

**SISTEM PAKAR UNTUK MENDAGNOSA PENYAKIT HATI
PADA ORGAN HATI MANUSIA DENGAN METODE
DEMPYER SHAFER**

SKRIPSI



**Disusun Oleh :
JUMIATI TARADI
12.18.161**

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
2016**

**LEMBAR PERSETUJUAN
SISTEM PAKAR UNTUK MENDAGNOSA PENYAKIT HATI PADA
ORGAN HATI MANUSIA DENGAN METODE DEMPTER
SHAFR**

SKRIPSI

*Disusun dan Diajukan untuk melengkapi dan memenuhi persyaratan guna
mencapai Gelar Sarjana Komputer Strata Satu (S-1)*



Disusun Oleh :

JUMIATI TARADI

NIM : 12.18.161

Diperiksa dan Disetujui

Dosen Pembimbing I

Dosen Pembimbing II

Joseph Dedy Irawan, ST, MT
NIP. 197404162005011002

Sandy Nataly Mantja, S.Kom
NIP.P. 1030800418

Mengetahui

Ketua Program Studi Teknik Informatika S-1

Joseph Dedy Irawan, ST, MT
NIP. 197404162005011002

**PROGRAM STUDI TEKNIK INFORMATIKA S-1
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG**

2016

**LEMBARAN KEASLIAN
PERNYATAAN KEASLIAN SKRIPSI**

Saya yang bertanda tangan dibawah ini :

Nama : Jumiati Taradi

NIM : 12.18.161

Program Studi : Teknik Informatika S-1

Fakultas : Fakultas Teknologi Industri

Menyatakan dengan sesungguhnya bahwa Skripsi saya yang berjudul :

“Sistem Pakar Untuk Mendiagnosa Penyakit Hati Pada Organ Hati Manusia Dengan Metode Dempster Shafer “

Adalah skripsi sendiri bukan duplikasi serta mengutip atau menyadur seluruhnya karya orang lain kecuali dari sumber aslinya.

Malang, 12 Desember 2015

Yang membuat pernyataan


Jumiati Taradi
12.18.161



Abstraksi

SISTEM PAKAR UNTUK MENDIAGNOSA PENYAKIT HATI PADA ORGAN HATI MANUSIA DENGAN METODE *DEMPSTER SHAFER*

Jumiati Taradi

Program Studi Teknik Informatika S-1
Fakultas Teknologi Industri
Institut Teknologi Nasional Malang
Jl. Raya Karanglo Km. 2 Tasikmadu Malang
Email : Jumiataradi@gmail.com

Dosen Pembimbing : 1. Joseph Dedy Irawan, ST. MT

2. Sandy Nataly Mantja, S.Kom

Abstrak

Hati memiliki banyak fungsi yang kompleks dan beragam, fungsi hati adalah sebagai filter semua darah yang datang dari usus melalui vena porta, kemudian menyimpannya dan mengubah bahan-bahan makanan yang diterima vena porta. Dari penjelasan tersebut, dapat diketahui peran atau fungsi hati, termasuk gangguan dari fungsi hati itu sendiri. Akan tetapi, masyarakat pada umumnya tidak mengetahui informasi tentang bahaya dari penyakit hati yang disebabkan oleh virus atau bakteri yang terbawa oleh makanan dan minuman yang mereka konsumsi sehari-hari, sehingga sering kali masyarakat mengabaikan hal-hal yang dapat merusak fungsi hati tersebut dan dapat menimbulkan penyakit hati. Jenis-jenis penyakit hati antara lain yaitu Hepatitis, Sirosis, Kanker Hati.

Penyakit-penyakit hati akut akan banyak mempengaruhi fungsi-fungsi hati. Oleh karena itu, gejala-gejala klinik yang timbul pada kerusakan hati akut menjadi kompleks, sehingga seorang pakar penyakit dalam terkadang mengalami kesulitan dalam menentukan penyakit yang diderita oleh pasien karena adanya beberapa gejala-gejala yang mirip pada beberapa penyakit. Maka dari itu penelitian ini dilakukan bertujuan untuk membangun sebuah system pakar yang dapat digunakan untuk menentukan penyakit hati, serta memberikan informasi kepada masyarakat tentang penyakit hati dari gejala, penyebab dan solusinya. Metode yang digunakan dalam aplikasi system pakar untuk mendiagnosa penyakit hati adalah metode Dempster Shafer.

Sistem pakar dengan metode Dempster-Shafer dapat dipergunakan untuk mendiagnosa penyakit organ hati pada manusia dengan masukkan data berupa pertanyaan yang dirasakan oleh pasien, hasil diagnosa yang sama antara perhitungan sistem dengan menggunakan metode Dempster-Shafer dan pengetahuan pakar. Hasil pengujian fungsional sistem dari semua form didapatkan hasil bahwa semua fungsi-fungsi yang ada dapat berjalan dan berfungsi 100%.

Kata Kunci : *system Pakar, Penyakit Hati, Dempster Shafer*

KATA PENGANTAR

Puji syukur alhamdulillah penulis panjatkan kehadirat Allah SWT atas berkat, rahmat, taufik dan hidayah-Nya, penyusunan skripsi yang berjudul “**Sistem Pakar Untuk Mendiagnosa Penyakit Hati Pada Organ Hati Manusia Dengan Metode Dempster Shafer**” dapat diselesaikan dengan baik. Shalawat serta salam senantiasa tercurah kepada junjungan Nabi besar Muhammad SAW beserta keluarga, sahabat, kerabat, dan pengikut beliau hingga akhir zaman.

Penulis menyadari bahwa dalam proses penulisan skripsi ini banyak mengalami kendala, namun berkat bantuan, bimbingan, kerjasama dari berbagai pihak dan berkah dari Allah SWT sehingga kendala – kendala yang dihadapi tersebut dapat diatasi. Untuk itu penulis menyampaikan ucapan terima kasih dan penghargaan kepada Bapak dan Ibu yang senantiasa mendoakan, memberikan bantuan moril, materi, dan nasehat selama penulis menjalani pendidikan. Selanjutnya ucapan terima kasih penulis sampaikan pula kepada:

1. Bapak Dr. Ir. Lalu Mulyadi, MTA, selaku Rektor Institut Teknologi Nasional Malang.
2. Bapak Ir. Anang Subardi, MT, selaku Dekan Fakultas Teknologi Industri, Institut Teknologi Nasional Malang.
3. Bapak Joseph Dedy Irawan, ST, MT, selaku Ketua Program Studi Teknik Informatika, Institut Teknologi Nasional Malang, dan selaku Dosen Pembimbing I, yang selalu memberikan bimbingan dan masukan.
4. Bapak Sonny Prasetyo, ST, MT, selaku Sekertaris Program Studi Teknik Informatika, Institut Teknologi Nasional Malang.
5. Sandy Nataly Mantja, S.Kom, selaku Dosen Pembimbing II, yang selalu memberikan bimbingan dan masukan.
6. Semua dosen Program Studi Teknik Informatika yang telah membantu dalam penulisan dan masukan.
7. Semua teman-teman yang telah memberikan doa dan dukungannya dalam menyelesaikan skripsi ini.

Dengan segala kerendahan hati, penulis menyadari masih banyak terdapat kekurangan-kekurangan, sehingga penulis mengharapkan adanya saran dan kritik yang bersifat membangun demi kesempurnaan skripsi ini.

Malang, 12 Januari 2016

Penulis

DAFTAR ISI

HALAMAN JUDUL	i
LEMBAR PERSETUJUAN.....	ii
LEMBARAN KEASLIAN	iii
Abstraksi	iv
KATA PENGANTAR	v
DAFTAR ISI.....	vii
DAFTAR GAMBAR.....	ix
DAFTAR TABEL	xi
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 Penelitian.....	2
1.6 Sistematika Penulisan	3
BAB II LANDASAN TOERI	4
2.1 Sistem Pakar.....	4
2.2 Metode Dempstr Shafer.....	5
2.3 Visual Studio 2010 (VB.NET).....	9
2.3.1 kelebihan dan kekurangan visual basic.....	10
2.3.2 Sejarah Visual Basic.....	11
2.4 SQL Server 2008	15
BAB III ANALISIS DAN PERANCANGAN	18
3.1 Desain Sistem.....	18
3.1.1 Perancangan Blok Diagram	18
3.1.2 Data Flow Diagram	18

3.1.2.1 DFD Level 0.....	19
3.1.2.2 DFD Lever 1.....	19
3.1.2 Flowchart.....	20
3.1.3 Rancangan Data Base.....	21
3.1.3 Struktur Menu.....	24
3.1.4 Perancangan Tampilan.....	24
BAB IV IMPLEMENTASI DAN PENGUJIAN.....	28
4.1 Implementasi.....	28
4.1.1 <i>Form</i> Utama.....	28
4.1.2 <i>Form</i> Login.....	29
4.1.3 <i>Form</i> data Gejala.....	29
4.1.4 <i>Form</i> data Penyakit.....	31
4.1.5 <i>Form</i> data Aturan.....	33
4.1.6 <i>Form</i> Konsultasi.....	35
4.2 Pengujian Sistem.....	35
4.2.1 Pengujian oleh Pembuat.....	36
BAB V PENUTUP.....	38
5.1 Kesimpulan.....	38
5.2 Saran.....	38
DAFTAR PUSTAKA.....	39
LAMPIRAN.....	40

DAFTAR GAMBAR

Gambar 2. 1 Konsep dasar sistem pakar.....	4
Gambar 2. 2 Tampilan awal <i>Visual Studio 2010</i>	10
Gambar 2. 3. Tampilan awal <i>Microsoft SQL server 2008</i>	17
Gambar 3. 1 Blok Diagram.....	18
Gambar 3. 2 DFD level 0.....	19
Gambar 3. 3 DFD level 1.....	19
Gambar 3. 4 <i>Flowchart</i> sistem.....	20
Gambar 3. 5 <i>Flowchart Inferensi Dempster-Shafer</i>	21
Gambar 3. 6. Struktur Menu.....	24
Gambar 3. 7. Rancangan <i>Form</i> Utama.....	24
Gambar 3. 8. Rancangan <i>form login</i>	25
Gambar 3. 9. Rancangan tampilan <i>form</i> penyakit.....	25
Gambar 3. 10. Rancangan tampilan <i>form</i> aturan.....	26
Gambar 3. 11. Rancangan tampilan <i>form</i> gejala.....	26
Gambar 3. 12. Rancangan <i>form</i> konsultasi.....	27
Gambar 4. 1. Tampilan <i>Form</i> Utama.....	28
Gambar 4. 2 <i>Form Login</i>	29
Gambar 4. 3 Konfirmasi <i>login</i> gagal.....	29
Gambar 4. 4 Tampilan Data Gejala.....	30
Gambar 4. 5 Tampilan <i>input</i> data gejala.....	30
Gambar 4. 6 Tampilan <i>edit</i> data gejala.....	31
Gambar 4. 7 Tampilan <i>form</i> data penyakit.....	31
Gambar 4. 8 Tampilan hapus data penyakit.....	32
Gambar 4. 9 Tampilan simpan data gejala.....	32
Gambar 4. 10 Tampilan data yang sudah tersimpan.....	33
Gambar 4. 11 Tampilan <i>form</i> data aturan.....	33
Gambar 4. 12 Tampilan data aturan.....	34

Gambar 4. 13 Tampilan data aturan yang sudah tersimpan.....	34
Gambar 4. 14 Tampilan hapus data aturan.	35
Gambar 4. 15 Tampilan <i>form</i> konsultasi	35

DAFTAR TABEL

Tabel 2. 1	<i>Range Belief dan Plausibility</i>	5
Tabel 2. 2	Tabel perhitungan dua gejala.....	9
Tabel 3. 1.	Tabel login.....	21
Tabel 3. 2	Tabel gejala	22
Tabel 3. 3	Tabel penyakit	22
Tabel 3. 4	Tabel pertanyaan	22
Tabel 3. 5	Tabel aturan	23
Tabel 3. 6	Tabel hasil konsultasi	23
Tabel 3. 7	Tabel diagnosa.....	23
Tabel 4. 1	Hasil Pengujian Fungsi.....	36
Tabel 4. 2	Pengujian Terhadap Pengguna.	37

BAB I

PENDAHULUAN

1.1 Latar Belakang

Saat ini perkembangan teknologi informasi ini telah merambah ke berbagai sektor termasuk kesehatan. Meskipun dunia kesehatan dan medis merupakan bidang yang bersifat *information-intensive*, akan tetapi penggunaan teknologi komputer relatif tertinggal. Sering perkembangan ilmu kedokteran mengalami kemajuan yang sangat pesat yang ditandai dengan ditemukannya penyakit-penyakit baru yang belum teridentifikasi sebelumnya. Salah satu penyakit yang berkembang saat ini yaitu penyakit pada organ hati.

Kesehatan organ hati sangat penting maknanya bagi tubuh manusia. Hati sebagai organ yang memiliki tugas utama sebagai penetral racun ditubuh menjadikan racun-racun yang selama ini masuk melalui tubuh kita dari makanan atau lingkungan mampu dinetralisir oleh hati. Manusia tidak akan hidup tanpa organ hati tersebut. Salah satu penyakit yang menyerang hati adalah Hepatitis yang terdiri atas berbagai macam tipe.

Perhitungan ketidakpastian diperlukan dalam sistem pakar untuk dapat meyakinkan pasien (*user*) dalam hasil diagnosa yang dihasilkan sehingga sistem pakar yang dibuat benar-benar seperti layaknya diagnosa seorang pakar atau dokter. Perhitungan ketidakpastian dalam sistem pakar dapat dilakukan dengan beberapa metode ketidakpastian. Salah satunya adalah dengan menggunakan metode *Dempster Shafer*. Metode ini dapat digunakan untuk mencari persentase kemungkinan penyakit yang diderita pasien (*user*) dengan mendiagnosa gejala yang dirasakan. Diharapkan dengan penggunaan metode ini dapat meminimalisir ketidakpastian sehingga dapat menghasilkan diagnosa yang valid. Banyak jenis penyakit yang disebabkan oleh orga hati pada manusia. Mengetahui jenis penyakit tersebut terlebih dahulu kita mengetahui gejala yang dialami. Berdasar permasalahan yang ditimbulkan di atas maka penulis membuat

program Sistem Pakar Untuk Mengdiagnosa Penyakit Hati, agar menderit dapat mengetahui jenis penyakit yang di deritanya.

1.2 Rumusan Masalah

Bagaimana cara untuk menerapkan sistem pakar untuk mendiagnosa penyakit organ hati pada manusia dengan metode *Dempster Shafer* ?

1.3 Batasan Masalah

Adapun batasan masalah dalam penelitian ini adalah sebagai berikut :

1. Metode yang digunakan adalah *Dempster Shafer*.
2. Menggunakan bahasa pemrograman *VB.NET* 2010
3. Data yang digunakan adalah *Ms.sql server* 2008
4. Data yang digunakan adalah berupa data penyakit hati dari Rumah sakit.

1.4 Tujuan

Adapun tujuan dari membuat sistem pakar untuk mendiagnosa penyakit organ hati ini adalah sebagai beriku :

1. Agar bisa informasi kepada user tentang gejala dan memberikan informasi tentang penyakit yang diderita oleh user.
2. Mempermudah pakar untuk menginputkan data. Baik data gejala, data penyakit dan data solusi.

1.5 Metodologi Penelitian

Dalam peneletian ini terdapat beberapa metode penelitian yang penulis tuliskan untuk mendukung penelitian yang penulis lakukan yaitu, sebagai berikut.

1. Studi Literatur

Studi literatur dalam penelitian ini dilakukan untuk mendukung teori yang digunakan dalam penelitian ini dengan mengumpulkan referensi-referensi dari sumber yang bisa dipertanggung jawabkan baik berupa buku, karya tulis ilmiah, paper, jurnal, dan artikel yang berkaitan dengan teori yang penelitian yang penulis bahas.

2. **Analisa Permasalahan**
Analisis permasalahan dilakukan untuk mempelajari permasalahan yang ada dan menyelesaikan permasalahan dengan menggunakan teori-teori yang telah dipelajari dalam studi literatur
3. **Perancangan Sistem**
Pada tahap ini dilakukan perancangan sistem seperti perancangan blok diagram, perancangan struktur tabel dalam database, perancangan struktur menu, perancangan *flowchart* aplikasi, dan perancangan *Data Flow Diagram* (DFD) dari Aplikasi Sistem Pakar Untuk Mendiagnosa Penyakit Pada Organ Hati Manusia Dengan Metode *Dempster-Shafer*.

1.6 Sistematika Penulisan

Untuk mempermudah memahami pembahasan pada penulisan skripsi ini, maka sistematika penulisan yang di peroleh sebagai berikut:

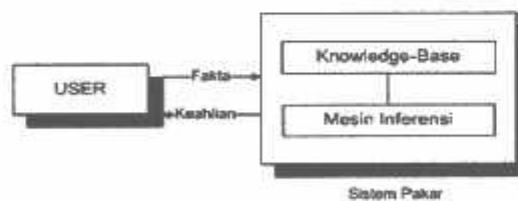
- BAB I** : Pendahuluan
Berisi latar belakang, perumusan masalah, batasan masalah, tujuan penelitian, metode penelitian, dan sistematika penulisan
- BAB II** : Landasan Teori
Berisi dasar teori mengenai permasalahan yang berhubungan dengan penelitian ini
- BAB III** : Analisa dan Perancangan Sistem
Menguraikan tentang analisa sistem dan perancangan sistem yang akan dibuat dalam penelitian.
- BAB IV** : Implementasi dan Pengujian
Menguraikan implementasi dan hasil pengujian sistem yang sudah dilakukan dalam penelitian.
- BAB V** : Kesimpulan dan Saran
berisi kesimpulan yang diambil berkaitan dengan sistem yang dikembangkan dan saran untuk pengembangan sistem lebih lanjut.
-

BAB II LANDASAN TOERI

2.1 Sistem Pakar

Sistem pakar merupakan program komputer untuk dapat meniru proses pemikiran dan pengetahuan pakar untuk menyelesaikan suatu masalah yang spesifik. Implementasi sistem pakar banyak digunakan untuk kepentingan masyarakat karena sistem pakar dipandang sebagai cara penyimpanan pengetahuan pakar dalam bidang tertentu ke dalam suatu program, sehingga dapat memberikan keputusan dan melakukan penalaran secara cerdas. Sistem pakar merupakan cabang dari kecerdasan buatan dan juga merupakan bidang ilmu yang muncul seiring perkembangan ilmu komputer saat ini. Sistem ini bekerja untuk mengadopsi pengetahuan manusia ke komputer yang menghubungkan dasar pengetahuan dengan sistem inferensi untuk menggantikan fungsi seorang pakar dalam menyelesaikan suatu masalah.

Dalam penyusunannya, sistem pakar mengkombinasikan kaidah-kaidah penarikan kesimpulan (*inference rules*) dengan basis pengetahuan tertentu yang diberikan oleh satu atau lebih pakar dalam bidang tertentu. Kombinasi dari kedua hal tersebut disimpan dalam komputer, yang selanjutnya digunakan dalam proses pengambilan keputusan untuk penyelesaian masalah tertentu. Sistem pakar dibuat dengan mendapatkan pengetahuan dari seorang pakar, kemudian dilakukan pengkodean ke bentuk yang dapat diproses oleh komputer untuk menyelesaikan persoalan yang sejenis. Sistem pakar sangat tergantung pada suatu bidang dalam menyusun penyelesaian persoalan yang dihadapi oleh sistem. Konsep dasar sistem pakar dapat dilihat pada Gambar 2.1.



Gambar 2. 1 Konsep dasar sistem pakar

2.2 Metode Dempster Shafer

Metode Dempster Shafer merupakan generalisasi dari teori Bayesian mengenai probabilitas subjektif. Ketika teori Bayesian memerlukan probabilitas untuk setiap pertanyaan yang ada, teori *Dempster-Shafer* mengizinkan untuk menentukan derajat kepercayaan (*degree of belief*) untuk satu pertanyaan pada probabilitas untuk pertanyaan yang berhubungan. Secara umum teori Dempster-Shafer ditulis dalam suatu interval : [*Belief,Plausibility*]

Belief (Bel) adalah ukuran kekuatan *evidence* dalam mendukung suatu hipotesa, jika bernilai 0 maka mengindikasikan bahwa tidak ada *evidence*, dan jika bernilai 1 menunjukkan adanya kepastian. Menurut Giarratano dan Riley fungsi *belief* dapat diformulasikan sebagai:

$$Bel(X) = \sum_{Y \subseteq X} m(Y) \dots \dots \dots (2.1)$$

Maka *Plausibility* (Pl), yang dinotasikan sebagai

$$Pls(X) = 1 - Bel(X') = 1 - \sum_{Y \subseteq X'} m(Y) \dots \dots \dots (2.2)$$

Dengan : $Bel(X) = Belief(X)$

$Pls(X) = Plausibility(X)$

$m(X) = mass\ function\ dari\ (X)$ $m(Y) = mass\ function\ dari\ (Y)$

Plausibility juga bernilai 0 sampai 1, jika kita yakin akan X' maka dapat dikatakan $Belief(X') = 1$ sehingga dari rumus di atas nilai $Pls(X) = 0$. Beberapa kemungkinan range antara *Belief* dan *Plausibility* adalah sebagai berikut :

Tabel 2. 1 Range *Belief* dan *Plausibility*

Kemungkinan	Keterangan
[1,1]	Semua Benar
[0,0]	Semua Salah
[0,1]	Ketidakpastian
[Bel,1] where $0 < Bel < 1$	Cenderung Mendukung
[0,Pls] where $0 < Pls < 1$	Cenderung Menolak
[Bel,Pls] where $0 < Bel \leq Pls < 1$	Cenderung Mendukung dan Menolak

Pada teori *Dempster-Shafer* juga dikenal adanya *frame of discernment* yang dinotasikan dengan Θ . FOD ini merupakan semesta pembicaraan dari sekumpulan hipotesis sehingga sering disebut dengan *environment*.

$$\Theta = \{\theta_1, \theta_2, \dots, \theta_n\} \dots \dots \dots (2.3)$$

Dengan : Θ = FOD atau *environment*

$\theta_1 \dots \theta_n$ = elemen/unsur bagian dalam *environment*

Environment mengandung elemen-elemen yang menggambarkan kemungkinan sebagai jawaban dan hanya ada satu yang akan sesuai dengan jawaban yang dibutuhkan. Kemungkinan ini dalam teori *Dempster-Shafer* disebut dengan *power set* dan dinotasikan dengan $P(\Theta)$, setiap elemen dalam *power set* ini memiliki nilai interval antara 0 sampai 1.

$$m = P(\Theta) \rightarrow [0,1]$$

maka dapat dirumuskan:

$$\sum_{X \in P(\Theta)} m(X) = 1 \approx \sum_{X \in P(\Theta)} m(X) = 1 \dots \dots \dots (2.4)$$

dengan $P(\Theta)$ = *power set* dan $m(X)$ = *mass function* dari (X)

sebagai contoh:

$$P(\text{hostile}) = 0,7$$

$$P(\text{non-hostile}) = 1 - 0,7 = 0,3$$

Pada contoh di atas *belief* dari *hostile* adalah 0,7 sedangkan *disbelief hostile* adalah 0,3. dalam teori *Dempster-Shafer*, *disbelief* dalam *environment* biasanya dinotasikan $m(\theta)$.

Maka *mass function* (m) dalam teori *Dempster-Shafer* adalah tingkat kepercayaan dari suatu *evidence* (gejala), sering disebut dengan *evidence measure* sehingga dinotasikan dengan (m).

Pada aplikasi sistem pakar dalam satu penyakit terdapat sejumlah *evidence* yang akan digunakan pada faktor ketidakpastian dalam pengambilan keputusan untuk diagnosa suatu penyakit. Untuk mengatasi sejumlah *evidence* tersebut pada teorim *Dempster-Shafer* menggunakan aturan yang lebih dikenal dengan *Dempster's Rule of Combination*.

$$m1 \oplus m2(Z) = \sum_{X \cap Y = Z} m1(X)m2(Y) \dots \dots \dots (2.5)$$

Dengan :

$m1 \oplus m2(Z)$ = mass function dari evidence (Z)

$m1(X)$ = mass function dari evidence (X)

$m2(Y)$ = mass function dari evidence (Y)

\oplus = operator *direct sum*

Secara umum formulasi untuk *Dempster's Rule of Combination* adalah:

$$m1 \oplus m2(Z) = \frac{\sum_{X \cap Y = Z} m1(X)m2(Y)}{1 - k} \dots \dots \dots (2.6)$$

Dengan : k = Jumlah *evidential conflict*.

Besarnya jumlah *evidential conflict* (k) dirumuskan dengan:

$$k = \sum_{X \cap Y = \emptyset} m1(X)m2(Y) \dots \dots \dots (2.7)$$

sehingga bila Persamaan (2.7) disubstitusikan ke Persamaan (2.6) akan menjadi:

$$m1 \oplus m2(Z) = \frac{\sum_{X \cap Y = Z} m1(X)m2(Y)}{1 - \sum_{X \cap Y = \emptyset} m1(X)m2(Y)} \dots \dots \dots (2.8)$$

Dengan :

$m1 \oplus m2(Z) = \text{mass function dari evidence (Z)}$

$m1(X) = \text{mass function dari evidence (X)}$

$m2(Y) = \text{mass function dari evidence (Y)}$

$k = \text{jumlah evidential conflict}$

Contoh:

$\Theta = \{K_o, K_r, K_s, K_v\}$

Dengan :

$K_o = \text{Kanker Ovarium}$

$K_r = \text{Kanker Rahim}$

$K_s = \text{Kanker Serviks}$

$K_v = \text{Kanker Vulva}$

Andaikan seorang pasien mengalami 2 gejala :

1. Tercium bau amis pada vagina dengan nilai probabilitas 0,8. Gejala tersebut adalah gejala dari penyakit : Kanker *Ovarium*, Kanker Rahim dan Kanker *Serviks*.

$$m1 \{K_o, K_r, K_s\} = 0,8$$

$$m1(\theta) = 1 - 0,8 = 0,2$$

2. Nyeri pinggang dengan nilai probabilitas 0,9. Gejala tersebut adalah gejala dari penyakit : Kanker Rahim, Kanker *Serviks* dan Kanker *Vulva*.

$$m2 \{K_r, K_s, K_v\} = 0,9$$

$$m2(\theta) = 1 - 0,9 = 0,1$$

Maka munculnya gejala kedua yaitu nyeri pada pinggang, maka harus dilakukan penghitungan densitas baru untuk beberapa kombinasi ($m3$). Untuk

memudahkan perhitungan maka himpunan-himpunan bagian yang terbentuk dimasukkan ke dalam tabel. Kolom pertama diisi dengan gejala yang pertama (m1). Maka baris pertama diisi dengan gejala yang kedua (m2) . Sehingga diperoleh nilai m3 sebagai hasil kombinasi m1 dan m2.

Tabel 2. 2 Tabel perhitungan dua gejala

		{Kr,Ks,Kv} (0,9)	θ	(0,1)
{Ko,Kr,Ks}	(0,8)	{Kr,Ks} (0,72)	{Ko,Kr,Ks}	(0,08)
θ	(0,2)	{Kr,Ks,Kv} (0,18)	θ	(0,02)

Sehingga dapat dihitung :

$$m3 \{Kr,Ks\} = \frac{0,72}{1-0,02} = 0,73$$

$$1-0,02$$

$$m3 \{Kr,Ks,Kv\} = \frac{0,18}{1-0,02} = 0,18$$

$$1-0,02$$

$$m3 \{Ko,Kr,Ks\} = \frac{0,08}{1-0,02} = 0,08$$

$$1-0,02$$

$$m3 \{ \theta \} = \frac{0,02}{1-0,02} = 0,02$$

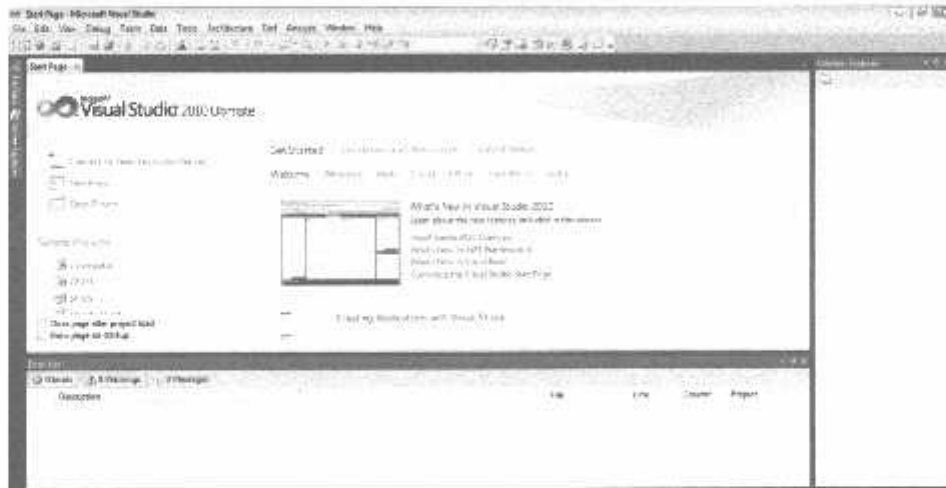
$$1-0,02$$

Dari hasil perhitungan nilai densitas m3 kombinasi di atas dapat dilihat bahwa terdapat gejala pertama yaitu {Ko,Kr,Ks} = 0.8, namun setelah ada gejala kedua yaitu {Kr,Ks,Kv}= 0.9. Sedangkan nilai {Kr,Ks}=0.73 . Jadi dari perhitungan di atas, nilai densitas yang paling tinggi adalah m3 {Kr,Ks}=0.73

2.3 Visual Studio 2010 (VB.NET)

Visual Basic.NET merupakan salah satu bahasa pemrograman yang dapat digunakan untuk membuat aplikasi. Bahasa pemrograman ini menyediakan beberapa tool untuk otomatisasi proses pengembangan, yaitu visual tool yang digunakan untuk melakukan beberapa operasi pemrograman dan desain umum dan juga fasilitas-fasilitas lain yang dapat menunjang dalam pemrograman.

Visual Basic.NET merupakan bagian dari Visual Studio.NET. visual Studio.NET merupakan suatu lingkungan (*Environment*) terintegrasi untuk membangun dan melakukan uji coba (*Testing and Debugging*) berbagai macam aplikasi. Diantaranya adalah aplikasi Windows, Web, Control, Class serta aplikasi Console. Dengan Visual Studio.NET, kita akan dapat lebih mudah membuat aplikasi karena dalam Visual Studio.Net ada dukungan fasilitas baru yang dapat ditambahkan, antara lain *Integrated Development Environment (IDE)*, *Microsoft Intelisense*, debugging yang lebih baik kemampuan dalam XML *Web Service*. Dngan VB.NET, User dapat mendesain, mengeksekusi dan men-debug program aplikasi yang telah dibuat. Adapun tampilan awal dari Visual studio 2010 dapat dilihat pada Gambar 2.2.



Gambar 2. 2 Tampilan awal *Visual Studio* 2010

2.3.1 kelebihan dan kekurangan visual basic

Berikut kelebihan pada Visual Studio 2010 :

1. pengguna dengan dasar pemrograman apapun bisa dengan mudah menggunakan
2. ketika kita melakukan instalasi program lain yang mendukung penggunaannya dalam VB, maka komponen dari program tersebut bisa di masukkan dalam daftar komponen VB.
3. Ketika kita melakukan kesalahan penulisan kode, VB secara otomatis membetulkannya dan tidak perlu kita bersusah payah mendeklarasikan komponen ke dalam list code.

Berikut kekurangan pada Visual Studio 2010 :

1. Sifatnya komersial
2. File VB sering menjadi target serangan virus
3. Dokumentasi sebenarnya lengkap dalam MSDN, namun untuk instalasi MSDN membutuhkan biaya dan space hardisk yang besar.

2.3.2 Sejarah Visual Basic

Visual Basic 1.0 dikenalkan pada tahun 1991. Konsep pemrograman dengan metode *drag-and-drop* untuk membuat tampilan aplikasi Visual Basic ini diadaptasi dari *prototype generator form* yang dikembangkan oleh Alan Cooper dan perusahaannya, dengan nama *Tripod*. Microsoft kemudian mengontrak Cooper dan perusahaannya untuk mengembangkan Tripod menjadi sistem *form* yang dapat diprogram untuk Windows 3.0, di bawah kode nama *Ruby*. Tripod tidak memiliki bahasa pemrograman sama sekali. Ini menyebabkan Microsoft memutuskan untuk mengkombinasikan Ruby dengan bahasa pemrograman Basic untuk membuat Visual Basic.

- a. 20 Mei 1991: Microsoft merilis Visual Basic versi 1.0 untuk Windows di Windows World 1991, Atlanta, Amerika Serikat. Programmer dapat membuat antar muka pengguna dengan mudah. Pembuatan aplikasi bisa lebih cepat dibandingkan sebelumnya. Kode ditulis oleh programmer untuk memberikan aksi terhadap Event dari pengguna (bagaimana pengguna merespon kepada object yang terdapat di antar muka).
 - b. 2. 1 September 1992: Microsoft mengumumkan Microsoft Visual Basic for MS-DOS dalam edisi Standard dan Professional. Seperti Visual Basic untuk Windows, versi ini mengkombinasikan kemudahan pendisainan secara grafis dengan kekuatan dan keunggulan dalam banyak hal di pemrograman
 - c. 2 November 1992: Microsoft mengumumkan kemampuan dari Visual Basic versi 2.0 untuk Windows pada edisi Professional dan Standard. Versi ini menyertakan lebih dari 300 fitur baru dan peningkatan untuk pengembangan aplikasi yang lebih cepat, pengaksesan ke fitur-fitur tingkat lanjut di Windows, dan produktivitas pengembang yang lebih besar lagi. Di antaranya: MDI Forms, ODBC, dan variabel
-

- d. 14 Mei 1993: Microsoft mengumumkan Visual Basic versi 3.0 untuk edisi Standard dan Professional. Versi 3.0 menyediakan kemudahan akses ke berbagai sumber data yang banyak dengan mengintegrasikan mesin database Microsoft Access Database for Windows 1.1 dan kemampuan di bidang aplikasi melalui Object Linking and Embedding (OLE) 2.0. Juga ditambahkan tools baru berupa controls baru, penggunaan yang lebih mudah, dan sebuah peningkatan standarisasi control bagi
 - e. 29 Juni 1993: Microsoft mengumumkan bahwa Microsoft Visual Basic for Applications (VBA) akan diintegrasikan ke dalam Microsoft Excel 5.0 dan Microsoft Project 4.0. Visual Basic akan melayani bahasa Macro yang umum di versi yang akan datang dari aplikasi-aplikasi untuk Windows dan Macintosh. Fitur Macro yang umum untuk aplikasi dan OLE 2.0, keduanya merupakan elemen penting di visi dan strategi Microsoft untuk aplikasi yang dapat diprogram —yang akan memberikan kemampuan bagi pengguna untuk mengintegrasikan aplikasi-aplikasi, tugas-tugas otomatis, dan pembuatan solusi yang bersifat kostumais. VBA memiliki mesin pengembangan tangguh dan fleksibel yang sama yang membuat Visual Basic menjadi terkenal, teknologi pemrograman yang memperoleh
 - f. 14 November 1994: Visual Basic versi 4.0 untuk Windows didemonstrasikan di Fall/COMDEX '94 di Las Vegas. Demonstrasi ini berfokus pada perannya sebagai aplikasi berbasis Windows 32-bit pertama yang menggunakan kontrol kostumais OLE (OCXs) — komponent perangkat lunak yang dapat digunakan kembali yang ditetapkan dengan spesifikasi OLE. Pada versi ini diperkenalkan: Class, OXC's, dan programmer dapat membuat add-ins sendiri dengan
 - g. 12 September 1995: Visual Basic versi 4.0 untuk MS-DOS, Microsoft Windows NT, dan Windows 95 diumumkan. Untuk meningkatkan edisi Standard dan Professional, maka Edisi Enterprise yang baru diperkenalkan untuk memenuhi kebutuhan perusahaan dan team pengembang. Semua versi menyertakan dukungan teknologi OLE yang lebih dikembangkan, perbaikan akses data, sebuah lingkungan pengembangan yang bersifat visual, dan migrasi ke Windows 95 dan Windows NT untuk aplikasi-aplikasi yang sudah
-

- h. 7 Desember 1995: Microsoft mengumumkan Microsoft Visual Basic Scripting Edition (VBScript), sebuah bahasa skrip Internet berbasis pada Visual Basic. VBScript merupakan sebuah skrip dengan performansi yang tinggi yang didisain untuk membuat isi yang bersifat aktif pada suatu situs. VBScript membolehkan pengembang untuk mengaitkan dan mengotomatiskan berbagai jenis object di halaman situs, termasuk object OLE. VBScript menjadi sebuah spesifikasi yang diusulkan berlisensi gratis yang tersedia bagi komunitas
 - i. 3 Februari 1997: Microsoft membuka sistem pemrograman Visual Basic versi 5.0, Edisi Professional — tool versi terakhir yang paling terkenal di dunia untuk membangun aplikasi yang tangguh (Rapid Application Development RAD) tool. Fitur-fitur Visual Basic versi 5.0 terdiri dari sejumlah performansi yang signifikan dan peningkatan produktivitas yang akan membuat para pengembang lebih efisien dan fleksibel serta cepat dalam memberikan solusi bagi pengguna akhir. Fitur-fitur tingkat lanjut — seperti kompilasi berjenis native code, akses database berkecepatan tinggi, dan sebuah lingkungan pengembangan yang ditingkatkan — membuat Visual Basic versi 5.0 menjadi versi yang paling hebat. Sebagai tambahan, kemampuan teknologi pembuatan komponen Microsoft ActiveX di Visual Basic versi 5.0 akan membolehkan lebih dari 3 juta pengembang menggunakan Visual Basic dalam membuat aplikasi yang berbasis pada komponen yang bertujuan untuk digunakan di Internet, intranets, dan lingkungan tradisional client/
 - j. 10 Maret 1997: Versi terakhir yang dirilis dari Visual Basic 5.0, Edisi Pembuatan Kontrol tersedia. Edisi Pembuatan Kontrol adalah cara mudah untuk membuat ActiveX Controls bagi Internet, intranets, dan aplikasi client/
 - k. 15 Juni 1998: Microsoft mengumumkan Visual Basic versi 6.0, dan dimasukkan ke dalam Microsoft Visual Studio versi 6.0. Fitur-fitur Visual Basic versi 6.0 menyediakan pengaksesan data secara terintegrasi dan bersifat grafis ke sumber data (data source) ODBC atau OLE DB manapun, dan perangkat tambahan database yang didisain untuk database Oracle dan Microsoft SQL Server™. Fitur unggulan di versi ini adalah: ActiveX Data Objects (ADO) untuk memanipulasi dan membuat database. Fitur
-

Pengembangan Situs membawa kemudahan dalam penggunaan, model pemrograman berbasis komponen dari Visual Basic untuk membuat HTML – dan Dynamic HTML (DHTML) – berbasis aplikasi. Fitur-fitur baru ini — dikombinasikan dengan optimisasi performansi, pengembangan aplikasi yang disederhanakan dan debugging, dan dukungan untuk Microsoft teknologi server — membuat Visual Basic versi 6.0 sebuah pilihan yang ideal untuk membangun aplikasi berskala

- l. 2 September 1998: Visual Studio 6.0, solusi tool pengembangan lengkap berskala perusahaan tersedia. Visual Studio 6.0 menyertakan Visual Basic versi 6.0, Microsoft Visual C++, Microsoft Visual FoxPro, Microsoft Visual InterDev Web, dan Microsoft Visual J++. Bersamaan dengan diluncurkannya Developer Days '98, lebih dari 115 perusahaan papan atas mengumumkan dukungan terhadap Visual Studio 6.
 - m. 28 April 1999: Lingkungan pengembangan VBA versi 6.0 dan Software Development Kit (SDK) VBA (SDK) versi 6.0 tersedia bagi vendor perangkat lunak pihak ketiga melalui program lisensi VBA. VBA 6.0 merupakan sebuah teknologi pengembangan yang tangguh untuk aplikasi yang dikemas secara kostumais dan merupakan sebuah komponen utama di Microsoft Office
 - n. 15 Februari 2000: Steve Ballmer, presiden yang baru saja ditunjuk dan CEO dari Microsoft Corporation, berbicara kepada lebih dari 2.000 pengembang dalam pidatonya pada Visual Basic Insiders Technical Summit (VBITS), menyatakan visi Microsoft terhadap situs yang dapat diprogram dan memaksa kembali komitmen perusahaan yang dalam terhadap para pengembang Visual
 - o. 11 Juli 2000: Microsoft menyampaikan Tool Platform .NET untuk XML Web Services. Dalam pidatonya yang ditujukan pada acara Microsoft Professional Developers Conference (PDC) 2000 yang kedelapan, Paul Maritz, wakil presiden kelompok dari Platforms Group di Microsoft, memperlihatkan Microsoft .NET Framework dan Microsoft Visual Studio .NET. Ini merupakan versi terakhir dari perangkat pengembangan yang digunakan di seluruh dunia. Visual Studio .NET menyediakan dukungan untuk pengembangan drag-and-drop terhadap XML Web Services. Secara bersama,
-

kedua produk tersebut menyediakan produktivitas yang tinggi kepada para pengembang, suatu lingkungan yang terdiri dari banyak bahasa untuk pembangunan, pengantaran, dan pengintegrasian XML Web Services pada platform Microsoft .

- p. 13 November 2000: Microsoft mengumumkan kemampuan Visual Studio .NET versi Beta 1. Dalam pidato acara COMDEX/Fall 2000, Bill Gates mengumumkan kemampuan versi beta pertama Visual Studio.NET dan .NET Framework, dua teknologi kunci untuk memungkinkan para pengembang membangun XML Web services pada platform .NET. XML Web services merupakan aplikasi dan komponen yang dibuat tersedia melewati situs dengan menggunakan XML dan Simple Object Access Protocol (SOAP), dan kunci yang dapat diprogram untuk membangun generasi masa depan Internet. Microsoft membuat Visual Studio .NET Beta 1 dan .NET Framework tersedia bagi jutaan pelanggan dan rekan.
- q. 13 Februari 2002: Microsoft mengumumkan kemampuan dari Visual Studio .NET versi akhir.

2.4 SQL Server 2008

MySQL merupakan software yang tergolong *data base server* dan bersifat *Open Source*. *Open Source* menyatakan bahwa software ini dilengkapi dengan source code (kode yang dipakai untuk membuat MySql), selain tentu saja bentuk executable-nya atau kode yang dapat dijalankan secara langsung dalam sistem operasi, dan bisa diperoleh dengan cara mengunduh di Internet secara gratis.

MySQL adalah sebuah implementasi dari sistem manajemen basis data relasional (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL (*General Public License*). Setiap pengguna dapat secara bebas menggunakan MySQL, namun dengan batasan perangkat lunak tersebut tidak boleh dijadikan produk turunan yang bersifat komersial.

MySQL sebenarnya merupakan turunan salah satu konsep utama dalam basisdata yang telah ada sebelumnya; SQL (Structured Query Language). SQL adalah sebuah konsep pengoperasian basisdata, terutama untuk pemilihan atau

seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis.

MySQL memiliki beberapa keistimewaan, antara lain :

1. Portabilitas MySQL dapat berjalan stabil pada berbagai sistem operasi seperti Windows, Linux, FreeBSD, Mac Os X Server, Solaris, Amiga, dan masih banyak lagi.
 2. Perangkat lunak sumber terbuka, MySQL didistribusikan sebagai perangkat lunak sumber terbuka, dibawah lisensi GPL sehingga dapat digunakan secara gratis.
 3. Multi-user, MySQL dapat digunakan oleh beberapa pengguna dalam waktu yang bersamaan tanpa mengalami masalah atau konflik.
 4. *Performance tuning* MySQL memiliki kecepatan yang menakjubkan dalam menangani query sederhana, dengan kata lain dapat memproses lebih banyak SQL per satuan waktu.
 5. Ragam tipe data MySQL memiliki ragam tipe data yang sangat kaya, seperti signed / unsigned integer, float, double, char, text, date, timestamp, dan lain-lain.
 6. Perintah dan Fungsi MySQL memiliki operator dan fungsi secara penuh yang mendukung perintah Select dan Where dalam perintah (query).
 7. Keamanan MySQL memiliki beberapa lapisan keamanan seperti level subnetmask, nama host, dan izin akses user dengan sistem perizinan yang mendetail serta sandi terenkripsi.
 8. Skalabilitas dan Pembatasan MySQL mampu menangani basis data dalam skala besar, dengan jumlah rekaman (records) lebih dari 50 juta dan 60 ribu tabel serta 5 milyar baris. Selain itu batas indeks yang dapat ditampung mencapai 32 indeks pada tiap tabelnya.
 9. Konektivitas MySQL dapat melakukan koneksi dengan klien menggunakan protokol TCP/IP, Unix socket (UNIX), atau Named Pipes (NT).
-

10. Lokalisasi, MySQL dapat mendeteksi pesan kesalahan pada klien dengan menggunakan lebih dari dua puluh bahasa. Meski pun demikian, bahasa Indonesia belum termasuk di dalamnya.
11. Antar Muka MySQL memiliki antar muka (interface) terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API (Application Programming Interface).
12. Klien dan Peralatan MySQL dilengkapi dengan berbagai peralatan (tool) yang dapat digunakan untuk administrasi basis data, dan pada setiap peralatan yang ada disertakan petunjuk online.

Struktur tabel MySQL memiliki struktur tabel yang lebih fleksibel dalam menangani ALTER TABLE, dibandingkan basis data lainnya semacam PostgreSQL ataupun Oracle. Adapun gambar tampilan awal dari *Microsoft SQL Server 2008* dapat ditunjukkan pada Gambar 2.2.



Gambar 2. 3. Tampilan awal *Microsoft SQL server 2008*

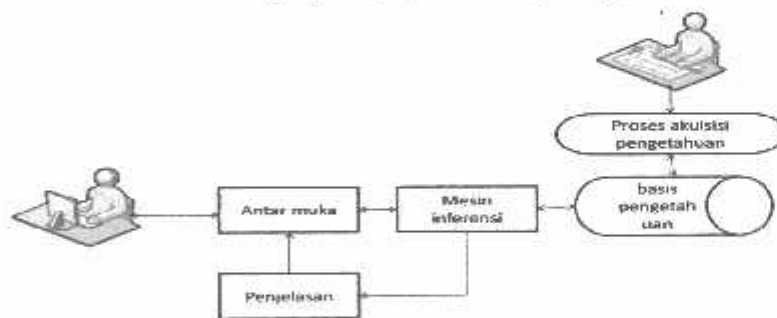
BAB III ANALISA PERANCANGAN SISTEM

3.1 Desain Sistem

Desain sistem merupakan gambaran dari sistem yang akan di implementasikan menjadi sebuah aplikasi. Adapun desain sistem pada penelitian ini meliputi perancangan Blok Diagram, rancangan *database*, *Data Flow Diagram*, *Flowchart Inferensi empster-Shafer*, *Flowchart sistem*, struktur menu dan rancangan tampilan pada aplikasi.

3.1.1 Perancangan Blok Diagram

Pengguna memerlukan informasi dari sistem pakar, maka mesin inferensi akan mengajukan serangkaian pertanyaan yang harus dijawab oleh pengguna melalui suatu *user-interface* (antarmuka pengguna). Jawaban yang dikumpulkan oleh mesin *inferensi* kemudian dipakai untuk menemukan informasi dari basis-pengetahuan, informasi ini kemudian diteruskan ke pengguna. Apabila pengguna ingin mengetahui lebih rinci tentang informasi yang diperolehnya maka dapat melakukan klarifikasi melalui fasilitas penjelas (*explanation facility*).



Gambar 3. 1 Blok Diagram

3.1.2 Data Flow Diagram

Data flow diagram (DFD) merupakan gambaran proses aliran data antara entitas di dalam sebuah sistem yang terlibat didalamnya.

3.1.2.1 DFD Level 0

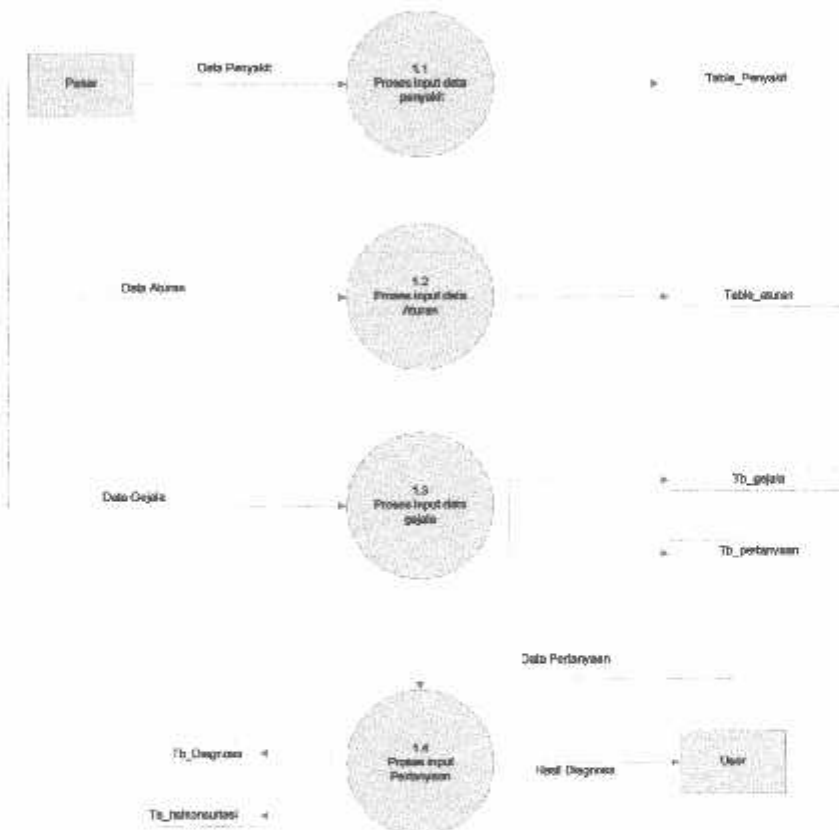
DFD level 0 ini adalah gambaran proses atau aliran data secara umum didalam aplikasi yang dibuat. DFD level 0 ditunjukkan pada Gambar 3.2 berikut :



Gambar 3. 2 DFD level 0

3.1.2.2 DFD Level 1

DFD level 1 berikut adalah gambaran proses secara lebih spesifik. DFD level 1 ditunjukkan pada Gambar 3.3.



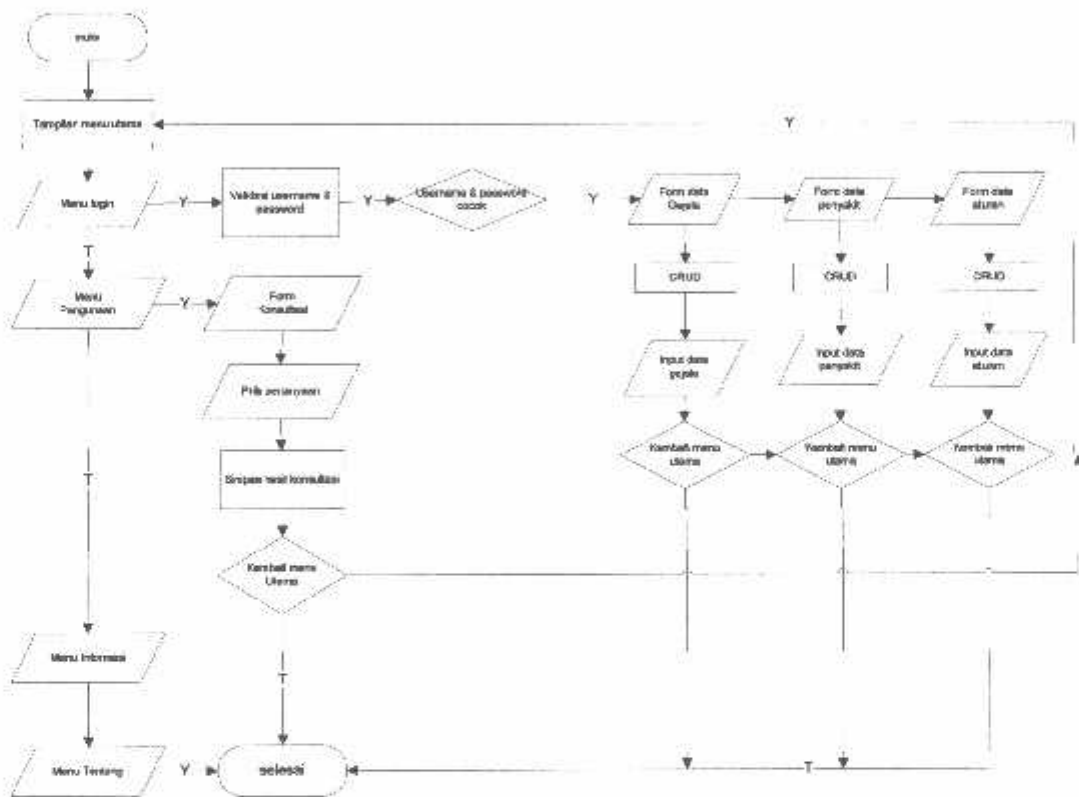
Gambar 3. 3 DFD level 1

3.1.2 Flowchart

Flowchart adalah suatu teknik untuk menyusun rencana sistem dari program. Selain itu flowchart adalah bentuk simbol yang menunjukkan sebuah alur sistem yang menunjukkan urutan tindakan dalam proses dalam bentuk yang lebih mudah dikomunikasikan.

a. Flowchart Sistem

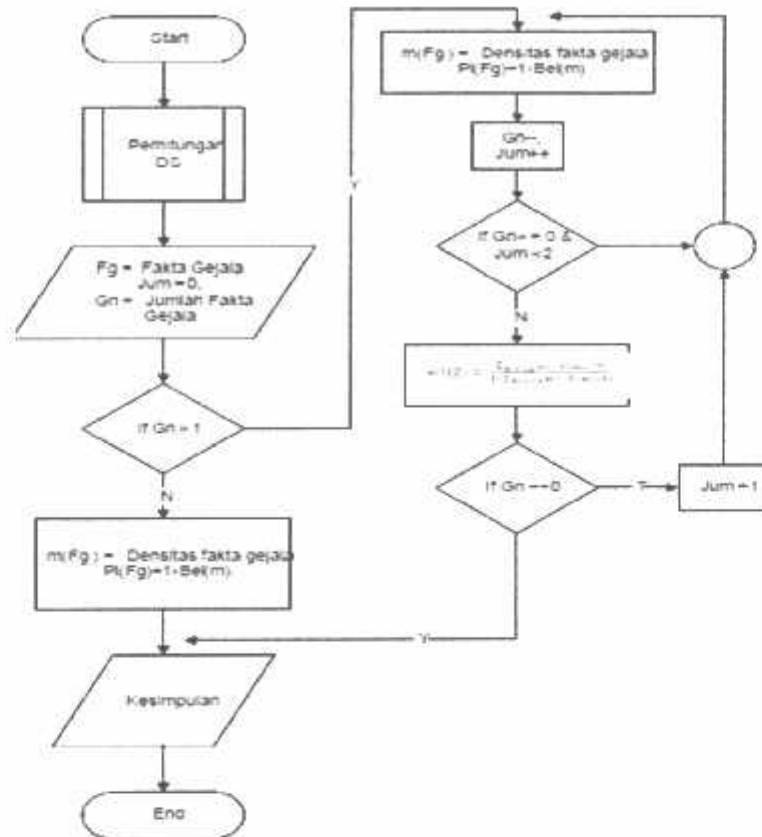
Flowchart sistem ini menjelaskan urutan dari prosedur yang ada di dalam sistem. Bagan alir sistem menunjukkan apa yang dikerjakan oleh sistem. Seperti pada Gambar 3.4.



Gambar 3. 4 *flowchart* sistem

b. Flowchart Inferensi Dempster-Shafer

Dempster-Shafer dan analisa hipotesa hasil dari perhitungan akhir yang kemudian dijadikan sebagai kesimpulan. Kesimpulan yaitu berupa sistem pakar mendiagnosa penyakit hati dan nilai. Untuk proses perhitungan inferensi *Dempster-Shafer* sebagai penarikan kesimpulan. Seperti pada Gambar 3.5.



Gambar 3.5 Flowchart Inferensi Dempster-Shafer

3.1.3 Rancangan Data Base

Dalam rancangan database ini ada beberapa tabel yang digunakan sebagai penyimpanan dan pengolahan data dari sistem pakar untuk mendiagnosa penyakit pada orga hati manusia.

1. Tabel login

Pada tabel login ini berisi data username dan password yang digunakan untuk melakukan login ke sistem.

Tabel 3.1. Tabel login

Nama Fiel	Tipe Data	Panjang/Nilai
Username	Varchar	20
Password	Varchar	50
Hak_akses	Varchar	50

2. Tabel gejala

Pada tabel gejala berisi data ID gejala, Nama gejala dan No gejala, yang digunakan untuk mengetahui gejala yang dipilih oleh user.

Tabel 3. 2 Tabel gejala

Nama fiel	Tipe Data	Panjang/Nilai
ID_gejala	Varchar	20
Nama_gejala	Varchar	50
No_gejala	Int	-

3. Tabel Penyakit

Pada tabel penyakit berisi data ID penyakit dan Nama Penyakit, Penyebab, Solusi dan No Penyakit.

Tabel 3. 3 Tabel penyakit

Nama fiel	Tipe Data	Panjang/Nilai
ID_penyakit	Varchar	20
Nama_penyakit	Varchar	50
Penyebab	Varchar	200
Solusi	Varchar	200
No_penyakit	Int	-

4. Tabel Pertanyaan

Pada tabel pertanyaan berisi data ID Pertanyaan, Pertanyaan, ID Gejala dan No Pertanyaan.

Tabel 3. 4 Tabel pertanyaan

Nama fiel	Tipe Data	Panjang/Nilai
ID_pertanyaan	Varchar	20
Pertanyaan	Varchar	200
ID_gejala	Varchar	50
No_pertanyaan	Int	-

5. Tabel aturan

Pada tabel aturan berisi data ID_aturan, ID_pertanyaan, ID_gejala, ID_penyakit, Nilai Denstas dan No aturan.

Tabel 3. 5 Tabel aturan

Nama fiel	Tipe Data	Panjang/Nilai
ID_aturan	varchar	20
ID_pertanyaan	Varchar	20
ID_gejala	Varchar	20
ID_penyakit	Varchar	20
Nilai_Densta	float	-
No_aturan	Int	-

6. Tabel hasil konsultasi

Pada tabel konsultasi berisi data ID_pertanyaan, dan jawaban.

Tabel 3. 6 Tabel hasil konsultasi

Nama fiel	Tipe Data	Panjang/Nilai
ID_pertanyaan	varchar	20
Jawaban	varchar	20

7. Table diagnose

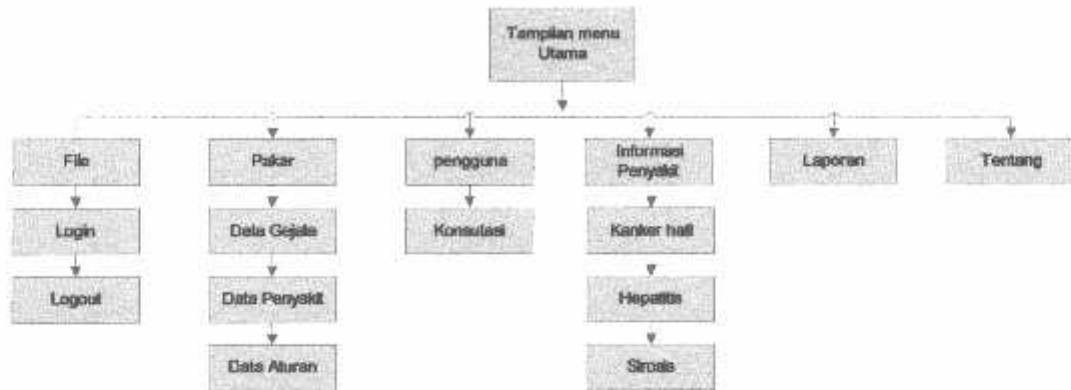
Pada tabel konsultasi berisi data Nam penyakit, Penyebab dan solusi.

Tabel 3. 7 Tabel diagnosa

Nama fiel	Tipe Data	Panjang/Nilai
Nama penyakit	varchar	20
Penyebab	varchar	200
Solusi	varchar	200

3.1.3 Struktur Menu

Struktur menu adalah menggambarkan aplikasi untuk dapat mengakses kemana saja dan terdapat fasilitas apa saja. Struktur menu ditunjukkan pada Gambar 3.6.

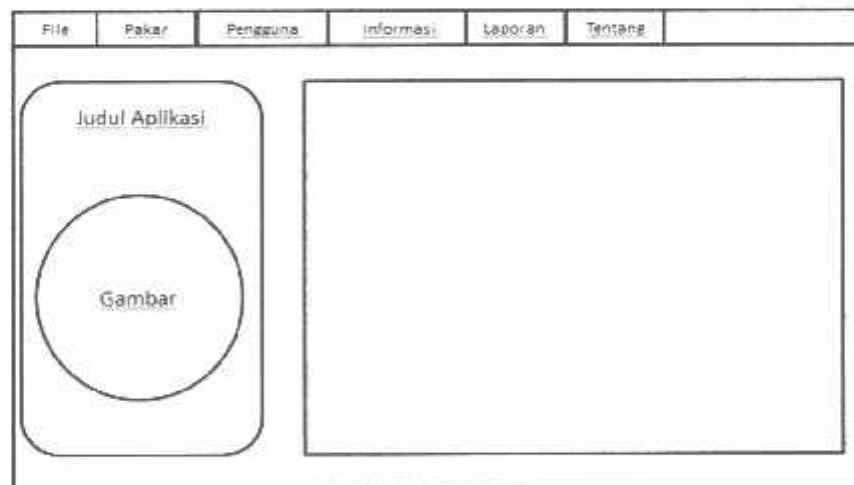


Gambar 3. 6. Struktur Menu

3.1.4 Perancangan Tampilan

a. *Form* Utama

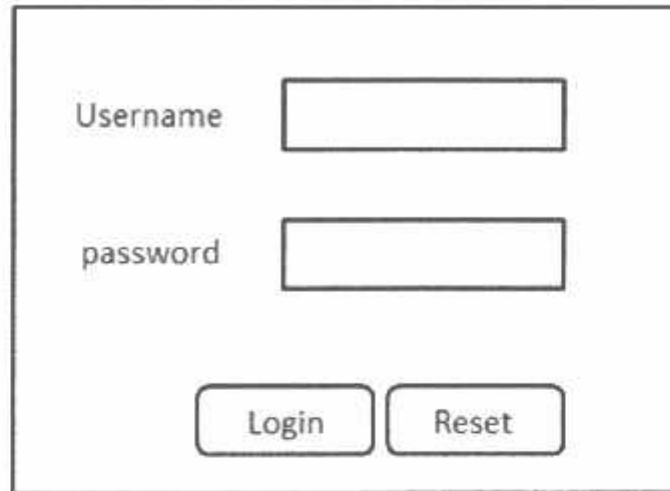
Form halaman utama merupakan tampilan utama yang berfungsi untuk mengakses menu-menu yang ada di dalam aplikasi sistem pakar untuk mendiagnosa penyakit organ hati pada manusia. Rancangan tampilan *form* utama ditunjukkan pada Gambar 3.7.



Gambar 3. 7. Rancangan *Form* Utama

b. *Form Login*

login adalah *form* yang digunakan oleh seorang pakar untuk bisa masuk didalam sistem. Rancangan tampilan *form login* ditunjukkan pada Gambar 3.8.

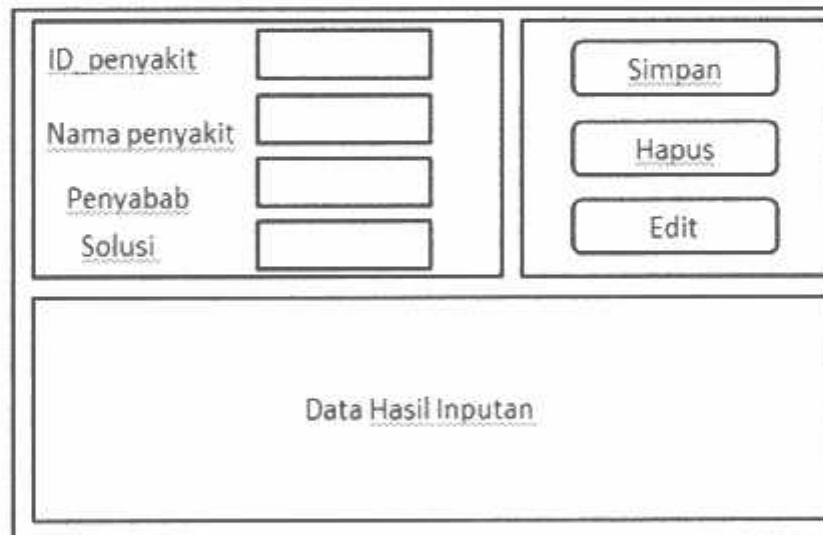


The image shows a login form layout. It consists of a rectangular container. Inside, there are two input fields: the first is labeled 'Username' and the second is labeled 'password'. Below these fields are two buttons: 'Login' and 'Reset'.

Gambar 3. 8. Rancangan *form login*

c. *Form Penyakit*

Form penyakit yang digunakan oleh seorang pakar untuk memasukkan data penyakit didalam sistem. Rancangan form ditunjukkan pada Gambar 3.9.

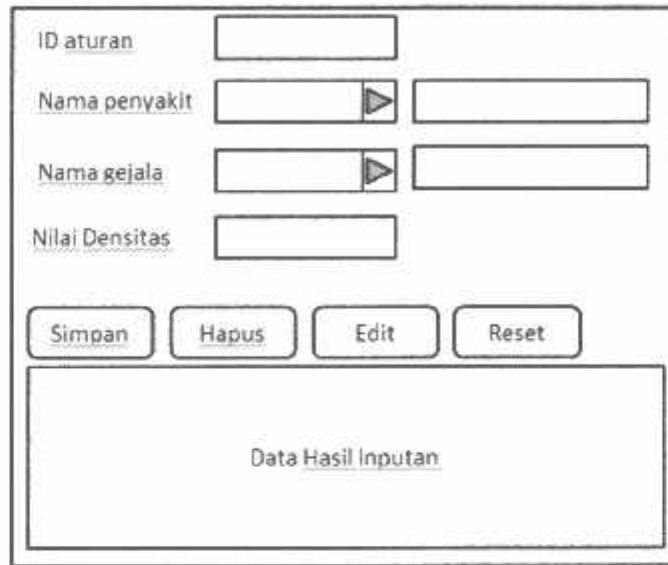


The image shows a disease form layout. It is divided into two main sections. The top section contains four input fields labeled 'ID penyakit', 'Nama penyakit', 'Penyabab', and 'Solusi'. To the right of these fields are three buttons: 'Simpan', 'Hapus', and 'Edit'. The bottom section is a large rectangular area labeled 'Data Hasil Inputan'.

Gambar 3. 9. Rancangan tampilan *form* penyakit

d. *Form Aturan*

Form aturan merupakan *form* yang digunakan oleh pakar untuk memasukkan data aturan di dalam sistem. Rancangan *form* ditunjukkan pada Gambar 3.10.



The image shows a form titled 'Form Aturan'. It contains the following elements:

- Input field for 'ID aturan'.
- Input field for 'Nama penyakit' with a right-pointing arrow.
- Input field for 'Nama gejala' with a right-pointing arrow.
- Input field for 'Nilai Densitas'.
- Four buttons: 'Simpan', 'Hapus', 'Edit', and 'Reset'.
- A large rectangular area labeled 'Data Hasil Inputan'.

Gambar 3. 10. Rancangan tampilan *form* aturan

e. *Form Gejala*

Form gejala digunakan oleh seorang pakar untuk memasukkan data gejala di dalam sistem. Rancangan *form* ditunjukkan pada Gambar 3.11.



The image shows two side-by-side form designs for 'Form Gejala'.

The left form has:

- Buttons: 'Input data', 'Hapus', and 'Edit'.
- A large rectangular area labeled 'Hasil Inputan'.

The right form has:

- Input field for 'ID gejala'.
- Input field for 'Nama gejala'.
- Input field for 'Pertanyaan'.
- Buttons: 'Simpan' and 'Close'.

Gambar 3. 11. Rancangan tampilan *form* gejala

f. *Form* Konsultasi

Form konsultasi merupakan *form* yang digunakan oleh seorang pengguna untuk melakukan konsultasi. Rancangan *form* ditunjukkan pada Gambar 3.12.

The image shows a wireframe of a consultation form. It is divided into several sections:

- Top Left:** A box containing the text "Pilih Pertanyaan" and a "Next" button below it.
- Top Middle:** A vertical stack of three buttons: "konsultasi", "Proses konsultasi", and "Batal konsultasi".
- Top Right:** A box containing the text "Data konsultasi".
- Center:** A large rectangular area containing the text "Data diagnosa".
- Bottom:** A "Simpan" button.

Gambar 3. 12. Rancangan *form* konsultasi

BAB IV IMPLEMENTASI DAN PENGUJIAN

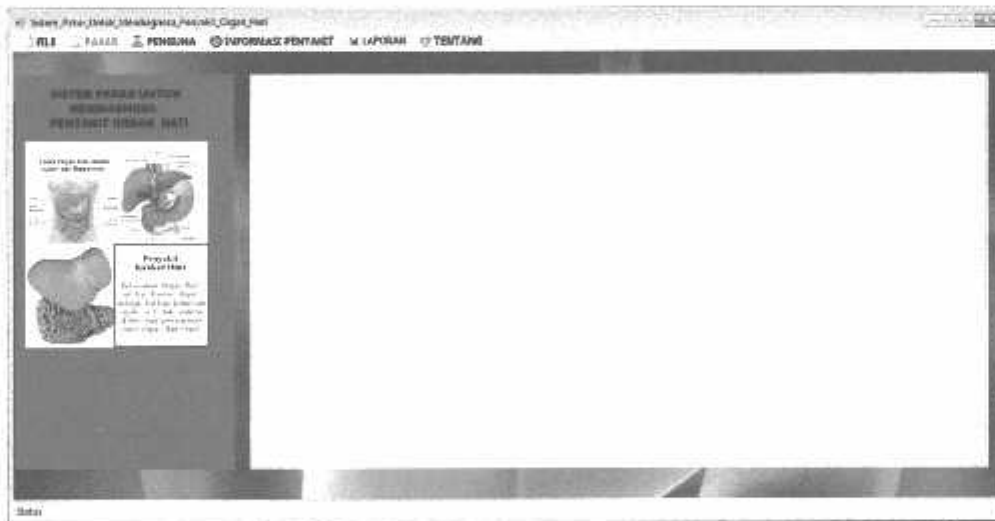
4.1 Implementasi

Tahap implementasi merupakan proses pengubahan analisa dan perancangan yang telah disusun sebelumnya menjadi suatu aplikasi yang siap untuk dijalankan. Implementasi aplikasi Sistem pakar untuk mendiagnosa penyakit hati pada organ hati manusia dengan metode dempter shafer.

Aplikasi Sistem pakar untuk mendiagnosa penyakit hati pada organ hati manusia dengan metode dempter shafer dibuat menggunakan bahasa pemrograman Visual Studio 2010 dengan di integrasikan dengan *database* MySQL Server 2008. Berikut penjelasan bagian-bagian dari Aplikasi sistem pakar untuk mendiagnosa penyakit hati pada organ hati manusia dengan metode dempter shafer.

4.1.1 Form Utama

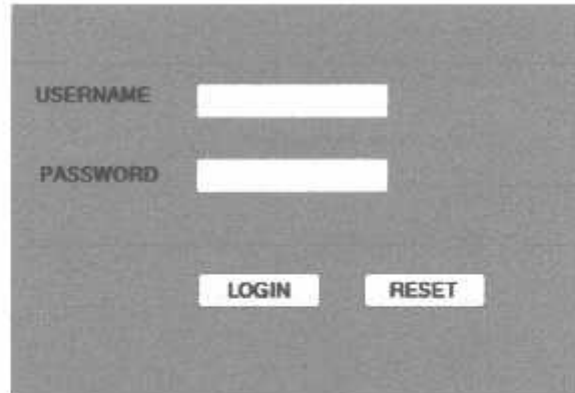
Form utama adalah pertama kali aplikasi dijalankan. Form ini terdiri dari beberapa menu di antaranya menu file ada login dan logout, menu pakar ada data gejala, data penyakit dan data aturan, menu pengguna ada konsultasi, menu Informasi, laporan dan tentang, dapat di lihat pada Gambar 4.1.



Gambar 4. 1. Tampilan *Form* Utama

4.1.2 Form Login

form login seorang pakar akan memasukan *username* dan *password* yang sudah tersimpan dalam database agar pakar dapat menginputkan data, dan apabila *username* dan *password* salah maka sistem akan menampilkan pesan bahwa *username* dan *password* salah. Adapun tampilan form login pada Gambar 4.2.

A screenshot of a login form. It features a dark gray background. At the top left, the label 'USERNAME' is followed by a white rectangular input field. Below it, the label 'PASSWORD' is followed by another white rectangular input field. At the bottom center, there are two white rectangular buttons: 'LOGIN' on the left and 'RESET' on the right.

Gambar 4. 2 Form Login

Tetapi jika *Username* dan *Password* tidak sesuai dengan data yang ada di dalam database, maka program akan memberikan konfirmasi bahwa *Username* atau *Password* tidak valid. Lihat Gambar 4.3.



Gambar 4. 3 Konfirmasi login gagal

4.1.3 Form data Gejala

form data gejala ini digunakan untuk memasukan data gejala yang digunakan oleh pakar. Tampilan form data gejala seperti pada Gambar 4.4.



Gambar 4. 4 Tampilan Data Gejala

Jika *user* atau pakar ingin menambahkan data gejala baru, maka seorang pakar harus klik *Button* tambah data, kemudian akan muncul *form* seperti pada Gambar 4.5.

Gambar 4. 5 Tampilan *input* data gejala

Jika seorang pakar mengubah data yang sudah tersimpan pada *form* data gejala, maka pakar harus pilih data yang akan diubah kemudian klik *button edit*. Maka akan muncul *form* seperti pada Gambar 4.6.

Gambar 4. 6 Tampilan *edit* data gejala

4.1.4 Form data Penyakit

Form data input data Penyakit memasukan data didalam *form* dan di simpn di *database*. Yang ditunjukkan pada Gambar 4.7.

ID Penyakit	Nama Penyakit	Penyakit	Gula
001	Diabetes	Penyakit yang disebabkan oleh ketidakmampuan pankreas memproduksi insulin yang cukup	Diabetes tipe 1
002	Asam Lambung	Kondisi di mana asam lambung berlebihan dan menyebabkan iritasi pada dinding lambung	Asam lambung
003	Demam Berdarah Dengue	Penyakit yang disebabkan oleh virus dengue yang ditularkan oleh nyamuk	Demam berdarah dengue

Gambar 4. 7 Tampilan *form* data penyakit

Jika seorang pakar menghapus data penyakit, maka pakar terlebih dulu mengisi ID_ penyakit , kemudian klik *button* hapus akan muncul perintah seperti pada Gambar 4.8. Jika klik *yes* maka data penyakit tersebut terhapus.



Gambar 4. 8 Tampilan hapus data penyakit

Jika pakar tambahkan data penyakit, maka seorang pakar harus mengisi data penyakit berupa Nama Penyakit, Penyebab dan Solusi. Kemudian klik *button* simpan, Maka akan muncul perintakan seperti ditunjukkan pada Gambar 4.9.



Gambar 4. 9 Tampilan simpan data gejala.

Jika pada klik *button* Ok, Maka data penyakit tersebut sudah tersimpan pada form data penyakit seperti yang ditunjukkan pada Gambar 4.10.



Gambar 4. 10 Tampilan data yang sudah tersimpan

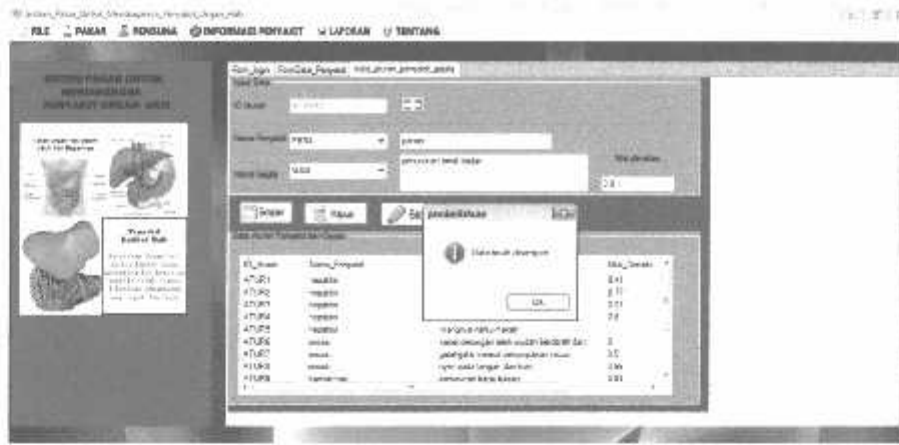
4.1.5 Form data Aturan

Form data aturan atau *rule* adalah *form* yang digunakan untuk oleh pakar menentukan aturan dasar dahulu yang terdiri dari data gejala, dan dan penyakit, yang ditunjukkan pada Gambar 4.11.



Gambar 4. 11 Tampilan *form* data aturan

Jika pakar menambah data aturan, maka pakar harus mengisi data berupa data ID aturan, nama penyakit, nama gejala dan menentukan nilai Densitas. Seperti yang ditunjukkan pada Gambar 4.12.



Gambar 4. 12 Tampilan data aturan.

Jika pada klik *button* Ok, Maka data penyakit tersebut sudah tersimpan pada *form* data aturan seperti yang ditunjukkan pada Gambar 4.13.



Gambar 4. 13 Tampilan data aturan yang sudah tersimpan

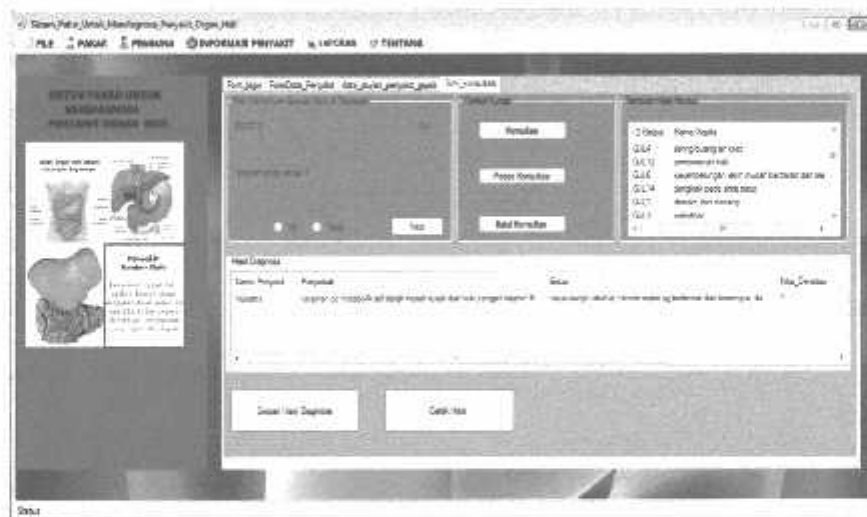
Jika pakar menghapus data aturan, maka pakar terlebih dulu pilih data aturan yang akan di hapus kemudian pilih *button* hapus. Seperti yang ditunjukkan pada Gambar 4.14. jika klik ok maka data tersebut sudah terhapus dari *form* data aturan.



Gambar 4. 14 Tampilan hapus data aturan.

4.1.6 Form Konsultasi

Form data konsultasi di halaman ini *user* akan menjawab pertanyaan yang diberikan oleh *system* untuk mengetahui gejala yang yang dirasakan oleh *user* tersebut. Yang ditunjukkan pada Gambar 4.15.



Gambar 4. 15 Tampilan form konsultasi

4.2 Pengujian Sistem

Pengujian sistem merupakan sebuah tahapan yang dilakukan untuk mengetahui hasil dari implementasi sistem yang sudah dibuat dalam hal ini adalah aplikasi sistem pakar untuk mendiagnosa penyakit hati pada organ hati manusia dengan metode dempter shafer yang bertujuan untuk mengetahui hasil diagnosa dari gejala yang di rasakan oleh user atau pasien. Pengujian dilakukan dengan menggunakan teknik pengujian *Black-box*. Pengujian sistem meliputi pengujian

yang dilakukan oleh pembuat dan pengujian yang dilakukan oleh pengguna (*user*).

4.2.1 Pengujian oleh Pembuat

Pengujian oleh pembuat merupakan pengujian yang dilakukan oleh pembuat sebelum sistem diujicobakan pada pengguna. Pengujian yang dilakukan oleh pembuat meliputi pengujian modul atau fungsi, pengujian sistem operasi, dan pengujian metode.

4.2.1.1 Pengujian Fungsi

Pengujian modul atau fungsi-fungsi yang terdapat dalam sistem sesuai dengan kebutuhan fungsional sistem. Pengujian dilakukan oleh pembuat perangkat lunak dan dalam pengujian modul, perangkat lunak dijalankan di beberapa komputer yang berbeda dengan sistem operasi yang berbeda. Tabel 4.1 menunjukkan hasil pengujian kebutuhan fungsional sistem yang telah dibuat.

Tabel 4. 1 Hasil Pengujian Fungsi

No.	Modul (Fungsi)	Sistem Operasi			
		Windows XP		Windows 7 Ultimate	
		✓	x	✓	X
1	Masuk aplikasi pada halaman utama	✓		✓	
2	Masuk halaman login	✓		✓	
	a. Tampilkan menu pakar, pengguna, informasi, laporan, tentang	✓		✓	
	b. Sistem menampilkan form data gejala	✓		✓	
	c. Sistem menampilkan form data penyakit	✓		✓	
	d. Sistem menampilkan form data aturan	✓		✓	
	e. Sistem menampilkan form data konsultasi	✓		✓	
	f. Sistem menampilkan form data informs penyakit	✓		✓	
3	g. Sistem menampilkan form data tentang	✓		✓	
	a. Form login masukan username dan password	✓		✓	
	b. Form data gejala bias CRUD	✓		✓	
	c. Form penyakit simpan, hapus dan edit	✓		✓	

d. Form data aturan CRUD	✓		✓	
e. Form konsultasi bisa menampilkan data gejala, data pertanyaan. Dan nilai densitas.	✓		✓	

Hasil pengujian semua modul(fungsi) yang dilakukan dalam dua sistem operasi berbeda yaitu *Windows XP* dan *Windows 7 Ultimate 32 bit*, didapatkan hasil pengujian semua fungsi berjalan 100%.

4.2.1.2 Pengujian Terhadap Pengguna

Pengujian dilakuka untuk mengetahui apakah sistem sudah berjalan dengan baik atau belum. Pengujian dilakukan terhadap 10 responden. Berupa Kuisisioner berisi 7 pertanyaan tentang aplikasi yang dibuat penulis. Hasil dari pertanyaan terhadap responden dapat dilihat pada Tabel 4.2.

Tabel 4. 2 Pengujian Terhadap Pengguna.

No	Pertanyaan	Penilaian		
		Baik	Cukup	Kurang
1	Apakah aplikasi ini bisa membantu user atau pengguna	40%	40%	20%
2	Apakah aplikasi ini mudah dioperasikan	50%	30%	20%
3	Struktur menu bisa dengan mudah di pahami	60%	30%	10%
4	Apakah sudah memiliki fungsi yang jelas	30%	40%	30%
5	Apakah tampilan aplikasi ini sudah enak dilihat atau sudah artistik	60%	40%	10%
6	Metode yang digunakan dalam aplikasi sudah jelas	30%	50%	20%
7	Apakah Aplikasi bisa membantu user dalam menginputkan data gejala, data penyakit dan data aturan	60%	30%	10%

BAB V

PENUTUP

5.1 Kesimpulan

Dari penelitian yang dilakukan dan pembahasan bab-bab sebelumnya, maka dapat disimpulkan bahwa:

1. Sistem pakar dengan metode *Dempster-Shafer* dapat dipergunakan untuk mendiagnosa penyakit organ hati pada manusia dengan memasukkan data berupa pertanyaan yang dirasakan oleh pasien, hasil diagnosa yang sama akan diperoleh dari perhitungan sistem dengan menggunakan metode *Dempster-Shafer* pada pengetahuan pakar.
2. Hasil pengujian fungsional sistem dari semua form didapatkan hasil bahwa semua fungsi-fungsi yang ada dapat berjalan dan berfungsi 100%.

DAFTAR PUSTAKA

- A.M.Hirin. 2012. Belajar Tuntas *VB.NET* 2011. Jakarta. Pretasi Pustaka Publisher
- Aristoteles. 2007. *Visual Programing Visual Basic*. Jakarta. Grafindo.
- Arhami, M., 2005, *Konsep Dasar Sistem Pakar*, ANDI. Yogyakarta.
- Ardi Pujiyanta, Ari Pujiantoro, 2012, Sistem Pakar Penentuan Jenis Penyakit Hati Dengan Metode Inferensi *Fuzzy* Tsukamoto, Universitas Ahmad Dahlan Yogyakarta.
- Esthi Dyah Rikhiana , Abdul Fadlil, 2013 Implementasi Sistem Pakar Untuk Mendiagnosa Penyakit Dalam Pada Manusia Menggunakan Metode *Dempster Shafer*. Universitas Ahmad Dahla.
- Priyono, Agus. 2011. Sistem Pakar Untuk Mendiagnosa Kerusakan Mesin Lokomoti Kereta Api Dengan Menggunakan Metode *Dempster-Shafer* (Skripsi-S1). Yogyakarta : Universitas Ahmad Dahlan.
- Sri Kusumadewi, *Artificial Intelligence Teknik dan Aplikasinya*, Graha Ilmu Yogyakarta

LAMPIRAN


**BERITA ACARA UJIAN SKRIPSI
FAKULTAS TEKNOLOGI INDUSTRI**

NAMA : Jumiati Taradi
NIM : 1218161
JURUSAN : Teknik Informatika S-1
JUDUL : Sistem Pakar Untuk Mendiagnosa Penyakit Hati Pada Organ Hati
Manusia Dengan Metode Dempster Shafer

Dipertahankan dihadapan Majelis Penguji Skripsi Jenjang Strata Satu (S-1) pada :
Hari : Sabtu
Tanggal : 16 Januari 2016
Nilai : 74(B+)

Panitia Ujian Skripsi :

Ketua Majelis Penguji



Joseph Dedy Irawan, ST, MT
NIP. 197404162005011002

Anggota Penguji :

Dosen Penguji I

Dosen Penguji II



Ahmad Paisol, ST, MT
NIP.P. 1031000431





Rofita El Maghfiroh, S.SI., M.Sc
NIP.P. 1031500505


FORMULIR PERBAIKAN SKRIPSI

Dalam pelaksanaan ujian skripsi jenjang Strata I Program Studi Teknik Informatika, maka perlu adanya perbaikan skripsi untuk mahasiswa :

NAMA : Jumiati Taradi
NIM : 1218161
JURUSAN : Teknik Informatika S-1
JUDUL : Sistem Pakar Untuk Mendiagnosa Penyakit Hati Pada Organ Hati Manusia Dengan Metode Dempster Shafer

No	Penguji	Tanggal	Uraian	Paraf
1.	Penguji I	16 Januari 2016	1. Di batasan malasa cantumkan nama rumah sakit 2. Pengujian di lakukan pada pakarnya.	
2.	Penguji II	16 Januari 2016	1. Dengan, dan, sehingga, sedangkan, dan kata hubungan lainnya. 2. BAB II metode dempster shafer dan contohnya di BAB II di jelaskan sampai ketemu penyakitnya.	

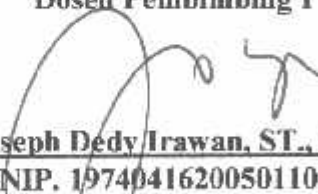
Dosen Penguji I


Ahmad Faisol, ST, MT
NIP.P. 1031000431

Dosen Penguji II


Rofila El Maghfiroh, S.SI., M.Sc
NIP.P. 1031500505

Dosen Pembimbing I


Joseph Dedy Irawan, ST., MT.
NIP. 197404162005011002

Dosen Pembimbing II


Sandy Nataly Mantia, S.Kom
NIP.P. 1030800418



PT. BNI (PERSERO) MALANG
BANK NIAGA MALANG

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

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

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, 23 Oktober 2015

Nomor : ITN-593/L.INF/TA/2015
Lampiran : —
Perihal : Bimbingan Skripsi

Kepada : Yth. Bpk/Ibu Joseph Dedy Irawan, ST, MT
Dosen Pembina Program Studi Teknik Informatika S-1
Institut Teknologi Nasional
Malang

Dengan Hormat,
Sesuai dengan permohonan dan persetujuan dalam proposal skripsi untuk mahasiswa :

Nama : JUMIATI TARADI
Nim : 1218161
Prodi : Teknik Informatika S-1
Fakultas : Teknologi Industri

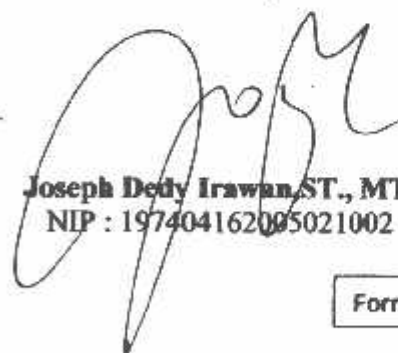
Maka dengan ini pembimbingan kami serahkan sepenuhnya kepada Saudara/i selama waktu 6 (enam) bulan, terhitung mulai tanggal :

23 Oktober 2015 S/D 23 Maret 2016

Sebagai satu syarat untuk menempuh Ujian Akhir Sarjana Teknik, Program Studi Teknik Informatika S-1.

Demikian agar maklum dan atas perhatian serta bantuannya kami sampaikan terima kasih.

Mengetahui
Program Studi Teknik Informatika S-1
Ketua,



Joseph Dedy Irawan, ST., MT.
NIP : 197404162095021002

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 : 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, 23 Oktober 2015

Nomor : ITN-593/LINF/TA/2015
Lampiran : —
Perihal : Bimbingan Skripsi

Kepada : Yth. Bpk/Ibu Sandy Nataly Mantja, S.Kom
Dosen Pembina Program Studi Teknik Informatika S-1
Institut Teknologi Nasional
Malang

Dengan Hormat,
Sesuai dengan permohonan dan persetujuan dalam proposal skripsi untuk mahasiswa :

Nama : JUMIATI TARADI
Nim : 1218161
Prodi : Teknik Informatika S-1
Fakultas : Teknologi Industri

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

23 Oktober 2015 S/D 23 Maret 2016

Sebagai satu syarat untuk menempuh Ujian Akhir Sarjana Teknik, Program Studi Teknik Informatika S-1.
Demikian agar maklum dan atas perhatian serta bantuannya kami sampaikan terima kasih.

Mengetahui
Program Studi Teknik Informatika S-1
Ketua,



Joseph Dedy Irawan, ST., MT.
NIP : 197404162005021002

Form S-4a

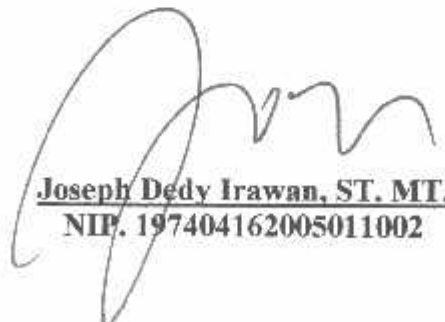
FORMULIR BIMBINGAN SKRIPSI

Nama : Jumiati Taradi
NIM : 12.18.161
Masa Bimbingan : 23 Oktober 2015 s/d 23 Maret 2016
Judul Skripsi : Sistem Pakar Untuk Mendiagnosa Penyakit Hati Pada Organ Hati Manusia Dengan Metode Dempster Shafer

No	Tanggal	Uraian	Paraf pembimbing
1	19-11-2015	Desain Sistem	
2	21-11-2015	BAB III	
3	22-11-2015	Acc BAB I -III	
4	09-12-2015	Revisi BAB IV dan BAB V	
5	11-12-2015	Acc Makalah Seminar Hasil	
6	07-01-2016	Pengujian	
7	10-01-2016	Daftar Pustakan	
8	12-01-2016	Acc Laporan dari BAB I sampai BAB V	




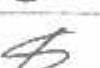


Malang, 26 Januari 2016

Dosen Pembimbing I


Joseph Dedy Irawan, ST. MT.
NIP. 197404162005011002

FORMULIR BIMBINGAN SKRIPSI

Nama : Jumiati Taradi
NIM : 12.18.161
Masa Bimbingan : 23 Oktober 2014 s/d 23 Maret 2015
Judul Skripsi : Sistem Pakar Untuk Mendiagnosa Penyakit Hati Pada Organ Hati Manusia Dengan Metode Dempster Shafer

No	Tanggal	Uraian	Paraf pembimbing
1	07-11-2015	Ganti metode dan Format Penulisan	
2	22-11-2015	Revisi BAB I	
3	23-11-2015	Acc makalah Proges	
4	10-12-2015	Revisi Program	
5	12-12-2015	Acc makalah seminar hasil	
6	12-01-2016	Acc BAB IV, BAB V dan Program	

Malang, 26 Januari 2016

Dosen Pembimbing II



Sandy Nataly Mantja, S.Kom
NIP.P. 1030800418

Code module

```
Imports System.Data.Sql
Imports System.Data.SqlClient
Module Module1
    'Module Mod_Koneksi
    Public koneksi As New SqlConnection
    Public adapter As New SqlCommand
    Public reader As SqlDataReader
    Public server As String
    Public Sub konek_db()
        server = "Data Source=JHUMY-PC\SQLEXPRESS;Initial Catalog=db_sp;Integrated
Security=True"
        koneksi = New SqlConnection(server)
        koneksi.Open()
    End Sub
End Module

Module Module2
    Public Function Tampilkan(ByVal xForm As Form)
        For i As Integer = 0 To
Sistem_Pakar_Untuk_Mendiagnosa_Penyakit_Organ_Hati.TabControl1.TabPages.Count - 1
            If Sistem_Pakar_Untuk_Mendiagnosa_Penyakit_Organ_Hati.TabControl1.TabPages(i).Name
= xForm.Name Then
                Sistem_Pakar_Untuk_Mendiagnosa_Penyakit_Organ_Hati.TabControl1.SelectedTab =
Sistem_Pakar_Untuk_Mendiagnosa_Penyakit_Organ_Hati.TabControl1.TabPages(xForm.Name)
                Return xForm
            End If
        Next i

Sistem_Pakar_Untuk_Mendiagnosa_Penyakit_Organ_Hati.TabControl1.TabPages.Add(xForm.Name,
xForm.Text)
        Sistem_Pakar_Untuk_Mendiagnosa_Penyakit_Organ_Hati.TabControl1.SelectedTab =
Sistem_Pakar_Untuk_Mendiagnosa_Penyakit_Organ_Hati.TabControl1.TabPages(xForm.Name)
        xForm.TopLevel = False
        xForm.Parent =
Sistem_Pakar_Untuk_Mendiagnosa_Penyakit_Organ_Hati.TabControl1.TabPages(xForm.Name)
```

```

xForm.Show()
Return xForm
End Function
Public Function MyParent(ByVal xForm As Form)
    With xForm
        .FormBorderStyle = Windows.Forms.BorderStyle.None
        .ControlBox = False
        .Parent =
Sistem_Pakar_Untuk_Mendiagnosa_Penyakit_Organ_Hati.TabControl1.TabPages(xForm.Name)
        .Height =
Sistem_Pakar_Untuk_Mendiagnosa_Penyakit_Organ_Hati.TabControl1.TabPages(xForm.Name).Height
        .Width =
Sistem_Pakar_Untuk_Mendiagnosa_Penyakit_Organ_Hati.TabControl1.TabPages(xForm.Name).Width
    End With
    Return xForm
End Function
Public Function Tutup(ByVal xForm As Form)
    xForm.Dispose()
Sistem_Pakar_Untuk_Mendiagnosa_Penyakit_Organ_Hati.TabControl1.TabPages(xForm.Name).Dispose()
    Return xForm
End Function
End Module

```

Code login

```

Public Class Form_login
    Private Sub txt_user_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles txt_user.TextChanged
        konek_db()
        Dim pilih As New SqlCommand("Select * From tb_loginuser where usemama = '" & txt_user.Text & "'")
        Dim reader As SqlClient.SqlDataReader
        pilih.Connection = koneksi
        reader = pilih.ExecuteReader
    End Sub
End Class

```

```

Try
  If reader.Read() Then
    level.Text = reader.GetString(6)
  Else
    level.Text = ""
  End If
Catch ex As Exception
End Try
End Sub

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button1.Click
  konek_db()
  Dim login As String
  adapter.CommandText = ("Select * From tb_loginuser where username=" & txt_user.Text & "
And password=" & txt_pass.Text & """)
  adapter.Connection = koneksi
  reader = adapter.ExecuteReader
  If reader.Read() Then
    login = reader!hak_akses

Sistem_Pakar_Untuk_Mendiagnosa_Penyakit_Organ_Hati.LogoutToolStripMenuItem.Enabled =
True
  Sistem_Pakar_Untuk_Mendiagnosa_Penyakit_Organ_Hati.LoginToolStripMenuItem.Enabled
= False
  If login = "admin" Then
    Sistem_Pakar_Untuk_Mendiagnosa_Penyakit_Organ_Hati.Enabled = True
    Sistem_Pakar_Untuk_Mendiagnosa_Penyakit_Organ_Hati.pakar.Enabled = True
    Close()
  End If
Else
  MsgBox("Username Atau Password Salah", MsgBoxStyle.Critical,
"Peringatan")
End If
End Sub

Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button2.Click
  txt_user.Clear()
  txt_pass.Clear()
End Sub

```

Code form konsultasi

```

Public Class form_konsultasi
  Dim hp_m1, hp_p1, hp_m2, hp_p2, hp_m3, hp_p3, hp_m4, hp_p4, hp_m5, hp_p5, hp_m6, hp_p6,
hp_m7, hp_p7 As Double 'hp_m8, hp_p8, hp_m9, hp_p9
  Dim siros_m1, siros_p1, siros_m2, siros_p2, siros_m3, siros_p3, siros_m4, siros_p4, siros_m5,

```

```

siros_p5 As Double
Dim kh_m1, kh_p1, kh_m2, kh_p2, kh_m3, kh_p3, kh_m4, kh_p4, kh_m5, kh_p5, kh_m6, kh_p6,
kh_m7, kh_p7 As Double
Public Sub Dempster_Shafer_kanker_hati()
    konek_db()
    Dim adapter As New SqlClient.SqlCommand("Select * from Table_aturan where ID_Penyakit =
'PEN3' and ID_pertanyaan = '" & Label1.Text & "'")
    Dim reader As SqlClient.SqlDataReader
    adapter.Connection = koneksi
    reader = adapter.ExecuteReader
    reader.Read()

    If rb_ya.Checked = True And Label1.Text = "PERT9" Then
        kh_m1 = reader!Nilai_Densitas
        kh_p1 = 1 - hp_m1
    ElseIf rb_tidak.Checked = True And Label1.Text = "PERT9" Then
        kh_m1 = 0
        kh_p1 = 1 - kh_m1
    ElseIf rb_ya.Checked = True And Label1.Text = "PERT9" Then
        kh_m2 = 0
        kh_p2 = 1 - kh_m2
        kh_m3 = kh_m1 * kh_m2 + kh_m1 * kh_p2 + kh_p1 * kh_m2
        kh_p3 = 1 - kh_m3
    ElseIf rb_ya.Checked = True And Label1.Text = "PERT11" Then
        kh_m4 = reader!Nilai_Densitas
        kh_p4 = 1 - kh_m4
        kh_m5 = kh_m3 * kh_m4 + kh_m3 * kh_p4 + kh_p3 * kh_m4
        kh_p5 = 1 - kh_m5
    ElseIf rb_tidak.Checked = True And Label1.Text = "PERT12" Then
        kh_m4 = 0
        kh_p4 = 1 - kh_m4
        kh_m5 = kh_m3 * kh_m4 + kh_m3 * kh_p4 + kh_p3 * kh_m4
        kh_p5 = 1 - kh_m5
    ElseIf rb_ya.Checked = True And Label1.Text = "PERT12" Then
        kh_m6 = reader!Nilai_Densitas
        kh_p6 = 1 - kh_m6
        kh_m7 = kh_m5 * kh_m6 + kh_m5 * kh_p6 + kh_p5 * kh_m6
        kh_p7 = 1 - kh_m7
    ElseIf rb_tidak.Checked = True And Label1.Text = "PERT12" Then
        kh_m6 = 0
        kh_p6 = 1 - kh_m6
        kh_m7 = kh_m5 * kh_m6 + kh_m5 * kh_p6 + kh_p5 * kh_m6
        kh_p7 = 1 - kh_m7
    End If
    reader.Close()
    koneksi.Close()
End Sub

```

```

Public Sub Dempster_Shafer_sirosis()
    konek_db()

    Dim adapter As New SqlClient.SqlCommand("Select * from Table_aturan where ID_Penyakit =
'PEN2' and ID_pertanyaan = '" & Label1.Text & "'")
    Dim reader As SqlClient.SqlDataReader
    adapter.Connection = koneksi
    reader = adapter.ExecuteReader
    reader.Read()

    If rb_ya.Checked = True And Label1.Text = "PERT6" Then
        siros_m1 = reader!Nilai_Densitas
        siros_p1 = 1 - siros_m1
    ElseIf rb_tidak.Checked = True And Label1.Text = "PERT6" Then
        siros_m1 = 0
        siros_p1 = 1 - siros_m1
    ElseIf rb_ya.Checked = True And Label1.Text = "PERT7" Then
        siros_m2 = reader!Nilai_Densitas
        siros_p2 = 1 - siros_m2
        siros_m3 = siros_m1 * siros_m2 + siros_m1 * siros_p2 + siros_p1 * siros_m2
        siros_m3 = 1 - siros_m3
    ElseIf rb_tidak.Checked = True And Label1.Text = "PERT7" Then
        siros_m2 = 0
        siros_p2 = 1 - siros_m2
        siros_m3 = siros_m1 * siros_m2 + siros_m1 * siros_p2 + siros_p1 * siros_m2
        siros_p3 = 1 - siros_m3
    ElseIf rb_ya.Checked = True And Label1.Text = "PERT8" Then
        siros_m4 = reader!Nilai_Densitas
        siros_p4 = 1 - siros_m4
        siros_m5 = siros_m3 * siros_m4 + siros_m3 * siros_p4 + siros_p3 * siros_m4
        siros_p5 = 1 - siros_m5
    ElseIf rb_tidak.Checked = True And Label1.Text = "PERT8" Then
        siros_m4 = 0
        siros_p4 = 1 - siros_m4
        siros_m5 = siros_m3 * siros_m4 + siros_m3 * siros_p4 + siros_p3 * siros_m4
        siros_p5 = 1 - siros_m5
    End If
    reader.Close()
    koneksi.Close()
End Sub
Public Sub Dempster_Shafer_hepatitis()
    konek_db()
    Dim adapter As New SqlClient.SqlCommand("Select * from Table_aturan where ID_Penyakit =
'PEN1' and ID_pertanyaan = '" & Label1.Text & "'")
    Dim reader As SqlClient.SqlDataReader
    adapter.Connection = koneksi
    reader = adapter.ExecuteReader

```

```

reader.Read()

If rb_ya.Checked = True And Label1.Text = "PERT1" Then
    hp_m1 = reader!Nilai_Densitas
    hp_p1 = 1 - hp_m1
ElseIf rb_tidak.Checked = True And Label1.Text = "PERT1" Then
    hp_m1 = 0
    hp_p1 = 1 - hp_m1
ElseIf rb_ya.Checked = True And Label1.Text = "PERT2" Then
    hp_m2 = reader!Nilai_Densitas
    hp_p2 = 1 - hp_m2
    hp_m3 = hp_m1 * hp_m2 + hp_m1 * hp_p2 + hp_p1 * hp_m2
    hp_m3 = 1 - hp_m3
ElseIf rb_tidak.Checked = True And Label1.Text = "PERT2" Then
    hp_m2 = 0
    hp_p2 = 1 - hp_m2
    hp_m3 = hp_m1 * hp_m2 + hp_m1 * hp_p2 + hp_p1 * hp_m2
    hp_p3 = 1 - hp_m3
ElseIf rb_ya.Checked = True And Label1.Text = "PERT3" Then
    hp_m4 = reader!Nilai_Densitas
    hp_p4 = 1 - hp_m4
    hp_m5 = hp_m3 * hp_m4 + hp_m3 * hp_p4 + hp_p3 * hp_m4
    hp_p5 = 1 - hp_m5
ElseIf rb_tidak.Checked = True And Label1.Text = "PERT3" Then
    hp_m4 = 0
    hp_p4 = 1 - hp_m4
    hp_m5 = hp_m3 * hp_m4 + hp_m3 * hp_p4 + hp_p3 * hp_m4
    hp_p5 = 1 - hp_m5
ElseIf rb_ya.Checked = True And Label1.Text = "PERT5" Then
    hp_m6 = reader!Nilai_Densitas
    hp_p6 = 1 - hp_m6
    hp_m7 = hp_m5 * hp_m6 + hp_m5 * hp_p6 + hp_p5 * hp_m6
    hp_p7 = 1 - hp_m7
ElseIf rb_tidak.Checked = True And Label1.Text = "PERT5" Then
    hp_m6 = 0
    hp_p6 = 1 - hp_m6
    hp_m7 = hp_m5 * hp_m6 + hp_m5 * hp_p6 + hp_p5 * hp_m6
    hp_p7 = 1 - hp_m7
End If
reader.Close()
koneksi.Close()
End Sub
Public Sub Hepa_ds()
    Dim adapter As New SqlClient.SqlCommand("Insert " & _
" into Table_hslperhitungan values(" & _
" 'PEN1', " & hp_m7 & ")")
    adapter.Connection = koneksi

```

```

    adapter.ExecuteNonQuery()
End Sub
Public Sub sh_ds()
    konek_db()
    Dim adapter As New SqlClient.SqlCommand("Insert " & _
" into Table_hslperhitungan values(" & _
" 'PEN2' , " & siros_m5 & "'")
    adapter.Connection = koneksi
    adapter.ExecuteNonQuery()
    koneksi.Close()
End Sub
Public Sub pertanyaan()
    konek_db()
    Dim cmd As New SqlClient.SqlCommand("select * from tb_pertanyaan where no_pertanyaan =
" & Label2.Text & "'")
    Dim reader As SqlClient.SqlDataReader
    cmd.Connection = koneksi
    reader = cmd.ExecuteReader
    reader.Read()
    Label3.Text = reader!pertanyaan
    Label1.Text = reader!ID_pertanyaan
    koneksi.Close()
End Sub
Public Sub simpan_hasil_konsultasi()
    konek_db()
    Dim jwb As String
    If rb_ya.Checked = True Then
        jwb = "Ya"
    ElseIf rb_tidak.Checked = True Then
        jwb = "Tidak"
    End If

    Dim cmd As New SqlClient.SqlCommand("Insert " & _
" into Table_hslkonsultasi values(" & _
Label1.Text & ",'" & jwb & "'")
    Try
        cmd.Connection = koneksi
        cmd.ExecuteNonQuery()
    Catch ex As Exception
    End Try
    koneksi.Close()
End Sub
Public Sub update_hasil_konsultasi()
    Dim jwb As String
    If rb_ya.Checked = True Then
        jwb = "Ya"

```

```

ElseIf rb_tidak.Checked = True Then
    jwb = "Tidak"
End If
konek_db()
Dim adapter As New SqlClient.SqlCommand("update " & _
" Table_hslkonsultasi set jawaban = " & jwb & " where and ID_pertanyaan = " & Label1.Text
& " ")
Try
    adapter.Connection = koneksi
    adapter.ExecuteNonQuery()
Catch ex As Exception
End Try
End Sub
Public Sub hapus_perhitungan()
    Dim adapter As New SqlClient.SqlCommand("Delete from Table_hslperhitungan ")
    Try
        adapter.Connection = koneksi
        adapter.ExecuteNonQuery()
    Catch ex As Exception
        MsgBox(ex)
    End Try
End Sub
Dim i As Integer = 0
Public Sub tampil_kesimpulan_ds()
    konek_db()
    Dim max_ds As Double

    max_ds = Math.Max(hp_m7, Math.Max(siros_m5, kh_m7))
    MsgBox(max_ds)

    Dim adapter As New SqlClient.SqlCommand("SELECT
Table_penyakit>Nama_Penyakit,Table_penyakit.Penyebab,Table_penyakit.Solusi,Table_aturan.Nilai_
Densitas " & _
" FROM Table_penyakit INNER JOIN Table_aturan ON Table_penyakit.ID_Penyakit =
Table_aturan.ID_Penyakit where Table_aturan.Nilai_Densitas = " & max_ds & " ")

    Dim reader As SqlClient.SqlDataReader
    adapter.Connection = koneksi
    reader = adapter.ExecuteReader
    Do While reader.Read
        Lv_diagnosa.Items.Add(reader!Nama_Penyakit)
        Lv_diagnosa.Items(i).SubItems.Add(reader!Penyebab)
        Lv_diagnosa.Items(i).SubItems.Add(reader!Solusi)
        Lv_diagnosa.Items(i).SubItems.Add(max_ds)
        i = i + 1
    Loop
    reader.Close()

```

```

koneksi.Close()
End Sub
Private Sub Button5_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles hasil_diagnosa.Click
    konek_db()
    Dim penyakit, penyebab, sosuli As String
    Dim a As Integer
    For a = 0 To 3
        Try
            'gejala = Lv_hasil_konsul.Items(a).SubItems(0).Text
            penyakit = Lv_diagnosa.Items(a).SubItems(0).Text
            penyebab = Lv_diagnosa.Items(a).SubItems(1).Text
            sosuli = Lv_diagnosa.Items(a).SubItems(2).Text
            'densitas = Lv_diagnosa.Items(a).SubItems(4).Text
            adapter.CommandText = ("insert into tb_diagnosa values(" & penyakit & ", " & penyebab
& ", " & sosuli & ")")

            adapter.Connection = koneksi
            adapter.ExecuteNonQuery()
        Catch ex As Exception
            Exit For
        End Try
    Next
    MsgBox("Data konsultasi telah disimpan", MsgBoxStyle.Information)

    hasil_diagnosa.Enabled = True
    cetak.Enabled = True

End Sub

Private Sub Button4_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button4.Click
    konek_db()
    Label2.Text = 1
    adapter.CommandText = ("select * from tb_pertanyaan")
    adapter.Connection = koneksi
    reader = adapter.ExecuteReader
    reader.Read()
    Label3.Text = reader!pertanyaan
    Label1.Text = reader!ID_pertanyaan
    gb_konsul.Enabled = True
    btn_proses_konsul.Enabled = True
    btn_batal.Enabled = True

    reader.Close()
    koneksi.Close()
End Sub

```

```
Dim id_gejala As String
```

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click  
    If rb_tidak.Checked = False And rb_ya.Checked = False Then  
        MessageBox.Show("Silahkan pilih gejala !!!", "Pemberitahuan", MessageBoxButtons.OK, MessageBoxIcon.Warning)  
    Else  
        Label2.Text = Label2.Text + 1  
        Try  
            simpan_hasil_konsultasi()  
            pertanyaan()  
        Catch ex As Exception  
            MessageBox.Show("Pertanyaan selesai, silahkan klik tombol 'Proses Hasil Konsultasi' ", "Pemberitahuan", MessageBoxButtons.OK, MessageBoxIcon.Information)  
            Label2.Text = 14  
            btn_proses_konsul.Enabled = True  
        End Try  
        rb_tidak.Checked = False  
        rb_ya.Checked = False  
    End If  
End Sub
```

```
Private Sub btn_proses_konsul_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn_proses_konsul.Click  
    konek_db()  
    'Lv_hasil_konsul.Items  
    adapter.CommandText = ("SELECT tb_gejala.ID_gejala, tb_gejala>Nama_gejala FROM  
tb_gejala INNER JOIN (Table_hslkonsultasi INNER JOIN tb_pertanyaan ON  
Table_hslkonsultasi.ID_pertanyaan = tb_pertanyaan.ID_pertanyaan) ON tb_gejala.ID_gejala =  
tb_pertanyaan.ID_gejala " & _  
" WHERE ((Table_hslkonsultasi.[jawaban])='Ya')")  
  
    adapter.Connection = koneksi  
    reader = adapter.ExecuteReader  
  
    Dim i As Integer = 0  
    Do While reader.Read()  
        Lv_hasil_konsul.Items.Add(reader!ID_gejala)  
        Lv_hasil_konsul.Items(i).SubItems.Add(reader>Nama_gejala)  
        i = i + 1  
    Loop  
  
    Lv_diagnosa.Items.Clear()  
    tampil_kesimpulan_ds()  
  
    gb_konsul.Enabled = False
```

```
reader.Close()
koneksi.Close()
End Sub
```

```
Private Sub form_konsultasi_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
```

```
End Sub
```

```
Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn_batal.Click
```

```
    If gb_konsul.Text = "Peranyaan Sesuai Yang di Rasakan" Then
        If MessageBox.Show("Batalkan konsultasi ???", "Batal Konsul",
            MessageBoxButtons.OKCancel, MessageBoxIcon.Question) = Windows.Forms.DialogResult.OK
        Then
```

```
            Try
```

```
                adapter.Connection = koneksi
                adapter.ExecuteNonQuery()
```

```
            Catch ex As Exception
```

```
                MsgBox(ex)
```

```
            End Try
```

```
            Button3_Click(sender, e)
```

```
        End If
```

```
    End If
```

```
End Sub
```

```
Private Sub rb_ya_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles rb_ya.CheckedChanged, rb_tidak.CheckedChanged
```

```
    Dempster_Shafer_hepatitis()
```

```
    Dempster_Shafer_sirosis()
```

```
    Dempster_Shafer_kanker_hati()
```

```
End Sub
```

Code form aturan

```
Imports System.Data.Sql
Imports System.Data.SqlClient
```

```
Public Class data_aturan
```

```
    Dim no As String
```

```
    Dim id As String
```

```
    Dim pertanyaan As String
```

```
Public Sub Tampil()
```

```
    dt_penyakitgejala.Items.Clear()
```

```
    konek_db()
```

```

    adapter.CommandText = ("SELECT Table_aturan.ID_Aaturan, tb_gejala>Nama_gejala,
Table_penyakit>Nama_Penyakit, Table_aturan.Nilai_Densitas " & _
" FROM (Table_aturan INNER JOIN Table_penyakit ON Table_aturan.ID_Penyakit =
Table_penyakit.ID_Penyakit) INNER JOIN tb_gejala ON Table_aturan.ID_gejala =
tb_gejala.ID_gejala order by no_aturan asc")
    adapter.Connection = koneksi
    reader = adapter.ExecuteReader
    Dim i As Integer = 0
    Do While reader.Read()
        dt_penyakitgejala.Items.Add(reader!ID_Aaturan)
        dt_penyakitgejala.Items(i).SubItems.Add(reader!Nama_Penyakit)
        dt_penyakitgejala.Items(i).SubItems.Add(reader!Nama_gejala)
        dt_penyakitgejala.Items(i).SubItems.Add(reader!Nilai_Densitas)
        i = i + 1
    Loop
End Sub
Public Sub cb_penyakit()
    jenis_penyakit.Items.Clear()
    konek_db()
    adapter.CommandText = ("select ID_Penyakit from Table_penyakit order by ID_Penyakit asc")
    adapter.Connection = koneksi
    reader = adapter.ExecuteReader
    Dim i As Integer = 0
    Do While reader.Read()
        jenis_penyakit.Items.Add(reader!ID_Penyakit)
        i = i + 1
    Loop
End Sub
Public Sub Cb_gejala()
    Cmb_gejala.Items.Clear()
    konek_db()
    adapter.CommandText = ("select ID_gejala from tb_gejala order by ID_gejala asc")
    adapter.Connection = koneksi
    reader = adapter.ExecuteReader
    Dim i As Integer = 0
    Do While reader.Read()
        Cmb_gejala.Items.Add(reader!ID_Gejala)
        i = i + 1
    Loop
End Sub
Private Sub data_aturan_penyakit_gejala_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
    Tampil()
End Sub
Private Sub Button4_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)

```

Handles Btn id tambah.Click

```
konek_db()
adapter.CommandText = ("select MAX(no_aturan)from Table_aturan")
adapter.Connection = koneksi
Try
    reader = adapter.ExecuteReader
    reader.Read()
    no = reader.GetValue(0) + 1
    id_aturan.Text = "ATUR" + no
Catch ex As Exception
    no - 1
    id_aturan.Text = "ATUR" + "1"
End Try
Cb_gejala()
cb_penyakit()
End Sub
```

Private Sub jenis_penyakit_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles jenis_penyakit.SelectedIndexChanged

```
konek_db()
adapter.CommandText = ("select Nama_Penyakit from Table_penyakit where ID_Penyakit=" &
jenis_penyakit.Text & "")
adapter.Connection = koneksi
reader = adapter.ExecuteReader
```

```
Dim i As Integer = 0
Do While reader.Read()
    txt_penyakit.Text = reader!Nama_Penyakit
    i = i + 1
Loop
```

End Sub

Private Sub Cmb_gejala_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Cmb_gejala.SelectedIndexChanged

```
konek_db()
adapter.CommandText = ("SELECT tb_gejala.Nama_gejala, tb_pertanyaan.ID_pertanyaan " &
" FROM tb_pertanyaan INNER JOIN tb_gejala ON tb_pertanyaan.ID_gejala = tb_gejala.ID_gejala
where tb_gejala.ID_gejala = " & Cmb_gejala.Text & "")
```

```
adapter.Connection = koneksi
reader = adapter.ExecuteReader
```

```
Dim i As Integer = 0
Do While reader.Read()
    txt_gejala.Text = reader!Nama_Gejala
    pertanyaan = reader!ID_pertanyaan
    i = i + 1
Loop
```

```
Loop  
End Sub
```

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)  
Handles Btn_simapn.Click
```

```
    konek_db()  
    adapter.CommandText = ("insert into Table_aturan values(" & id_aturan.Text & " " & _  
        " " & pertanyaan & " " & _  
        " " & Cmb_gejala.Text & " " & _  
        " " & jenis_penyakit.Text & "" & _  
        " " & txt_nilai.Text & "" & _  
        " " & no & ")")
```

```
Try
```

```
    adapter.Connection = koneksi  
    adapter.ExecuteReader()  
    MsgBox("Data telah disimpan", MsgBoxStyle.Information, "pemberitahuan")
```

```
Catch ex As Exception
```

```
    MsgBox("data masih kosong", MsgBoxStyle.Critical, "Error")
```

```
End Try
```

```
    data_aturan_penyakit_gejala_Load(sender, e)
```

```
End Sub
```

```
Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)  
Handles Btn_hapus.Click
```

```
    konek_db()  
    adapter.CommandText = ("Delete from Table_aturan where ID_Aturan = " & id_aturan.Text &  
    """)
```

```
Try
```

```
    adapter.Connection = koneksi  
    adapter.ExecuteNonQuery()  
    MsgBox("Data telah dihapus", MsgBoxStyle.Information, "Pemberitahuan")  
    data_aturan_penyakit_gejala_Load(sender, e)
```

```
Catch ex As Exception
```

```
    MsgBox(ex)
```

```
End Try
```

```
End Sub
```

```
Private Sub dt_penyakitgejala_SelectedIndexChanged(ByVal sender As System.Object, ByVal e  
As System.EventArgs) Handles dt_penyakitgejala.SelectedIndexChanged
```

```
    id_aturan.Text = dt_penyakitgejala.SelectedItems(0).SubItems(0).Text  
    txt_penyakit.Text = dt_penyakitgejala.SelectedItems(0).SubItems(1).Text  
    txt_gejala.Text = dt_penyakitgejala.SelectedItems(0).SubItems(2).Text  
    txt_nilai.Text = dt_penyakitgejala.SelectedItems(0).SubItems(3).Text  
    jenis_penyakit.Enabled = False  
    Cmb_gejala.Enabled = False  
    Btn_id_tambah.Enabled = False  
    Btn_simapn.Enabled = False
```

```

    Btn_edit.Enabled = True
    Btn_hapus.Enabled = True
End Sub

Private Sub Btn_edit_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Btn_edit.Click
    Dim perintah As New SqlCommand("update Table_aturan set Nilai_Densitas = " &
txt_nilai.Text & " where ID_Aaturan = " & id_aturan.Text & "")
    Try
        perintah.Connection = koneksi
        perintah.ExecuteNonQuery()
        MsgBox("Data telah diubah", MsgBoxStyle.Information, "Pemberitahuan")
        data_aturan_penyakit_gejala_Load(sender, e)
    Catch ex As Exception
        MsgBox("Pastikan data telah diisi dengan baik dan benar", MsgBoxStyle.Critical, "Error")
    End Try
End Sub

Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button3.Click
    txt_nilai.Clear()
    id_aturan.Clear()
    txt_penyakit.Clear()
    txt_gejala.Clear()
    Cmb_gejala.Items.Clear()
    jenis_penyakit.Items.Clear()
    Cmb_gejala.Text = ""
    jenis_penyakit.Text = ""
    Btn_id_tambah.Enabled = True
    Btn_edit.Enabled = False
    Btn_hapus.Enabled = False
    Btn_simapn.Enabled = True
End Sub

```