Perancangan basis data dilakukan agar Sistem Pakar untuk mendiagnosa penyakit kelamin manusia memiliki basis data yang efisien dalam penggunaan ruang penyimpanan, mudah mengubah data yang sudah ada.

3.2.2.1 Hubungan antar Entitas Pembentuk Sistem dalam Diagram-ER

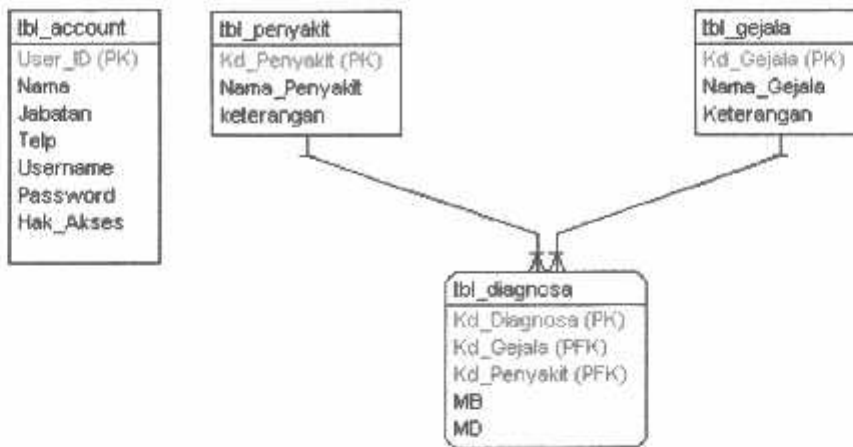
Diagram-ER digunakan untuk memodelkan struktur data dan hubungan antar entitas. Untuk menjelaskan hubungan antar entitas satu dengan entitas lain dalam sistem pakar ini, akan ditunjukkan dalam diagram-ER dalam Gambar 3.4 dibawah ini :



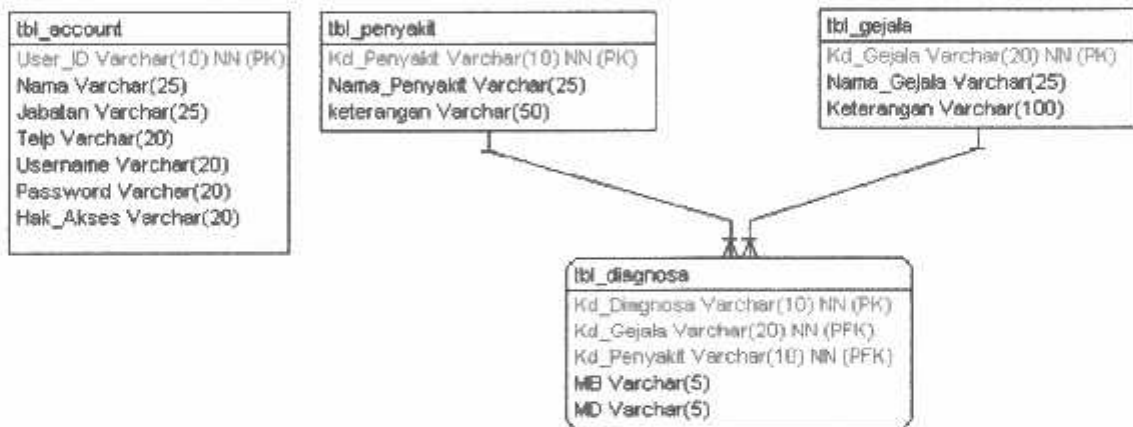
Gambar 3.5 ERD Sistem Pakar untuk Mendiagnosa penyakit yang disebabkan nyamuk

3.2.2.2 Relasi Antar Tabel.

Relasi antar tabel pada basisdata sistem pakar untuk mendiagnosa penyakit yang disebabkan nyamuk digambarkan dalam bentuk konsep atau *Conceptual Data Model (CDM)* dan dalam bentuk fisik atau *Physical Data Model (PDM)*, seperti yang ditunjukkan pada gambar berikut:



Gambar 3.6 *Conceptual Data Model (CDM)*



Gambar 3.7 *Physical Data Model (PDM)*

3.2.2.3 Struktur Tabel Database

Berdasarkan *diagram* di atas, maka tabel – tabel yang digunakan antara lain :

1. Tabel *account*.

Tabel *account* digunakan untuk menyimpan data *user* yang memiliki hak akses aplikasi.

Tabel 3.1 Tabel *Account*.

Name	Type	Length	Decimals	Allow Null	
User_ID	varchar	10	0	<input type="checkbox"/>	
Nama	varchar	25	0	<input checked="" type="checkbox"/>	
Jabatan	varchar	25	0	<input checked="" type="checkbox"/>	
Telp	varchar	15	0	<input checked="" type="checkbox"/>	
Username	varchar	20	0	<input checked="" type="checkbox"/>	
Password	varchar	20	0	<input checked="" type="checkbox"/>	
Hak_Akses	varchar	15	0	<input checked="" type="checkbox"/>	

2. Tabel Diagnosa.

Tabel diagnosa digunakan untuk menyimpan data diagnosa, data gejala, data penyakit, serta nilai MB dan MD.

Tabel 3.2 Tabel Diagnosa

Name	Type	Length	Decimals	Allow Null	
Kd_Diagnosa	varchar	10	0	<input type="checkbox"/>	
Kd_Penyakit	varchar	10	0	<input checked="" type="checkbox"/>	
Kd_Gejala	varchar	10	0	<input checked="" type="checkbox"/>	
MB	varchar	5	0	<input checked="" type="checkbox"/>	
MD	varchar	5	0	<input checked="" type="checkbox"/>	

3. Tabel Gejala.

Tabel gejala digunakan untuk menyimpan data gejala dan data penyakit.

Tabel 3.3 Tabel Gejala

Name	Type	Length	Decimals	Allow Null	
Kd_Gejala	varchar	10	0	<input type="checkbox"/>	
Nama_Gejala	varchar	25	0	<input checked="" type="checkbox"/>	
Keterangan	varchar	100	0	<input checked="" type="checkbox"/>	

4. Tabel Memo.

Tabel memo digunakan untuk menyimpan informasi penyakit yang disebabkan oleh gigitan nyamuk.


Tabel 3.4 Tabel Memo.



5. Tabel Penyakit.

Tabel penyakit digunakan untuk menyimpan data jenis penyakit yang disebabkan gigitan nyamuk.

Tabel 3.5 Tabel Penyakit.

Name	Type	Length	Decimals	Allow Null	
Kd_Penyakit	varchar	10	0	<input type="checkbox"/>	
Nama_Penyakit	varchar	25	0	<input checked="" type="checkbox"/>	
Keterangan	varchar	50	0	<input checked="" type="checkbox"/>	

3.2.3 Desain Modul

Desain tampilan digunakan untuk memberikan penjelasan tentang menu-menu yang terdapat pada sistem yang dibuat. Pertama kali *user* akan masuk ke halaman yang berisi pilihan menu *user*, *administrator area*, dan laporan.

Menu *user* terdiri atas :

1. Login : untuk mengakses menu *user account*, data penyakit, data gejala dan data diagnosa.
2. Konsultasi Penyakit : digunakan untuk melakukan proses konsultasi penyakit dengan sistem pakar.
3. Informasi Penyakit : berisi tentang semua informasi penyakit yang disebabkan oleh gigitan nyamuk.
4. *Logout* : untuk keluar dari *user account*.

5. *Exit* : untuk keluar dari sistem pakar.

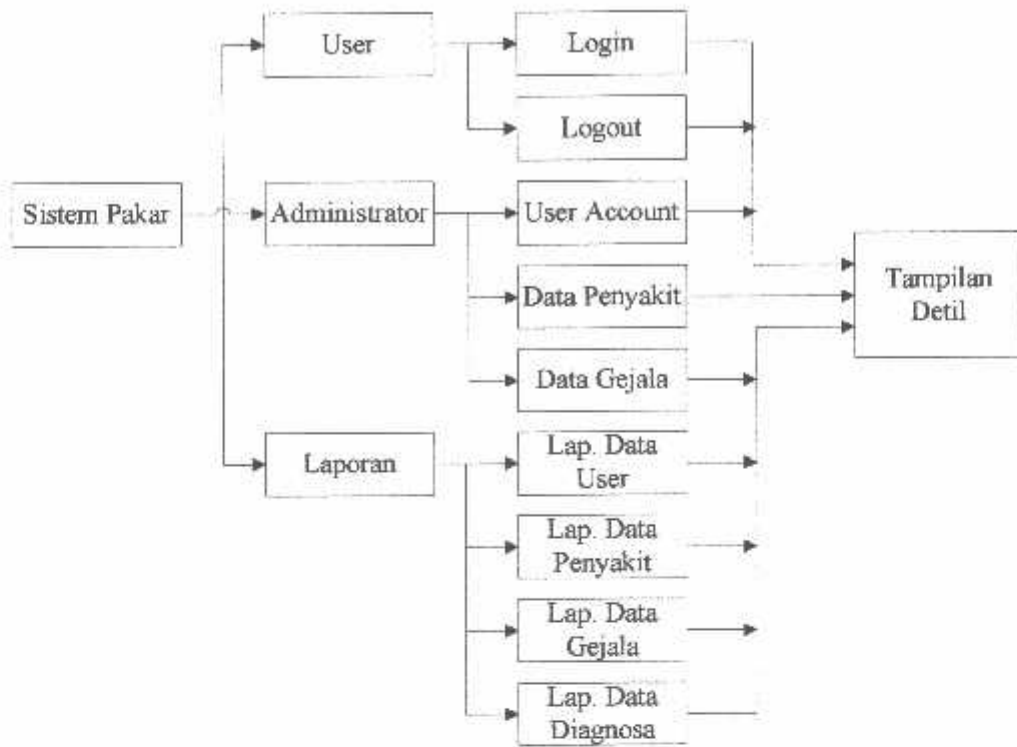
Administrator Area terdiri atas :

1. *User account* : digunakan untuk pemilihan hak akses sistem pakar.
2. *Data Penyakit* : digunakan oleh *administrator* untuk *update* data – data penyakit yang disebabkan oleh gigitan nyamuk.
3. *Data Gejala* : digunakan oleh *administrator* untuk *update* data – data gejala penyakit yang disebabkan oleh gigitan nyamuk.
4. *Data Diagnosa* : digunakan oleh pakar untuk *update* nama jenis penyakit dan gejalanya beserta nilai kepastian (MB) dan nilai ketidakpastian (MD).

Laporan terdiri atas :

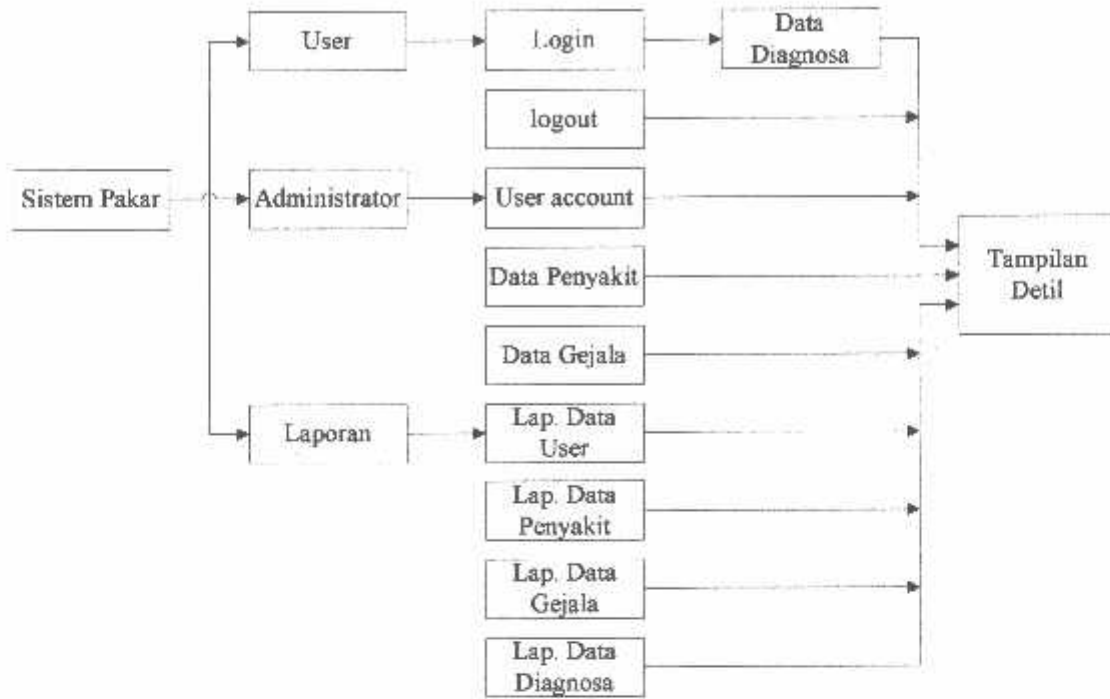
1. Laporan data *user* : berisi laporan tentang pemegang hak akses aplikasi sistem pakar tersebut.
2. Laporan data penyakit : berisi laporan tentang jenis penyakit yang disebabkan oleh gigitan nyamuk.
3. Laporan data gejala : berisi laporan tentang gejala – gejala penyakit yang disebabkan oleh gigitan nyamuk.
4. Laporan diagnosa : berisi laporan tentang data – data diagnosa penyakit yang disebabkan oleh gigitan nyamuk.

Bagan sistem pakar dibedakan menjadi 3, yaitu bagan *administrator*, bagan pakar dan bagan pasien.



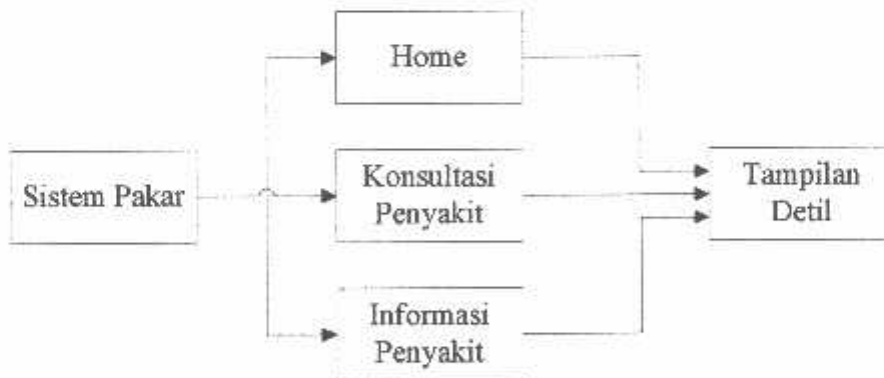
Gambar 3.8 Bagan sistem pakar untuk *Administrator*.

User administrator terlebih dahulu untuk melakukan *login* untuk dapat mengakses *user account*, data penyakit, data gejala yang digunakan untuk mengupdate *user account*, data – data penyakit dan data gejala. *User* pakar harus melakukan *login* terlebih dahulu. Pakar dapat membuka menu yang tidak bisa diakses oleh *user administrator* atau pasien.



Gambar 3.9 Bagan sistem pakar untuk *user* pakar.

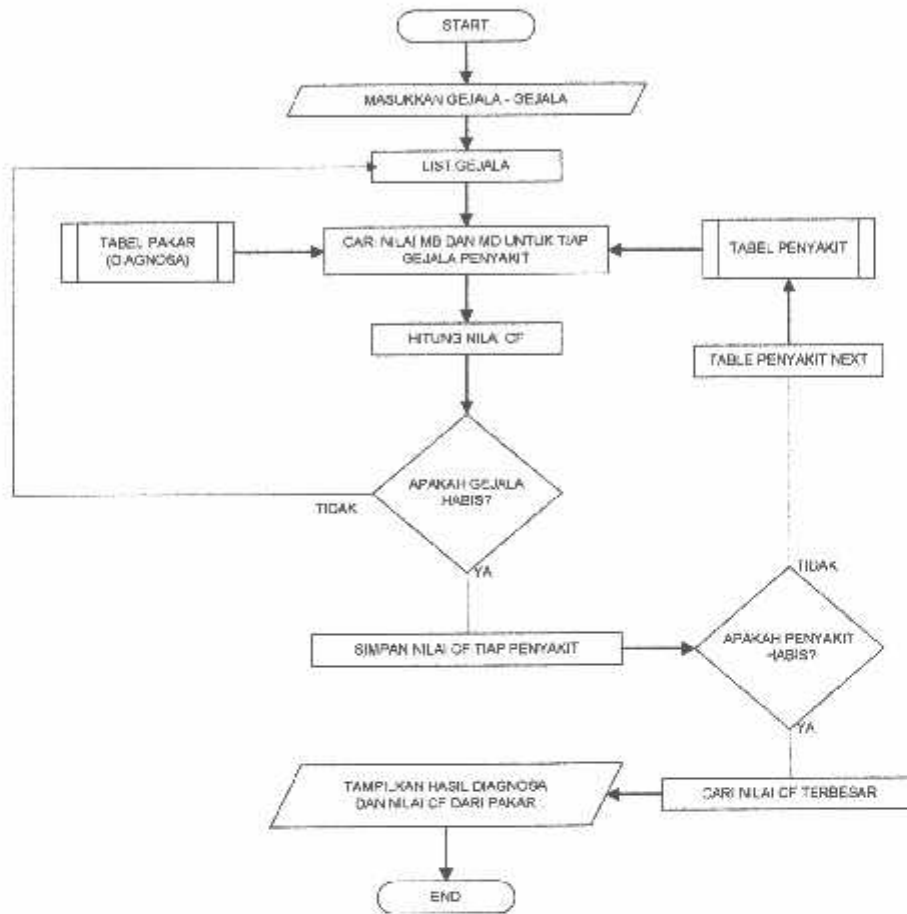
User pasien dapat langsung mengakses sistem tanpa harus melakukan *login* terlebih dahulu. *User* dapat memilih beberapa menu diantaranya konsultasi penyakit yang digunakan untuk melakukan proses konsultasi penyakit sedangkan informasi penyakit digunakan untuk mengetahui informasi tentang jenis penyakit yang disebabkan oleh gigitan nyamuk.



Gambar 3.10 Bagan sistem pakar untuk pasien

3.3.4 Flowchart

Flowchart proses diagnosa menggambarkan proses yang ada dalam form konsultasi. Flowchart proses dapat dilihat gambar di bawah ini :



Gambar 3.11 Algoritma Proses Diagnosa



BAB IV

IMPLEMENTASI DAN PENGUJIAN SISTEM

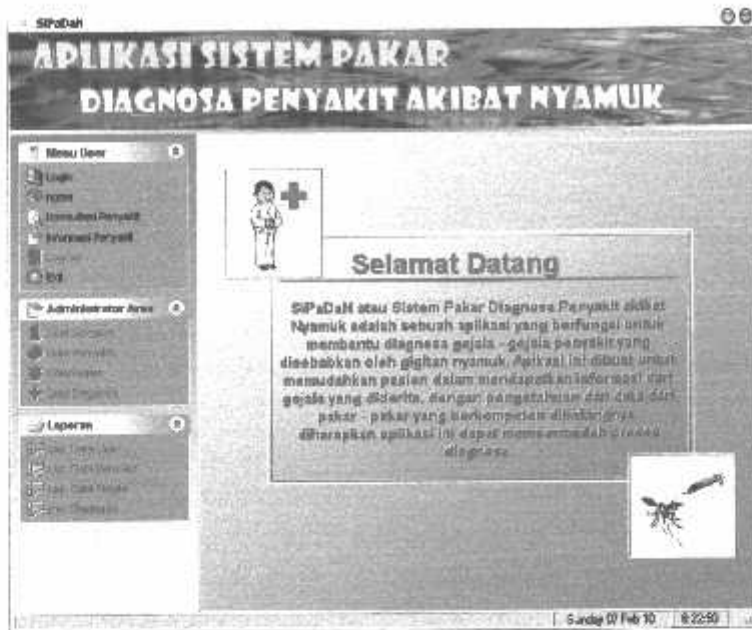
4.1 Implementasi Sistem

Implementasi pengembangan perangkat lunak merupakan proses perubahan spesifikasi sistem menjadi sistem yang dapat dijalankan. Tahap ini merupakan lanjutan dari proses perancangan sistem, yaitu proses pemrograman perangkat lunak sesuai dengan spesifikasi dan desain sistem.

Aplikasi sistem pakar ini menggunakan basis data MySQL yang berfungsi sebagai media penyimpanan data, yang terdiri dari beberapa tabel yang saling berhubungan. Sedangkan untuk menghubungkan antara informasi yang dibuat dengan basis data yang ada menggunakan Borland Delphi 7. Implementasi dari desain aplikasi berupa implementasi struktur data dari masing-masing proses. Sedangkan untuk menampilkan *Report* menggunakan *Quick Report*.

4.1.1 Menu Utama

Menu utama merupakan tampilan awal ketika sistem diakses. Pada menu ini terdapat pilihan menu diantaranya menu *User*, *Administrator Area*, dan Laporan. Menu utama dapat dilihat pada Gambar 4.1.



Gambar 4.1 *Form* Menu Utama

4.1.2 Menu User

Didalam menu *user* terdapat beberapa pilihan , yaitu *login* untuk mengakses *administrator* dan laporan, *home* berisi cerita singkat tentang aplikasi, konsultasi penyakit digunakan untuk melakukan konsultasi penyakit dengan sistem pakar, informasi penyakit berisi tentang informasi penyakit yang disebabkan oleh gigitan nyamuk, *logout* digunakan untuk keluar dari *administrator*, *exit* digunakan untuk keluar dari sistem pakar.



Gambar 4.2 Gambar Menu *User*

4.1.3 Administrator Area

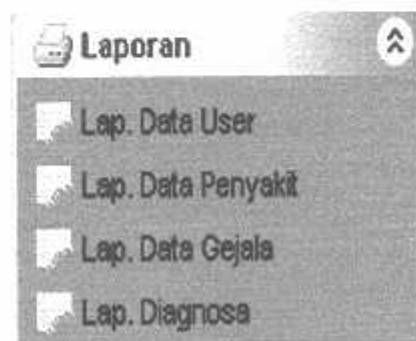
Didalam *Administrator area* terdapat menu *User account* yang berfungsi untuk penentuan pemegang sistem pakar, data penyakit digunakan untuk mengupdate jenis penyakit yang disebabkan oleh gigitan nyamuk, data gejala digunakan untuk mengupdate gejala – gejala penyakit yang disebabkan oleh gigitan nyamuk, data diagnosa digunakan pakar untuk mengupdate jenis penyakit dan gejalanya beserta nilai kepercayaan (MB) dan nilai ketidakpercayaan (MD).



Gambar 4.3 Gambar Menu *Administrator Area*

4.1.4 Menu Laporan

Menu laporan berisi laporan – laporan data *user*, laporan data penyakit, laporan data gejala, dan laporan diagnosa.



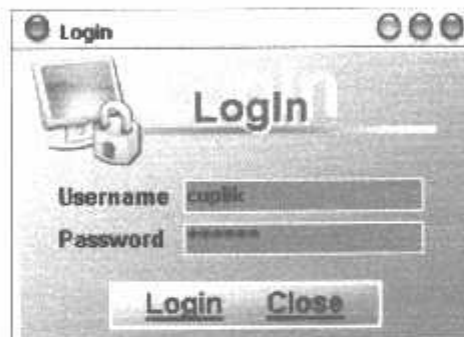
Gambar 4.4 Gambar Menu Laporan

4.2 Pengujian Sistem

Pengujian aplikasi ditujukan untuk menemukan kesalahan, sehingga aplikasi akan terbebas dari segala kesalahan pada saat dieksekusi oleh pengguna. Pengujian aplikasi meliputi *login*, *update data*, *hapus data* dan *konsultasi*. Pengujian dapat dilakukan dengan mengeksekusi aplikasi.

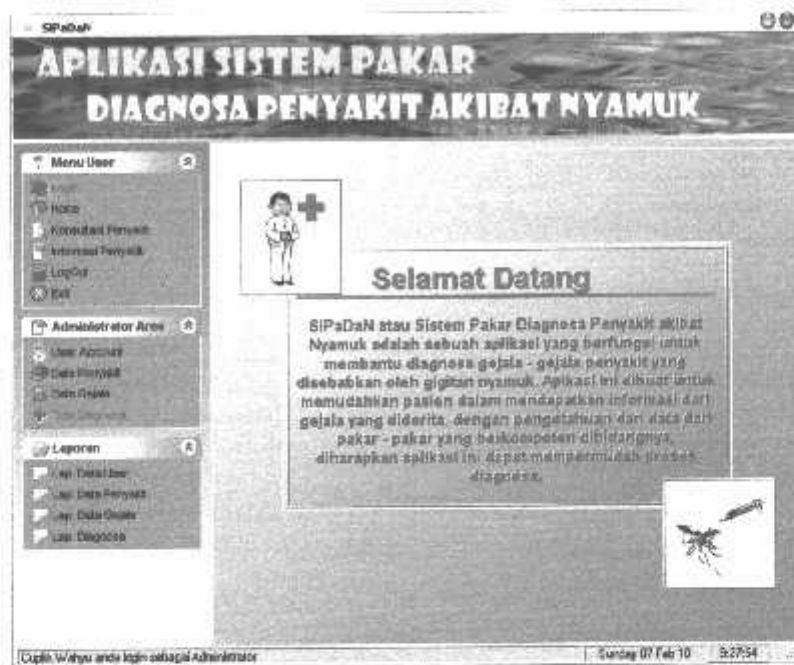
4.2.1 Pengujian *Login*

Sebelum masuk ke sistem *update data* dan *hapus data* harus melalui *login* terlebih dahulu seperti yang dilihat pada gambar 4.5.



Gambar 4.5 *Form Login*

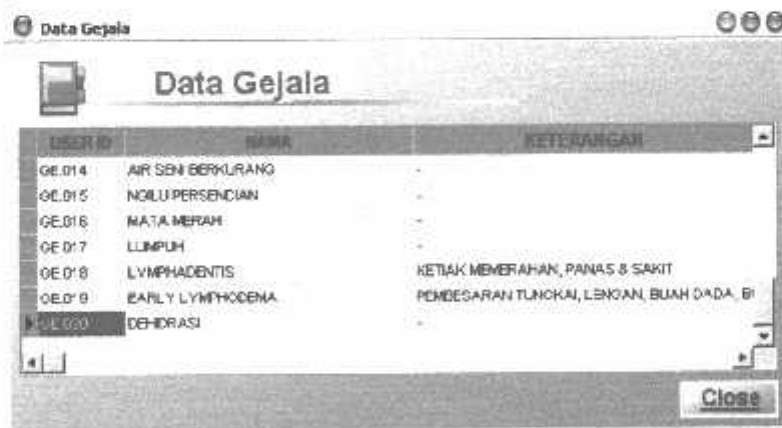
Setelah login berhasil, maka menu *administrator area* dan laporan akan aktif seperti terlihat pada gambar 4.6.



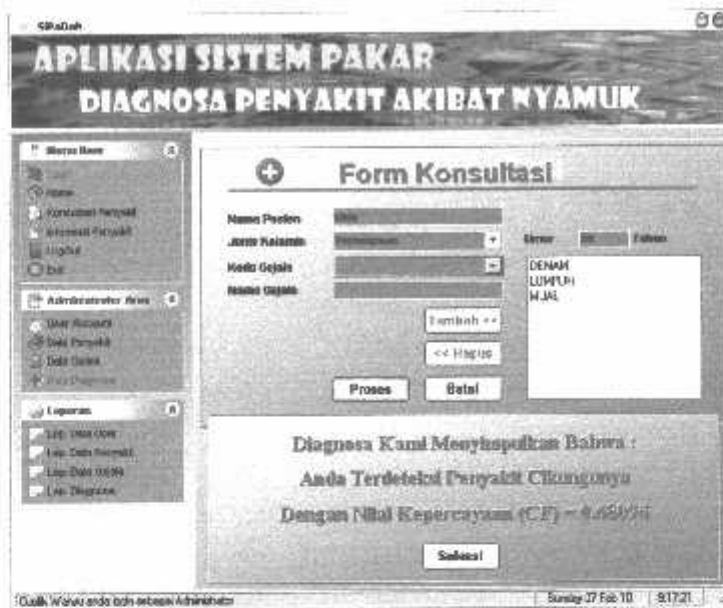
Gambar 4.6 Menu *Administrator* dan Laporan dapat diakses.

4.2.2 Pengujian Konsultasi Penyakit

Pada pengujian konsultasi penyakit, *user* akan melakukan konsultasi penyakit dengan cara memilih gejala yang dirasakan yang telah disediakan sistem seperti yang terlihat pada gambar 4.7.



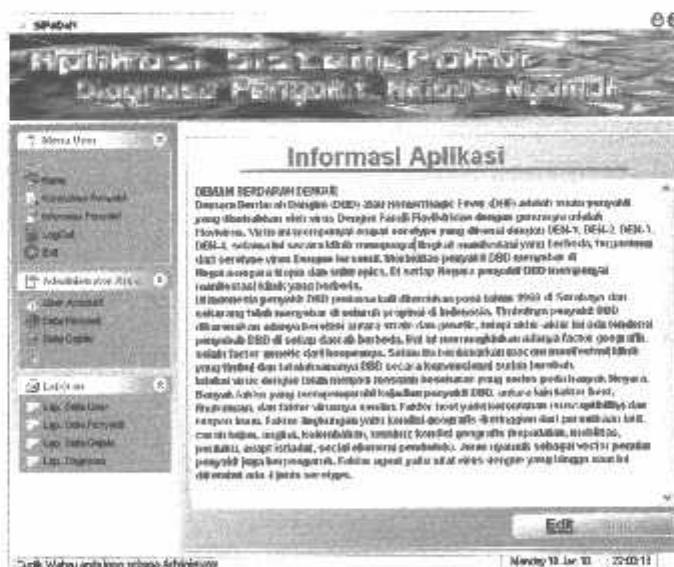
Gambar 4.7 *Form* Data Gejala untuk dipilih oleh *user*.



Gambar 4.9 Form Konsultasi dan Hasil diagnosa

4.2.3 Pengujian Informasi Penyakit

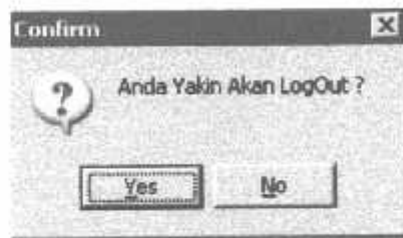
Di dalam menu Informasi Penyakit berisi tentang informasi jenis penyakit yang disebabkan oleh gigitan nyamuk dan gejala – gejalanya seperti yang terlihat pada gambar di bawah ini.



Gambar 4.10 Form Informasi Penyakit.

4.2.4 Pengujian *Logout*.

Logout digunakan *user* untuk keluar dari *administrator*. Apabila *user* akan keluar, maka sistem akan mengkonfirmasi *Yes* atau *No* seperti yang terlihat pada gambar 4.11.



Gambar 4.11 Konfirmasi *Logout*.

4.3 Pengujian Menu *Administrator Area*.

Administrator Area terdapat menu *user account*, data penyakit, data gejala, dan data diagnosa. Pada menu *administrator area* ini digunakan pakar dan *administrator* untuk mengisi data – data penyakit, data gejala dan data diagnosa.

4.3.1 Pengujian *User Account*.

Menu *User Account* digunakan *administrator* untuk menentukan siapa yang akan memegang hak akses aplikasi sistem pakar ini, seperti yang terlihat pada gambar 4.12.



Gambar 4.12 Form User Account.

Untuk menambahkan hak akses baru *user* harus memasukkan *user ID*, nama, jabatan, telepon, *username* dan *password* serta memilih hak akses pakar atau *administrator* lalu tekan tombol simpan. Jika penyimpanan berhasil maka sistem akan menampilkan pesan Data Sudah Tersimpan seperti yang terlihat pada gambar di bawah ini.



Gambar 4.13 Penyimpanan Data User Account berhasil

4.3.2 Pengujian Data Penyakit

Menu data penyakit digunakan *administrator* untuk mengupdate data – data penyakit yang disebabkan oleh gigitan nyamuk.



Gambar 4.14 Form Data Penyakit.

Untuk menyimpan data penyakit yang disebabkan nyamuk, *user* harus mengisi kode penyakit, nama penyakit dan keterangan jika diperlukan. Setelah itu tekan tombol simpan. Jika penyimpanan data berhasil, maka sistem akan menampilkan pesan Data Sudah tersimpan seperti yang terlihat pada gambar di bawah ini.



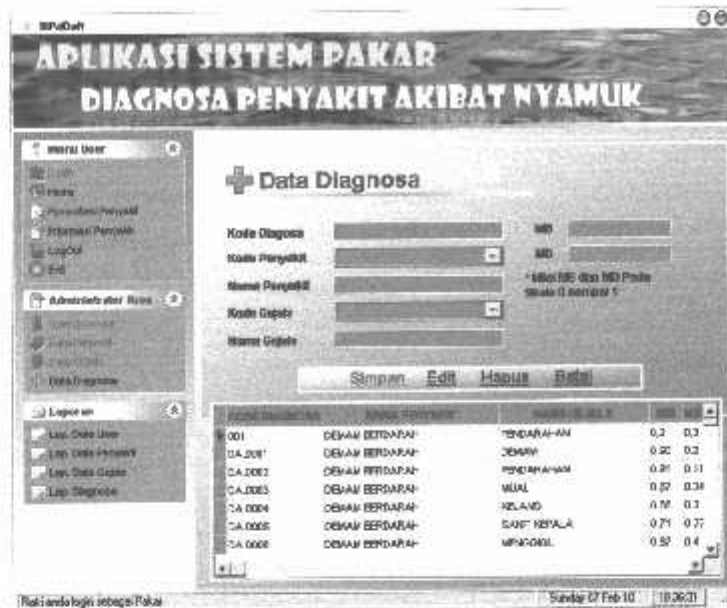
Gambar 4.15 Penyimpanan Data Penyakit

4.3.3 Pengujian Data Gejala.

Menu Data Gejala digunakan *administrator* untuk mengupdate gejala – gejala penyakit yang disebabkan nyamuk.



Gambar 4.16 Form Data Gejala.



Gambar 4.18 Form Data Diagnosa.

Untuk memasukkan data diagnosa baru *user* harus mengisi kode diagnosa, kode penyakit, nama penyakit, kode gejala, nama gejala dan nilai – nilai MB dan MD lalu tekan tombol simpan. Setelah tekan tombol simpan sistem akan menampilkan pesan Data Sudah Tersimpan seperti pada gambar 4.19.



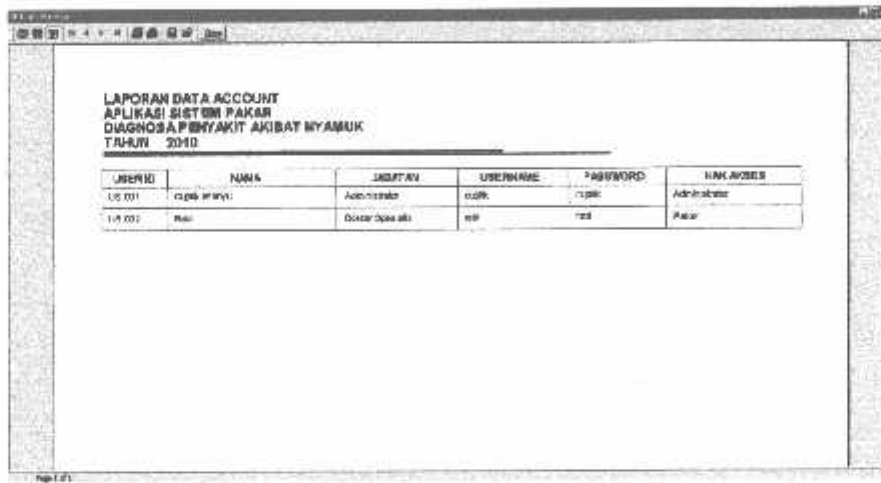
Gambar 4.19 Penyimpanan Data Diagnosa Baru.

4.4 Pengujian Menu Laporan.

Pada menu Laporan terdapat laporan Data *User*, Laporan Data Penyakit, Laporan Data Gejala, dan laporan Data Diagnosa seperti terlihat pada gambar dibawah ini.

1. Laporan Data Account

Dalam laporan data account berisi user ID, nama, jabatan, password, dan hak akses.

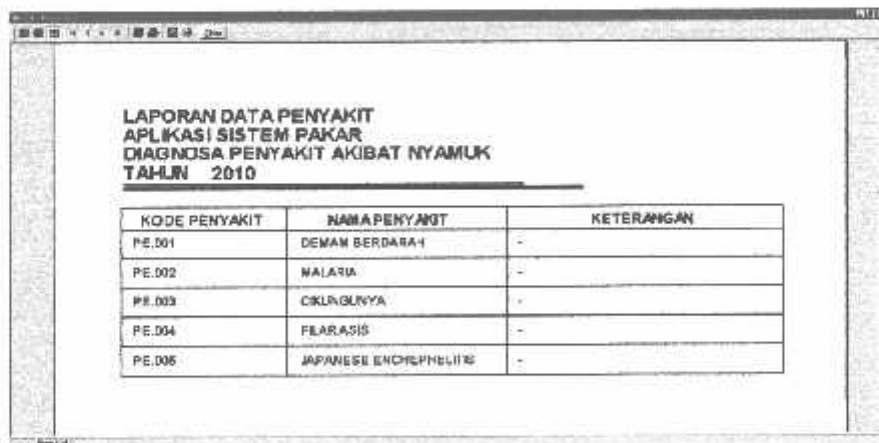


USERID	NAMA	JABATAN	USERNAME	PASSWORD	HAK AKSES
U-001	cupak@nyamuk	Admin	cupak	cupak	Admin
U-002	pacar	Dokter Spesialis	pacar	pacar	Pakar

Gambar 4.20 Laporan Data *user*

2. Laporan Data Penyakit.

Laporan ini berisi kode penyakit, nama penyakit dan keterangan.



KODE PENYAKIT	NAMA PENYAKIT	KETERANGAN
PE.001	DEMAM BERDARAH	-
PE.002	MALARIA	-
PE.003	CHIKUNGUNYA	-
PE.004	FEARASIS	-
PE.005	JAPANESE ENCEPHALITIS	-

Gambar 4.21 Laporan Data Penyakit

3. Laporan Data Gejala

Laporan ini berisi kode gejala, nama gejala, dan keterangan.

LAPORAN DATA GEJALA
 APLIKASI SISTEM PAKAR
 DIAGNOSA PENYAKIT AKIBAT NYAMUK
 TAHUN 2018

KODE GEJALA	NAMA GEJALA	KETERANGAN
GE 001	LEKEM	
GE 002	PERIBANDUAN	
GE 003	REPAK	
GE 004	SPANKS	
GE 005	ERANTHERAL	
GE 006	MEDGAGE	
GE 007	NYER DITOL	
GE 008	ULAT BIRU PERAN	
GE 009	UMAM PANGIT	REJENDRA ESTOKAN
GE 010	TRIMBATA TIRPAM	MULAWAN REKESIO
GE 011	LETA: E LERU	
GE 012	EDWZONGA	
GE 013	DAKOGGAL: K SADRAL	

Gambar 4.22 Laporan Data Gejala.

4. Laporan Data Diagnosa

Laporan ini berisi kode diagnosa, nama penyakit, nama gejala, serta nilai MB dan MD.

LAPORAN DATA DIAGNOSA
 APLIKASI SISTEM PAKAR
 DIAGNOSA PENYAKIT AKIBAT NYAMUK
 TAHUN 2018

KODE DIAGNOSA	NAMA PENYAKIT	NAMA GEJALA	MB	MD
DI 001	DENAM BERDARAH	REJENDRA	0.1	0.2
DI 002	DENAM BERDARAH	DENAM	0.2	0.3
DI 003	DENAM BERDARAH	PERIBANDUAN	0.4	0.5
DI 004	DENAM BERDARAH	REPAK	0.5	0.6
DI 005	DENAM BERDARAH	SPANKS	0.6	0.7
DI 006	DENAM BERDARAH	ERANTHERAL	0.7	0.8
DI 007	DENAM BERDARAH	MEDGAGE	0.8	0.9
DI 008	DENAM BERDARAH	NYER DITOL	0.9	1.0
DI 009	DENAM BERDARAH	ULAT BIRU PERAN	1.0	1.1
DI 010	DENAM BERDARAH	TRIMBATA	1.1	1.2
DI 011	DENAM BERDARAH	TRIMBATA TIRPAM	1.2	1.3
DI 012	DENAM BERDARAH	LETA: E LERU	1.3	1.4
DI 013	DENAM BERDARAH	EDWZONGA	1.4	1.5

Gambar 4.23 Laporan Data Diagnosa

4.5 Pengujian Sistem Pakar dengan Pakar dan sumber lain.

Pengujian ini digunakan untuk membandingkan hasil diagnose antara aplikasi sistem pakar dengan hasil diagnosa pakar dan dari sumber – sumber lain.

Tabel 4.1 Hasil pengujian terhadap 5 user.

No.	Nama Pasien	Gejala	Diagnosa Aplikasi	Diagnosa Pakar	Diagnosa dengan Perhitungan Manual
1.	Rizma N	<ul style="list-style-type: none"> ▪ Demam ▪ Lumpuh ▪ Mual 	Chikungunya CF = 0,463	Chikungunya	Chikungunya CF = 0,4629
<p>Diagnosa pakar diperoleh dari analisa gejala yang ada disertai dengan pemeriksaan laboratorium untuk memperkuat diagnosa.</p> <p>Prosentase nilai error diagnosa aplikasi dengan diagnosa perhitungan manual = 0,021 %</p>					
2.	Rini Dwi	<ul style="list-style-type: none"> ▪ Lymphadenitis ▪ Mata Merah ▪ Demam 	Chikungunya CF = 0,16	Chikungunya	Chikungunya CF = 0,1593
<p>Prosentase nilai error diagnosa aplikasi dengan diagnosa perhitungan manual = 0,043 %</p>					
3.	Ganang S	<ul style="list-style-type: none"> ▪ Demam ▪ Trombositopeni 	Malaria CF = 0,424	Malaria	Malaria CF = 0,4239
<p>Prosentase nilai error diagnosa aplikasi dengan diagnosa perhitungan manual = 0,023 %</p>					
4.	Nurwin L	<ul style="list-style-type: none"> ▪ Demam ▪ Nyilu Persendian ▪ Gangguan Kesadaran 	Malaria CF = 0,294	Malaria	Malaria CF = 0,290
<p>Prosentase nilai error diagnosa aplikasi dengan diagnosa perhitungan manual = 1,37 %</p>					

BAB V

PENUTUP

5.1 Kesimpulan

Kesimpulan yang didapat dari perancangan, implementasi, dan pengujian aplikasi sistem pakar untuk mendiagnosa penyakit yang disebabkan oleh gigitan nyamuk antara lain:

1. Penggunaan faktor kepastian pada sistem pakar untuk mendiagnosa penyakit yang disebabkan nyamuk dapat mendiagnosa gangguan kesehatan yang disebabkan oleh gigitan nyamuk dan solusi untuk mengatasi gangguan kesehatan tersebut dengan menerima masukkan gejala dari *user*.
2. Data penyakit, data gejala serta nilai – nilai kepastian dan ketidakpastian yang ada dalam sistem pakar ini dapat diperbaharui dengan mudah karena para medis dapat dengan mudah memasukkan pengetahuan dengan mengisi tabel yang ada dalam *data* diagnosa.
3. Dari pengujian pada aplikasi ini, hasil diagnosa pakar diperoleh dari analisa gejala pasien yang ada disertai dengan *test* laboratorium untuk memperkuat hasil diagnose.
4. Dari hasil diagnosa antara aplikasi sistem pakar dengan pakar mempunyai prosentase nilai kesamaan 100%, sedangkan hasil diagnosa aplikasi dengan nilai CF perhitungan manual masih mempunyai selisih karena pembulatan nilai.
5. Dari hasil pengujian terhadap 5 orang user didapatkan nilai eror :
 - Pengujian 1 mempunyai nilai eror sebesar 0,021 %

- Pengujian 2 mempunyai nilai eror sebesar 0,43 %
 - Pengujian 2 mempunyai nilai eror sebesar 0,023 %
 - Pengujian 2 mempunyai nilai eror sebesar 1,37 %
 - Pengujian 2 mempunyai nilai eror sebesar 0,046 %
6. Dari hasil pengujian antara diagnose aplikasi dengan diagnose perhitungan manual mempunyai rata – rata nilai eror sebesar 0,3006 %

5.2 Saran

Saran yang dapat diberikan untuk pengembangan sistem pakar untuk mendiagnosa penyakit yang disebabkan oleh gigitan nyamuk dengan menggunakan faktor kepastian:

1. Sistem pakar untuk mendiagnosa penyakit yang disebabkan nyamuk menggunakan faktor kepastian dapat dikembangkan dalam bentuk aplikasi yang lebih mendetail sehingga dapat diakses oleh lebih banyak *user*.
2. Dalam penghitungan *CF*, perlu dikembangkan lebih lanjut untuk mendapatkan *CF* yang mendekati nilai sesungguhnya.
3. Untuk mendapatkan hasil diagnosa yang akurat, disarankan *user* juga melakukan *test* laboratorium dan pemeriksaan pakar lebih lanjut.



DAFTAR PUSTAKA

- [1] Andi, 2003. *Pemrograman Borland Delphi 7*. Penerbit Andi dan Madcoms, Madiun
- [2] Bakti Husada. 1997. *Nyamuk dan kesehatan*. Penerbit Departemen Kesehatan, Jakarta.
- [3] Kusrini. 2006. *Sistem Pakar Teori dan Aplikasi*. Penerbit Andi, Yogyakarta
- [4] Arhami, M (2005). *Konsep Dasar Sistem Pakar*. Yogyakarta : Andi Offset.
- [5] Kusuma, Dewi, 2003. "Artificial Intelligence (Teknik dan Aplikasinya)". Penerbit Graha Ilmu.
- [6] "Gejala Penyakit yang disebabkan nyamuk dan penyembuhannya". <http://www.ppmplp.depkes.go.id/detil.asp?m=4&s=3&i=7>
- [7] Kadir, Abdul. 2004 . *Dasar Aplikasi Database MySQL - Delphi 7.0*. Penerbit Andi Offset, Yogyakarta





INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK ELEKTRO S-1
Jl. Karanglo km 2, Malang

**BERITA ACARA UJIAN SKRIPSI
FAKULTAS TEKNOLOGI INDUSTRI**

Nama : Cuplik Wahyu Ratri Tri Pamungkas
NIM : 04.12.673
Jurusan : Teknik Elektro S-1
Konsentrasi : Teknik Komputer dan Informatika
Judul Skripsi : **SISTEM PAKAR UNTUK MENDIAGNOSA PENYAKIT YANG
DISEBABKAN NYAMUK MENGGUNAKAN METODE
CERTAINTY**

Dipertahankan di hadapan Majelis Penguji Skripsi Jenjang Strata Satu (S-1) pada :

Hari : Rabu
Tanggal : 10 Februari 2010
Dengan Nilai : 84,75 (A) *EH*



Ir. H. Sidik Noertjahjono, MT.
NIP.Y 102 8700 163

Sekretaris Majelis Penguji

Ir. F. Yudi Limpraptono, MT.
NIP.Y 103 9500 274

Penguji I

Ir. TH. Mimin Mustikawati, MT
NIP. P. 1030000352

Penguji II

M. Ibrahim Ashari, ST. MT
NIP. P. 1030100358



INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK ELEKTRO S-1
Jl. Karanglo km 2, Malang

FORMULIR PERBAIKAN SKRIPSI

Dalam pelaksanaan ujian skripsi jenjang Strata Satu (S-1) Jurusan Teknik Elektro Konsentrasi Teknik Komputer dan Informatika, maka perlu adanya perbaikan skripsi untuk mahasiswa :

Nama : CUPLIK WAHYU RATRI TRI PAMUNGKAS
NIM : 04.12.673
Jurusan : Teknik Elektro S-1
Konsentrasi : Teknik Komputer dan Informatika
Masa Bimbingan :
Judul Skripsi : **SISTEM PAKAR UNTUK MENDIAGNOSA PENYAKIT YANG
DISEBABKAN NYAMUK MENGGUNAKAN METODE
CERTAINTY**

Tanggal	Uraian	Paraf
Penguji I 10 Februari 2010	Laporan kurang sempurna.	
Penguji II 10 Februari 2010	Penambahan kalimat pengantar pada gambar.	
Penguji II 10 Februari 2010	Tunjukkan rumus – rumus CF pada software	

Disetujui :

Penguji I

Ir. TH Mimin M. MT.
NIP.1030000352

Penguji II

M. Ibrahim Ashari, ST, MT
NIP. 1030100358

Mengetahui :

Dosen Pembimbing I

Ir. Yusuf Ismail Nakhoda, MT
NIP.Y : 101 8800 180

Dosen Pembimbing II

Dr. Eng. Arsyanto Soetejo, ST, MT
NIP.Y : 103 0800 417



PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG
INSTITUT TEKNOLOGI NASIONAL MALANG

FAKULTAS TEKNOLOGI INDUSTRI
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
PROGRAM PASCASARJANA MAGISTER TEKNIK

PT. BNI (PERSERO) MALANG
BANK NIAGA MALANG

Kampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145
Kampus II : Jl. Raya Karanglo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

Malang, 30 Januari 2010

Nomor : ITN-469/I.TA/2/10
Lampiran : -
Perihal : BIMBINGAN SKRIPSI
Kepada : Yth. Sdr./i. **IR. YUSUF ISMAIL NAKHODA, MT**
Dosen Institut Teknologi Nasional Malang

Dosen Pembimbing
Jurusan Teknik Elektro S-1
di
Malang

Dengan hormat
Sesuai dengan permohonan dan persetujuan dalam Proposal Skripsi
Untuk Mahasiswa :

Nama : CUPLIK WAHYU, R.T.P
Nim : 0412673
Fakultas : Teknologi Industri
Jurusan : Teknik Elektro S-1
Konsentrasi : Teknik **Komputer & Informatika**

Maka dengan ini pembimbingan tersebut kami serahkan sepenuhnya
kepada Saudara/i selama masa waktu (enam) 6 bulan, terhitung mulai
tanggal :

25 Desember 2009 s/d 25 Juni 2010

Sebagai satu syarat untuk menempuh ujian Sarjana Teknik,
Jurusan Teknik Elektro S-1
Demikian agar maklum dan atas perhatian serta bantuannya kami sampaikan
terima kasih



Ketua Jurusan
Teknik Elektro S-1

Ir. F. Yudi Limpraptono, MT
Nip. Y. 1039500274

Tembusan Kepada Yth.
1. Mahasiswa Yang Bersangkutan
2. Arsip
3. Coret yang tidak perlu

Form. S.4a



PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG
INSTITUT TEKNOLOGI NASIONAL MALANG

FAKULTAS TEKNOLOGI INDUSTRI
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
PROGRAM PASCASARJANA MAGISTER TEKNIK

PT. BNI (PERSERO) MALANG
BANK NIAGA MALANG

Kampus I : . . . Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145
Kampus II : Jl. Raya Karanglo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

Malang, 30 Januari 2010

Nomor : ITN-468/I.TA/2/10
Lampiran : -
Perihal : BIMBINGAN SKRIPSI
Kepada : Yth. Sdr./i. **DR. ENG. ARYUANTO S, ST, MT**
Dosen Institut Teknologi Nasional Malang

Dosen Pembimbing
Jurusan Teknik Elektro S-1
di
Malang

Dengan hormat
Sesuai dengan permohonan dan persetujuan dalam Proposal Skripsi
Untuk Mahasiswa :

Nama : CUPLIK WAHYU. R.T.P
Nim : 0412673
Fakultas : Teknologi Industri
Jurusan : Teknik Elektro S-1
Konsentrasi : Teknik **Komputer & Informatika**

Maka dengan ini pembimbingan tersebut kami serahkan sepenuhnya
kepada Saudara/i selama masa waktu (enam) 6 bulan, terhitung mulai
tanggal :

25 Desember 2009 s/d 25 Juni 2010

Sebagai satu syarat untuk menempuh ujian Sarjana Teknik,
Jurusan Teknik Elektro S-1
Demikian agar maklum dan atas perhatian serta bantuannya kami sampaikan
terima kasih



Ketua Jurusan
Teknik Elektro S-1

Ir. F. Yudi Limpraptono, MT
Nip. Y. 1039500274

Tembusan Kepada Yth :
1. Mahasiswa Yang bersangkutan
2. Arsip
3. Coret yang tidak perlu

Form. S 4a



FORMULIR BIMBINGAN SKRIPSI

Nama : CUPLIK WAHYU RATRI T.P
Nim : 04.12.673
Masa Bimbingan : 25 DESEMBER 2009 s/d 25 JUNI 2010
Judul Skripsi : SISTEM PAKAR UNTUK MENDIAGNOSA PENYAKIT YANG
DISEBABKAN NYAMUK MENGGUNAKAN METODE
CERTAINTY

No	Tanggal	Uraian	Paraf Pembimbing
1	22 Januari 2010	Maju dan demo program sekaligus perbaikan font judul pada aplikasi karena kurang jelas.	
2	24 Januari 2010	Maju dan demo program yang telah diperbaiki tampilannya.	
3	28 Januari 2010	Maju makalah seminar hasil dan revisi format penulisan serta ACC nakalah.	
4	2 Februari 2010	Maju laporan Bab I, II, III, IV, V dan revisi format penulisan.	
5	3 Februari 2010	Maju laporan Bab I, II, III, IV, V dan revisi penambahan table perbandingan diagnosa aplikasi dengan pakar.	
6	8 Februari 2010	Haju laporan Bab I, II, III, IV, V dan revisi kesimpulan serta ACC laporan.	

Malang,

Dosen pembimbing I


Ir. Yusuf Ismail Nabkoda. MT
NIP. Y. 1018800189

Form S-4b



FORMULIR BIMBINGAN SKRIPSI

Nama : CUPLIK WAHYU RATRI T.P
Nim : 04.12.673
Masa Bimbingan : 25 DESEMBER 2009 s/d 25 JUNI 2010
Judul Skripsi : SISTEM PAKAR UNTUK MENDIAGNOSA PENYAKIT YANG
DISEBABKAN NYAMUK MENGGUNAKAN *METODE
CERTAINTY*

No	Tanggal	Uraian	Paraf Pembimbing
1	22 Januari 2010	Cara penulisan nakalah seminar hasil dirapikan.	
2	24 Januari 2010	Setiap kalimat yang diambil dari refrensi diberi indek pustaka.	
3	28 Januari 2010	ACC makalah seminar hasil.	
4	2 Februari 2010	Maju Bab I, II, III, IV, V dan revisi format penulisan.	
5	3 Februaui 2010	ACC laporan bab I, II, III, IV, V.	
6			

Malang,

Dosen pembimbing II

Dr. Aryuanto S, ST MT
NIP. Y 1030800417

Form S-4b



**BERITA ACARA SEMINAR PROPOSAL SKRIPSI
 PROGRAM STUDI TEKNIK ELEKTRO S1**

KONSENTRASI		TEKNIK KOMPUTER & INFORMATIKA		
1.	Nama Mahasiswa	EUPLIK WAHYU RATRI TRI. P	NIM	DA.12.692
2.	Keterangan	Tanggal	Waktu	Tempat / Ruang
	Pelaksanaan	25 NOVEMBER 2009	13.30 - 14.15.	R. SEMINAR
Spesifikasi Judul (berilah tanda silang *)				
3.	a. Sistem Tenaga Elektrik	e. Embedded System	i. Sistem Informasi	
	b. Konversi Energi	f. Antar Muka	j. Jaringan Komputer	
	c. Sistem Kendali	g. Elektronika Telekomunikasi	k. Web	
	d. Tegangan Tinggi	h. Elektronika Instrumentasi	<input checked="" type="checkbox"/> l. Algoritma Cerdas	
4.	Judul Proposal yang diseminarkan Mahasiswa	SISTEM BAKAR UNTUK MENDIAGNOSA PENYAKIT YANG DISEBABKAN OLEH NGAMUK MENGGUNAKAN METODE CERTAINY		
5.	Perubahan Judul yang diusulkan oleh Kelompok Dosen Keahlian			
6.	Catatan :			
7.	Catatan :	<input checked="" type="checkbox"/> metode Cert <input checked="" type="checkbox"/> Algoritma *) diingkari a, b, c,		

```
//-----Data Account-----
```

```
procedure TFormUtama.edtIDChange(Sender: TObject);  
begin  
if edtID.Text="" then  
  lblSimpanUser.Enabled:=false  
else  
  lblSimpanUser.Enabled:=true;  
end;
```

```
procedure TFormUtama.ActBatalAccExecute(Sender: TObject);  
begin  
edtID.Text:='';  
edtNamaAcc.Text:='';  
edtJabatanAcc.Text:='';  
edtTelpAcc.Text:='';  
edtUsernameAcc.Text:='';  
edtpasswordAcc.Text:='';  
cmbHakAksesAcc.Text:='-----Pilih Hak Akses -----';  
edtID.Enabled:=true;  
edtID.SetFocus;  
editdata:=false;  
end;
```

```
procedure TFormUtama.ActSimpanAccExecute(Sender: TObject);  
var  
found:Boolean;  
kode:String;  
i:integer;  
begin  
if edtID.Text="" then  
  begin  
  application.MessageBox('Maaf User ID Belum Diisi...!', 'Cek Input', MB_IconInformation);  
  edtID.SetFocus;  
  end  
else if edtNamaAcc.Text="" then  
  begin  
  application.MessageBox('Maaf Nama User Belum Diisi...!', 'Cek Input', MB_IconInformation);  
  edtNamaAcc.SetFocus;  
  end  
else if edtJabatanAcc.Text="" then  
  begin  
  application.MessageBox('Maaf Jabatan User Belum Diisi...!', 'Cek Input', MB_IconInformation);  
  edtJabatanAcc.SetFocus;  
  end  
else if edtUsernameAcc.Text="" then  
  begin  
  application.MessageBox('Maaf Username Belum Diisi...!', 'Cek Input', MB_IconInformation);  
  edtUsernameAcc.SetFocus;  
  end  
else if edtpasswordAcc.Text="" then  
  begin  
  application.MessageBox('Maaf Password Belum Diisi...!', 'Cek Input', MB_IconInformation);  
  edtpasswordAcc.SetFocus;  
  end  
else if cmbHakAksesAcc.Text='-----Pilih Hak Akses -----' then  
  begin  
  application.MessageBox('Maaf Hak Akses Belum Diisi...!', 'Cek Input', MB_IconInformation);  
  cmbHakAksesAcc.SetFocus;  
  end  
else  
  begin  
  tbl_Account.Active:=true;  
  found:=false;  
  if(editdata=false)then begin  
  kode:=edtID.Text;  
  tbl_Account.First;  
  for i:=1 to tbl_Account.RecordCount do begin  
  if(trim(kode)=trim(tbl_Account.Fields[0].AsString))then begin  
    found:=true;  
  end else  
    tbl_Account.Next;  
  end;  
  if (found=true) then  
  begin  
  application.MessageBox('Maaf user ID Sudah Ada...!', 'Cek data', MB_IconInformation);  
  edtID.Text:='';  
  edtID.SetFocus;  
  end else begin
```

```

tbl_Account.append;
with tbl_Account do
begin
FieldValues['User_ID']:=edtID.Text;
FieldValues['Nama']:=edtNamaAcc.Text;
FieldValues['Jabatan']:=edtJabatanAcc.Text;
FieldValues['Telp']:=edtTelpAcc.Text;
FieldValues['Username']:=edtUsernameAcc.Text;
FieldValues['Password']:=edtpasswordAcc.Text;
FieldValues['Hak_Akses']:=cmbHakAksesAcc.Text;
Post;
end;
MessageDlg('Data Sudah Tersimpan....!',mtInformation, [mbOK], 0);
ActBatalAcc.Execute;
edtID.SetFocus;
editdata:=false;
end;

end else begin
tbl_Account.edit;
with tbl_Account do
begin
FieldValues['Nama']:=edtNamaAcc.Text;
FieldValues['Jabatan']:=edtJabatanAcc.Text;
FieldValues['Telp']:=edtTelpAcc.Text;
FieldValues['Username']:=edtUsernameAcc.Text;
FieldValues['Password']:=edtpasswordAcc.Text;
FieldValues['Hak_Akses']:=cmbHakAksesAcc.Text;
Post;
end;
MessageDlg('Data Sudah Tersimpan....!',mtInformation, [mbOK], 0);
ActBatalAcc.Execute;
edtID.SetFocus;
editdata:=false;
end;
end;

procedure TFormUtama.ActEditAccExecute(Sender: TObject);
begin
indexAcc:=1;
Application.CreateForm(TFormAcc,FormAcc);
FormAcc.Show;
FormUtama.Enabled:=false;
end;

procedure TFormUtama.ActHapusAccExecute(Sender: TObject);
begin
indexAcc:=2;
Application.CreateForm(TFormAcc,FormAcc);
FormAcc.Show;
FormUtama.Enabled:=false;
end;

//-----Data Penyakit-----

procedure TFormUtama.edtKodePenyChange(Sender: TObject);
begin
if edtKodePeny.Text='' then
LblSimpanPeny.Enabled:=false
else
LblSimpanPeny.Enabled:=true;
end;

procedure TFormUtama.ActBatalPenyakitExecute(Sender: TObject);
begin
edtKodePeny.Text:='';
edtNamaPeny.Text:='';
edtKetPeny.Text:='';
edtKodePeny.Enabled:=true;
edtKodePeny.SetFocus;
editdata:=false;
end;

procedure TFormUtama.ActSimpanPenyakitExecute(Sender: TObject);
var
found:Boolean;
kode:String;
i:integer;

```

```

begin
if edtKodePeny.Text="" then
begin
application.MessageBox('Maaf Kode Penyakit Belum Diisi...!', 'Cek Input', MB_IconInformation);
edtKodePeny.SetFocus;
end
else if edtNamaPeny.Text="" then
begin
application.MessageBox('Maaf Nama Penyakit Belum Diisi...!', 'Cek Input', MB_IconInformation);
edtNamaPeny.SetFocus;
end
else
begin
tbl_Penyakit.Active:=true;
found:=false;
if(editdata=false)then begin
kode:=edtKodePeny.Text;
tbl_Penyakit.First;
for i:=1 to tbl_Penyakit.RecordCount do begin
if(trim(kode)=trim(tbl_Penyakit.Fields[0].AsString))then begin
found:=true;
end else
tbl_Penyakit.Next;
end;
if (found=true) then
begin
application.MessageBox('Maaf Kode Penyakit Sudah Ada...!', 'Cek data', MB_IconInformation);
edtKodePeny.Text:="";
edtKodePeny.SetFocus;
end else begin
tbl_Penyakit.append;
with tbl_Penyakit do
begin
FieldValues['Kd_Penyakit']:=edtKodePeny.Text;
FieldValues['Nama_Penyakit']:=edtNamaPeny.Text;
FieldValues['Keterangan']:=edtKetPeny.Text;
Post;
end;
MessageDlg('Data Sudah Tersimpan...!', mtInformation, [mbOK], 0);
ActBatalPenyakit.Execute;
edtKodePeny.SetFocus;
editdata:=false;
end;
end else begin
tbl_Penyakit.edit;
with tbl_Penyakit do
begin
FieldValues['Nama_Penyakit']:=edtNamaPeny.Text;
FieldValues['Keterangan']:=edtKetPeny.Text;
Post;
end;
MessageDlg('Data Sudah Tersimpan...!', mtInformation, [mbOK], 0);
ActBatalPenyakit.Execute;
edtKodePeny.SetFocus;
editdata:=false;
end;
end;

procedure TFormUtama.ActEditPenyakit.Execute(Sender: TObject);
begin
indexPeny:=1;
Application.CreateForm(TFormPeny, FormPeny);
FormPeny.Show;
FormUtama.Enabled:=false;
end;

procedure TFormUtama.ActHapusPenyakit.Execute(Sender: TObject);
begin
indexPeny:=2;
Application.CreateForm(TFormPeny, FormPeny);
FormPeny.Show;
FormUtama.Enabled:=false;
end;

//=====Data Gejala=====

```

```

procedure TFormUtama.edtKodeGejChange(Sender: TObject);
begin
if edtKodeGej.Text="" then
  lblSimpanGej.Enabled:=false
else
  lblSimpanGej.Enabled:=true;
end;

procedure TFormUtama.ActBatalGejExecute(Sender: TObject);
begin
edtKodeGej.Text:="";
edtNamaGej.Text:="";
edtKetGej.Text:="";
edtKodeGej.Enabled:=true;
edtKodeGej.SetFocus;
editdata:=false;
end;

procedure TFormUtama.ActSimpanGejExecute(Sender: TObject);
var
found:Boolean;
kode:String;
i:integer;
begin
if edtKodeGej.Text="" then
  begin
  application.MessageBox('Maaf Kode Gejala Belum Diisi..!', 'Cek Input', MB_IconInformation);
  edtKodeGej.SetFocus;
  end
else if edtNamaGej.Text="" then
  begin
  application.MessageBox('Maaf Nama Gejala Belum Diisi..!', 'Cek Input', MB_IconInformation);
  edtNamaGej.SetFocus;
  end
else
  begin
tbl_Gejala.Active:=true;
found:=false;
if(editdata=false)then begin
kode:=edtKodeGej.Text;
tbl_Gejala.First;
for i=1 to tbl_Gejala.RecordCount do begin
if(trim(kode)=trim(tbl_Gejala.Fields[0].AsString))then begin
found:=true;
end else
tbl_Gejala.Next;
end;
if (found=true) then
begin
application.MessageBox('Maaf Kode Gejala Sudah Ada..!', 'Cek data', MB_IconInformation);
edtKodeGej.Text:="";
edtKodeGej.SetFocus;
end else begin
tbl_Gejala.append;
with tbl_Gejala do
begin
FieldValues['Kd_Gejala']:=edtKodeGej.Text;
FieldValues['Nama_Gejala']:=edtNamaGej.Text;
FieldValues['Keterangan']:=edtKetGej.Text;
Post;
end;
MessageDlg('Data Sudah Tersimpan ...!', mtInformation, [mbOK], 0);
ActBatalGej.Execute;
edtKodeGej.SetFocus;
editdata:=false;
end;

end else begin
tbl_Gejala.edit;
with tbl_Gejala do
begin
FieldValues['Nama_Gejala']:=edtNamaGej.Text;
FieldValues['Keterangan']:=edtKetGej.Text;
Post;
end;
MessageDlg('Data Sudah Tersimpan...!', mtInformation, [mbOK], 0);
ActBatalGej.Execute;
edtKodeGej.SetFocus;
editdata:=false;

```

```

end;
end;
end;

procedure TFormUtama.ActEditGejExecute(Sender: TObject);
begin
indexGej:=1;
Application.CreateForm(TFormGejala,FormGejala);
FormGejala.Show;
FormUtama.Enabled:=false;
end;

procedure TFormUtama.ActHapusGejExecute(Sender: TObject);
begin
indexGej:=2;
Application.CreateForm(TFormGejala,FormGejala);
FormGejala.Show;
FormUtama.Enabled:=false;
end;

//=====Diagnosa=====

procedure TFormUtama.edtKodeDiagChange(Sender: TObject);
begin
if edtKodeDiag.Text="" then
tblSimpanDiag.Enabled:=false
else
tblSimpanDiag.Enabled:=true;
end;

procedure TFormUtama.ActbatalDiagExecute(Sender: TObject);
begin
edtKodeDiag.Text:="";
BedtPenyakit.Text:="";
edtNmPeny.Text:="";
BedtGejala.Text:="";
edtNmGej.Text:="";
edtMB.Text:="";
edtMD.Text:="";
edtKodeDiag.Enabled:=true;
edtKodeDiag.SetFocus;
edtldata:=false;
end;

procedure TFormUtama.ActSimpanDiagExecute(Sender: TObject);
var
found:Boolean;
kode:String;
i:integer;
begin
if edtKodeDiag.Text="" then
begin
application.MessageBox('Maaf Kode Diagnosa Belum Diisi...!', 'Cek Input', MB_IconInformation);
edtKodeDiag.SetFocus;
end
else if BedtPenyakit.Text="" then
begin
application.MessageBox('Maaf Nama Penyakit Belum Diisi...!', 'Cek Input', MB_IconInformation);
BedtPenyakit.SetFocus;
end
else if BedtGejala.Text="" then
begin
application.MessageBox('Maaf Nama Gejala Belum Diisi...!', 'Cek Input', MB_IconInformation);
BedtGejala.SetFocus;
end
else if edtMB.Text="" then
begin
application.MessageBox('Maaf Nilai MB Belum Diisi...!', 'Cek Input', MB_IconInformation);
edtMB.SetFocus;
end
else if edtMD.Text="" then
begin
application.MessageBox('Maaf Nilai MD Belum Diisi...!', 'Cek Input', MB_IconInformation);
edtMD.SetFocus;
end
else
begin
tbl_Diagnosa.Active:=true;
found:=false;

```

```

if(editdata=false)then begin
kode:=edtKodeDiag.Text;
tbl_Diagnosa.First;
for i:=1 to tbl_Diagnosa.RecordCount do begin
if(trim(kode)=trim(tbl_Diagnosa.Fields[0].AsString))then begin
found:=true;
end else
tbl_Diagnosa.Next;
end;
if (found=true) then
begin
application.MessageBox('Maaf Kode Diagnosa Sudah Ada...!', 'Cek data', MB_IconInformation);
edtKodeDiag.Text:='';
edtKodeDiag.SetFocus;
end else begin
tbl_Diagnosa.append;
with tbl_Diagnosa do
begin
FieldValues['Kd_Diagnosa']:=edtKodeDiag.Text;
FieldValues['Kd_Penyakit']:=BedtPenyakit.Text;
FieldValues['Kd_Gejala']:=BedtGejala.Text;
FieldValues['MB']:=edtMB.Text;
FieldValues['MD']:=edtMD.Text;
Post;
end;
MessageDlg('Data Sudah Tersimpan...!', mtInformation, [mbOK], 0);
ActbatalDiag.Execute;
edtKodeDiag.SetFocus;
editdata:=false;
end;

end else begin
tbl_Diagnosa.edit;
with tbl_Diagnosa do
begin
FieldValues['Kd_Penyakit']:=BedtPenyakit.Text;
FieldValues['Kd_Gejala']:=BedtGejala.Text;
FieldValues['MB']:=edtMB.Text;
FieldValues['MD']:=edtMD.Text;
Post;
end;
MessageDlg('Data Sudah Tersimpan...!', mtInformation, [mbOK], 0);
ActbatalDiag.Execute;
edtKodeDiag.SetFocus;
editdata:=false;
end;
end;
vw_Diagnosa.Active:=false;
vw_Diagnosa.Active:=true;
end;

```

```

procedure TFormUtama.BedtPenyakitButtonClick(Sender: TObject);
begin
indexPeny:=3;
Application.CreateForm(TFormPeny, FormPeny);
FormPeny.Show;
FormUtama.Enabled:=false;
end;

```

```

procedure TFormUtama.BedtPenyakitChange(Sender: TObject);
var
i:integer;
begin
tbl_Penyakit.Active:=false;
tbl_Penyakit.Active:=true;
for i:=1 to tbl_Penyakit.RecordCount do begin
if (BedtPenyakit.Text=tbl_Penyakit.FieldValues['Kd_Penyakit']) then begin
edtNmPeny.Text:=tbl_Penyakit.FieldValues['Nama_Penyakit'];
end else
tbl_Penyakit.Next;
end;
end;
end;

```

```

procedure TFormUtama.BedtGejalaButtonClick(Sender: TObject);
begin
indexGej:=3;
Application.CreateForm(TFormGejala, FormGejala);
FormGejala.Show;

```

```

FormUtama.Enabled:=false;
end;

procedure TFormUtama.BedtGejalaChange(Sender: TObject);
var
i:integer;
begin
tbl_Gejala.Active:=false;
tbl_Gejala.Active:=true;
for i:=1 to tbl_Gejala.RecordCount do begin
if (BedtGejala.Text=tbl_Gejala.FieldValues['Kd_Gejala']) then begin
edtNmGej.Text:=tbl_Gejala.FieldValues['Nama_Gejala'];
end else
tbl_Gejala.Next;
end;
end;
end;

```

```

procedure TFormUtama.ActEditDiagExecute(Sender: TObject);
begin
indexDiag:=1;
Application.CreateForm(TFormDiagnosa,FormDiagnosa);
FormDiagnosa.Show;
FormUtama.Enabled:=false;
end;

```

```

procedure TFormUtama.ActHapusDiagExecute(Sender: TObject);
begin
indexDiag:=2;
Application.CreateForm(TFormDiagnosa,FormDiagnosa);
FormDiagnosa.Show;
FormUtama.Enabled:=false;
end;

```

//----- Login -----//

```

procedure TFormUtama.ActLoginExecute(Sender: TObject);
begin
Application.CreateForm(TFormLogin,FormLogin);
FormLogin.Show;
FormUtama.Enabled:=false;
end;

```

```

procedure TFormUtama.ActLogoutExecute(Sender: TObject);
begin
if MessageDlg('Anda Yakin Akan LogOut ?',
mtConfirmation, [mbYES, mbNO], 0) = mrYES then begin

```

```

GB_Awal.Visible:=true;
Gb_Penyakit.Visible:=false;
Gb_Gejala.Visible:=false;
Gb_Account.Visible:=false;
Gb_Diagnosa.Visible:=false;
Gb_Informasi.Visible:=false;
Gb_Konsultasi.Visible:=false;

```

```

FormUtama.GBUser.Items[0].Enabled:=true;
FormUtama.GBUser.Items[4].Enabled:=false;

```

```

FormUtama.GBAdmin.Items[0].Enabled:=false;
FormUtama.GBAdmin.Items[1].Enabled:=false;
FormUtama.GBAdmin.Items[2].Enabled:=false;
FormUtama.GBAdmin.Items[3].Enabled:=false;

```

```

FormUtama.GBLaporan.Items[0].Enabled:=false;
FormUtama.GBLaporan.Items[1].Enabled:=false;
FormUtama.GBLaporan.Items[2].Enabled:=false;
FormUtama.GBLaporan.Items[3].Enabled:=false;
end;
end;

```

```

procedure TFormLogin.IblLoginClick(Sender: TObject);
var
i:integer;
ketemu:boolean;
nama_akses:String;
begin
ketemu:=false;
FormUtama.tbl_Account.Active:=true;
FormUtama.tbl_Account.First;

```

```

for i:=1 to FormUtama.tbl_Account.RecordCount do begin
if FormUtama.tbl_Account.FieldValues['Username']=edtUsernameAcc.Text then begin
if FormUtama.tbl_Account.FieldValues['Password']=edtPasswordAcc.Text then begin
nama:=FormUtama.tbl_Account.FieldValues['Nama'];
akses:=FormUtama.tbl_Account.FieldValues['Hak_Akses'];
FormUtama.stsPnlForm.Caption:=nama+' anda login sebagai '+akses;
if akses='Administrator' then begin
FormUtama.GBUser.Items[0].Enabled:=false;
FormUtama.GBUser.Items[4].Enabled:=true;

FormUtama.GBAdmin.Items[0].Enabled:=true;
FormUtama.GBAdmin.Items[1].Enabled:=true;
FormUtama.GBAdmin.Items[2].Enabled:=true;
FormUtama.GBAdmin.Items[3].Enabled:=false;

FormUtama.GBLaporan.Items[0].Enabled:=true;
FormUtama.GBLaporan.Items[1].Enabled:=true;
FormUtama.GBLaporan.Items[2].Enabled:=true;
FormUtama.GBLaporan.Items[3].Enabled:=true;
FormUtama.lblEditInfo.Enabled:=true;

end else begin

FormUtama.GBUser.Items[0].Enabled:=false;
FormUtama.GBUser.Items[4].Enabled:=true;

FormUtama.GBAdmin.Items[0].Enabled:=false;
FormUtama.GBAdmin.Items[1].Enabled:=false;
FormUtama.GBAdmin.Items[2].Enabled:=false;
FormUtama.GBAdmin.Items[3].Enabled:=true;

FormUtama.GBLaporan.Items[0].Enabled:=true;
FormUtama.GBLaporan.Items[1].Enabled:=true;
FormUtama.GBLaporan.Items[2].Enabled:=true;
FormUtama.GBLaporan.Items[3].Enabled:=true;
FormUtama.lblEditInfo.Enabled:=false;
FormUtama.lblSimpanInfo.Enabled:=false;

end;
ketemu:=true;
close
end
end
else begin
FormUtama.tbl_Account.Next;
end;
end;
if ketemu=false then begin
MessageDlg('Username atau Password tidak valid!', mtError, [mbOK], 0);
edtUsernameAcc.Text:='';
edtPasswordAcc.Text:='';
edtUsernameAcc.SetFocus;
end;
end;

```

//-----Info-----

```

procedure TFormUtama.lblEditInfoClick(Sender: TObject);
begin
MemoInfo.ReadOnly:=false;
FormUtama.lblEditInfo.Enabled:=false;
FormUtama.lblSimpanInfo.Enabled:=true;
end;

```

```

procedure TFormUtama.lblSimpanInfoClick(Sender: TObject);
begin
tbl_memo.Active:=true;
tbl_memo.Edit;
tbl_memo.FieldValues['memo']:=MemoInfo.Lines.Text;
tbl_memo.Post;
MemoInfo.ReadOnly:=true;
FormUtama.lblEditInfo.Enabled:=true;
FormUtama.lblSimpanInfo.Enabled:=false;
end;

```

//-----Laporan-----

```

procedure TFormUtama.ActLapUserExecute(Sender: TObject);
begin

```

```

Application.CreateForm(TFormLapUser,FormLapUser);
FormLapUser.QuickRep1.Preview;
end;

procedure TFormUtama.ActLapPenyakitExecute(Sender: TObject);
begin
Application.CreateForm(TFormLapPenyakit,FormLapPenyakit);
FormLapPenyakit.QuickRep1.Preview;
end;

procedure TFormUtama.ActLapGejalaExecute(Sender: TObject);
begin
Application.CreateForm(TFormLapGejala,FormLapGejala);
FormLapGejala.QuickRep1.Preview;
end;

procedure TFormUtama.ActLapDiagnosaExecute(Sender: TObject);
begin
Application.CreateForm(TFormLapDiagnosa,FormLapDiagnosa);
FormLapDiagnosa.QuickRep1.Preview;
end;

//=====Konsultasi=====

procedure TFormUtama.edtNamaGejKonsChange(Sender: TObject);
begin
if edtNamaGejKons.Text="" then
  btnTambah.Enabled:=false
else
  btnTambah.Enabled:=true;
end;

procedure TFormUtama.HedtkodeGejKonsButtonClick(Sender: TObject);
begin
indexGej:=4;
Application.CreateForm(TFormGejala,FormGejala);
FormGejala.Show;
FormUtama.Enabled:=false;
end;

procedure TFormUtama.BedtkodeGejKonsChange(Sender: TObject);
var
i:integer;
begin
tbl_Gejala.Active:=false;
tbl_Gejala.Active:=true;
for i:=1 to tbl_Gejala.RecordCount do begin
if (BedtkodeGejKons.Text=tbl_Gejala.FieldValues['Kd_Gejala']) then begin
  edtNamaGejKons.Text:=tbl_Gejala.FieldValues['Nama_Gejala'];
end else
tbl_Gejala.Next;
end;
end;

procedure TFormUtama.btnTambahClick(Sender: TObject);
var
z:integer;
found:boolean;
begin
found:=false;
for z:=0 to ListGejala.Count-1 do begin
if (edtNamaGejKons.Text=ListGejala.Items.Strings[z]) then begin
found:=true;
end;
end;

if (found=true) then begin
MessageDlg('Maaf Nama Gejala Sudah Ada...!',mtInformation, [mbOK], 0);
BedtkodeGejKons.Text="";
edtNamaGejKons.Text="";
end else begin
ListGejala.Items.Add(edtNamaGejKons.Text);
BedtkodeGejKons.Text="";
edtNamaGejKons.Text="";
if ListGejala.Count>0 then begin
btnProses.Enabled:=true
end else
btnProses.Enabled:=false;
end;
end;

```

```

end;

procedure TFormUtama.btnhapusClick(Sender: TObject);
begin
ListGejala.Items.Delete(indexList);
btnhapus.Enabled:=false;
if ListGejala.Count>0 then begin
btnProses.Enabled:=true
end else
btnProses.Enabled:=false;
end;

procedure TFormUtama.ListGejalaClick(Sender: TObject);
begin
indexList:=ListGejala.ItemIndex;
btnhapus.Enabled:=true;
end;

procedure TFormUtama.btnBatalClick(Sender: TObject);
begin
edtNamaPasien.Text:='';
edtUmurPasien.Text:='';
cmbKelPasien.Text:='---Pilih Jenis Kelamin---';
BodtkkodeGejKons.Text:='';
edtNamaGejKons.Text:='';
btnTambah.Enabled:=false;
btnhapus.Enabled:=false;
btnProses.Enabled:=false;
ListGejala.Clear;
end;

procedure TFormUtama.btnProsesClick(Sender: TObject);
var
i,j,k,l:integer;
totMB,totMD,totMBAwal,TotMBAkhir,totMDAwal,TotMDAkhir,CF,CF1,CF2,CF3,CF4,CF5:Real;
kodePeny,kodeGej,gejala,MBS1,MDS1:string;
found:boolean;
begin
tbl_Penyakit.Active:=true;
tbl_Gejala.Active:=true;
tbl_Diagnosa.Active:=true;
tbl_Penyakit.First;
tbl_Gejala.First;
tbl_Diagnosa.First;
CF:=0;CF1:=0;CF2:=0;CF3:=0;CF4:=0;CF5:=0;

for i:=1 to tbl_Penyakit.RecordCount do begin
kodePeny:=tbl_Penyakit.FieldValues['Kd_Penyakit'];

for j:=0 to ListGejala.Count-1 do begin
gejala:=ListGejala.Items.Strings[j];

tbl_Gejala.First;
for k:=1 to tbl_Gejala.RecordCount do begin
if (trim(gejala)=trim(tbl_Gejala.Fields[1].AsString)) then begin
kodeGej:=tbl_Gejala.FieldValues['Kd_Gejala'];
end else begin
tbl_Gejala.Next;
end;
end;
found:=false;
tbl_Diagnosa.First;
for l:=1 to tbl_Diagnosa.RecordCount do begin
if (trim(kodePeny)=trim(tbl_Diagnosa.Fields[1].AsString)) and
(trim(kodeGej)=trim(tbl_Diagnosa.Fields[2].AsString)) then begin
found:=true;
end else begin
tbl_Diagnosa.Next;
end;
end; //end for diagnosa

if (found=true) then begin
TotMBAkhir:=StrToFloat(tbl_Diagnosa.FieldValues['MB']);
TotMDAkhir:=StrToFloat(tbl_Diagnosa.FieldValues['MD']);
TotMB:=totMBAwal+TotMBAkhir*(1-totMBAwal);
TotMD:=totMDAwal+TotMDAkhir*(1-totMDAwal);
totMBAwal:=totMB;
totMDAwal:=totMD;

```