

**PERANCANGAN DAN PEMBUATAN SISTEM PENDUKUNG  
KEPUTUSAN DIAGNOSA PENYAKIT MATA MENGGUNAKAN  
METODE FUZZY LOGIC**

**SKRIPSI**



**Disusun Oleh :**  
**Tirto Wicaksono**  
**NIM : 07.12.521**

**PROGRAM STUDI TEKNIK ELEKTRO S-1  
KONSENTRASI TEKNIK ELEKTRONIKA  
FAKULTAS TEKNOLOGI INDUSTRI  
INSTITUT TEKNOLOGI NASIONAL MALANG  
2012**

---

## LEMBAR PERSETUJUAN

### PERANCANGAN DAN PEMBUATAN SISTEM PENDUKUNG KEPUTUSAN DIAGNOSA PENYAKIT MATA MENGGUNAKAN METODE *FUZZY LOGIC*

#### SKRIPSI

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



**PROGRAM STUDI TEKNIK ELEKTRO S-1  
KONSENTRASI TEKNIK KOMPUTER DAN INFORMATIKA  
FAKULTAS TEKNOLOGI INDUSTRI  
INSTITUT TEKNOLOGI NASIONAL MALANG  
2012**

## SURAT PERNYATAAN ORISINALITAS

Yang bertanda tangan di bawah ini :

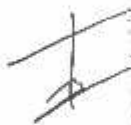
Nama : Tirta Wicaksono  
NIM : 07.12.521  
Program Studi : Teknik Komputer dan Informatika  
Konsentrasi : Teknik Elektronika S-1

Dengan ini menyatakan bahwa Skripsi yang saya buat adalah hasil karya sendiri, tidak merupakan plagiasi dari karya orang lain. Dalam Skripsi ini tidak memuat karya orang lain, kecuali dicantumkan sumbernya sesuai dengan ketentuan yang berlaku.

Demikian surat pernyataan ini saya buat, dan apabila di kemudian hari ada pelanggaran atas surat pernyataan ini, saya bersedia menerima sanksinya.

Malang, 31 Juli 2012

Yang membuat Pernyataan,

  
Tirta Wicaksono  
NIM .0712521



## ABSTRAK

### PERANCANGAN DAN PEMBUATAN SISTEM PENDUKUNG KEPUTUSAN DIAGNOSA PENYAKIT MATA MENGGUNAKAN METODE *FUZZY LOGIC*

Tirto Wicaksono, NIM 0712521

Dosen Pembimbing : M. Ibrahim Ashari, ST, MT dan Sandy Nataly M, S.Kom

Sebagai salah satu dari 5 indra manusia, mata merupakan indra penglihatan yang sangat berharga. Tetapi sebagian orang menganggap gejala gejala umum yang dialaminya seperti mata merah, gatal, mata berair, dll adalah hal biasa dan akan sembuh dengan sendirinya, pada kenyataannya gejala gejala umum tersebut mungkin merupakan gejala penyakit mata yang lebih serius.

Seorang dokter dalam mendiagnosa suatu penyakit, dia akan melihat gejala gejala apa yang terjadi pada pasien lalu menyimpulkan penyakit apa yang diderita pasien. Cara yang digunakan dokter tersebut memiliki kesamaan dengan metode fuzzy logic sehingga metode fuzzy logic dapat digunakan untuk membantu mendiagnosa suatu penyakit.

Skripsi ini akan membahas perancangan dan pembuatan suatu system pendukung keputusan yang akan membantu seorang dokter dalam mendiagnosa suatu penyakit mata, serta system ini juga digunakan untuk manajemen pasien pasien tersebut sehingga seorang dokter dapat lebih mudah memantau keadaan pasien-pasiennya.

**Kata Kunci:** *Sistem Pendukung Keputusan, Fuzzy Logic, Penyakit Mata.*

## KATA PENGANTAR

Puji syukur atas kehadiran Tuhan Yang Maha Kuasa atas segala limpahan berkat dan rahmat-Nya sehingga dapat menyelesaikan skripsi yang berjudul “Perancangan dan Pembuatan Sistem Pendukung Keputusan Diagnosa Penyakit Mata Menggunakan Metode *Fuzzy Logic*” ini dengan baik. Skripsi ini merupakan persyaratan kelulusan studi di jurusan Teknik Elektro S-1 konsentrasi Teknik Komputer dan Informatika ITN Malang dan untuk mencapai gelar sarjana teknik.

Keberhasilan penyelesaian laporan skripsi ini tidak lepas dari dukungan dan bantuan berbagai pihak. Untuk itu penulis menyampaikan terima kasih kepada:

1. Bapak Ir. Soeparno Djiwo, MT selaku Rektor ITN Malang.
2. Bapak Ir. Sidik Noertjahjono, MT selaku Dekan Fakultas Teknologi Industri.
3. Bapak Ir. Yusuf Ismail Nakhoda, MT selaku Ketua Jurusan Teknik Elektro S-1.
4. Bapak Dr. Eng. Aryuanto Soetedjo, ST, MT selaku Sekretaris Jurusan Teknik Elektro S-1.
5. Bapak M. Ibrahim Ashari, ST, MT selaku Dosen Pembimbing I.
6. Ibu Sandy Nataly M, S.Kom selaku Dosen Pembimbing II.
7. Semua pihak yang telah membantu penulis dalam menyelesaikan skripsi ini yang tidak bisa penulis sebutkan satu persatu.

Akan keterbatasan pengetahuan dalam menyelesaikan laporan ini, untuk itu penulis mengharapkan saran dan kritik yang membangun dari pembaca demi kesempurnaan laporan ini.

Harapan penulis semoga laporan ini memberikan manfaat bagi perkembangan ilmu pengetahuan dan pembaca.

Malang, Juli 2012

Penulis

## DAFTAR ISI

LEMBAR PERSETUJUAN .....	i
SURAT PERNYATAAN ORISINALITAS.....	ii
ABSTRAK.....	iii
KATA PENGANTAR.....	iv
DAFTAR ISI .....	v
DAFTAR GAMBAR.....	viii
DAFTAR TABEL .....	x
BAB I PENDAHULUAN .....	1
1.1. Latar Belakang .....	1
1.2. Rumusan Masalah.....	2
1.3. Tujuan .....	2
1.4. Batasan Masalah .....	2
1.5. Metodologi Penelitian.....	2
1.6. Sistematika Penulisan .....	3
BAB II TINJAUAN PUSTAKA .....	4
2.1. Sistem Pendukung Keputusan.....	4
2.2. <i>Fuzzy Logic</i> .....	5
2.2.1. Himpunan Fuzzy.....	5
2.2.2. Variabel Linguistik.....	8
2.2.3. Fuzzyfikasi.....	8
2.2.4. Inferencing (Rule Base).....	8
2.2.5. Defuzzyfikasi .....	9
2.2.6. Metodologi dan Desain Sistem Fuzzy .....	9
2.3. Microsoft SQL Server.....	10
2.4. Microsoft Visual Basic .NET.....	10
2.5. SQL.....	10
2.6. Microsoft Visual Studio.....	11

2.7.	Mata .....	12
2.7.1.	Organ Mata .....	12
2.7.2.	Sistem Kerja Mata .....	13
2.8.	Penyakit Mata .....	15
2.8.1.	Konjungtivitis .....	15
2.8.2.	Endoftalmitis .....	15
2.8.3.	Trachoma .....	16
2.8.4.	Blefaritis .....	16
2.8.5.	Dakriosistitis .....	16
2.8.6.	Ulkus Kornea .....	16
2.8.7.	Keratokonjungtivitis Sika .....	17
2.8.8.	Keratokonjungtivitis Epidermi .....	17
2.8.9.	Keratitis Neuropalitik .....	17
2.8.10.	Ulkus Serpens .....	18
2.8.11.	Selusitis Orbitalis (SO) .....	18
2.8.12.	Keratokonjungtivitas Vernalis (Kv) .....	18
BAB III PERANCANGAN DAN DESAIN SISTEM .....		19
3.1.	Deskripsi Umum Sistem .....	19
3.2.	Perancangan Sistem .....	19
3.2.1.	Proses Fuzzyfikasi Data .....	20
3.2.2.	Proses Learning .....	22
3.2.3.	Proses Membangun Tree .....	23
3.2.4.	Proses Menghitung Fuzzy Entropy Dan Information Gain .....	24
3.2.5.	Contoh Perhitungan Manual .....	27
3.3.	Data Flow Diagram .....	36
3.3.1.	Contex Diagram .....	36
3.3.2.	DVD LV 0 .....	37
3.3.3.	DVD LV 1 .....	37

BAB IV. IMPLEMENTASI DAN PEMBAHASAN.....	39
4.1. Lingkungan Implementasi .....	39
4.1.1. Perangkat Keras .....	39
4.1.2. Perangkat Lunak .....	39
4.2. Implementasi Program.....	39
4.2.1. Penyimpanan Rule.....	39
4.2.2. Proses Pemanggilan Rule Dalam Pendiagnosaan.....	41
4.2.3. Proses Pencocokan Bobot Diagnosa Dengan Bobot Rule .....	43
4.2.4. Proses Defuzzyfikasi .....	44
4.3. Implementasi Program .....	45
BAB V. PENUTUP .....	52
5.1. Kesimpulan .....	52
5.2. Saran .....	52
DAFTAR PUSTAKA .....	53
LAMPIRAN .....	54



## DAFTAR GAMBAR

Gambar 2.1. Pendefinisian kecepatan dalam bentuk logika <i>fuzzy</i> dan logika <i>Boolean</i> .....	6
Gambar 2.2. <i>S-function</i> .....	6
Gambar 2.3. $\pi$ - <i>function</i> .....	7
Gambar 2.4. <i>T-function</i> .....	7
Gambar 2.5. <i>Metodologi pengembangan sistem fuzzy</i> .....	9
Gambar 3.1. Alur Proses.....	19
Gambar 3.2. Proses Fuzzyfikasi Data.....	20
Gambar 3.3. Himpunan Fuzzy Atribut Gejala.....	21
Gambar 3.4. Proses Learning.....	22
Gambar 3.5. Proses Membangun Tree .....	23
Gambar 3.6. Perhitungan Fuzzy Entropy Keseluruhan Data.....	25
Gambar 3.7. Fuzzy Entropy Suatu Atribut.....	26
Gambar 3.8. Proses Perhitungan Information Gain.....	26
Gambar 3.9. Ekspansi Tree 1.....	31
Gambar 3.10. Ekspansi Tree 2.....	34
Gambar 3.11. Ekspansi Tree 3.....	35
Gambar 3.12. Context Diagram.....	36
Gambar 3.13. DVD LV 0 .....	37
Gambar 3.14. DVD LV 1 .....	38
Gambar 4.1. <i>Source code</i> penyimpanan rule <i>fuzzy</i> .....	41
Gambar 4.2. Prosedur Pengambilan Rule.....	43
Gambar 4.3. Prosedur Pencocokan Bobot.....	44
Gambar 4.4. Prosedur Defuzzyfikasi.....	45
Gambar 4.5. Tampilan Form Admin .....	46
Gambar 4.6. Tampilan Form Utama.....	46
Gambar 4.7. Tampilan Form Data Pasien .....	47
Gambar 4.8. Tampilan Form Data Pegawai .....	47
Gambar 4.9. Tampilan Form Data Penyakit.....	48

Gambar 4.10. Tampilan Form Data Gejala .....	48
Gambar 4.11. Tampilan Form Data Aturan.....	49
Gambar 4.12. Tampilan Form Proses Diagnosa.....	50
Gambar 4.13. Tampilan Form Hasil Diagnosa.....	50
Gambar 4.14. Tampilan Form Rekap Medik.....	51

## DAFTAR TABEL

Tabel 3.1. Gejala Dan Penyakit.....	21
Tabel 3.2. Contoh Data Pelatihan.....	27
Tabel 3.3. Fuzzy Input Atribut Gejala 1.....	28
Tabel 3.4. Fuzzy Input Atribut Gejala 2.....	28
Tabel 3.5. Fuzzy Input Atribut Gejala 3.....	28
Tabel 3.6. Nilai fuzzy Entropy Dan Information Gain 1.....	31
Tabel 3.7. Nilai fuzzy Entropy Dan Information Gain 2.....	28
Tabel 4.1. Tabel Kebenaran.....	51

# BAB I

## PENDAHULUAN

### 1.1. Latar Belakang

Mata merupakan organ yang sangat penting bagi kehidupan manusia, selama manusia hidup, mata menjadi jendela kehidupan dimana kita diberi anugerah untuk dapat menikmati indahnya hasil karya Tuhan. Maka sudah seharusnya kita menjaga salah satu dari kelima indra penting ini.

Orang biasa biasanya menganggap gejala-gejala penyakit pada mata mereka dengan remeh karena mereka hanya menganggap hal itu mungkin merupakan terkena debu yang berterbangan di sekeliling mereka, misalnya mata terasa nyeri, merah pada mata, mata berair, lelah pada mata. Padahal gejala-gejala tersebut mungkin merupakan gejala dari suatu penyakit mata yang serius yang pada akhirnya mungkin dapat menyebabkan kebutaan.

Untuk mendiagnosa penyakit mata dibutuhkan seorang ahli mata yaitu dokter mata, dokter dalam mendiagnosa penyakit pasti melihat gejala-gejala yang terjadi pada pasien. Kemudian dari gejala-gejala tersebut yang didapatkan akan digunakan oleh seorang dokter untuk menentukan penyakit mata apa yang dialami oleh pasien tersebut. Cara dokter dalam menentukan penyakit tersebut memiliki kesamaan dengan *fuzzy logic*, maka *fuzzy logic* dapat digunakan untuk membantu dokter maupun orang awam untuk mendiagnosa suatu penyakit mata.

Dalam *fuzzy logic* untuk diagnosa ini, Gejala-gejala penyakit mata diberikan bobot masing-masing untuk menentukan seberapa parah gejala tersebut lalu dari *crisp sets* yang didapatkan akan diubah menjadi *fuzzy sets*, dari *fuzzy sets-fuzzy sets* yang didapatkan dari masing-masing gejala digunakan dalam aturan *if-then* untuk menentukan seberapa parah penyakit tersebut.

### 1.2. Rumusan Masalah

Dalam skripsi ini, ruang lingkup permasalahan dibatasi pada bagaimana membuat program aplikasi dalam bentuk perangkat lunak (*software*) yang dipergunakan

untuk mendiagnosa penyakit mata beserta keterangannya dalam usaha memberikan masukan bagi dokter maupun pihak-pihak yang terkait.

Karena permasalahan penyakit mata ini cukup kompleks maka permasalahan pada skripsi ini dapat dirumuskan sebagai berikut :

1. Bagaimana membangun aplikasi dalam membantu mendiagnosa penyakit pada mata?
2. Bagaimana *fuzzy logic* dapat diterapkan dalam mendiagnosa penyakit mata?

### 1.3. Tujuan

Tujuan penulisan skripsi ini adalah :

1. Membuat program aplikasi sebagai alat bantu dalam mendiagnosa suatu penyakit mata berdasarkan gejala-gejala yang timbul.
2. Menerapkan *fuzzy logic* dalam proses mendiagnosa penyakit mata.

### 1.4. Batasan Masalah

Permasalahan yang akan dibahas dalam penulisan ini dibatasi pada:

1. Pembuatan aplikasi ini berdasarkan pada gejala-gejala yang umum dialami oleh pasien dan tidak termasuk tes laboratorium.
2. Program aplikasi ini menggunakan 3 kondisi tingkat *fuzzy sets*.
3. Solusi yang dihasilkan hanya bersifat rekomendasi, tidak merupakan solusi yang mutlak.
4. Aplikasi ini dibuat dengan Visual Basic .Net dengan database SQL SERVER.

### 1.5. Metodologi Penelitian

Adapun metode penelitian yang digunakan adalah sebagai berikut:

1. Studi literatur  
Pengumpulan data yang dilakukan dengan mencari bahan-bahan kepustakaan dan referensi dari berbagai sumber sebagai landasan teori yang ada hubungannya dengan permasalahan yang dijadikan objek penelitian.
  2. Analisa Kebutuhan Sistim  
Data dan informasi yang telah diperoleh akan dianalisa agar didapatkan kerangka global yang bertujuan untuk mendefinisikan kebutuhan sistim di mana nantinya akan digunakan sebagai acuan perancangan sistim.
-

### 3. Perancangan dan Implementasi

Berdasarkan data dan informasi yang telah diperoleh serta analisa kebutuhan untuk membangun sistim ini, akan dibuat rancangan kerangka global yang menggambarkan mekanisme dari sistim yang akan dibuat dan diimplementasikan kedalam sistim.

### 4. Eksperimen dan Evaluasi

Pada tahap ini, sistim yang telah selesai dibuat akan diuji coba, yaitu pengujian berdasarkan fungsionalitas program, dan akan dilakukan koreksi dan penyempurnaan program jika diperlukan.

## 1.6. Sistematika Penulisan

Untuk mempermudah dan memahami pembahasan penulisan skripsi ini, maka sistematika penulisan disusun sebagai berikut :

#### Bab I : Pendahuluan

Berisi Latar Belakang, Rumusan Masalah, Tujuan Penelitian, Pembatasan Permasalahan, Metode Penelitian dan Sistematika Penulisan.

#### Bab II : Tinjauan Pustaka

Berisi tentang landasan teori mengenai permasalahan yang berhubungan dengan penelitian yang dilakukan.

#### Bab III : Perancangan dan Analisa Sistim

Dalam bab ini berisi mengenai analisa kebutuhan sistim baik software maupun hardware yang diperlukan untuk membuat kerangka global yang menggambarkan mekanisme dari sistim yang akan dibuat. .

#### Bab IV : Pembuatan dan Pengujian Sistim

Berisi tentang implementasi dari perancangan sistim yang telah dibuat serta pengujian terhadap sistim tersebut.

#### Bab V : Penutup

Merupakan bab terakhir yang memuat intisari dari hasil pembahasan yang berisikan kesimpulan dan saran yang dapat digunakan sebagai pertimbangan untuk pengembangan penulisan selanjutnya.

---

## BAB II

### TINJAUAN PUSTAKA

#### 2.1. Sistem Pendukung Keputusan

Sistem pendukung keputusan (Inggris: *decision support systems* disingkat *DSS*) adalah bagian dari sistem informasi berbasis komputer (termasuk sistem berbasis pengetahuan (manajemen pengetahuan)) yang dipakai untuk mendukung pengambilan keputusan dalam suatu organisasi atau perusahaan.

Dapat juga dikatakan sebagai sistem komputer yang mengolah data menjadi informasi untuk mengambil keputusan dari masalah semi-terstruktur yang spesifik.

Menurut *Moore and Chang*, SPK dapat digambarkan sebagai sistem yang berkemampuan mendukung analisis *ad hoc* data, dan pemodelan keputusan, berorientasi keputusan, orientasi perencanaan masa depan, dan digunakan pada saat-saat yang tidak biasa.

Tahapan SPK:

- Definisi masalah
- Pengumpulan data atau elemen informasi yang relevan
- pengolahan data menjadi informasi baik dalam bentuk laporan grafik maupun tulisan
- menentukan alternatif-alternatif solusi (bisa dalam persentase)

Tujuan dari SPK:

- Membantu menyelesaikan masalah semi-terstruktur
- Mendukung manajer dalam mengambil keputusan
- Meningkatkan efektifitas bukan efisiensi pengambilan keputusan

Dalam pemrosesannya, SPK dapat menggunakan bantuan dari sistem lain seperti *Artificial Intelligence*, *Expert Systems*, *Fuzzy logic*, dll.

## 2.2. Fuzzy logic

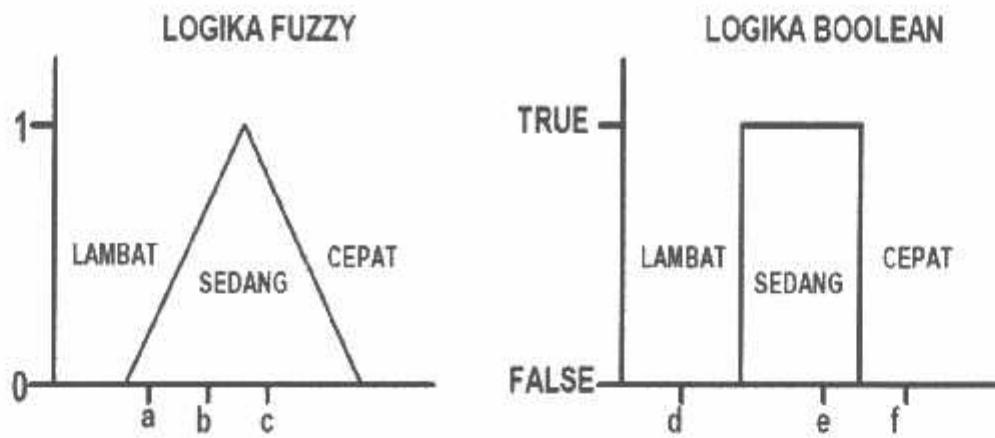
Jaringan Logika *fuzzy* adalah cabang dari sistem kecerdasan buatan (*Artificial Intelegent*) yang mengemulasi kemampuan manusia dalam berfikir ke dalam bentuk algoritma yang kemudian dijalankan oleh mesin 7 . Algoritma ini digunakan dalam berbagai aplikasi pemrosesan data yang tidak dapat direpresentasikan dalam bentuk biner. Logika *fuzzy* menginterpretasikan statemen yang samar menjadi sebuah pengertian yang logis.

Logika *Fuzzy* pertama kali diperkenalkan oleh *Prof. Lotfi Zadeh* seorang kebangsaan Iran yang menjadi guru besar di *University of California at Berkeley* pada tahun 1965 dalam papernya yang monumental. Dalam paper tersebut dipaparkan ide dasar *fuzzy set* yang meliputi *inclusion, union, intersection, complement, relation* dan *convexity*. Pelopor aplikasi *fuzzy set* dalam bidang kontrol, yang merupakan aplikasi pertama dan utama dari *fuzzy set* adalah *Prof. Ebrahim Mamdani* dan kawan-kawan dari *Queen Mary College London*. Penerapan kontrol *fuzzy* secara nyata di industri banyak dipelopori para ahli dari Jepang, misalnya *Prof. Sugeno* dari *Tokyo Institute of Technology*, *Prof. Yamakawa* dari *Kyusu Institute of Technology*, *Togay* dan *Watanabe* dari *Bell Telephone Labs*.

### 2.8.1. Himpunan Fuzzy

Himpunan *fuzzy* merupakan suatu pengembangan lebih lanjut tentang konsep himpunan dalam matematika. Himpunan *fuzzy* adalah rentang nilai-nilai. Masing-masing nilai mempunyai derajat keanggotaan (*membership*) antara 0 sampai dengan 1. Ungkapan logika *Boolean* menggambarkan nilai-nilai “benar” atau “salah”. Logika *fuzzy* menggunakan ungkapan misalnya : “sangat lambat”, “agak sedang”, “sangat cepat” dan lain-lain untuk mengungkapkan derajat intensitasnya. Ilustrasi antara keanggotaan *fuzzy* dengan *Boolean set* dapat dilihat pada gambar 2.1 dibawah ini :





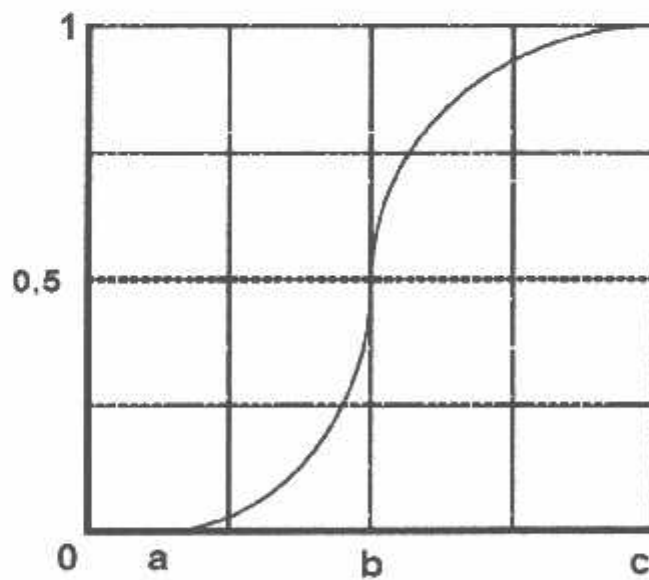
a=sangat lambat b=agak sedang c=sedikit cepat d=lambat e=sedang f=cepat

Gambar 2.1 Pendefinisian kecepatan dalam bentuk logika *fuzzy* dan logika *Boolean*

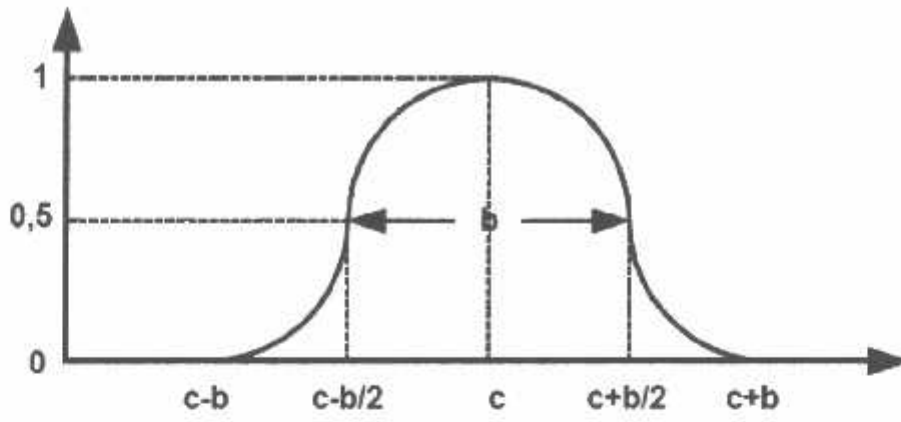
Logika *fuzzy* menggunakan satu set aturan untuk menggambarkan perilakunya. Aturan-aturan tersebut menggambarkan kondisi yang diharapkan dan hasil yang diinginkan dengan menggunakan *statemen IF... THEN*.

Penentuan keanggotaan suatu himpunan *fuzzy* tidak dibatasi oleh aturan-aturan tertentu. Contoh berikut ini adalah tiga macam keanggotaan yang dinyatakan dengan fungsi keanggotaan  $S$ ,  $\pi$ , dan  $T$  (*triangular*).

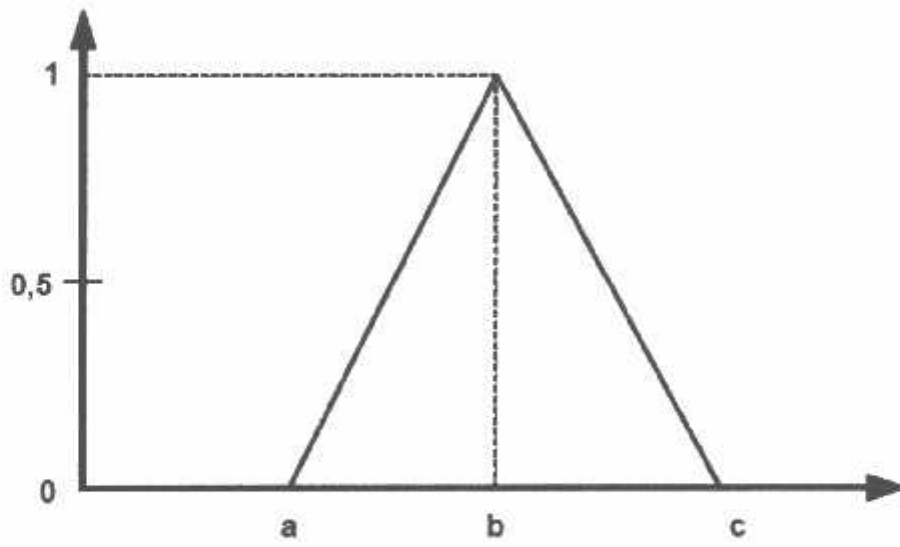
a. *S-function*



Gambar 2.2 *S-function*

b.  $\pi$ -functionGambar 2.3  $\pi$ -function

c. T-function



Gambar 2.4 T function

### 2.8.2. Variabel Linguistik

Suatu himpunan *fuzzy* bisa didefinisikan berdasarkan variabel linguistik tertentu. Variabel linguistik didefinisikan sebagai :

$$(u, T(u), U, R, S) \quad (2.8)$$

dengan  $U$  adalah nama variabel linguistik;  $T(u)$  adalah himpunan *term* (*linguistic value/linguistic label*) pada  $u$  dan masing-masing *term* didefinisikan dengan fungsi keanggotaan yang normal (mempunyai harga maksimum sama dengan 1) dan *convex* pada  $U$ ;  $R$  adalah aturan sintatik untuk menghasilkan nama nilai-nilai pada  $u$ ; dan  $S$  adalah aturan semantik untuk menghubungkan tiap nilai dengan artinya.

### 2.8.3. Fuzzyfikasi

Proses *fuzzyfikasi* merupakan proses untuk mengubah variabel *non fuzzy* (variabel numerik) menjadi variabel *fuzzy* (variabel linguistik). Nilai masukan-masukan yang masih dalam bentuk variabel numerik yang telah dikuantisasi sebelum diolah oleh pengendali *fuzzy* harus diubah terlebih dahulu ke dalam variabel *fuzzy*. Melalui fungsi keanggotaan yang telah disusun maka nilai-nilai masukan tersebut menjadi informasi *fuzzy* yang berguna nantinya untuk proses pengolahan secara *fuzzy* pula. Proses ini disebut *fuzzyfikasi*.

### 2.8.4. Inferencing (Rule Base)

Pada umumnya, aturan-aturan *fuzzy* dinyatakan dalam bentuk "IF...THEN" yang merupakan inti dari relasi *fuzzy*. Relasi *fuzzy*, dinyatakan dengan  $R$ , juga disebut implikasi *fuzzy*. Untuk mendapatkan aturan "IF...THEN" ada dua cara utama :

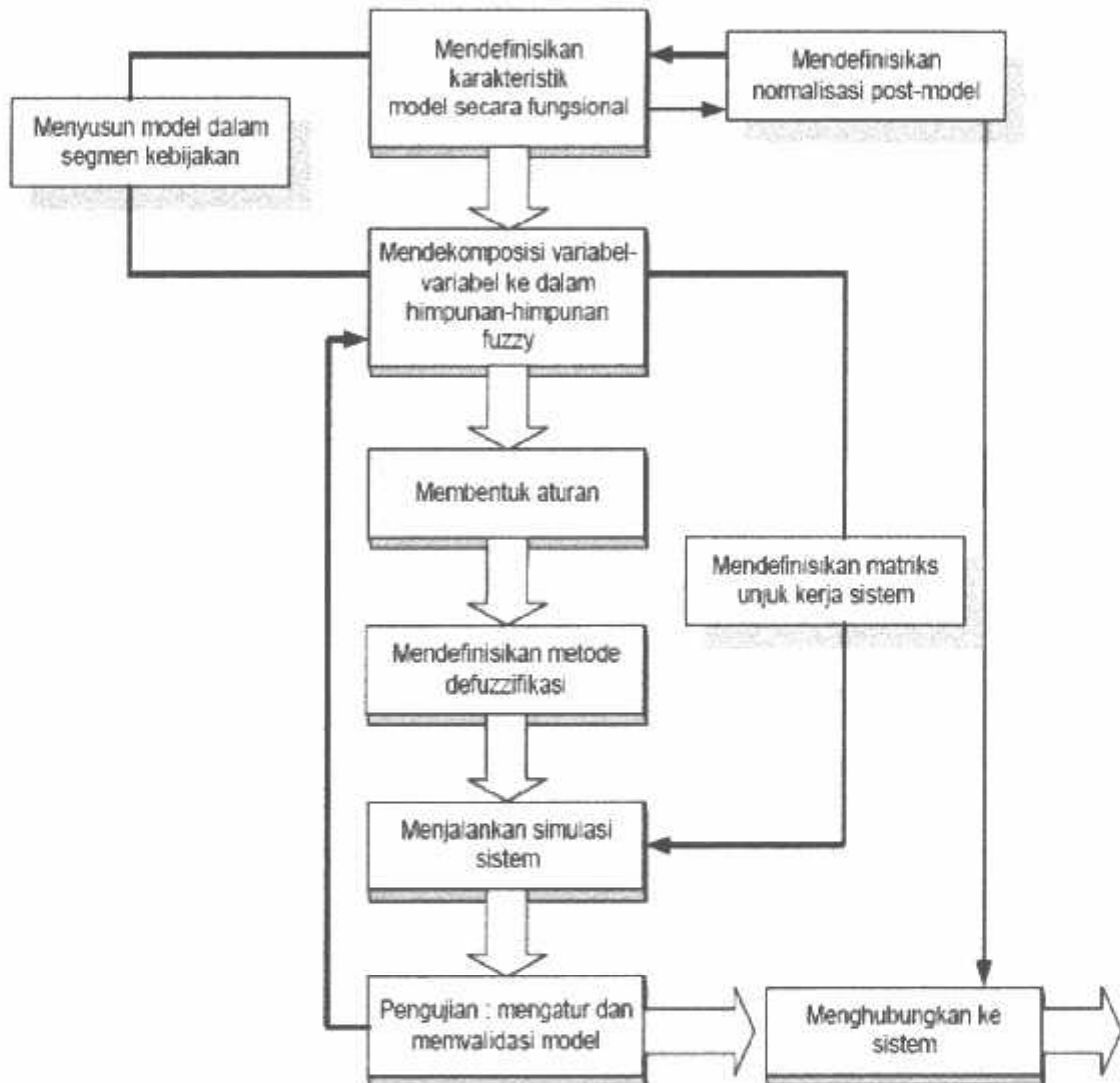
1. Menanyakan ke operator manusia yang dengan cara manual telah mampu mengendalikan sistem tersebut, dikenal dengan "*human expert*".
2. Dengan menggunakan algoritma pelatihan berdasarkan data-data masukan dan keluaran.

Dalam penalaran logika *fuzzy*, ada dua tipe utama untuk pengambilan keputusan *fuzzy* yaitu : *Generalized Modus Ponens (GMP)* dan *Generalized Modus Tolens (GMT)*. *GMP* disebut juga dengan *direct reasoning*, sedangkan *GMT* disebut juga *indirect reasoning*.

### 2.8.5. Defuzzyfikasi

Keputusan yang dihasilkan dari proses penalaran masih dalam bentuk *fuzzy*, yaitu berupa derajat keanggotaan keluaran. Hasil ini harus diubah kembali menjadi variabel numerik *non fuzzy* melalui proses defuzzyfikasi.

### 2.8.6. Metodologi Desain Sistem Fuzzy



Gambar 2.5 Metodologi pengembangan sistem fuzzy

### 2.3. Microsoft SQL Server

Server Microsoft SQL Server adalah sebuah sistem manajemen basis data relasional (RDBMS) produk Microsoft. Bahasa kueri utamanya adalah Transact-SQL yang merupakan implementasi dari SQL standar ANSI/ISO yang digunakan oleh Microsoft dan Sybase. Umumnya SQL Server digunakan di dunia bisnis yang memiliki basis data berskala kecil sampai dengan menengah, tetapi kemudian berkembang dengan digunakannya SQL Server pada basis data besar.

Microsoft SQL Server dan Sybase/ASE dapat berkomunikasi lewat jaringan dengan menggunakan protokol TDS (*Tabular Data Stream*). Selain dari itu, Microsoft SQL Server juga mendukung ODBC (*Open Database Connectivity*), dan mempunyai driver JDBC untuk bahasa pemrograman Java. Fitur yang lain dari SQL Server ini adalah kemampuannya untuk membuat basis data *mirroring* dan *clustering*.

### 2.4. Microsoft Visual Basic .NET

Pada Microsoft Visual Basic .NET adalah sebuah alat untuk mengembangkan dan membangun aplikasi yang bergerak di atas sistem .NET Framework, dengan menggunakan bahasa BASIC. Dengan menggunakan alat ini, para programmer dapat membangun aplikasi Windows Forms, Aplikasi web berbasis ASP.NET, dan juga aplikasi command-line. Alat ini dapat diperoleh secara terpisah dari beberapa produk lainnya (seperti Microsoft Visual C++, Visual C#, atau Visual J#), atau juga dapat diperoleh secara terpadu dalam Microsoft Visual Studio .NET. Bahasa Visual Basic .NET sendiri menganut paradigma bahasa pemrograman berorientasi objek yang dapat dilihat sebagai evolusi dari Microsoft Visual Basic versi sebelumnya yang diimplementasikan di atas .NET Framework. Peluncurannya mengundang kontroversi, mengingat banyak sekali perubahan yang dilakukan oleh Microsoft, dan versi baru ini tidak kompatibel dengan versi terdahulu.

### 2.5. SQL

SQL (Structured Query Language) adalah sebuah bahasa yang dipergunakan untuk mengakses data dalam basis data relasional. Bahasa ini secara de facto merupakan bahasa standar yang digunakan dalam manajemen basis data relasional. Saat ini hampir semua server basis data yang ada mendukung bahasa ini untuk melakukan manajemen datanya.

---

Secara umum, SQL terdiri dari dua bahasa, yaitu *Data Definition Language* (DDL) dan *Data Manipulation Language* (DML). Implementasi DDL dan DML berbeda untuk tiap sistem manajemen basis data (SMBD), namun secara umum implementasi tiap bahasa ini memiliki bentuk standar yang ditetapkan ANSI. Artikel ini akan menggunakan bentuk paling umum yang dapat digunakan pada kebanyakan SMBD.

DDL digunakan untuk mendefinisikan, mengubah, serta menghapus basis data dan objek-objek yang diperlukan dalam basis data, misalnya tabel, *view*, *user*, dan sebagainya. Secara umum, DDL yang digunakan adalah *CREATE* untuk membuat objek baru, *USE* untuk menggunakan objek, *ALTER* untuk mengubah objek yang sudah ada, dan *DROP* untuk menghapus objek. DDL biasanya digunakan oleh administrator basis data dalam pembuatan sebuah aplikasi basis data.

DML digunakan untuk memanipulasi data yang ada dalam suatu tabel. Perintah yang umum dilakukan adalah:

- *SELECT* untuk menampilkan data
- *INSERT* untuk menambahkan data baru
- *UPDATE* untuk mengubah data yang sudah ada
- *DELETE* untuk menghapus data

## 2.6. Microsoft Visual Studio

Microsoft Visual Studio merupakan sebuah perangkat lunak lengkap (suite) yang dapat digunakan untuk melakukan pengembangan aplikasi, baik itu aplikasi bisnis, aplikasi personal, ataupun komponen aplikasinya, dalam bentuk aplikasi console, aplikasi Windows, ataupun aplikasi Web. Visual Studio mencakup kompiler, SDK, Integrated Development Environment (IDE), dan dokumentasi (umumnya berupa MSDN Library). Kompiler yang dimasukkan ke dalam paket Visual Studio antara lain Visual C++, Visual C#, Visual Basic, Visual Basic .NET, Visual InterDev, Visual J++, Visual J#, Visual FoxPro, dan Visual SourceSafe.

Microsoft Visual Studio dapat digunakan untuk mengembangkan aplikasi dalam native code (dalam bentuk bahasa mesin yang berjalan di atas Windows) ataupun managed code (dalam bentuk Microsoft Intermediate Language di atas .NET Framework). Selain itu, Visual Studio juga dapat digunakan untuk mengembangkan aplikasi Silverlight, aplikasi Windows Mobile (yang berjalan di atas .NET Compact Framework).

---

Visual Studio kini telah menginjak versi Visual Studio 9.0.21022.08, atau dikenal dengan sebutan Microsoft Visual Studio 2008 yang diluncurkan pada 19 November 2007, yang ditujukan untuk platform Microsoft .NET Framework 3.5. Versi sebelumnya, Visual Studio 2005 ditujukan untuk platform .NET Framework 2.0 dan 3.0. Visual Studio 2003 ditujukan untuk .NET Framework 1.1, dan Visual Studio 2002 ditujukan untuk .NET Framework 1.0. Versi-versi tersebut di atas kini dikenal dengan sebutan Visual Studio .NET, karena memang membutuhkan Microsoft .NET Framework. Sementara itu, sebelum muncul Visual Studio .NET, terdapat Microsoft Visual Studio 6.0 (VS1998).

## 2.7. Mata

Mata adalah organ penglihatan yang mendeteksi cahaya. Yang dilakukan mata yang paling sederhana tak lain hanya mengetahui apakah lingkungan sekitarnya adalah terang atau gelap. Mata yang lebih kompleks dipergunakan untuk memberikan pengertian visual.

### 2.8.1. Organ Mata

Bagian-bagian pada organ mata bekerjasama mengantarkan cahaya dari sumbernya menuju ke otak untuk dapat dicerna oleh sistem saraf manusia. Bagian-bagian tersebut adalah:

- Kornea  
Merupakan bagian terluar dari bola mata yang menerima cahaya dari sumber cahaya.
  - Sklera  
Merupakan bagian dinding mata yang berwarna putih. Tebalnya rata-rata 1 milimeter tetapi pada irensi otot, menebal menjadi 3 milimeter.
  - Pupil dan iris  
Dari kornea, cahaya akan diteruskan ke pupil. Pupil menentukan kuantitas cahaya yang masuk ke bagian mata yang lebih dalam. Pupil mata akan melebar jika kondisi ruangan yang gelap, dan akan menyempit jika kondisi ruangan terang. Lebar pupil dipengaruhi oleh iris di sekelilingnya. Iris berfungsi sebagai diafragma. Iris inilah terlihat sebagai bagian yang berwarna pada mata.
  - Lensa  
Lensa mata menerima cahaya dari pupil dan meneruskannya pada retina. Fungsi
-

lensa mata adalah mengatur fokus cahaya, sehingga cahaya jatuh tepat pada bintik kuning retina. Untuk melihat objek yang jauh (cahaya datang dari jauh), lensa mata akan menipis. Sedangkan untuk melihat objek yang dekat (cahaya datang dari dekat), lensa mata akan menebal.

- Retina

Retina adalah bagian mata yang paling peka terhadap cahaya, khususnya bagian retina yang disebut bintik kuning. Setelah retina, cahaya diteruskan ke saraf optik.

- Saraf

Saraf yang memasuki sel tali dan kerucut dalam retina, untuk menuju ke otak.

### 2.8.2. Sistem kerja mata

Tatkala mengamati alam terbuka disekitar anda akan segera anda saksikan beragam benda terjauh dan terdekat dari anda dengan segala bentuk, warna, dan ukuran mereka. Pemandangan ini yang anda saksikan tanpa susah payah adalah hasil beragam reaksi rumit dalam tubuh anda. Kini marilah kita amati secara lebih dekat. Mata manusia memiliki cara kerja otomatis yang sempurna, mata dibentuk dengan 40 unsur utama yang berbeda dan kesemua bagian ini memiliki fungsi penting dalam proses melihat kerusakan atau ketiadaan salah satu fungsi bagiannya saja akan menjadikan mata mustahil dapat melihat.

Lapisan tembus cahaya di bagian depan mata adalah kornea, tepat dibelakangnya terdapat iris, selain member warna pada mata iris juga dapat merubah ukurannya secara otomatis sesuai kekuatan cahaya yang masuk, dengan bantuan otot yang melekat padanya. Misalnya ketika berada di tempat gelap iris akan membesar untuk memasukkan cahaya sebanyak mungkin. Ketika kekuatan cahaya bertambah, iris akan mengecil untuk mengurangi cahaya yang masuk ke mata. Sistem pengaturan otomatis yang berkeja pada mata bekerja sebagaimana berikut.

Ketika cahaya mengenai mata sinyal saraf terbentuk dan dikirimkan ke otak, untuk memberikan pesan tentang keberadaan cahaya, dan kekuatan cahaya. Lalu otak mengirim balik sinyal dan memerintahkan sejauh mana otot disekitar iris harus mengerut. Bagian mata lainnya yang bekerja bersamaan dengan struktur ini adalah lensa. Lensa bertugas memfokuskan cahaya yang memasuki mata pada lapisan retina di bagian belakang mata. Karena otot-otot disekeliling lensa cahaya yang datang ke mata dari berbagai sudut dan jarak berbeda dapat selalu difokuskan ke retina. Semua sistem yang telah kami sebutkan tadi berukuran lebih kecil, tapi jauh lebih unggul daripada

---



peralatan mekanik yang dibuat untuk meniru desain mata dengan menggunakan teknologi terbaru, bahkan sistem perekaman gambar buatan paling modern di dunia ternyata masih terlalu sederhana jika dibandingkan mata. Jika kita renungkan segala jerih payah dan pemikiran yang dicurahkan untuk membuat alat perekaman gambar buatan ini kita akan memahami betapa jauh lebih unggulnya teknologi penciptaan mata. Jika kita amati bagian-bagian lebih kecil dari sel sebuah mata maka kehebatan penciptaan ini semakin terungkap. Anggaplah kita sedang melihat mangkuk Kristal yang penuh dengan buah-buahan, cahaya yang datang dari mangkuk ini ke mata kita menembus kornea dan iris kemudian difokuskan pada retina oleh lensa jadi apa yang terjadi pada retina, sehinggasel-sel retina dapat merasakan adanya cahaya ketika partikel cahaya yang disebut foton mengenai sel-sel retina. Ketika itu mereka menghasilkan efek rantai layaknya sederetan kartu domino yang tersusun dalam barisan rapi. Kartu domino pertama dalam sel retina adalah sebuah molekul bernama 11-cis retinal. Ketika sebuah foton mengenainya molekul ini berubah bentuk dan kemudian mendorong perubahan protein lain yang berikatan kuat dengannya yakni rhodopsin. Kini rhodopsin berubah menjadi suatu bentuk yang memungkinkannya berikatan dengan protein lain yakni transdusin. Transdusin ini sebelumnya sudah ada dalam sel namun belum dapat bergabung dengan rhodopsin karena ketidaksesuaian bentuk. Penyatuan ini kemudian diikuti gabungan satu molekul lain yang bernama GTP kini dua protein yakni rhodopsin dan transdusin serta 1 molekul kimia bernama GTP telah menyatu tetapi proses sesungguhnya baru saja dimulai senyawa bernama GDP kini telah memiliki bentuk sesuai untuk mengikat satu protein lain bernama phosphodiesterase yang senantiasa ada dalam sel. Setelah berikatan bentuk molekul yang dihasilkan akan menggerakkan suatu mekanisme yang akan memulai serangkaian reaksi kimia dalam sel. Mekanisme ini menghasilkan reaksi ion dalam sel dan menghasilkan energy listrik energy ini merangsang saraf-saraf yang terdapat tepat di belakang sel retina. Dengan demikian bayangan yang ketika mengenai mata berwujud seperti foton cahaya ini meneruskan perjalanannya dalam bentuk sinyal listrik. Sinyal ini berisi informasi visual objek di luar mata. Agar mata dapat melihat sinyal listrik yang dihasilkan dalam retina harus diteruskan dalam pusat penglihatan di otak. Namun sel-sel saraf tidak berhubungan langsung satu sama lain ada celah kecil yang memisah titik-titik sambungan mereka lalu bagaimana sinyal listrik ini melanjutkan perjalanannya disini serangkaian mekanisme rumit terjadi energy listrik diubah menjadi energy kimia tanpa kehilangan informasi yang sedang dibawa dan dengan cara ini informasi diteruskan dari satu sel saraf ke sel

---

saraf berikutnya. Molekul kimia pengangkut ini yang terletak pada titik sambungan sel-sel saraf berhasil membawa informasi yang datang dari mata dari satu saraf ke saraf yang lain. Ketika dipindahkan ke saraf berikutnya sinyal ini diubah lagi menjadi sinyal listrik dan melanjutkan perjalanannya ke tempat titik sambungan lainnya dengan cara ini sinyal berhasil mencapai pusat penglihatan pada otak disini sinyal tersebut dibandingkan informasi yang ada di pusat memori dan bayangan tersebut ditafsirkan akhirnya kita dapat melihat mangkuk yang penuh buah-buahan sebagaimana kita saksikan sebelumnya karena adanya sistem sempurna yang terdiri atas ratusan komponen kecil ini dan semua rentetan peristiwa yang menakjubkan ini terjadi pada waktu kurang dari 1 detik.

## **2.8. Penyakit Mata**

### **2.8.1. Konjungtivitis**

Adalah iritasi/peradangan akibat infeksi pada bagian selaput yang melapisi mata. Gejalanya mata memerah, terasa nyeri, berair, gatal, keluar kotoran (belek), dan penglihatan (kabur). Penyakit yang mudah menular dan bisa berlangsung hingga berbulan-bulan ini disebabkan beberapa faktor, seperti infeksi virus atau bakteri, alergi (debu, serbuk, bulu, angin, atau asap), penggunaan lensa kontak yang kurang bersih, dan pemakaian lensa kontak jangka panjang. Penyakit ini dibagi menjadi 2 golongan yaitu Konjungtivitis Gonokokal dan Konjungtivitis Vernalis.

### **2.8.2. Endoftalmitis**

Merupakan infeksi yang terjadi di lapisan mata bagian dalam sehingga bola mata bernaah. Gejalanya berupa mata merah, nyeri, bahkan sampai mengalami gangguan penglihatan. Biasanya terjadi karena mata anak tertusuk sesuatu seperti lidi atau benda tajam lainnya. Infeksi ini cukup berat sehingga harus segera ditangani karena bisa menimbulkan kebutaan.

### **2.8.3. Trachoma**

Adalah infeksi pada mata yang disebabkan bakteri *Chlamydia trachomatis*. Bakteri ini berkembang biak di lingkungan yang kotor atau bersanitasi buruk. Lantaran itulah, trakoma sering menyerang anak-anak, terutama di berbagai negara berkembang. Pemaparan bakteri berlangsung saat anak menggunakan alat atau benda yang sudah tercemari *Chlamydia* seperti sapu tangan atau handuk. Gejala trakoma adalah mata

---

merah, mengeluarkan kotoran (belean), pembengkakan kelopak mata dan kelenjar getah bening, serta kornea kelihatan keruh. Penyakit ini sangat menular.

#### **2.8.4. Blefaritis**

Adalah suatu peradangan pada kelopak mata karena terjadinya produksi minyak yang berlebihan. Tidak diketahui persis mengapa produksi minyak bisa menjadi berlebihan. Sayangnya kelebihan minyak ini ada di dekat kelopak mata yang juga sering didatangi bakteri. Gejala blefaritis berupa mata merah, nyeri, panas, gatal, berair, ada luka di bagian kelopak mata dan membengkak. Pada beberapa kasus sampai terjadi kerontokan bulu mata. Ada dua jenis blefaritis yaitu blefaritis anterior dan blefaritis posterior. Yang pertama merupakan peradangan di kelopak mata bagian luar depan yaitu di tempat melekatnya bulu mata. Penyebabnya adalah bakteri stafilokokus. Yang kedua adalah peradangan di kelopak mata bagian dalam, yaitu bagian kelopak mata yang bersentuhan dengan mata. Penyebabnya adalah kelainan pada kelenjar minyak.

#### **2.8.5. Dakriosistitis**

Penyebab dakriosistitis adalah penyumbatan yang terjadi pada duktus nasolakrimalis yaitu saluran yang mengalirkan air mata ke hidung. Faktor alergilah yang menyebabkan terjadinya sumbatan pada saluran tersebut. Akibatnya adalah infeksi di sekitar kantung air mata yang menimbulkan nyeri, warna merah dan bengkak, bahkan bisa sampai mengeluarkan nanah dan penderita mengalami demam. Infeksi yang ringan biasanya akan cepat sembuh walau tetap ada pembengkakan. Sementara yang tergolong parah dapat menyebabkan kemerahan dan penebalan di atas kantung air mata. Jika terus berlanjut akan terbentuk kantung nanah.

#### **2.8.6. Ulkus Kornea**

Adalah infeksi pada kornea bagian luar. Biasanya terjadi karena jamur, virus, protozoa atau karena beberapa jenis bakteri, seperti stafilokokus, pseudomonas atau pneumokokus. Penyebab awal bisa karena mata kelilipan atau tertusuk benda asing. UK terkadang terjadi di seluruh permukaan kornea sampai ke bagian dalam dan belakang kornea. UK yang memburuk dapat menyebabkan komplikasi infeksi di bagian kornea yang lebih dalam, perforasi kornea (terjadi lubang), kelainan letak iris (selaput pelangi) dan kerusakan mata. Gejalanya mata merah, nyeri, gatal, berair, muncul kotoran mata,

---

peka terhadap cahaya (photo phobia) , pada bagian kornea tampak bintik nanah warna kuning keputihan, dan gangguan penglihatan.

#### **2.8.7. Keratokonjungtivitis Sika**

Adalah suatu keadaan dimana permukaan kornea kering dan konjungtiva. Terjadi pada penyakit-penyakit yang menyebabkan defisiensi komponen lemak air mata, kelenjar air mata, musin, akibat penguapan berlebihan, atau karena parut kornea atau hilangnya mikrovili kornea. Bila terjadi bersama arthritis reumatoid dan penyakit autoimun lain, disebut sebagai sindrom Sjogren. Gejalanya adalah gatal, mata seperti berpasir, silau, dan penglihatan kadang-kadang kabur. Terdapat gejala sekresi mukus yang berlebihan, sukar menggerakkan kelopak mata, mata tampak kering, dan terdapat erosi kornea. Pada pemeriksaan terdapat edema konjungtiva bulbi, hiperemis, menebal, dan kusam. Kadang terdapat benang mukus kekuning-kuningan pada forniks konjungtiva bawah. Keluhan berkurang bila mata dipejamkan.

#### **2.8.8. Keratokonjungtivitis Epidemik**

Adalah suatu peradangan kornea dan konjungtiva yang disebabkan oleh reaksi alergik terhadap adenovirus tipe 8. Biasanya unilateral, penyakit ini dapat timbul sebagai suatu epidemik. Gejalanya adalah demam, mata terasa seperti ada benda asing, kadang-kadang dapat ditemukan nyeri periorbita, sera mengalami tajam penglihatan menurun.

#### **2.8.9. Keratitis Neuropalitik**

Adalah peradangan kornea yang terjadi sebagai akibat hilangnya fungsi saraf sensorik kornea. Gejalanya adalah penurunan tajam penglihatan, tidak nyeri dan silau. Penyakit ini juga menyebabkan mata jarang berkedip karena hilangnya reflek berkedip dan permukaan kornea menjadi kusam.

#### **2.8.10. Ulkus Serpens**

Adalah ulkus kornea sentral yang berjalan cepat dan kebanyakan disebabkan oleh kuman pneumokok. Penyakit ini banyak terdapat pada petani, buruh tambang, orang-orang jompo, orang-orang dengan kesehatan yang buruk, atau pecandu alkohol dan obat bius. Biasanya ulkus ini terjadi karena didahului oleh trauma yang merusak apitel kornea dan akibat cacat kornea tersebut maka mudah terjadi invasi kuman kedalam kornea. Gejalanya adalah nyeri mata, silau, dan tajam penglihatan menurun.

---

### **2.8.11. Selulitis Orbitalis (So)**

Adalah suatu peradangan yang disebabkan oleh suatu kuman pada jaringan orbita. Penyakit ini biasanya terjadi akut yang disebabkan oleh kuman piogenik, dapat juga terjadi secara kronik, misalnya oleh karena lesi jamur dan sarkoidosis. Biasanya terjadi pada orang dengan gizi buruk atau pada anak-anak yang menderita sinusitis. Gejalanya adalah panas, penurunan visus, sakit saat menggerakkan bola mata.

### **2.8.12. Keratokonjungtivitis Vernalis (Kv)**

Konjungtivitis vernalis adalah salah satu bentuk dari konjungtivitis yang disebabkan oleh faktor alergi, disamping juga dipengaruhi oleh faktor, yakni; iklim, usia, dan jenis kelamin. Penyakit ini biasanya mengenai pasien muda antara 3-25 tahun. Pada laki-laki biasanya dimulai pada usia dibawah 10 tahun. Pada umumnya penderita konjungtivitis vernalis mengeluh gatal, mata merah, bengkak, dan mengeluarkan sekret atau kotoran. Konjungtivitis karena virus atau alergi mengeluarkan kotoran yang jernih.

---

## BAB III

### PERANCANGAN DAN DESAIN SISTEM

#### 3.1.Deskripsi Umum Sistem

Secara umum sistem yang dibangun adalah perangkat lunak untuk mendiagnosa penyakit mata dengan mengimplementasikan metode *fuzzy logic*. Sistem ini bertujuan untuk membantu seorang ahli dalam hal ini dokter dalam mendiganosa suatu penyakit mata yang mungkin diderita pasien berdasar gejala-gejala umum yang timbul. Hasil diagnosa dari sistem ini bersifat rekomendasi pada sang ahli atau dokter karena tujuan dari sistem ini adalah membantu ahli atau dokter tersebut bukan menggantikannya.

#### 3.2.Perancangan Sistem

Pada perancangan sistem ini, diperlukan beberapa tahapan yang perlu dilalui agar dapat membentuk sistem pendukung keputusan diagnosa penyakit mata dengan menggunakan metode *fuzzy logic*. Dalam proses diagnosa ini memerlukan input berupa data gejala-gejal yang umum dialami oleh pasien yang kemudian ditransformasikan ke dalam bentuk data *fuzzy* dan kemudian data tersebut mengalami proses learning. Alur sistem ini digambarkan dalam bentuk flowchart seperti pada gambar 3.1 berikut ini.



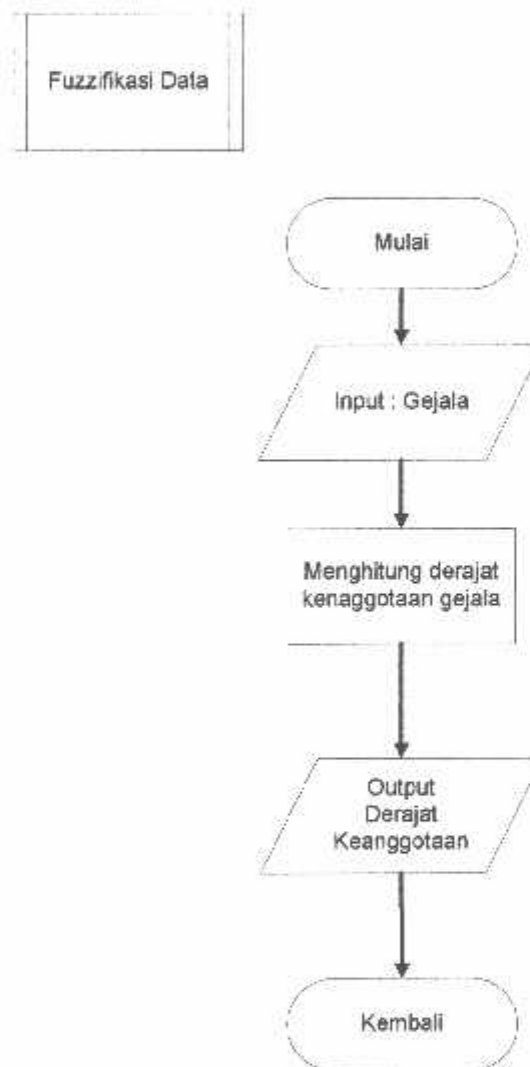
Gambar 3.1 Alur Proses

Berikut ini adalah penjelasan pada alur sistem:

1. Tahap awal adalah memasukkan data penderita penyakit mata.
2. Data tersebut diubah ke dalam bentuk data *fuzzy*.
3. Data pelatihan mengalami proses learning untuk membangun model klasifikasi dengan tree.

### 3.2.1. Proses Fuzzifikasi Data

Pada proses awal craps input diubah menjadi *fuzzy* input berdasarkan fungsi keanggotaan seperti pada gambar 3.2 berikut.



Gambar 3.2 proses fuzzifikasi data

Crips input gejala akan diubah kedalam *fuzzy* input yaitu tingkat keseringan gejala muncul. Data awal yang diperoleh dari beberapa sumber dan literatur adalah sebagai berikut.

Tabel 3.1 gejala dan penyakit

	rasa muntah	nyeri	berak	gatal	leher kerdas	pembelahan lidah	lengket	demam	otak	rasa ada benda asing
KONJUNKTIVITIS	✓	✓	✓	✓	✓	✓	✓			
STRABISME/INSIDIVITAS VISUALIS (SV)	✓		✓	✓	✓		✓		✓	
ENDOKRINITIS	✓	✓				✓			✓	
KEKUALIFIKASIAN (K)		✓				✓	✓	✓		
TRAKOMA	✓				✓		✓			
BAKTERIEMIA	✓			✓						✓
DIABETES MELITUS	✓						✓	✓		
LEUKEMIA	✓	✓		✓	✓			✓		
SEPSIS/NEKROTORIK						✓		✓		
SEPSIS/ORGANISITIS/EPIDEMIO		✓				✓		✓		✓
SEPSIS/ORGANISITIS/EPIDEMIO				✓		✓		✓		✓
LEUKEMIA		✓				✓		✓		✓

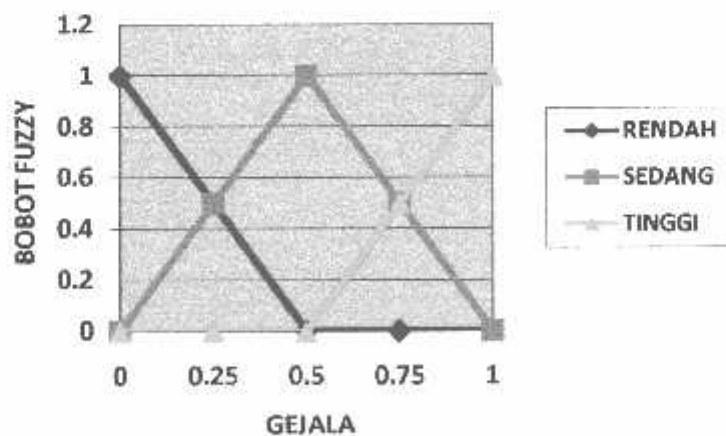
Atribut gejala dibagi menjadi 3 kelompok atau linguistic term yaitu rendah, sedang dan tinggi. Dari pembagian itu dapat ditentukan nilai derajat keanggotaan dari himpunan *fuzzy* rendah, sedang dan tinggi menggunakan persamaan sebagai berikut.

$$\mu_{rendah}(x) = \begin{cases} 1 & ; x < 0 \\ \frac{0.5 - x}{0.5} & ; 0 \leq x \leq 0.5 \\ 0 & ; x > 0.5 \end{cases}$$

$$\mu_{sedang}(x) = \begin{cases} 0 & ; x \leq 0 \text{ atau } x \geq 1 \\ \frac{x - 0.5}{0.5} & ; 0 \leq x \leq 0.5 \\ \frac{1 - x}{0.5} & ; 0.5 \leq x \leq 1 \end{cases}$$

$$\mu_{tinggi}(x) = \begin{cases} 0 & ; x < 0.5 \\ \frac{x - 0.5}{0.5} & ; 0.5 \leq x \leq 1 \\ 1 & ; x > 1 \end{cases}$$

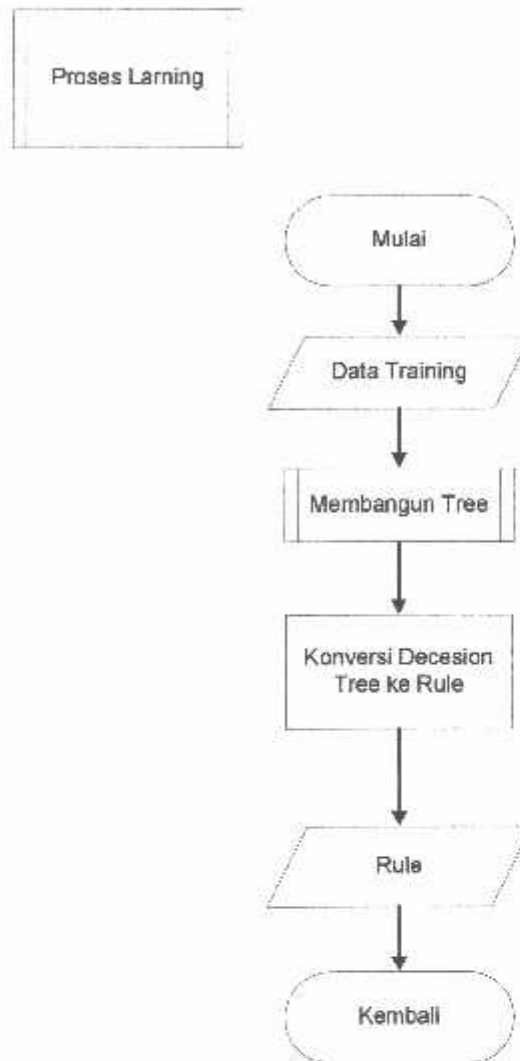
Himpunan *fuzzy* untuk setiap *linguistic term* atribut gejala menggunakan kurva seperti pada gambar 3.3.

Gambar 3.3 himpunan *fuzzy* atribut gejala



### 3.2.2. Proses Learning

Setelah sistem mendapatkan data yang telah difuzzifikasi, proses yang akan terjadi pada sistem selanjutnya adalah proses learning, yaitu membentuk proses diagnosa dengan decision tree seperti pada gambar 3.4 berikut.



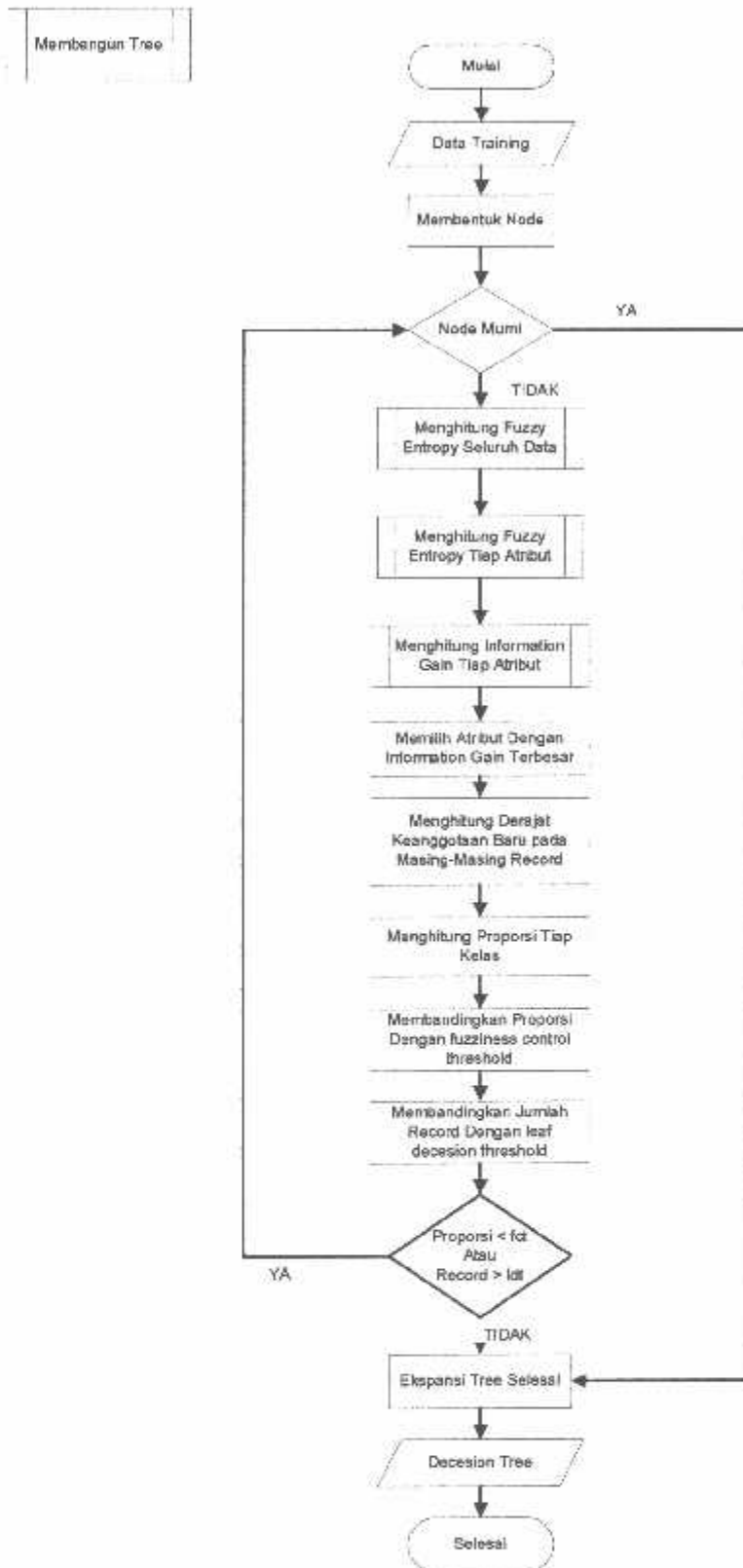
Gambar 3.4 proses learning

Berdasar gambar diatas, langkah-langkah dalam proses learning adalah sebagai berikut:

1. Sistem mendapatkan data training
2. Sistem membangun *fuzzy* decision training
3. Hasil dari prmbangunan tree di konversi kedalam bentuk rule.
4. Sistem menampilkan rule yang terbentuk

### 3.2.3. Proses membangun tree

Proses pembangunan tree apat dilihat pad gambar 3.5 berikut.



Gambar 3.5 proses membangun tree

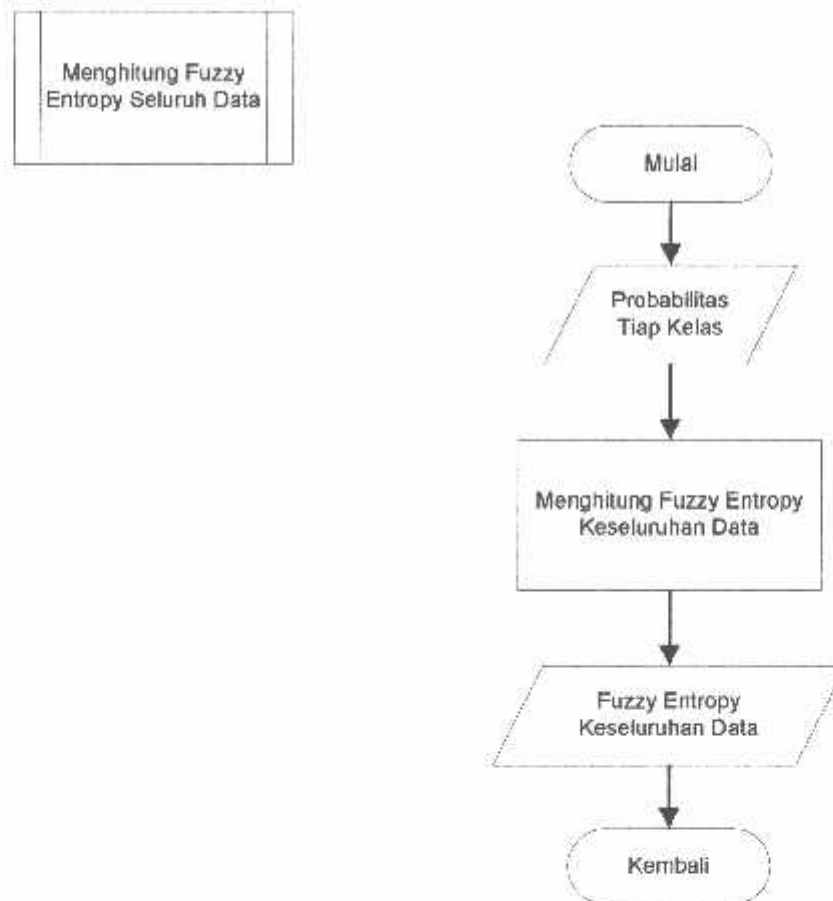
Proses membangun tree meliputi

1. Sistem mendapatkan data training
2. Sistem membentuk *node*.
3. *Node* dikatakan murni apabila sudah tidak ada lagi data yang akan diekspansi. Jika *node* sudah murni, maka proses pembangunan tree selesai. Jika *node* belum murni maka dilanjutkan ke langkah 4.
4. Menentukan *node* dengan menghitung *fuzzy entropy* dan *information gain* masing masing atribut kemudian memilih atribut dengan *information gain* terbesar yang akan menjadi *node*.
5. Atribut yang menjadi root memiliki derajat keanggotaan 1. Kemudian menghitung derajat keanggotaan *subnode* yang baru yaitu derajat keanggotaan *subnode* dikalikan dengan derajat keanggotaan atribut.
6. Menghitung proporsi kelas pada tiap *node*.
7. Membandingkan proporsi tiap kelas dengan *fuzziness control threshold*.
8. Bila proporsi lebih kecil dari pada *fuzziness control threshold* dan jumlah record lebih besar dari pada *leaf decision threshold*, maka dilakukan proses ekspansi tree. Bila proporsi lebih besar dari pada *fuzziness control threshold* dan jumlah record lebih kecil daripada *fuzziness control threshold* maka ekspansi tree berhenti dilakukan.
9. Mengecek kembali apakah *node* sudah murni atau belum. Jika sudah murni proses berhenti dan decision tree yang terbentuk adalah model tree yang dihasilkan. Jika belum, maka proses membagi *node* (langkah 4 sampai 8), dilakukan berulang-ulang sehingga semua *node* yang terbentuk menjadi murni dan sudah tidak ada lagi *node* yang bisa dibagi.

#### 3.2.4. Proses Menghitung *Fuzzy Entropy* dan *Information Gain*

Proses penghitungan *fuzzy entropy* dan *information gain* digunakan untuk menentukan atribut yang dijadikan root *node*. Sebelum menghitung *fuzzy entropy* atribut, maka dilakukan perhitungan *fuzzy entropy* keseluruhan. Langkah-langkahnya adalah sebagai berikut:

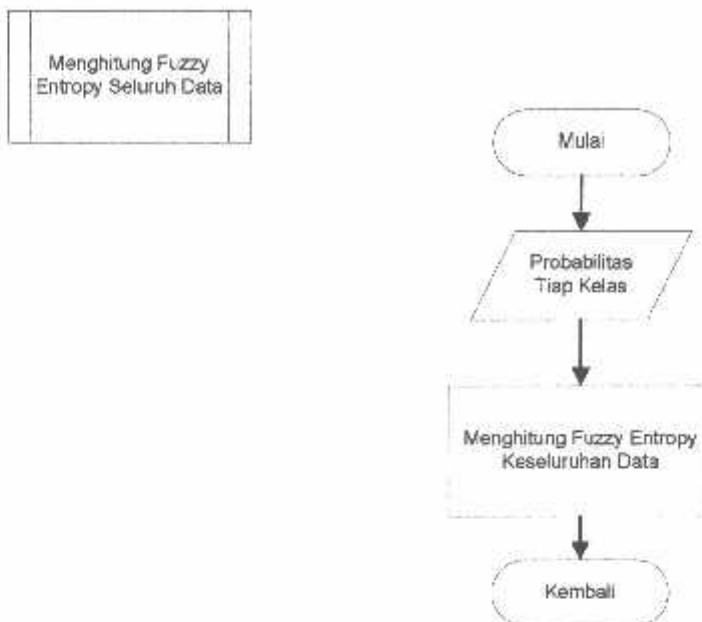
1. Menginputkan probabilitas tiap kelas untuk menghitung *fuzzy entropy* keseluruhan.
  2. Melakukan perhitungan *fuzzy entropy* keseluruhan atribut.
  3. Setelah dilakukan perhitungan akan mendapatkan nilai output yaitu *fuzzy entropy* keseluruhan data.
-



Gambar 3.6 perhitungan *Fuzzy Entropy* keseluruhan Data

Setelah mendapat output *fuzzy entropy* keseluruhan data, maka proses selanjutnya adalah proses perhitungan *fuzzy entropy* masing-masing atribut berdasarkan derajat keanggotaannya. Langkah-langkah pada proses ini adalah sebagai berikut:

1. Memasukkan nilai input keanggotaan atribut tiap kelas.
2. Melakukan perhitungan *fuzzy entropy* tiap atribut.
3. Setelah dilakukan perhitungan maka akan didapatkan output *fuzzy entropy* suatu atribut.

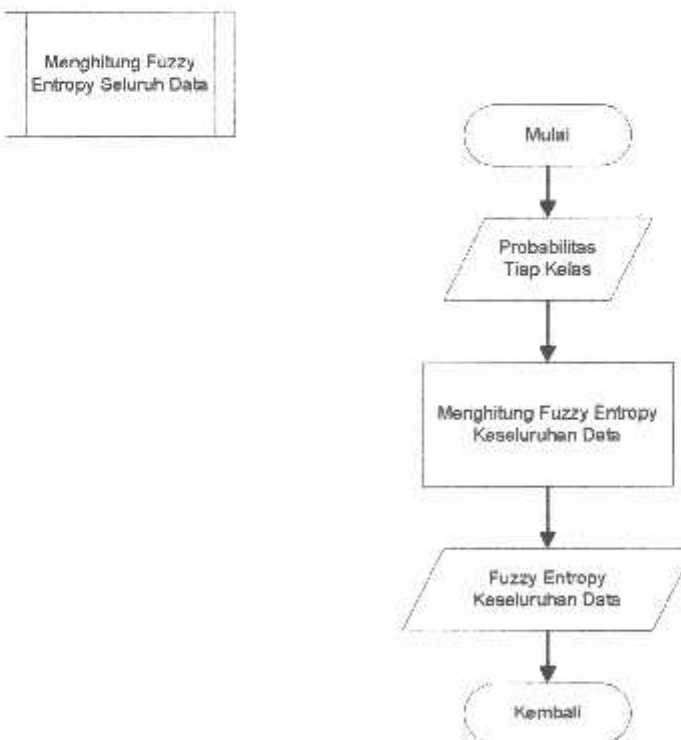


Gambar 3.7 Perhitungan *Fuzzy Entropy* Suatu Atribut

Proses selanjutnya adalah penghitungan *information gain* masing-masing atribut.

Langkah-langkah proses ini adalah sebagai berikut:

1. Memasukkan input *fuzzy entropy* keseluruhan atribut dan *fuzzy entropy* tiap atribut.
2. Melakukan perhitungan *information gain*
3. Setelah dilakukan perhitungan akan didapatkan output *information gain* masing-masing atribut.



Gambar 3.8 Proses Perhitungan *Information Gain*

### 3.2.5. Contoh perhitungan Manual (contoh untuk penyakit konjungtivitis)

Berikut ini merupakan contoh perhitungan manual untuk klasifikasi pendiagnosaan penyakit mata (konjungtivitis) dengan metode *fuzzy logic* menggunakan 10 data pelatihan dimana status 1 menandakan pasien terkena penyakit konjungtivitis sedangkan status 0 pasien tidak terkena penyakit konjungtivitis. Data pelatihan tersebut dapat dilihat pada tabel berikut.

Tabel 3.2 Contoh Data Pelatihan

Data Input penyakit KONJUNGTIVITIS				
NO	GEJALA 1	GEJALA 2	GEJALA 3	STATUS
1	mata merah	nyeri	berair	1
2	nyeri	berair	gatal	1
3	demam	bengkak	penglihatan kabur	0
4	mata merah	berair	keluar kotoran	1
5	silau	bengkak	mata merah	0
6	keluar kotoran	penglihatan kabur	mata merah	1
7	gatal	silau	terasa ada benda asing	0
8	mata merah	gatal	keluar kotoran	1
9	terasa ada benda asing	mata merah	nyeri	1
10	bengkak	keluar kotoran	demam	0

#### a. Proses Fuzzifikasi

Data input pelatihan tersebut kemudian diubah menjadi *fuzzy* input berdasarkan fungsi keanggotaan masing masing atribut. Dalam contoh tersebut pada penyakit konjungtivitis mata merah bernilai 0.75, nyeri bernilai 0.75, berair bernilai 0.5, gatal bernilai 0.5, keluar kotoran bernilai 0.25, penglihatan kabur bernilai 0.25, dan yang lainnya tidak bernilai. Yang selanjutnya nilai – nilai tersebut di fuzzifikasi menjadi nilai fuzzy dalam bentuk persen sesuai kurva pada gambar 3.3, maka didapat fuzzy input dri masing – masing gejala.

Tabel 3.3 *fuzzy* input atribut gejala 1

FUZZY INPUT ATRIBUT GEJALA 1					
NO	GEJALA	RENDAH	SEDANG	TINGGI	STATUS
1	mata merah	0	0.5	0.5	1
2	nyeri	0	0.5	0.5	1
3	demam	0	0	0	0
4	mata merah	0	0.5	0.5	1
5	silau	0	0	0	0
6	keluar kotoran	0.5	0.5	0	1
7	gatal	0	1	0	0
8	mata merah	0	0.5	0.5	1
9	terasa ada benda asing	0	0	0	1
10	bengkak	0	0	0	0
	TOTAL	0.5	3.5	2	

Tabel 3.4 *fuzzy* input atribut gejala 2

FUZZY INPUT ATRIBUT GEJALA 2					
NO	GEJALA	RENDAH	SEDANG	TINGGI	STATUS
1	nyeri	0	0.5	0.5	1
2	berair	0	1	0	1
3	bengkak	0	0	0	0
4	berair	0	1	0	1
5	bengkak	0	0	0	0
6	penglihatan kabur	0.5	0.5	0	1
7	silau	0	0	0	0
8	gatal	0	1	0	1
9	mata merah	0	0.5	0.5	1
10	keluar kotoran	0.5	0.5	0	0
	TOTAL	1	5	1	

Tabel 3.5 *fuzzy* atribut gejala 3

FUZZY INPUT ATRIBUT GEJALA 3					
NO	GEJALA	RENDAH	SEDANG	TINGGI	STATUS
1	berair	0	1	0	1
2	gatal	0	1	0	1
3	penglihatan kabur	0.5	0.5	0	0
4	keluar kotoran	0.5	0.5	0	1
5	mata merah	0	0.5	0.5	0
6	mata merah	0	0.5	0.5	1
7	terasa ada benda asing	0	0	0	0
8	keluar kotoran	0.5	0.5	0	1
9	nyeri	0	0.5	0.5	1
10	demam	0	0	0	0
	TOTAL	1.5	5	1.5	

b. Perhitungan *Fuzzy Entropy* dan *Information Gain*

Setelah diketahui *fuzzy* input masing masing atribut, selanjutnya adalah perhitungan mencari *fuzzy entropy* keseluruhan data.

$$\begin{aligned} H_f(s) &= -\left(\frac{6}{10} \log_2 \frac{6}{10}\right) - \left(\frac{4}{10} \log_2 \frac{4}{10}\right) \\ &= -(0.6 \times -0.73697) - (0.4 \times -1.32193) \\ &= 0.442182 + 0.528772 \\ &= 0.970954 \end{aligned}$$

Perhitungan selanjutnya adalah mencari *fuzzy entropy* dan *information gain* masing masing atribut.

1. Information gain atribut gejala 1

$$\begin{aligned} H_f(G1, r) &= -\left(\frac{0.5}{0.5} \log_2 \frac{0.5}{0.5}\right) - \left(\frac{0}{0.5} \log_2 \frac{0}{0.5}\right) \\ &= -(1 \times 0) - (-0) \\ &= 0 + 0 \\ &= 0 \end{aligned}$$

$$\begin{aligned} H_f(G1, s) &= -\left(\frac{2.5}{3.5} \log_2 \frac{2.5}{3.5}\right) - \left(\frac{1}{3.5} \log_2 \frac{1}{3.5}\right) \\ &= -(0.714 \times -0.48543) - (0.286 \times -1.80735) \\ &= 0.3466 - 0.517 \\ &= 0.8636 \end{aligned}$$

$$\begin{aligned} H_f(G1, t) &= -\left(\frac{2}{2} \log_2 \frac{2}{2}\right) - \left(\frac{0}{2} \log_2 \frac{0}{2}\right) \\ &= -(1 \times 0) - (-0) \\ &= 0 + 0 \\ &= 0 \end{aligned}$$

$$\begin{aligned} G_f(s, G1) &= 0.970954 - \left(\frac{0.5}{10} \times 0\right) - \left(\frac{3.5}{10} \times 0.8636\right) - \left(\frac{0.5}{10} \times 0\right) \\ &= 0.970954 - 0 - 0.30226 - 0 \\ &= 0.668694 \end{aligned}$$

2. Information gain atribut gejala 2

$$\begin{aligned} H_f(G2, r) &= -\left(\frac{0.5}{1} \log_2 \frac{0.5}{1}\right) - \left(\frac{0.5}{1} \log_2 \frac{0.5}{1}\right) \\ &= -(0.5 \times -1) - (0.5 \times -1) \\ &= 0.5 + 0.5 \\ &= 1 \end{aligned}$$



$$\begin{aligned}
 H_f(G2, s) &= -\left(\frac{4.5}{5} \log_2 \frac{4.5}{5}\right) - \left(\frac{0.5}{5} \log_2 \frac{0.5}{5}\right) \\
 &= -(0.9 \times -0.152) - (0.1 \times -3.32193) \\
 &= 0.1368 + 0.332193 \\
 &= 0.468993
 \end{aligned}$$

$$\begin{aligned}
 H_f(G2, t) &= -\left(\frac{1}{1} \log_2 \frac{1}{1}\right) - \left(\frac{0}{1} \log_2 \frac{0}{1}\right) \\
 &= -(1 \times 0) - (-0) \\
 &= 0 + 0 \\
 &= 0
 \end{aligned}$$

$$\begin{aligned}
 G_f(s, G2) &= 0.970954 - \left(\frac{1}{10} \times 1\right) - \left(\frac{5}{10} \times 0.468993\right) - \left(\frac{1}{10} \times 0\right) \\
 &= 0.970954 - 0.1 - 0.234496 - 0 \\
 &= 0.636458
 \end{aligned}$$

### 3. Information gain atribut gejala 3

$$\begin{aligned}
 H_f(G3, r) &= -\left(\frac{1}{1.5} \log_2 \frac{1}{1.5}\right) - \left(\frac{0.5}{1.5} \log_2 \frac{0.5}{1.5}\right) \\
 &= -(0.67 \times -0.58496) - (0.33 \times -1.58496) \\
 &= 0.3919232 + 0.5230368 \\
 &= 0.91496
 \end{aligned}$$

$$\begin{aligned}
 H_f(G3, s) &= -\left(\frac{4}{5} \log_2 \frac{4}{5}\right) - \left(\frac{1}{5} \log_2 \frac{1}{5}\right) \\
 &= -(0.8 \times 0.32193) - (0.2 \times -2.32193) \\
 &= 0.257544 + 0.464386 \\
 &= 0.72193
 \end{aligned}$$

$$\begin{aligned}
 H_f(G3, t) &= -\left(\frac{1}{1.5} \log_2 \frac{1}{1.5}\right) - \left(\frac{0.5}{1.5} \log_2 \frac{0.5}{1.5}\right) \\
 &= -(0.67 \times -0.58496) - (0.33 \times -1.58496) \\
 &= 0.3919232 + 0.5230368 \\
 &= 0.91496
 \end{aligned}$$

$$\begin{aligned}
 G_f(s, G3) &= 0.970954 - \left(\frac{1.5}{10} \times 0.91496\right) - \left(\frac{5}{10} \times 0.72193\right) - \left(\frac{1.5}{10} \times 0.91496\right) \\
 &= 0.970954 - 0.137244 - 0.360965 - 0.137244 \\
 &= 0.335501
 \end{aligned}$$

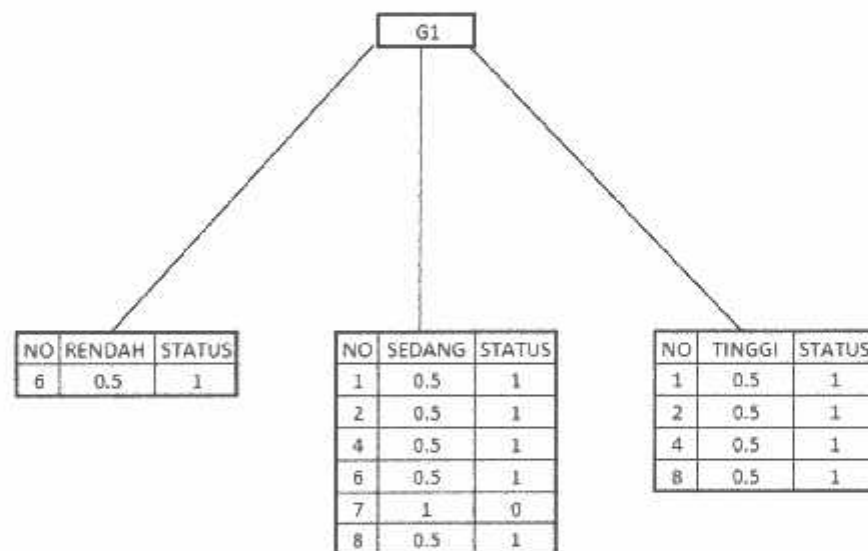
c. Pembentukan *Tree*

Berdasarkan perhitungan diatas, maka didapat nilai *fuzzy entropy* dan *information gain* setiap atribut dalam tabel berikut.

Tabel 3.6. nilai *fuzzy entropy* dan *information gain* 1

HASIL PENCARIAN NILAI	HASIL
<i>FUZZY ENTROPY</i>	0.970954
<i>IG ATRIBUT GEJALA 1</i>	0.668694
<i>IG ATRIBUT GEJALA 2</i>	0.636458
<i>IG ATRIBUT GEJALA 3</i>	0.335501

Dari hasil tersebut dipilih atribut dengan nilai *information gain* terbesar yaitu 0.668694, atribut gejala 1 kemudian digunakan untuk mengekspansi tree atau menjadi *root node*. Maka hasil ekspansi tree yang terbentuk berdasarkan atribut gejala 1 adalah seperti gambar 3.9. berikut.



Gambar 3.9. Ekspansi Tree 1

Perhitungan tiap kelas yang ada pada tiap *node* sebagai berikut.

1. *Node* RENDAH

$$C1 = 0.5$$

$$C0 = 0$$

Proporsi kelas 1

$$\frac{0.5}{0.5 + 0} \times 100\% = 100\%$$

Proporsi kelas 0

$$\frac{0}{0.5 + 0} \times 100\% = 0\%$$

2. *Node* SEDANG

$$CI = 0.5 - 0.5 + 0.5 + 0.5 + 0.5 = 2.5$$

$$C0 = 1$$

Proporsi kelas 1

$$\frac{2.5}{2.5 + 1} \times 100\% = 71.43\%$$

Proporsi kelas 0

$$\frac{1}{2.5 + 1} \times 100\% = 28.57\%$$

3. *Node* TINGGI

$$CI = 0.5 + 0.5 + 0.5 + 0.5 = 2$$

$$C0 = 0$$

Proporsi kelas 1

$$\frac{2}{2 + 0} \times 100\% = 100\%$$

Proporsi kelas 0

$$\frac{0}{2 + 0} \times 100\% = 0\%$$

Fuzziness Control Threshold (FTC) yang digunakan adalah 75% sehingga *node* SEDANG yang akan diekspansi lagi karena proporsi *node* lebih kecil dari FTC. Sedangkan *node* RENDAH dan TINGGI berhenti diekspansi karena proporsinya lebih besar dari FTC.

Proses ekspansi tree selanjutnya yaitu dengan menghitung *Fuzzy Entropy* dan *Information Gain* data data tiap *node*.

*Node* SEDANG

$$\begin{aligned} H_f(s) &= -\left(\frac{5}{6} \log_2 \frac{5}{6}\right) - \left(\frac{1}{6} \log_2 \frac{1}{6}\right) \\ &= -(0.83 \times -0.26303) - (0.17 \times -2.58496) \\ &= 0.2183149 + 0.4394432 \\ &= 0.6577581 \end{aligned}$$

Perhitungan selanjutnya adalah mencari *fuzzy entropy* dan *information gain* masing masing atribut.

Information gain atribut gejala 2

$$\begin{aligned} H_f(G2, r) &= -\left(\frac{0.5}{0.5} \log_2 \frac{0.5}{0.5}\right) - \left(\frac{0}{0.5} \log_2 \frac{0}{0.5}\right) \\ &= -(1 \times 0) - (-0) \\ &= 0 + 0 \\ &= 0 \end{aligned}$$

$$\begin{aligned} H_f(G2, s) &= -\left(\frac{4}{4} \log_2 \frac{4}{4}\right) - \left(\frac{0}{4} \log_2 \frac{0}{4}\right) \\ &= -(1 \times 0) - (-0) \\ &= 0 + 0 \\ &= 0 \end{aligned}$$

$$\begin{aligned} H_f(G2, t) &= -\left(\frac{0.5}{0.5} \log_2 \frac{0.5}{0.5}\right) - \left(\frac{0.5}{0.5} \log_2 \frac{0.5}{0.5}\right) \\ &= -(1 \times 0) - (-0) \\ &= 0 + 0 \\ &= 0 \end{aligned}$$

$$\begin{aligned} G_f(s, G2) &= 0.65775181 - \left(\frac{0.5}{6} \times 0\right) - \left(\frac{4}{6} \times 0\right) - \left(\frac{0.5}{6} \times 0\right) \\ &= 0.65775181 - 0 - 0 - 0 \\ &= 0.65775181 \end{aligned}$$

Information gain atribut gejala 3

$$\begin{aligned} H_f(G3, r) &= -\left(\frac{1}{1} \log_2 \frac{1}{1}\right) - \left(\frac{0}{1} \log_2 \frac{0}{1}\right) \\ &= -(1 \times 0) - (-0) \\ &= 0 + 0 \\ &= 0 \end{aligned}$$

$$\begin{aligned} H_f(G3, s) &= -\left(\frac{3.5}{3.5} \log_2 \frac{3.5}{3.5}\right) - \left(\frac{0}{3.5} \log_2 \frac{0}{3.5}\right) \\ &= -(1 \times 0) - (-0) \\ &= 0 + 0 \\ &= 0 \end{aligned}$$

$$\begin{aligned} H_f(G3, t) &= -\left(\frac{0.5}{0.5} \log_2 \frac{0.5}{0.5}\right) - \left(\frac{0}{0.5} \log_2 \frac{0}{0.5}\right) \\ &= -(1 \times 0) - (-0) \end{aligned}$$


---

$$- 0 + 0$$

$$= 0$$

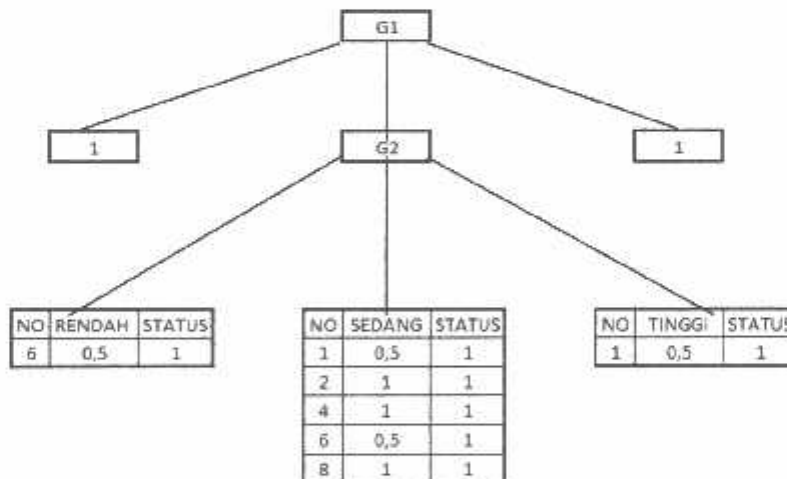
$$\begin{aligned} G_f(s, G3) &= 0.65775181 - \left(\frac{1}{6} \times 0\right) - \left(\frac{3.5}{6} \times 0\right) - \left(\frac{0.5}{6} \times 0\right) \\ &= 0.65775181 - 0 - 0 - 0 \\ &= 0.65775181 \end{aligned}$$

Berdasarkan perhitungan diatas, maka didapat nilai *fuzzy entropy* dan *information gain* setiap atribut dalam tabel 3.7. berikut.

Tabel 3.7. nilai *fuzzy entropy* dan *information gain* 2

HASIL PENCARIAN NILAI	HASIL
<i>FUZZY ENTROPY</i>	0.65775181
<i>IG ATRIBUT GEJALA 2</i>	0.65775181
<i>IG ATRIBUT GEJALA 3</i>	0.65775181

Dari hasil tersebut dipilih atribut dengan nilai *information gain* terbesar yaitu 0.65775181. atribut gejala 2 atau atribut gejala 3 kemudian digunakan untuk mengekspansi tree atau menjadi *root node*. Maka jika dipilih atribut gejala 2 hasil ekspansi tree yang terbentuk berdasarkan atribut gejala 2 adalah seperti gambar 3.10. berikut.



Gambar 3.10. Ekspansi Tree 2

Perhitungan tiap kelas yang ada pada tiap *node* sebagai berikut.

1. *Node* RENDAH

$$C1 = 0.5$$

$$C0 = 0$$

Proporsi kelas 1

$$\frac{0.5}{0.5 + 0} \times 100\% = 100\%$$

Proporsi kelas 0

$$\frac{0}{0.5 + 0} \times 100\% = 0\%$$

2. *Node* SEDANG

$$C1 = 0.5 + 1 + 1 + 0.5 + 1 = 4$$

$$C0 = 0$$

Proporsi kelas 1

$$\frac{4}{4 + 0} \times 100\% = 100\%$$

Proporsi kelas 0

$$\frac{0}{4 + 0} \times 100\% = 0\%$$

3. *Node* TINGGI

$$C1 = 0.5$$

$$C0 = 0$$

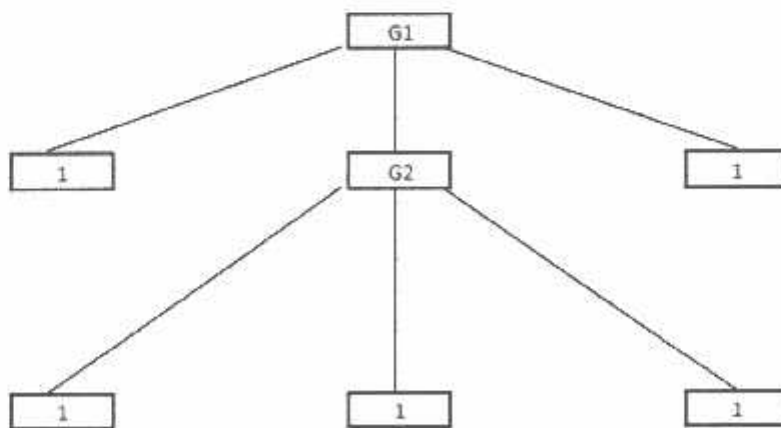
Proporsi kelas 1

$$\frac{0.5}{0.5 + 0} \times 100\% = 100\%$$

Proporsi kelas 0

$$\frac{0}{0.5 + 0} \times 100\% = 0\%$$

Fuzziness Control Threshold (FTC) yang digunakan adalah 75% sehingga *node* pada root gejala 2 tidak akan diekspansi lagi karena proporsi *node* lebih besar dari FTC. Maka hasil Tree yang dihasilkan adalah seperti gambar 3.11. berikut.



Gamabar 3.11. Ekspani Tree 3

Berdasarkan perhitungan pembentukan *fuzzy decision tree* dengan algoritma ID3 diperoleh sebuah model yang terdiri atas 5 aturan dengan menggunakan training set. Aturan klasifikasi yang diperoleh adalah sebagai berikut.

1. IF Gejala 1 rendah THEN terkena penyakit konjungtivitis.
2. IF Gejala 1 tinggi THEN terkena penyakit konjungtivitis.
3. IF Gejala 1 sedang AND Gejala 2 rendah THEN terkena penyakit konjungtivitis
4. IF Gejala 1 sedang AND Gejala 2 sedang THEN terkena penyakit konjungtivitis
5. IF Gejala 1 sedang AND Gejala 2 tinggi THEN terkena penyakit konjungtivitis

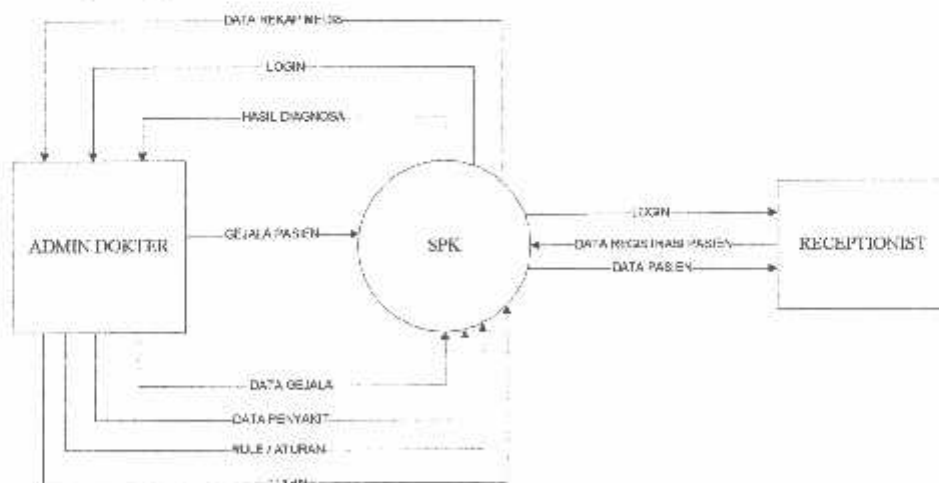
### 3.3.Data Flow Diagram

Data Flow Diagram (DFD) adalah suatu diagram yang menggunakan notasi-notasi untuk menggambarkan arus dari data sistem, yang penggunaannya sangat membantu untuk memahami sistem secara logika, tersruktur dan jelas.

DFD merupakan alat bantu dalam menggambarkan atau menjelaskan sistem yang sedang berjalan logis.

#### 3.3.1. Context Diagram

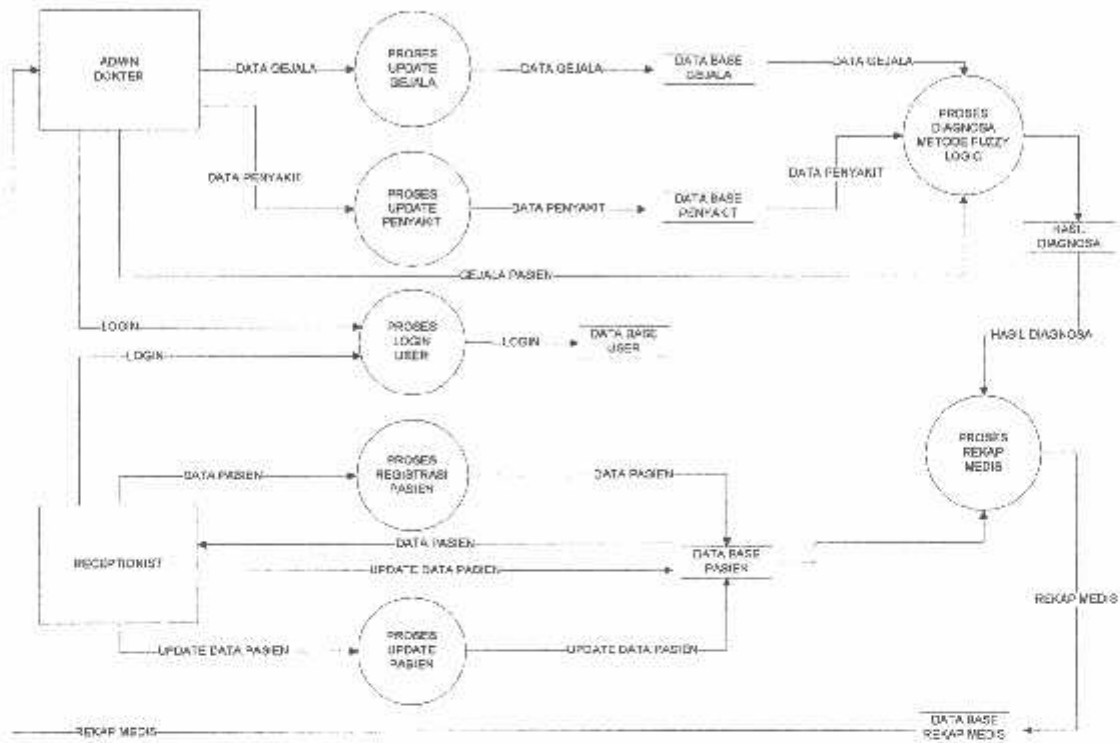
Context Diagram adalah suatu diagram yang digunakan untuk mendesain sistem yang memberikan gambaran detail mengenai semua informasi yang diterima ataupun dihasilkan dari suatu aktivitas. Diagram ini menggambarkan sebuah sistem/aktivitas pada bagian tengah tanpa informasi internal tentang sistem/aktivitas tersebut. Diagram dapat dilihat pada gambar 3.12. berikut.



Gambar 3.12. Context Diagram

### 3.3.2. DFD LV 0

Data Flow Diagram (DFD) Level 0 merupakan penjabaran proses pada diagram konteks (Context Diagram) yang memuat proses-proses yang ada dalam sistem secara garis besar dan keseluruhan. Diagram arus data level 0 juga mencantumkan kesatuan luar yang berhubungan dengan sistem. Diagram dapat dilihat pada gambar 3.13. berikut.

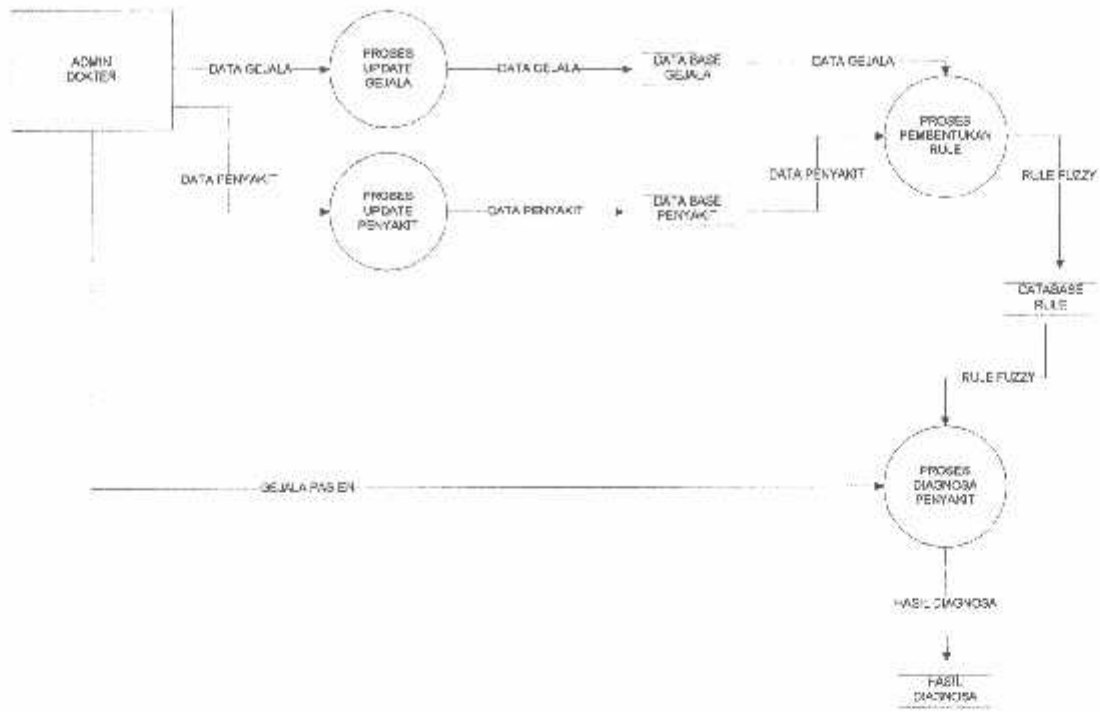


Gambar 3.13. DFD Level 0

### 3.3.3. DFD LV 1

Data Flow Diagram (DFD) Level 1 merupakan penjabaran proses pada Data Flow Diagram (DFD) Level 0. Di man proses yang dijabarkan adalah proses diagnosa penyakit. Diagram dapat dilihat pada gambar 3.14. berikut.





Gambar 3.14. DFD Level 1

## BAB IV IMPLEMENTASI DAN PEMBAHASAN

### 4.1 Lingkungan Implementasi

Lingkungan implementasi yang akan dijelaskan pada sub bab ini adalah lingkungan perangkat keras dan lingkungan perangkat lunak yang digunakan dalam *sistem* pendukung keputusan diagnosa penyakit mata menggunakan metode *fuzzy logic*.

#### 4.1.1. Perangkat Keras

Perangkat keras yang digunakan dalam *sistem* pendukung keputusan diagnosa penyakit mata menggunakan metode *fuzzy logic* adalah :

1. Prosesor AMD Turion™ II X2 M500 (2,2 GHz, 1MB L2 Cache)
2. Memori 4 GB RAM
3. Harddisk dengan kapasitas 250 GB
4. Monitor 14,0" HD LED LCD
5. Keyboard
6. Mouse

#### 4.1.2. Perangkat Lunak

Perangkat lunak yang digunakan dalam *sistem* pendukung keputusan diagnosa penyakit mata menggunakan metode *fuzzy logic* ini adalah :

1. Sistem Operasi *Windows XP SP 3*
2. *Microsoft Visual Studio 2008*
3. *Microsoft SQL Server 2005*

### 4.2. Implementasi Program

Pada sub bab ini akan dibahas mengenai implementasi dari *sistem* pendukung keputusan diagnosa penyakit mata menggunakan metode *fuzzy logic*.

#### 4.2.1. Penyimpanan Rule

Dalam mendiagnosa diperlukan dua atribut yaitu gejala dan penyakit. Dan dalam *sistem* pendukung keputusan diagnosa penyakit mata menggunakan metode *fuzzy logic* ini setiap penyakit mempunyai beberapa gejala yang mempunyai bobot masing – masing. Sehingga setiap gejala mempunyai bobotnya sendiri pada tiap gejala. Apabila

gejala tersebut termasuk ringan pada suatu penyakit maka gejala tersebut memiliki nilai *fuzzy* 0.25, apabila gejala tersebut termasuk ringan pada suatu penyakit maka gejala tersebut memiliki nilai 0.5, dan apabila gejala tersebut termasuk ringan pada suatu penyakit maka gejala tersebut memiliki nilai 1.

Prosedurnya dapat dilihat pada gambar 4.1

```

Private Sub btnSimpan_Click(ByVal sender As Sistem.Object, ByVal
e As Sistem.EventArgs) Handles btnSimpan.Click
    If (A.Text.Trim() = "") And (btnSimpan.Text <> "Edit") Then
        DevComponents.DotNetBar.MessageBoxEx.Show("Cek
Inputan, Inputan Tidak Boleh Kosong !", "Kesalahan",
MessageBoxButtons.OK, MessageBoxIcon.Error)
    End Sub
End If
Dim status As Double
If (RadioButton1.Checked) Then
    status = 0.25
ElseIf (RadioButton2.Checked) Then
    status = 0.5
Else
    status = 1
End If
If (btnSimpan.Text = "Simpan") Then
    Try
        Me.Tb_ruleTableAdapter.InsertQuery(A.SelectedValue,
Convert.ToInt16(ComboBoxEx1.SelectedValue), status)

        DevComponents.DotNetBar.MessageBoxEx.Show("Proses
Penyimpanan Berhasil !", "Informasi", MessageBoxButtons.OK,
MessageBoxIcon.Information)
    Catch ex As Exception
        DevComponents.DotNetBar.MessageBoxEx.Show("Terjadi
Kesalahanm, Proses Penyimpanan Gagal !", "Kesalahan",
MessageBoxButtons.OK, MessageBoxIcon.Error)
    End Try
End If

```

```

End Try
btnRefresh_Click(sender, e)
ElseIf (btnSimpan.Text = "Edit") Then
    Panellsi.Enabled = True
    btnSimpan.Text = "Update"
    A.SelectedValue = Kode_penyakitTextBox.Text
Else
    Try

        Me.Tb_ruleTableAdapter.InsertQuery(A.SelectedValue,
Convert.ToInt16(ComboBoxEx1.SelectedValue), status)

        DevComponents.DotNetBar.MessageBoxEx.Show("Proses
Update Berhasil !", "Informasi", MessageBoxButtons.OK,
MessageBoxIcon.Information)

        Catch ex As Exception
            DevComponents.DotNetBar.MessageBoxEx.Show("Terjadi
Kesalahanm, Proses Update Gagal !", "Kesalahan", MessageBoxButtons.OK,
MessageBoxIcon.Error)

        End Try

        btnRefresh_Click(sender, e)

    End If
End Sub

```

Gambar 4.1 *Source code* penyimpanan rule *fuzzy*.

#### 4.2.2. Proses Pemanggilan Rule Dalam Pendiagnosaan

Dalam proses pendiagnosaan rule yang tadi telah di simpan dipanggil untuk mencocokkan nilai data input gejala – gejala yang dialami pasien dengan data rule yang telah disimpan.

Prosedurnya dapat dilihat pada gambar 4.2

```

Sub getRule()
    rule = New Dictionary(Of String, Dictionary(Of String,
Double))

```

```

lsHasil.Items.Clear()
For i = 0 To lsGejala.CheckedItems.Count - 1

    Dim tmpCode As String =
lsGejala.CheckedItems.Item(i).ToString.Split("|")(0)
    Dim tmpStr As String =
lsGejala.CheckedItems.Item(i).ToString.Split("|")(1)
    lsHasil.Items.Add(tmpStr)
    Dim dt As DataTable
    Dim dtS As New
dataDataSetTableAdapters.tb_ruleTableAdapter()
    dt = dtS.GetDataByGejala(Convert.ToInt16(tmpCode))

    If dt.Rows.Count > 0 Then
        For j = 0 To dt.Rows.Count - 1
            If (rule.ContainsKey(dt.Rows(j).Item("kode_penyakit")))
Then
rule(dt.Rows(j).Item("kode_penyakit")).Add(tmpCode,
dt.Rows(j).Item("nilai"))
            Else
                Dim tmpCodeArr As Dictionary(Of String, Double) =
New Dictionary(Of String, Double)
                tmpCodeArr.Add(tmpCode, dt.Rows(j).Item("nilai"))
                rule.Add(dt.Rows(j).Item("kode_penyakit"),
tmpCodeArr)
            End If
        Next j
    End If
Next i
End Sub

```

Gambar 4.2 Prosedur Pengambilan Rule

#### 4.2.3. Proses Pencocokan Hasil Bobot Diagnosa Dengan Bobot Rule

Gejala – gejala yang dialami pasien diproses sehingga mendapatkan suatu bobot yang kemudian di bandingkan dengan bobot – bobot penyakit yang telah di simpan dalam rule.

Prosedurnya dapat dilihat pada gambar 4.3

```

Private Function getHasil(ByVal SourceDictionary As
Dictionary(Of String, Double)) As Dictionary(Of String, Double)

    Dim inDict As Dictionary(Of String, Double) =
SourceDictionary
    Dim outDict As New Dictionary(Of String, Double)
    Dim currentMaxKey As String = String.Empty
    Dim currentMaxValue As Integer = 0

    Do While inDict.Keys.Count > 0
        For Each currentKVP As KeyValuePair(Of String,
Double) In inDict
            If currentKVP.Value >= currentMaxValue Then
                currentMaxKey = currentKVP.Key
                currentMaxValue = currentKVP.Value
            End If
        Next

        outDict.Add(currentMaxKey, currentMaxValue)

        inDict.Remove(currentMaxKey)
        currentMaxValue = 0
    Loop

    Return outDict

```

```
End Function
```

Gambar 4.3 Prosedur Pencocokan Bobot

#### 4.2.4. Procs Defuzzifikasi

Setelah proses pencocokan bobot, Apabila ada bobot yang cocok maka bobot tersebut defuzzifikasi untuk mengeluarkan nama penyakit yang cocok dengan gejala pasien. Namun apabila tidak ada bobot yang cocok maka penyakit tersebut belum dikenali oleh rule sehingga tidak ada hasil defuzzifikasi yang muncul.

Prosedurnya dapat dilihat pada gambar 4.4

```
Sub defuzzifikasi()
    Dim result As Dictionary(Of String, Double) = New
Dictionary(Of String, Double)

    For Each rowPenyakit As KeyValuePair(Of String,
Dictionary(Of String, Double)) In rule
        Console.WriteLine("Key = {0}, Value = {1}",
rowPenyakit.Key, rowPenyakit.Value)
        Dim fuzzValue As Double = 0
        For Each rowGejala As KeyValuePair(Of String, Double)
In rowPenyakit.Value
            Console.WriteLine("-->> Key = {0}, Value = {1}",
rowGejala.Key, rowGejala.Value)
            fuzzValue = fuzzValue + rowGejala.Value
        Next rowGejala
        result.Add(rowPenyakit.Key, fuzzValue)
    Next rowPenyakit
    Dim Newresult As Dictionary(Of String, Double) =
getHasil(result)
    For Each rowS As KeyValuePair(Of String, Double) In
Newresult
        Console.WriteLine(">>> Key = {0}, Value = {1}",
rowS.Key, rowS.Value)
```

```

Next rowS
Dim lsHasil As List(Of String) = Newresult.Keys.ToList
Me.Tb_pasienTableAdapter.Fill(Me.DataDataSet.tb_pasien)

Dim dt As DataTable
Dim dtS As New
dataDataSetTableAdapters.tb_penyakitTableAdapter()
dt = dtS.GetDataByKode(lsHasil(0))

If dt.Rows.Count > 0 Then
    For i = 0 To dt.Rows.Count - 1
        B.Text = dt.Rows(i).Item("nama_penyakit")

    Next i
Else
    B.Text = "PENYAKIT TIDAK DIKENALI"
End If

End Sub

```

Gambar 4.4. Prosedur Deffuzifikasi

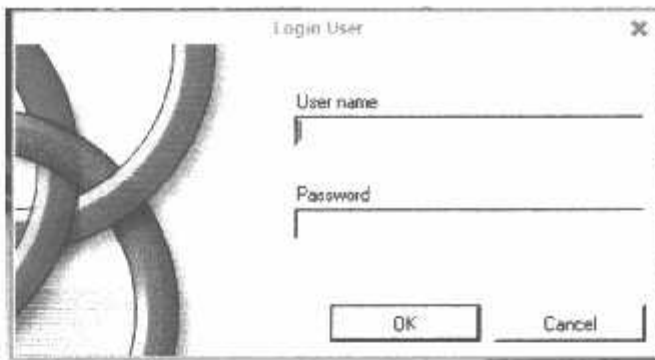
### 4.3. Implementasi Program

Dalam *sistem* pendukung keputusan diagnosa penyakit mata menggunakan metode *fuzzy logic*, terdapat beberapa *Form* yang terdiri dari :

1. Tampilan *Form* Admin

Pada halaman *login*, akan diminta nama *user* dan *password* untuk mengakses aplikasi. *Form User* dapat dilihat Dalam Gambar 4.5





Gambar 4.5. Tampilan *Form Admin*

## 2. Tampilan *Form Utama*

Menampilkan menu-menu yang dapat dipilih oleh *user* untuk dapat menjalankan program. Terdapat pada gambar 4.6.



Gambar 4.6. Tampilan *Form Utama*

## 3. Form Data Pasien

*Form* ini berfungsi untuk memasukkan atau mengubah data pasien yang datang untuk berobat. Terdapat pada Gambar 4.7

No	Nama	Alamat	Jk	Usia	No. Telp	Tanggal Masuk
1	Peter	Kabang	L	20	123	19/01/2022
2	Peter	Kabang	L	20	456	19/02/2022

Gambar 4.7. Tampilan Form Data Pasien

#### 4. Form Data Pegawai

Form data pegawai ini berfungsi untuk mendaftarkan admin (dokter) dan operator yang digunakan untuk login ke sistem. *Form* ini dapat dilihat pada Gambar 4.8.

No	Nama	Status
1	Admin	Admin
2	Operator	Operator
3	Tika	Admin

Gambar 4.8. Tampilan *Form* Data Pegawai.

#### 5. Form Data Penyakit

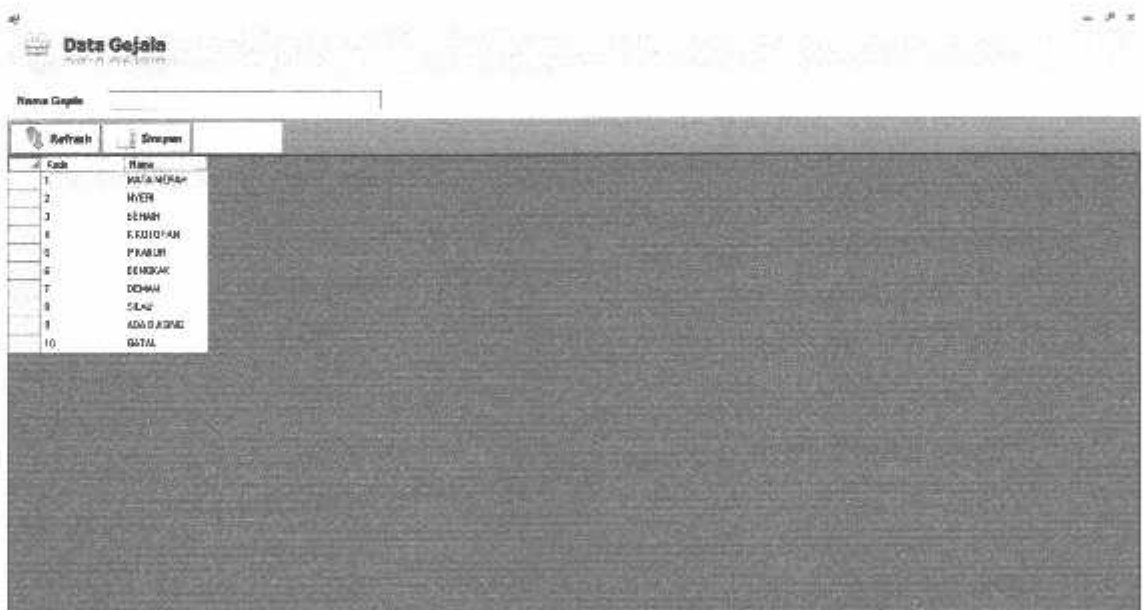
Form Data Penyakit ini digunakan untuk menginputkan jenis – jenis penyakit mata yang nantinya akan digunakan sebagai hasil rule. *Form* ini dapat dilihat pada Gambar 4.9.



Gambar 4.9. Tampilan *Form* Data Penyakit

#### 6. *Form* Data Gejala

*Form* Data Gejala digunakan untuk memasukkan gejala – gejala umum yang timbul pada pasien, data tersebut nantinya akan di berikan nilai pada setiap penyakit yang memiliki gejala yang cocok . Tampilan dari *Form* Pelatihan ini dapat dilihat pada Gambar 4.10.



Gambar 4.10. Tampilan *Form* Data Gejala

## 7. Form Data Aturan

Pada *Form Data Aturan* adalah untuk membuat rule dengan merelasikan penyakit dengan gejala – gejalanya dan tiap gejala memiliki suatu nilai tertentu yang nantinya akan diubah menjadi bobot *fuzzy* yang akan diproses dalam proses pendiagnosaan pasien nantinya. Tampilan dari *Form Data Aturan* ini dapat dilihat pada Gambar 4.11.

Gejala	Bobot	Status
UJER PANG	NYERI	0.25
UJER PANG	TAKUR	0.5
UJER PANG	SULAM	0.75
UJER PANG	MATA MERAH	0.25
UJER PANG	NYERI	0.5
UJER PANG	BERAK	0.25
UJER PANG	KOTORAN	0.75
UJER PANG	TAKUR	0.25
UJER PANG	DASIL	0.25
TRAKOM	MATA MERAH	0.5
TRAKOM	KOTORAN	0.25
TRAKOM	BENCAC	0.25
OPHTALMIA	MATA MERAH	0.5
OPHTALMIA	NYERI	0.75
OPHTALMIA	TAKUR	0.25
OPHTALMIA	DENGAM	0.25
OPHTALMIA	TAKUR	0.5
OPHTALMIA	SULAM	0.75
KUNJINGTUMBUH	MATA MERAH	0.75
KUNJINGTUMBUH	NYERI	0.75
KUNJINGTUMBUH	BERAK	0.5
KUNJINGTUMBUH	KOTORAN	0.25
KUNJINGTUMBUH	TAKUR	0.75

Gambar 4.11. Tampilan *Form Data Aturan*

## 8. Form Proses Diagnosa

*Form Proses Diagnosa* digunakan oleh dokter untuk membantu proses pendiagnosaan pasien dengan cara memilih gejala apa saja yang dialami oleh pasien. Form ini dapat dilihat pada gambar 4.12.

**Proses Diagnosa**

Name Pasien:  Baru

Kode	Nama	Alamat	JK	Usia	Tinggi
1	Pasien 1	Malang	L	20	123
2	Pasien 2	Malang	P	20	156

Kode:

Nama:

Alamat:

File Gejala

- MATA MERAH
- MATA BERAIR
- MATA KOTORAN
- MATA RABUH
- MATA BENGKAK
- MATA DEMAM
- MATA BERAIR
- MATA BERSING
- MATA BERGATAL

Gambar 4.12. Tampilan Form Proses Diagnosa

Setelah dokter menginputkan gejala, selanjutnya sistem akan memproses dan akan menampilkan nama penyakit. Penyakit yang direkomendasikan tidak bersifat mutlak, apabila dokter merasa tidak cocok dengan penyakit tersebut dokter dapat mengganti dengan penyakit lain. Dalam form hasil diagnosa ini juga disediakan kolom keterangan sebagai catatan – catatan dokter tentang si pasien. Form ini dapat dilihat pada gambar 4.13.

**Proses Diagnosa**

Name Pasien:  Baru

Kode	Nama	Alamat	JK	Usia	Tinggi
1	Pasien 1	Malang	L	20	123
2	Pasien 2	Malang	P	20	156

Kode:

Nama:

Alamat:

Hasil Diagnosa

MATA MERAH  
MATA BERAIR  
MATA KOTORAN  
MATA BENGKAK  
MATA DEMAM  
MATA BERAIR  
MATA BERSING  
MATA BERGATAL

Keterangan:

Gambar 4.13. Tampilan Form Hasil Diagnosa

## 9. Form Rekap Medik

Form Rekap Medik digunakan untuk memantau keadaan pasien dari terakhir pasien tersebut berobat. Dengan demikian dokter dapat mengetahui perkembangan kesehatan pasien serta mengetahui keadaan terakhir pasien ataupun obat yang dokter berikan sebelumnya. Dalam form ini juga disediakan kolom keterangan untuk memasukkan keadaan pasien terbaru oleh dokter. Form ini dapat dilihat pada gambar 4.14.

Nama Pasien						Riwayat Rekap Medik		
Kode	Nama	Alamat	JK	Usia	Telp	Tanggal	Penyakit	Keterangan
1	Pasien 1	Malang	L	20	123	11/08/2017	U KORNEA	KOROSI LEMAH, ABC ZMG, 11 LL, PL, JK SHARI STII MKM
2	Pasien 2	Malang	P	20	456	11/08/2017	U KORNEA	MENSAK, ABC 1 NG, PL, JK SHARI
						11/08/2017	K SIKSA	PAPAH, PL, JK ZMG, JK BEHAR, PUYER SSB, JK BEHAR S

Penyakit: K SIKSA  
Kategori:

Kode:   
Nama:   
Alamat:

Gambar 4.14. Tampilan Form Rekap Medik

## 10. Tabel Hasil Tes Sistem

Sistem dites apakah dapat memprediksi sesuai dengan data penyakit yang ada pada literatur. Tabel dapat di lihat pada table 4.1.

Tabel 4.1. Tabel Kebenaran

NO	DESIJA	PENYAKIT LITERATUR	HASIL
1	MATA MERAH, NYERI, BERAIR, GATAL, KELUAR KOTORAN, PENGLIHATAN	KONJUNKTIVITIS	BENAR
2	MATA MERAH, BERAIR, GATAL, KELUAR KOTORAN, BENGEK, SILAU	KERATOKONJUNKTIVITIS VERNALIS (KV)	BENAR
3	MATA MERAH, NYERI, PENGLIHATAN KABUR	ENDOFTALMITIS	SALAH
4	NYERI, PENGLIHATAN KABUR, BENGEK, DEMAM	SELUKITIS ORBITALIS (SO)	BENAR
5	MATA MERAH, KELUAR KOTORAN, BENGEK	TRAKOMA	SALAH
6	MATA MERAH, GATAL, BENGEK, TERASA ADA BENDA ASING	BLEFARITIS	BENAR
7	MATA MERAH, NYERI, BENGEK, DEMAM	DAKRIOSITIS	SALAH
8	MATA MERAH, NYERI, GATAL, KELUAR KOTORAN	ULUS KORNEA	BENAR
9	PENGLIHATAN KABUR	KERATITIS NEUROPARALITIK	SALAH
10	NYERI, PENGLIHATAN KABUR, DEMAM, TERASA ADA BENDA ASING	KERATOKONJUNKTIVITIS EPIDEMIA	BENAR
11	GATAL, PENGLIHATAN KABUR, SILAU, TERASA ADA BENDA ASING	KERATOKONJUNKTIVITIS SIKSA	BENAR
12	NYERI, PENGLIHATAN KABUR, SILAU	ULUS SERIKENS	BENAR

Dari tabel diatas maka sistem dapat memprediksi penyakit sebesar 66,7%

## BAB V

### PENUTUP

#### 1.1. Kesimpulan

Setelah dilakukan perancangan dan pembangunan *sistem* pendukung keputusan diagnosa penyakit mata menggunakan metode *fuzzy logic*, dilakukan serangkaian uji coba dan didapatkan hasil rekomendasi penyakit mata berdasar gejala – gejala yang dialami pasien berdasar rule yang dibuat. Setelah dilakukan analisa dan evaluasi terhadap hasil uji coba maka didapatkan beberapa kesimpulan sebagai berikut :

1. Sistem mampu untuk memberikan rekomendasi penyakit kepada dokter berdasarkan gejala – gejala yang dialami oleh pasien sesuai rule *fuzzy* yang sebelumnya telah ditentukan dengan nilai error 30%..
2. Sistem mampu untuk menerapkan metode *fuzzy logic* pada proses pendiagnosaan penyakit mata oleh dokter.
3. Hasil yang diberikan memiliki nilai kebrnaran 70% maka hasil diagnosa bersifat rekomendasi sehingga dokter dapat mengganti hasil diagnosa sesuai dengan pendapat dokter.

#### 5.2. Saran

Perangkat lunak yang dibuat pada tugas akhir ini masih dapat dikembangkan lebih lanjut dengan bentuk sebagai berikut :

1. Penambahan database gejala dan penyakit demi penyempurnaan sistem.
  2. Dilakukan penelitian tentang penggunaan metode lain yang dapat mempercepat proses diagnosa.
  3. Dapat dilakukan penelitian untuk mencoba metode *fuzzy logic* ke jenis penyakit lain.
  4. Dapat ditambahkan ke *sistem* informasi di rumah sakit atau praktek dokter bersma untuk kemudahan penyimpanan rekap medis pasien dan proses pendiagnosaan pasien penyaakit mata.
-

**DAFTAR PUSTAKA**

1. Fakultas Kedokteran Universitas Indonesia, 2008, Sari Ilmu Penyakit Mata, Jakarta, Gaya Baru.
  2. Prof. dr. H. Sidarta Iwas, SpM , 2009, Ikhtisar Ilmu Penyakit Mata, Jakarta, Balai Penerbit FKUI.
  3. Jan Jantzen, 1998, Tutorial On *Fuzzy logic*, Denmark, Technical University of Denmark, Department of Automation.
  4. Khaled Hammouda and Prof. Fakhreddine Karray, 1996, A Comparative Study of Data Clustering Techniques, Canada , Department of Systems Design Engineering University of Waterloo.
  5. Sri Kusumadewi, 2002, Analisis dan Desain Sistem *Fuzzy* Menggunakan Toolbox Matlab, Yogyakarta, Graha Ilmu.
  6. Ir. Yuniar Supriadi, 2011, Semua Bisa Menjadi Programmer VB6 Hingga VB2008 BASIC, Jakarta, PT. Elex Media Komputindo.
  7. Daihani, Dadan Umar. 2001. *Komputerisasi Pengambilan Keputusan*. Jakarta: PT Elex Media Komputindo.
  8. Kusumadewi, Sri dan Purnomo, Hari. 2004. *Aplikasi Logika Fuzzy untuk Pendukung Keputusan*. Edisi Pertama. Yogyakarta: Graha Ilmu.
  9. [www.medicastore.com](http://www.medicastore.com), tanggal akses 5 Mei 2011.
  10. [http://id.wikipedia.org/wiki/Sistim\\_pakar](http://id.wikipedia.org/wiki/Sistim_pakar), tanggal akses 5 Mei 2011.
  11. [http://id.wikipedia.org/wiki/Visual\\_Basic\\_.NET](http://id.wikipedia.org/wiki/Visual_Basic_.NET), tanggal akses 5 Mei 2011.
  12. [http://id.wikipedia.org/wiki/Microsoft\\_SQL\\_Server](http://id.wikipedia.org/wiki/Microsoft_SQL_Server), tanggal akses 5 Mei 2011.
  13. <http://id.wikipedia.org/wiki/SQL>, tanggal akses 5 Mei 2011.
  14. [http://id.wikipedia.org/wiki/Microsoft\\_Visual\\_Studio](http://id.wikipedia.org/wiki/Microsoft_Visual_Studio), tanggal akses 5 Mei 2011.
  15. [www. Mathworks.com](http://www.Mathworks.com), 2004, Fuzzy logic Toolbox User's Guide Version 2, The MathWork Inc.
-





**BERITA ACARA UJIAN SKRIPSI  
FAKULTAS TEKNOLOGI INDUSTRI**

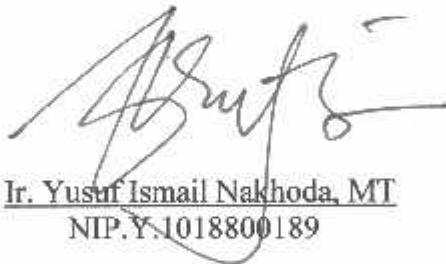
NAMA : TIRTO WICAKSONO  
NIM : 07.12.521  
PROGRAM STUDI : Teknik Elektro S-1  
KONSENTRASI : Teknik Komputer dan Informatika  
MASA BIMBINGAN : Semester Genap Tahun Akademik 2011 - 2012  
JUDUL : PERANCANGAN DAN PEMBUATAN SISTEM  
PENDUKUNG KEPUTUSAN DIAGNOSA  
PENYAKIT MATA MENGGUNAKAN METODE  
FUZZY LOGIC

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

Hari : Selasa  
Tanggal : 31 Juli 2012  
Dengan Nilai : 78,05 (B+) *cc*

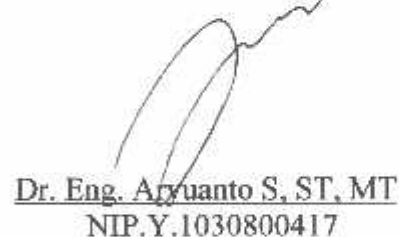
**PANITIA UJIAN SKRIPSI**

Ketua Majelis Penguji,



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

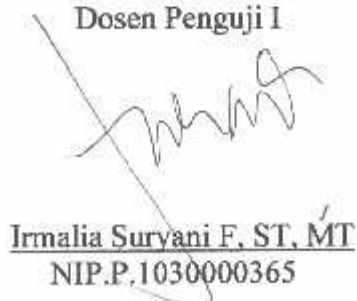
Sekretaris Majelis Penguji,



Dr. Eng. Agyuanto S, ST, MT  
NIP.Y.1030800417

**ANGGOTA PENGUJI**

Dosen Penguji I



Irmalia Suryani F, ST, MT  
NIP.P.1030000365

Dosen Penguji II



Bima Aulia, ST



## FORMULIR PERBAIKAN UJIAN SKRIPSI

Dalam pelaksanaan ujian skripsi jenjang Strata 1 Program Studi Teknik Elektro Konsentrasi Teknik Komputer dan Informatika, maka perlu adanya perbaikan skripsi untuk mahasiswa :

NAMA : TIRTO WICAKSONO  
NIM : 07.12.521  
PROGRAM STUDI : Teknik Komputer dan Informatika S-1  
JUDUL : PERANCANGAN DAN PEMBUATAN SISTEM  
PENDUKUNG KEPUTUSAN DIAGNOSA PENYAKIT  
MATA MENGGUNAKAN METODE FUZZY LOGIC

No	Penguji	Tanggal	Uraian	Paraf
1.	Penguji I	31 Juli 2012	1. BAB IV Pengujian Aplikasi 2. Perbaikan Fungsi Login	
2.	Penguji II	31 Juli 2012	1. Perbaikan Login 2. Pengujian Terhadap Pemakai	

Disetujui :

Dosen Penguji I

Irmalia Suryani Faradisa, ST, MT  
NIP.P. 1030000365

Dosen Penguji II

Bima Aulia, ST

Mengetahui :

Dosen Pembimbing I

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

Dosen Pembimbing II

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

## SCRIPT FORM GEJALA

```
Public Class FormGejala
    Inherits DevComponents.DotNetBar.Metro.MetroForm
    Private Sub FormGejala_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        btnRefresh_Click(sender, e)

    End Sub

    Private Sub btnRefresh_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnRefresh.Click
        btnSimpan.Enabled = True
        btnSimpan.Text = "Simpan"
        btnHapus.Enabled = False
        'TODO: This line of code loads data into the
'DataDataSet.tb_gejala' table. You can move, or remove it, as needed.
        Me.tb_gejalaTableAdapter.Fill(Me.DataDataSet.tb_gejala)

        A.Clear()
        PanelIsi.Enabled = True
    End Sub

    Private Sub btnSimpan_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnSimpan.Click
        If (A.Text.Trim() = "") And (btnSimpan.Text <> "Edit") Then
            DevComponents.DotNetBar.MessageBoxEx.Show("Cek inputan, Inputan
Tidak Boleh Kosong!", "Kesalahan", MessageBoxButtons.OK,
MessageBoxIcon.Error)
            Exit Sub
        End If
        If (btnSimpan.Text = "Simpan") Then
            Try
                Me.tb_gejalaTableAdapter.InsertQuery(A.Text.Trim())
                DevComponents.DotNetBar.MessageBoxEx.Show("Proses
Penyimpanan Berhasil!", "Informasi", MessageBoxButtons.OK,
MessageBoxIcon.Information)
            Catch ex As Exception
                DevComponents.DotNetBar.MessageBoxEx.Show("Terjadi
Kesalahanm, Proses Penyimpanan Gagal!", "Kesalahan", MessageBoxButtons.OK,
MessageBoxIcon.Error)
            End Try
            btnRefresh_Click(sender, e)
        ElseIf (btnSimpan.Text = "Edit") Then
            PanelIsi.Enabled = True
            btnSimpan.Text = "Update"
            A.Text = Nama_gejalaTextBox.Text
        Else
            Try
                Me.tb_gejalaTableAdapter.UpdateQuery(A.Text.Trim(),
Kode_gejalaTextBox.Text, Kode_gejalaTextBox.Text)
                DevComponents.DotNetBar.MessageBoxEx.Show("Proses Update
Berhasil!", "Informasi", MessageBoxButtons.OK, MessageBoxIcon.Information)
            Catch ex As Exception
                DevComponents.DotNetBar.MessageBoxEx.Show("Terjadi
Kesalahanm, Proses Update Gagal!", "Kesalahan", MessageBoxButtons.OK,
MessageBoxIcon.Error)
            End Try
        End Try
    End Sub
End Class
```

---

```

        btnRefresh_Click(sender, e)
    End If

End Sub

Private Sub dtGrid_CellClick(ByVal sender As System.Object, ByVal e As
System.Windows.Forms.DataGridViewCellEventArgs) Handles
dtGrid.CellContentClick
    PanelIsi.Enabled = False
    btnHapus.Enabled = True
    btnSimpan.Text = "Edit"
    A.Clear()
    PanelEx1.Refresh()
End Sub

Private Sub btnHapus_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnHapus.Click
    If (DevComponents.DotNetBar.MessageBoxEx.Show("Yakin Untuk
Menghapus Data?", "Konfirmasi.", MessageBoxButtons.YesNo,
MessageBoxIcon.Question) = Windows.Forms.DialogResult.Yes) Then
        Try
            Me.Tb_gejalaTableAdapter.DeleteQuery(Kode_gejalaTextBox.Text)
            DevComponents.DotNetBar.MessageBoxEx.Show("Proses
Penghapusan Data Berhasil !", "Informasi", MessageBoxButtons.OK,
MessageBoxIcon.Information)
        Catch ex As Exception
            DevComponents.DotNetBar.MessageBoxEx.Show("Terjadi
Kesalahan, Proses Penghapusan Data Gagal !", "Kesalahan",
MessageBoxButtons.OK, MessageBoxIcon.Error)
        End Try
        btnRefresh_Click(sender, e)
    End If
End Sub
End Class

```

---

## SCRIPT FORM HASIL

```
Public Class FormHasil
    Inherits DevComponents.DotNetBar.Metro.MetroForm

    Private Sub FormHasil_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        Me.tb_pasienTableAdapter.Fill(Me.DataDataSet.tb_pasien)
    End Sub

    Private Sub A_TextChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles A.TextChanged
        If Trim(A.Text) = "" Then Exit Sub
        Me.tb_pasienTableAdapter.FillByKode(Me.DataDataSet.tb_pasien,
A.Text)
        DataGridViewX1.Refresh()
    End Sub

    Private Sub Kode_pasienTextBox1_TextChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Kode_pasienTextBox1.TextChanged
        If Trim(Kode_pasienTextBox1.Text) = "" Then Exit Sub
        Me.tb_hasilTableAdapter1.Fill(Me.DataDataSet.tb_hasil,
Kode_pasienTextBox1.Text)
        DataGridViewX2.Refresh()
    End Sub

    Private Sub ButtonX3_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles ButtonX3.Click
        If (B.Text.Trim <> "") Then
            Me.tb_rekapmedikTableAdapter1.InsertQuery(Convert.ToInt16(Kode_pasienTextBo
x1.Text.Trim), Now, FormLogin.UsernameTextBox.Text, C.Text.Trim,
B.Text.Trim)
            End If
            C.Clear()
            Kode_pasienTextBox1_TextChanged(sender, e)
            DevComponents.DotNetBar.MessageBoxEx.Show("Proses Penyimpanan
Berhasil !", "Informasi", MessageBoxButtons.OK, MessageBoxIcon.Information)

        End Sub

    Private Sub DataGridViewX2_CellContentClick(ByVal sender As
System.Object, ByVal e As System.Windows.Forms.DataGridViewCellEventArgs)
Handles DataGridViewX2.CellContentClick
        btnHapus.Enabled = True
    End Sub

    Private Sub DataGridViewX1_CellContentClick(ByVal sender As
System.Object, ByVal e As System.Windows.Forms.DataGridViewCellEventArgs)
Handles DataGridViewX1.CellContentClick
        btnHapus.Enabled = False
    End Sub

    Private Sub btnHapus_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnHapus.Click
        If (DevComponents.DotNetBar.MessageBoxEx.Show("Yakin Untuk
Menghapus Data?", "Konfirmasi.", MessageBoxButtons.YesNo,
MessageBoxIcon.Question) = Windows.Forms.DialogResult.Yes) Then
```

---

```
        Try
            Me.Tb_hasilTableAdapter1.DeleteQuery(Label1.Text)
            DevComponents.DotNetBar.MessageBoxEx.Show("Proses
Pengahapusan Data Berhasil !", "Informasi", MessageBoxButtons.OK,
MessageBoxIcon.Information)
        Catch ex As Exception
            DevComponents.DotNetBar.MessageBoxEx.Show("Terjad
Kesalahan, Proses Pengahapusan Data Gagal !", "Kesalahan",
MessageBoxButtons.OK, MessageBoxIcon.Error)
        End Try
        btnHapus.Enabled = False
        Me.Tb_pasienTableAdapter.Fill(Me.DataDataSet.tb_pasien)
        DataGridViewX2.Refresh()
    End If
End Sub
End Class
```

---

## SCRIPT FORM LOGIN

```
Public Class FormLogin

    Inherits DevComponents.DotNetBar.Metro.MetroForm

    Private Sub OK_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles OK.Click
        If (UsernameTextBox.Text.Trim = "" Or PasswordTextBox.Text.Trim =
"") Then
            Exit Sub
        End If

        If (Tb_pegawaiTableAdapter1.LoginUser(UsernameTextBox.Text.Trim,
PasswordTextBox.Text.Trim) > 0) Then
            Dim dt As DataTable
            Dim dtS As New
dataDataSetTableAdapters.tb_pegawaiTableAdapter()
            dt = dtS.GetDataByKode(UsernameTextBox.Text.Trim)

            If dt.Rows.Count > 0 Then
                FormMain.ButtonItem1.Enabled = False
                FormMain.ButtonItem2.Enabled = False
                FormMain.ButtonItem3.Enabled = False
                FormMain.ButtonItem4.Enabled = False
                FormMain.ButtonItem5.Enabled = False
                FormMain.ButtonItem6.Enabled = False
                FormMain.ButtonItem7.Enabled = False

                If dt.Rows(0).Item("status").ToString.Trim.Equals("Admin")
Then
                    FormMain.ButtonItem1.Enabled = False
                    FormMain.ButtonItem2.Enabled = True
                    FormMain.ButtonItem3.Enabled = True
                    FormMain.ButtonItem4.Enabled = True
                    FormMain.ButtonItem5.Enabled = True
                    FormMain.ButtonItem6.Enabled = True
                    FormMain.ButtonItem7.Enabled = True
                Else
                    FormMain.ButtonItem1.Enabled = True
                    FormMain.ButtonItem2.Enabled = False
                    FormMain.ButtonItem3.Enabled = False
                    FormMain.ButtonItem4.Enabled = False
                    FormMain.ButtonItem5.Enabled = False
                    FormMain.ButtonItem6.Enabled = False
                    FormMain.ButtonItem7.Enabled = False
                End If
                Hide()
                PasswordTextBox.Clear()
                FormMain.Show()
            End If
        End If
    End Sub
End Class
```

```
        Else
            DevComponents.DotNetBar.MessageBoxEx.Show("Terjadi Kesalahanm,
Proses Login Gagal !", "Kesalahan", MessageBoxButtons.OK,
MessageBoxIcon.Error)
        End If

    End Sub

    Private Sub Cancel_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Cancel.Click
        Me.Close()
    End Sub

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

    End Sub
End Class
```

---



## SCRIPT FORM MAIN

```
Public Class FormMain
    Inherits DevComponents.DotNetBar.Metro.MetroForm

    Private Sub FormMain_FormClosed(ByVal sender As Object, ByVal e As
System.Windows.Forms.FormClosedEventArgs) Handles Me.FormClosed

    End Sub

    Private Sub FormMain_FormClosing(ByVal sender As Object, ByVal e As
System.Windows.Forms.FormClosingEventArgs) Handles Me.FormClosing
        End
    End Sub
    Private Sub FormMain_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load

    End Sub

    Private Sub ButtonItem1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles ButtonItem1.Click
        FormPasien.Show()
    End Sub

    Private Sub ButtonItem2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles ButtonItem2.Click
        FormPegawai.Show()
    End Sub

    Private Sub ButtonItem3_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles ButtonItem3.Click
        FormPenyakit.Show()
    End Sub

    Private Sub ButtonItem4_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles ButtonItem4.Click
        FormGejala.Show()
    End Sub

    Private Sub ButtonItem5_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles ButtonItem5.Click
        FormRule.Show()
    End Sub

    Private Sub ButtonItem6_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles ButtonItem6.Click
        FormUji.Show()
    End Sub

    Private Sub ButtonItem7_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles ButtonItem7.Click
        FormHasil.Show()
    End Sub
End Class
```

---

## SCRIPT FORM PASIEN

```
Public Class FormPasien
    Inherits DevComponents.DotNetBar.Metro.MetroForm

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

    End Sub

    Private Sub btnRefresh_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnRefresh.Click
        btnSimpan.Enabled = True
        btnSimpan.Text = "Simpan"
        btnHapus.Enabled = False
        'TODO: This line of code loads data into the
'DataDataSet.tb_pasien' table. You can move, or remove it, as needed.
Me.Tb_pasienTableAdapter.Fill(Me.DataDataSet.tb_pasien)

        A.Clear()
        B.Clear()
        D.Text = 0
        E1.Value = 0
        PanelIsi.Enabled = True
    End Sub

    Private Sub btnSimpan_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnSimpan.Click
        If (A.Text.Trim() = "" Or B.Text.Trim() = "" Or E1.Text.Trim = "")
And (btnSimpan.Text <> "Edit") Then
            DevComponents.DotNetBar.MessageBoxEx.Show("Cek Inputan, Inputan
Tidak Boleh Kosong !", "Kesalahan", MessageBoxButtons.OK,
MessageBoxIcon.Error)
            Exit Sub
        End If
        Dim JK As String
        If (C1.Checked) Then
            JK = "L"

        Else
            JK = "P"

        End If
        If (btnSimpan.Text = "Simpan") Then
            Try

                Me.Tb_pasienTableAdapter.InsertQuery(A.Text.Trim(),
B.Text.Trim, JK, D.Value, E1.Text.Trim, Now)
                DevComponents.DotNetBar.MessageBoxEx.Show("Proses
Penyimpanan Berhasil !", "Informasi", MessageBoxButtons.OK,
MessageBoxIcon.Information)
            Catch ex As Exception
                DevComponents.DotNetBar.MessageBoxEx.Show("Terjadi
Kesalahan, Proses Penyimpanan Gagal !", "Kesalahan", MessageBoxButtons.OK,
MessageBoxIcon.Error)
            End Try
        End If
    End Sub
End Class
```

---

```

        End Try
        btnRefresh_Click(sender, e)
    ElseIf (btnSimpan.Text = "Edit") Then
        PanelIsi.Enabled = True
        btnSimpan.Text = "Update"
        A.Text = NamaTextBox.Text.Trim
        B.Text = AlamatTextBox.Text.Trim
        C1.Checked = False
        C2.Checked = False
        If (JkTextBox.Text.Trim = "L") Then
            C1.Checked = True
        Else
            C2.Checked = True
        End If
        D.Value = UsiaTextBox.Text.Trim
        E1.Text = TelpTextBox.Text.Trim

    Else
        Try
            Me.Tb_pasienTableAdapter.UpdateQuery(A.Text.Trim(),
            B.Text.Trim, JK, D.Value, E1.Text.Trim, Now, Kode_pasienTextBox.Text,
            Kode_pasienTextBox.Text)
            DevComponents.DotNetBar.MessageBoxEx.Show("Proses Update
            Berhasil !", "Informasi", MessageBoxButtons.OK, MessageBoxIcon.Information)
            Catch ex As Exception
                DevComponents.DotNetBar.MessageBoxEx.Show("Terjadi
                Kesalahanm, Proses Update Gagal !", "Kesalahan", MessageBoxButtons.OK,
                MessageBoxIcon.Error)
            End Try
            btnRefresh_Click(sender, e)
        End If
    End Sub

    Private Sub btnHapus_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles btnHapus.Click
        If (DevComponents.DotNetBar.MessageBoxEx.Show("Yakin Untuk
        Menghapus Data?", "Konfirmasi.", MessageBoxButtons.YesNo,
        MessageBoxIcon.Question) = Windows.Forms.DialogResult.Yes) Then
            Try

            Me.Tb_pasienTableAdapter.DeleteQuery(Kode_pasienTextBox.Text)
            DevComponents.DotNetBar.MessageBoxEx.Show("Proses
            Penghapusan Data Berhasil !", "Informasi", MessageBoxButtons.OK,
            MessageBoxIcon.Information)
            Catch ex As Exception
                DevComponents.DotNetBar.MessageBoxEx.Show("Terjadi
                Kesalahanm, Proses Penghapusan Data Gagal !", "Kesalahan",
                MessageBoxButtons.OK, MessageBoxIcon.Error)
            End Try
            btnRefresh_Click(sender, e)
        End If
    End Sub

    Private Sub dtGrid_CellContentClick(ByVal sender As System.Object,
    ByVal e As System.Windows.Forms.DataGridViewCellEventArgs) Handles
    dtGrid.CellContentClick
        PanelIsi.Enabled = False
        btnHapus.Enabled = True
        btnSimpan.Text = "Edit"
        A.Clear()
        B.Clear()

```

---

```
D.Value = 0
E1.Value = 0
PanelEx1.Refresh()
End Sub
End Class
```

---

## SCRIPT FORM PEGAWAI

```
Public Class FormPegawai
    Inherits DevComponents.DotNetBar.Metro.MetroForm

    Private Sub FormPegawai_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        'TODO: This line of code loads data into the
'DataDataSet.tb_pegawai' table. You can move, or remove it, as needed.

        C.Items.Add("Admin")
        C.Items.Add("Operator")
        C.SelectedIndex = 0
        btnRefresh_Click(sender, e)
    End Sub

    Private Sub btnRefresh_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnRefresh.Click
        btnSimpan.Enabled = True
        btnSimpan.Text = "Simpan"
        btnHapus.Enabled = False
        'TODO: This line of code loads data into the
'DataDataSet.tb_pasien' table. You can move, or remove it, as needed.
        Me.Tb_pegawaiTableAdapter.Fill(Me.DataDataSet.tb_pegawai)

        A.Clear()
        B.Clear()

        PanelIsi.Enabled = True
    End Sub

    Private Sub btnSimpan_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnSimpan.Click
        If (A.Text.Trim() = "" Or B.Text.Trim() = "" Or C.Text.Trim = "")
And (btnSimpan.Text <> "Edit") Then
            DevComponents.DotNetBar.MessageBoxEx.Show("Cek Inputan, Inputan
Tidak Boleh Kosong !", "Kesalahan", MessageBoxButtons.OK,
MessageBoxIcon.Error)
            Exit Sub
        End If

        If (btnSimpan.Text = "Simpan") Then
            Try

                Me.Tb_pegawaiTableAdapter.InsertQuery(A.Text.Trim(),
B.Text.Trim, C.Text)
                DevComponents.DotNetBar.MessageBoxEx.Show("Proses
Penyimpanan Berhasil !", "Informasi", MessageBoxButtons.OK,
MessageBoxIcon.Information)
                Catch ex As Exception
                    DevComponents.DotNetBar.MessageBoxEx.Show("Terjadi
Kesalahanm, Proses Penyimpanan Gagal !", "Kesalahan", MessageBoxButtons.OK,
MessageBoxIcon.Error)
                End Try
                btnRefresh_Click(sender, e)
            ElseIf (btnSimpan.Text = "Edit") Then
                PanelIsi.Enabled = True
                btnSimpan.Text = "Update"
                A.Text = NamaTextBox.Text.Trim
            End If
        End If
    End Sub
End Class
```

---

```

        B.Text = PasswordsTextBox.Text.Trim
        C.Text = StatusTextBox.Text.Trim

    Else
        Try
            Me.Tb_pegawaiTableAdapter.UpdateQuery(A.Text.Trim(),
            B.Text.Trim, C.Text.Trim, Kode_pegawaiTextBox.Text,
            Kode_pegawaiTextBox.Text)
            DevComponents.DotNetBar.MessageBoxEx.Show("Proses Update
            Berhasil !", "Informasi", MessageBoxButtons.OK, MessageBoxIcon.Information)
        Catch ex As Exception
            DevComponents.DotNetBar.MessageBoxEx.Show("Terjadi
            Kesalahanm, Proses Update Gagal !", "Kesalahan", MessageBoxButtons.OK,
            MessageBoxIcon.Error)
        End Try
        btnRefresh_Click(sender, e)
    End If
End Sub

Private Sub btnHapus_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnHapus.Click
    If (DevComponents.DotNetBar.MessageBoxEx.Show("Yakin Untuk
    Menghapus Data?", "Konfirmasi.", MessageBoxButtons.YesNo,
    MessageBoxIcon.Question) = Windows.Forms.DialogResult.Yes) Then
        Try
            Me.Tb_pegawaiTableAdapter.DeleteQuery(Kode_pegawaiTextBox.Text)
            DevComponents.DotNetBar.MessageBoxEx.Show("Proses
            Penghapusan Data Berhasil !", "Informasi", MessageBoxButtons.OK,
            MessageBoxIcon.Information)
        Catch ex As Exception
            DevComponents.DotNetBar.MessageBoxEx.Show("Terjadi
            Kesalahanm, Proses Penghapusan Data Gagal !", "Kesalahan",
            MessageBoxButtons.OK, MessageBoxIcon.Error)
        End Try
        btnRefresh_Click(sender, e)
    End If
End Sub

Private Sub dtGrid_CellContentClick(ByVal sender As System.Object,
ByVal e As System.Windows.Forms.DataGridViewCellEventArgs) Handles
dtGrid.CellContentClick
    PanelIsi.Enabled = False
    btnHapus.Enabled = True
    btnSimpan.Text = "Edit"
    A.Clear()
    B.Clear()

    PanelEx1.Refresh()

End Sub
End Class

```

---

## SCRIPT FORM PENYAKIT

```
Public Class FormPenyakit
    Inherits DevComponents.DotNetBar.Metro.MetroForm

    Private Sub FormPenyakit_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        'TODO: This line of code loads data into the
'DataDataSet.tb_penyakit' table. You can move, or remove it, as needed.
        btnRefresh_Click(sender, e)

    End Sub

    Private Sub btnRefresh_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnRefresh.Click
        btnSimpan.Enabled = True
        btnSimpan.Text = "Simpan"
        btnHapus.Enabled = False
        'TODO: This line of code loads data into the
'DataDataSet.tb_gejala' table. You can move, or remove it, as needed.
        Me.tb_penyakitTableAdapter.Fill(Me.DataDataSet.tb_penyakit)

        A.Clear()
        PanelIsi.Enabled = True
    End Sub

    Private Sub btnSimpan_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnSimpan.Click
        If (A.Text.Trim() = "") And (btnSimpan.Text <> "Edit") Then
            DevComponents.DotNetBar.MessageBoxEx.Show("Cek Inputan, Inputan
Tidak Boleh Kosong !", "Kesalahan", MessageBoxButtons.OK,
MessageBoxIcon.Error)
            Exit Sub
        End If
        If (btnSimpan.Text = "Simpan") Then
            Try
                Me.tb_penyakitTableAdapter.InsertQuery(A.Text.Trim())
                DevComponents.DotNetBar.MessageBoxEx.Show("Proses
Penyimpanan Berhasil !", "Informasi", MessageBoxButtons.OK,
MessageBoxIcon.Information)
            Catch ex As Exception
                DevComponents.DotNetBar.MessageBoxEx.Show("Terjadi
Kesalahan, Proses Penyimpanan Gagal !", "Kesalahan", MessageBoxButtons.OK,
MessageBoxIcon.Error)
            End Try
            btnRefresh_Click(sender, e)
        ElseIf (btnSimpan.Text = "Edit") Then
            PanelIsi.Enabled = True
            btnSimpan.Text = "Update"
            A.Text = Nama_penyakitTextBox.Text
        Else
            Try
                Me.tb_penyakitTableAdapter.UpdateQuery(A.Text.Trim(),
Kode_penyakitTextBox.Text, Kode_penyakitTextBox.Text)
                DevComponents.DotNetBar.MessageBoxEx.Show("Proses Update
Berhasil !", "Informasi", MessageBoxButtons.OK, MessageBoxIcon.Information)
            Catch ex As Exception
            End Try
        End If
    End Sub
End Class
```

---

```

        DevComponents.DotNetBar.MessageBoxEx.Show("Terjadi
Kesalahanm, Proses Update Gagal !", "Kesalahan", MessageBoxButtons.OK,
MessageBoxIcon.Error)
    End Try
    btnRefresh_Click(sender, e)
End If
End Sub

Private Sub btnHapus_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnHapus.Click
    If (DevComponents.DotNetBar.MessageBoxEx.Show("Yakin Untuk
Menghapus Data?", "Konfirmasi.", MessageBoxButtons.YesNo,
MessageBoxIcon.Question) = Windows.Forms.DialogResult.Yes) Then
        Try

Me.Tb_penyakitTableAdapter.DeleteQuery(Kode_penyakitTextBox.Text)
            DevComponents.DotNetBar.MessageBoxEx.Show("Proses
Penghapusan Data Berhasil !", "Informasi", MessageBoxButtons.OK,
MessageBoxIcon.Information)
        Catch ex As Exception
            DevComponents.DotNetBar.MessageBoxEx.Show("Terjadi
Kesalahanm, Proses Penghapusan Data Gagal !", "Kesalahan",
MessageBoxButtons.OK, MessageBoxIcon.Error)
        End Try
        btnRefresh_Click(sender, e)
    End If
End Sub

Private Sub dtGrid_CellContentClick(ByVal sender As System.Object,
ByVal e As System.Windows.Forms.DataGridViewCellEventArgs) Handles
dtGrid.CellContentClick
    PanelTsi.Enabled = False
    btnHapus.Enabled = True
    btnSimpan.Text = "Edit"
    A.Clear()
    PanelEx1.Refresh()
End Sub
End Class

```

---



## SCRIPT FORM RULE

```
Public Class FormRule
    Inherits DevComponents.DotNetBar.Metro.MetroForm

    Private Sub FormRule_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        'TODO: This line of code loads data into the 'DataDataSet.tb_rule'
table. You can move, or remove it, as needed.
        Me.Tb_ruleTableAdapter.Fill(Me.DataDataSet.tb_rule)

        btnRefresh_Click(sender, e)
    End Sub

    Private Sub ListBox1_SelectedIndexChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs)

    End Sub

    Private Sub btnRefresh_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnRefresh.Click
        'TODO: This line of code loads data into the
'DataDataSet.tb_view_rule' table. You can move, or remove it, as needed.
        Me.Tb_view_ruleTableAdapter.Fill(Me.DataDataSet.tb_view_rule)
        'TODO: This line of code loads data into the
'DataDataSet.tb_gejala' table. You can move, or remove it, as needed.
        Me.Tb_gejalaTableAdapter.Fill(Me.DataDataSet.tb_gejala)
        'TODO: This line of code loads data into the
'DataDataSet.tb_penyakit' table. You can move, or remove it, as needed.
        Me.Tb_penyakitTableAdapter.Fill(Me.DataDataSet.tb_penyakit)
        Dim dt As DataTable
        Dim dtDta As New dataDataSetTableAdapters.tb_gejalaTableAdapter()

        btnSimpan.Enabled = True
        btnSimpan.Text = "Simpan"
        btnHapus.Enabled = False
        PanelIsi.Enabled = True

    End Sub

    Private Sub btnSimpan_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnSimpan.Click
        If (A.Text.Trim() = "") And (btnSimpan.Text <> "Edit") Then
            DevComponents.DotNetBar.MessageBoxEx.Show("Cek Inputan, Inputan
Tidak Boleh Kosong!", "Kesalahan", MessageBoxButtons.OK,
MessageBoxIcon.Error)
            Exit Sub
        End If
        Dim status As Double
        If (RadioButton1.Checked) Then
            status = 0.25
        ElseIf (RadioButton2.Checked) Then
            status = 0.5
        Else
            status = 0.75
        End If
    End Sub
```

---

```

If (btnSimpan.Text = "Simpan") Then
    Try

        Me.Tb_ruleTableAdapter.InsertQuery(A.SelectedValue,
Convert.ToInt16(ComboBoxEx1.SelectedValue), status)

        DevComponents.DotNetBar.MessageBoxEx.Show("Proses
Penyimpanan Berhasil !", "Informasi", MessageBoxButtons.OK,
MessageBoxIcon.Information)
        Catch ex As Exception
            DevComponents.DotNetBar.MessageBoxEx.Show("Terjadi
Kesalahanm, Proses Penyimpanan Gagal !", "Kesalahan", MessageBoxButtons.OK,
MessageBoxIcon.Error)
        End Try
        btnRefresh_Click(sender, e)
    ElseIf (btnSimpan.Text = "Edit") Then
        PanelIsi.Enabled = True
        btnSimpan.Text = "Update"
        A.SelectedValue = Kode_penyakitTextBox.Text
    Else
        Try

            Me.Tb_ruleTableAdapter.InsertQuery(A.SelectedValue,
Convert.ToInt16(ComboBoxEx1.SelectedValue), status)

            DevComponents.DotNetBar.MessageBoxEx.Show("Proses Update
Berhasil !", "Informasi", MessageBoxButtons.OK, MessageBoxIcon.Information)
            Catch ex As Exception
                DevComponents.DotNetBar.MessageBoxEx.Show("Terjadi
Kesalahanm, Proses Update Gagal !", "Kesalahan", MessageBoxButtons.OK,
MessageBoxIcon.Error)
            End Try
            btnRefresh_Click(sender, e)
        End If
    End Sub

    Private Sub btnHapus_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnHapus.Click
        If (DevComponents.DotNetBar.MessageBoxEx.Show("Yakin Untuk
Menghapus Data?", "Konfirmasi.", MessageBoxButtons.YesNo,
MessageBoxIcon.Question) = Windows.Forms.DialogResult.Yes) Then
            Try

                Me.Tb_ruleTableAdapter.DeleteQuery(Kode_penyakitTextBox.Text,
Kode_gejalaTextBox.Text)
                DevComponents.DotNetBar.MessageBoxEx.Show("Proses
Penghapusan Data Berhasil !", "Informasi", MessageBoxButtons.OK,
MessageBoxIcon.Information)
                Catch ex As Exception
                    DevComponents.DotNetBar.MessageBoxEx.Show("Terjadi
Kesalahanm, Proses Penghapusan Data Gagal !", "Kesalahan",
MessageBoxButtons.OK, MessageBoxIcon.Error)
                End Try
                btnRefresh_Click(sender, e)
            End If
        End Sub

```

---

```
Private Sub dtGrid_CellContentClick(ByVal sender As System.Object,  
ByVal e As System.Windows.Forms.DataGridViewCellEventArgs) Handles  
dtGrid.CellContentClick  
    PanelIsi.Enabled = False  
    btnHapus.Enabled = True  
    btnSimpan.Text = "Edit"  
    PanelEx1.Refresh()  
End Sub  
  
Private Sub E_SelectedIndexChanged(ByVal sender As System.Object, ByVal  
e As System.EventArgs)  
  
End Sub  
  
Private Sub RadioButton3_CheckedChanged(ByVal sender As System.Object,  
ByVal e As System.EventArgs) Handles RadioButton3.CheckedChanged  
  
End Sub  
End Class
```

---

## SCRIPT FORM UJI

```
Public Class FormUji
    Inherits DevComponents.DotNetBar.Metro.MetroForm
    Private Sub FrmUji_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        btnHapus_Click(sender, e)
    End Sub

    Private Sub LabelX1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs)

    End Sub

    Private Sub A_TextChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs)

    End Sub

    Private Sub ButtonX1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles ButtonX1.Click
        FormPasien.ShowDialog()
    End Sub

    Private Sub A_TextChanged_1(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles A.TextChanged
        If Trim(A.Text) = "" Then Exit Sub
        Me.Tb_pasienTableAdapter.FillByKode(Me.DataDataSet.tb_pasien,
A.Text)
        DataGridViewX1.Refresh()
    End Sub

    Private Sub btnHapus_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnHapus.Click
        'TODO: This line of code loads data into the
'DataDataSet.tb_pasien' table. You can move, or remove it, as needed.
        Me.Tb_pasienTableAdapter.Fill(Me.DataDataSet.tb_pasien)
        Dim dt As DataTable
        Dim dtS As New dataDataSetTableAdapters.tb_gejalaTableAdapter()
        dt = dtS.GetData()
        lsGejala.Items.Clear()
        If dt.Rows.Count > 0 Then
            For i = 0 To dt.Rows.Count - 1
                lsGejala.Items.Add(dt.Rows(i).Item("kode_gejala") & "|" &
dt.Rows(i).Item("nama_gejala"))
            Next i
        End If
    End Sub

    Private Sub btnSimpan_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnSimpan.Click
        getRule()
        defuzzifikasi()
        pnlGejala.Hide()
        pnlHasil.Show()
    End Sub
```

---

```
Dim rule As Dictionary(Of String, Dictionary(Of String, Double)) = New  
Dictionary(Of String, Dictionary(Of String, Double))
```

```
Sub getRule()  
    rule = New Dictionary(Of String, Dictionary(Of String, Double))  
    lsHasil.Items.Clear()  
    For i = 0 To lsGejala.CheckedItems.Count - 1  
  
        Dim tmpCode As String =  
lsGejala.CheckedItems.Item(i).ToString.Split("!")(0)  
        Dim tmpStr As String =  
lsGejala.CheckedItems.Item(i).ToString.Split("|")(1)  
        lsHasil.Items.Add(tmpStr)  
        Dim dt As DataTable  
        Dim dtS As New dataDataSetTableAdapters.tb_ruleTableAdapter()  
        dt = dtS.GetDataByGejala(Convert.ToInt16(tmpCode))  
  
        If dt.Rows.Count > 0 Then  
            For j = 0 To dt.Rows.Count - 1  
                If (rule.ContainsKey(dt.Rows(j).Item("kode_penyakit"))) Then  
                    rule(dt.Rows(j).Item("kode_penyakit")).Add(tmpCode,  
dt.Rows(j).Item("nilai"))  
                Else  
                    Dim tmpCodeArr As Dictionary(Of String, Double) =  
New Dictionary(Of String, Double)  
                    tmpCodeArr.Add(tmpCode, dt.Rows(j).Item("nilai"))  
                    rule.Add(dt.Rows(j).Item("kode_penyakit"),  
tmpCodeArr)  
                End If  
  
                Next j  
            End If  
        Next i  
    End Sub
```

```
Private Function getHasil(ByVal SourceDictionary As Dictionary(Of  
String, Double)) As Dictionary(Of String, Double)
```

```
    Dim inDict As Dictionary(Of String, Double) = SourceDictionary  
    Dim outDict As New Dictionary(Of String, Double)  
    Dim currentMaxKey As String = String.Empty  
    Dim currentMaxValue As Integer = 0  
  
    Do While inDict.Keys.Count > 0  
        For Each currentKVP As KeyValuePair(Of String, Double) In  
inDict  
            If currentKVP.Value >= currentMaxValue Then  
                currentMaxKey = currentKVP.Key  
                currentMaxValue = currentKVP.Value  
            End If  
        Next  
  
        outDict.Add(currentMaxKey, currentMaxValue)  
  
        inDict.Remove(currentMaxKey)  
        currentMaxValue = 0  
    Loop
```

---

```

Return outDict

End Function

Sub defuzzifikasi()
    Dim result As Dictionary(Of String, Double) = New Dictionary(Of
String, Double)

    For Each rowPenyakit As KeyValuePair(Of String, Dictionary(Of
String, Double)) In rule
        Console.WriteLine("Key = {0}, Value = {1}", rowPenyakit.Key,
rowPenyakit.Value)
        Dim fuzzValue As Double = 0
        For Each rowGejala As KeyValuePair(Of String, Double) In
rowPenyakit.Value
            Console.WriteLine("-->> Key = {0}, Value = {1}",
rowGejala.Key, rowGejala.Value)
            fuzzValue = fuzzValue + rowGejala.Value
        Next rowGejala
        result.Add(rowPenyakit.Key, fuzzValue)
    Next rowPenyakit
    Dim Newresult As Dictionary(Of String, Double) = getHasil(result)
    For Each rows As KeyValuePair(Of String, Double) In Newresult
        Console.WriteLine(">>> Key = {0}, Value = {1}", rows.Key,
rows.Value)
    Next rows
    Dim lsHasil As List(Of String) = Newresult.Keys.ToList
    Me.Tb_pasienTableAdaptor.Fill(Me.DataDataSet.tb_pasien)

    Dim dt As DataTable
    Dim dtS As New dataDataSetTableAdapters.tb_penyakitTableAdapter()
    dt = dtS.GetDataByKode(lsHasil(0))

    If dt.Rows.Count > 0 Then
        For i = 0 To dt.Rows.Count - 1
            B.Text = dt.Rows(i).Item("nama_penyakit")

        Next i
    Else
        B.Text = "PENYAKIT TIDAK DIKENALI"
    End If

End Sub

Private Sub LabelX5_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs)

End Sub

Private Sub PanelIsi_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles PanelIsi.Click

End Sub

Private Sub ButtonX3_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles ButtonX3.Click
    If (B.Text.Trim <> "") Then
        Me.Tb_rekapmedikTableAdapter1.InsertQuery(Convert.ToInt16(Kode_pasienTextBo

```

```
xl.Text.Trim), Now, FormLogin.UsernameTextBox.Text, C.Text.Trim,
B.Text.Trim)
    End If
    ButtonX2_Click(sender, e)
    DevComponents.DotNetBar.MessageBoxEx.Show("Proses Penyimpanan
Berhasil !", "Informasi", MessageBoxButtons.OK, MessageBoxIcon.Information)
    End Sub

Private Sub ButtonX2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles ButtonX2.Click
    C.Clear()
    A.Clear()
    B.Clear()
    lsGejala.Items.Clear()
    lsHasil.Items.Clear()
    pnlGejala.Show()
    pnlHasil.Hide()
    btnHapus_Click(sender, e)
End Sub
End Class
```

---