

SKRIPSI

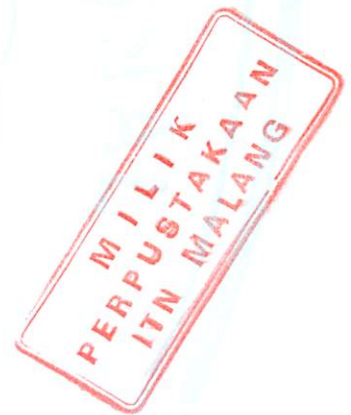
FUZZY INFERENCE SYSTEM UNTUK MENDIAGNOSA PENYAKIT INFEKSI PADA MANUSIA MENGUNAKAN METODE TSUKAMOTO



Disusun Oleh :

PUTRI SARI S.

NIM: 06.12.588



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

LEMBAR PERSETUJUAN
FUZZY INFERENCE SYSTEM UNTUK MENDIAGNOSA
PENYAKIT INFEKSI PADA MANUSIA
MENGGUNAKAN METODE TSUKAMOTO

SKRIPSI

*Disusun dan Diajukan Untuk Melengkapi dan Memenuhi Syarat
Guna Mencapai Gelar Sarjana Teknik*

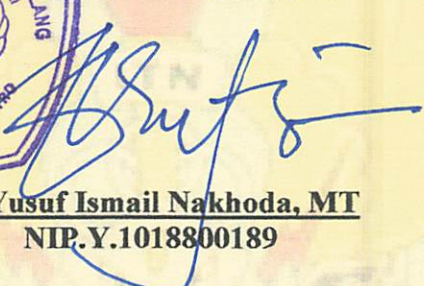
Disusun oleh :

PUTRI SARI S.

NIM : 06.12.588

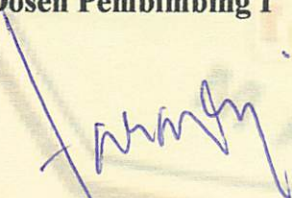


Mengetahui,
Ketua Prodi T. Elektro S-1

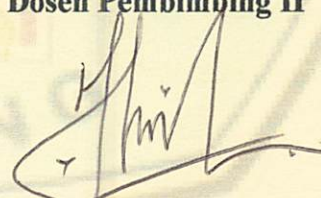

Ir. Yusuf Ismail Nakhoda, MT
NIP.Y.1018800189

Diperiksa dan Disetujui,

Dosen Pembimbing I


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

Dosen Pembimbing II


Yuli Wahyuni. ST, MT
NIP.P. 1031200456

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

Lembar Persembahan

Bismillahirrohmanirrohim

Syukur Alhamdulillah, ku panjatkan segala puji syukur kepada Allah Robbul'Alamiin, atas segala rahmat, hidayah dan izin-Nya, sehingga masih diberi kesempatan untuk bernaung dibawah lindungan-Nya dan diberi kesempatan untuk menyelesaikan laporan Skripsi ini.

Hasil karya ini aku persembahkan kepada semua yang menjadi bagian dalam hidupku :

- Kedua orang tuaku yang aku sayangi, terimakasih sudah melahirkan, mendidik, membesarkan dan memberikan perhatian serta doa dan dukungannya dan terimakasih untuk kesabarannya selama ini.
- Untuk adekku satu-satunya (Ayu "Bebek") terimakasih buat bantuan menghitung rumus-rumus dan doanya.
- Untuk Keluarga besar di Malang, Madiun dan Kertosono, terimakasih buat semua perhatian, nasehat serta doanya.
- Untuk teman-teman Kozemake (Anggie, Atik, Dini, Diana, Dayu, Nonik, Lina, Ratna, dan Malida) terimakasih selalu menyemangati dan mengingatkan untuk menyelesaikan kuliah serta terimakasih doanya.
- Juga untuk si duo (Diyah dan Nita) terimakasih sudah banyak merepotkan selama ini.
- Teman-teman di Lab Elka Digital yang tidak bisa aku sebutkan satu persatu, terimakasih banyak buat bantuan tempat mengerjakan skripsiku.
- *Untuk seseorang yang selalu setia mendampingi, mengantar dan memberikanku semangat terimakasih untukmu (Ai dYka).*
- Terakhir untuk semua teman-teman seperjuangan, terimakasih untuk saran dan kritiknya. Selamat datang di dunia kerja..

SURAT PERNYATAAN ORISINALITAS

Yang bertanda tangan di bawah ini :

Nama : Putri Sari S.

NIM : 0612588

Program Studi : Teknik Elektro S1

Konsentrasi : Teknik Informatika dan Komputer S1

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, Agustus 2012

Yang membuat Pernyataan,



Putri Sari S.

NIM :0612588

FUZZY INFERENCE SYSTEM UNTUK MENDIAGNOSA PENYAKIT INFEKSI PADA MANUSIA MENGUNAKAN METODE TSUKAMOTO

Putri Sari S., 06.12.588

Email : ce_aquarius_1607@yahoo.co.id

Jurusan Teknik Elektro

Konsentrasi Teknik Komputer & Informatika S-1

Fakultas Teknologi Industri

Institut Teknologi Nasional Malang

**Dosen Pembimbing : 1. Irmalia Suryani F. ST, MT
2. Yuli Wahyuni. ST, MT.**

ABSTRAKSI

Gejala merupakan suatu unsur penting dalam menentukan suatu pasien mengidap penyakit tertentu. Beberapa penyakit mungkin memiliki gejala yang sama, namun yang membedakan adalah intensitasnya atau rasa sakitnya. Karena terjadinya masalah tersebut maka di buatlah software yang mengambil cara kerja dokter untuk menganalisa penyakit pasien.

Cara kerja software menggunakan fuzzy inference system menggunakan metode tsukamoto dalam menentukan prediksi awal suatu penyakit infeksi pada manusia dengan memasukkan gejala yang diderita yang dapat menangani masalah ketidakpastian seperti intensitas gejala seperti yang berbeda-beda. Sistem operasi yang bekerja dalam software ini adalah Microsoft Windows XP, dengan database My SQL sebagai media penyimpanan, dan menggunakan bahasa pemrograman Delphi 7.0.

Dengan Metode tsukamoto maka di dapat domain yang mendekati hasil diagnosa dokter adalah intensitas fuzzy RENDAH memiliki domain $[0, 4]$, SEDANG memiliki domain $[2, 8]$, dan TINGGI memiliki domain $[4, 10]$ sehingga di dihasilkan rata-rata adalah 71%.

Kata Kunci: gejala, fuzzy inference system, metode tsukamoto.

Abstract

Symptoms is an important element in determine a patient's specific disease. Some diseases may have similar symptoms, but the difference is the intensity or the pain. Because of these problems then create software that takes the doctor to analyze the workings of a patient's disease.

The workings of the software using the fuzzy inference system using the Tsukamoto method in determining the initial prediction of an infectious disease in humans by entering suffered symptoms that can deal with uncertainties such as the intensity of symptoms vary. Operating systems that work in this software is Microsoft Windows XP, the My SQL database as storage media, and use the programming language Delphi 7.0.

With Tsukamoto method is in the domain can approach the intensity of the doctor's diagnosis is to have LOW fuzzy domain $[0, 4]$, IS has a domain $[2, 8]$, and HIGH have the domain $[4, 10]$ that generated an average of 71 %.

Keyword: symptom, fuzzy inference system, Tsukamoto methods

KATA PENGANTAR

Alhamdulillah, puji syukur kehadiran-Mu Ya Allah yang telah memberikan Rahmat dan HidayahNya, sehingga penulis dapat menyelesaikan skripsi dengan judul **“Fuzzy Inference System Untuk Mendiagnosa Penyakit Infeksi Pada Manusia Menggunakan Metode Tsukamoto”** dengan lancar. Skripsi merupakan persyaratan kelulusan Studi di Program Studi 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 penyusun menyampaikan terima kasih kepada :

1. Allah Robbul’Alamiin, atas segala rahmat, hidayah dan izin-Nya, sehingga masih diberi kesempatan untuk bernaung dibawah lindungan-Nya dan diberi kesempatan untuk menyelesaikan laporan Skripsi ini.
2. Nabi Muhammad S.A.W, yang telah membawa umat manusia dari kehidupan jahiliyah menuju kehidupan dengan kejayaan Islam.
3. Bapak Ir. Soeparno Djiwo, MT selaku Rektor Institut Teknologi Nasional Malang.
4. Bapak Ir. Sidik Noetjahjono, MT selaku Dekan Fakultas Teknologi Industri Institut Teknologi Nasional Malang
5. Bapak Yusuf Ismail Nahkoda ST, MT selaku Ketua Jurusan Teknik Elektro S-1.
6. Bapak Dr. Eng. Aryuanto Soetejo, ST, MT selaku Sekretaris Jurusan Teknik Elektro S-1.
7. Ibu Irmalia Suryani F. ST, MT selaku Dosen Pembimbing I.
8. Ibu Yuli Wahyuni. ST, MT selaku Dosen Pembimbing II.
9. Serta semua pihak yang telah membantu, yang tidak bisa disebutkan satu persatu.

Penulis telah berusaha semaksimal mungkin dan menyadari sepenuhnya akan keterbatasan pengetahuan dalam menyelesaikan laporan ini. Untuk itu penyusun mengharapkan saran dan kritik yang membangun dari pembaca demi kesempurnaan laporan ini.

Harapan penulis semoga laporan skripsi ini memberikan manfaat bagi perkembangan ilmu pengetahuan bangsa dan negara.

Malang, Agustus 2012

Penulis

DAFTAR ISI

HALAMAN JUDUL	i
LEMBAR PERSETUJUAN	ii
LEMBAR PERSEMBAHAN	iii
ABSTRAK	iv
KATA PENGANTAR	v
DAFTAR ISI	vi
DAFTAR TABEL	ix
DAFTAR GAMBAR	x
BAB I PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	2
1.3. Tujuan dan Manfaat Penelitian	2
1.4. Batasan Masalah	2
1.5. Metode Penelitian	3
1.6. Sistematika Penelitian	3
BAB II TINJAUAN PUSTAKA	5
2.1. Logika Fuzzy	5
2.1.1. Himpunan Fuzzy	5
2.1.2. Notasi-Notasi Himpunan Fuzzy	5
2.1.3. Fungsi Keanggotaan	7
2.1.4. Operator Dasar Zadeh Untuk Operasi Himpunan Fuzzy	14
2.1.5. Fungsi Implikasi	15
2.1.6. Alasan Digunakannya Logika Fuzzy	16
2.1.7. Jenis – Jenis dari Fuzzy logic	17
2.2 Fuzzy Inference System	17
2.2.1. Metode Tsukamoto	18
2.2.2. Fuzzifikasi	19
2.2.3. Inferensi	20
2.2.4. Komposisi dan Defuzzifikasi	20
2.3 Borland Delphi 7	20
2.3.1 Sejarah Borland Delphi	20

2.3.2 Kelebihan Borland Delphi 7.0	22
2.3.3 Kebutuhan Sistem	22
2.3.4 IDE Delphi	23
2.4. My SQL	24
2.4.1. Sepintas Mengenai MySQL	24
2.4.2. Alasan memilih MySQL	25
2.5. Perangkat Analisa dan Perancangan	28
2.5.1. Pemodelan Data	28
2.5.2. Metode Pengembangan Sistem	29
2.6 Penyakit Infeksi	29
2.6.1. Penyakit Infeksi Menyebar Melalui Kontak	30
2.6.2. Penyakit Infeksi Menyebar melalui udara	31
2.6.3. Penyakit Infeksi Menyebar Melalui Vektor & Pembawa	31
2.7. Probabilitas	32
BAB III METODOLOGI DAN PERENCANAAN	33
3.1. Pengumpulan Data	33
3.2. Deskripsi umum sistem	33
3.3. Perancangan Basis Penalaran	34
3.3.1. DFD	34
3.3.2. Perancangan Tabel Pengetahuan	36
3.3.3. Tabel Keputusan	37
3.3.4. Pengalihan Tabel Keputusan ke Bentuk <i>If-Then Rule</i>	38
3.3.5. Pembelajaran / <i>Learning</i>	38
3.4. Perancangan Mesin Inferensi	39
3.4.1. Inferensi	39
3.4.2. Menentukan Variabel Linguistik	40
3.4.3. Menentukan Fungsi Keanggotaan (Membership Function)	41
3.4.4. Penanganan Gejala yang tidak dimasukkan oleh user	44
3.4.5. Komposisi dan Defuzzifikasi	44
3.5. Perancangan Basis Data	45
3.5.1. Perancangan Tabel Database	45
3.6. Diagram Alir Sistem	48
3.7. Perancangan Antarmuka Pemakai (<i>User Interface</i>)	53
3.8. Pembuatan Desain Interface	54

3.9. Struktur Basis Pengetahuan (<i>Knowledge Base</i>)	58
BAB IV IMPLEMENTASI DAN PEMBAHASAN	60
4.1. Lingkungan Implementasi	60
4.1.1 Lingkungan Perangkat Keras	60
4.1.2. Lingkungan Perangkat Lunak	60
4.2. Implementasi Program	60
4.2.1 Pada User	61
4.2.1.1 Proses search	61
4.2.1.2 Proses Diagnosa	62
4.2.1.2.1 Proses Input User	62
4.2.1.2.2 Proses Input Gejala	63
4.2.1.2.3 Input Intensitas Gejala	64
4.2.1.2.4 Hasil Diagnosa	64
4.2.2 Untuk Admin	65
4.2.2.1 Proses Login	65
4.2.2.2 Edit Gejala	65
4.2.2.3 Edit Penyakit	67
4.2.2.4 Edit Rule	67
4.2.2.5 Edit Data Hak akses	68
4.2.2.6 Edit user	69
4.3. Analisa Data	70
BAB V PENUTUP	72
5.1. Kesimpulan	72
5.2. Saran	72
DAFTAR PUSTAKA	
LAMPIRAN-LAMPIRAN	

DAFTAR TABEL

BAB III METODOLOGI DAN PERENCANAAN

Tabel 3.1. Tabel Daftar Jenis Penyakit	36
Tabel 3.2. Tabel Daftar Nama Gejala	36
Tabel 3.3. Tabel Variabel Linguistik dan Domain dari Intensitas	40
Tabel 3.4. Tabel Variabel Linguistik dan Domain Peluang Penyakit	40
Tabel 3.5. Tabel Gejala (tbGejala)	45
Tabel 3.6. Tabel Rule If Gejala (tbRGejala_If)	45
Tabel 3.7. Tabel Rule Then	45
Tabel 3.8. Tabel Jenis Penyakit	45
Tabel 3.9. Tabel Membership Function (tbFuzzy)	46
Tabel 3.10. Contoh Tabel Gejala Demam Berdarah	46
Tabel 3.11. Contoh Tabel Penyakit (tbPenyakit)	47
Tabel 3.12. Contoh Tabel Membership Function (tbFuzzy)	47
Tabel 3.13. Contoh Tabel Rule If Gejala (tbRGejala_If)	47
Tabel 3.14. Contoh Tabel Rule Then Gejala (tbRGejala_Then)	47
Tabel 3.15. Tabel Peraturan Penyakit dan Gejala	58

BAB IV IMPLEMENTASI DAN PEMBAHASAN

Tabel 4.1. Tabel Pengujian	70
----------------------------------	----

DAFTAR GAMBAR

BAB II TINJAUAN PUSTAKA	5
Gambar 2.1 Contoh pemetaan input output	5
Gambar 2.2 Himpunan Fuzzy pada Variabel Temperatur	6
Gambar 2.3 Representasi Linear Naik	8
Gambar 2.4 Representasi Linear Turun	8
Gambar 2.5 Kurva Segitiga	9
Gambar 2.6 Kurva Trapesium	9
Gambar 2.7 Kurva ‘Bahu’	10
Gambar 2.8 Himpunan Fuzzy dengan Kurva-S: PERTUMBUHAN	10
Gambar 2.9 Himpunan Fuzzy dengan Kurva-S: PENYUSUTAN	11
Gambar 2.10 Karakteristik Fungsi Kurva-S Dalam Bentuk Skema	11
Gambar 2.11 Karakteristik Fungsional Kurva PI	12
Gambar 2.12 Karakteristik Fungsional Kurva BETA	13
Gambar 2.13 Karakteristik Fungsional GAUSS	14
Gambar 2.14 Fungsi Implikasi: MIN	15
Gambar 2.15 Fungsi Implikasi: DOT	16
Gambar 2.16 Konsep Umum Kronologi Proses Pembangunan FIS	18
Gambar 2.17 Inferensi dengan Menggunakan Metode Tsukamo	19
Gambar 2.18 Fuzzifikasi	19
Gambar 2.19 Gambar Tampilan Borland Delphi 7.0	21
Gambar 2.20 Bagian-bagian IDE	23
BAB III METODOLOGI DAN PERANCANGAN	33
Gambar 3.1 DFD level 0	34
Gambar 3.2. Fungsi Keanggotaan untuk intensitas gejala.....	41
Gambar 3.3. Fungsi Keanggotaan untuk peluang penyakit	42
Gambar 3.4. Diagram Alir Sistem Pada User	48
Gambar 3.5. Diagram Alir “Proses Search”.....	49
Gambar 3.6 Diagram Alir “Proses Diagnosa”.....	51
Gambar 3.7. Diagram Alir Untuk admin	52
Gambar 3.8. Form Menu	54
Gambar 3.9. Form Login	54
Gambar 3.10 Form Search	54

Gambar 3.11 Form Hasil Search	55
Gambar 3.12 Form Diagnosa	55
Gambar 3.13 Form Intensitas	56
Gambar 3.14 Form Diagnosa	56
Gambar 3.15 Form Edit Gejala	56
Gambar 3.16 form edit penyakit	57
Gambar 3.17 form edit rule	57
BAB IV IMPLEMENTASI DAN PEMBAHASAN	60
Gambar 4.1 Tampilan menu	61
Gambar 4.2 Proses Search	61
Gambar 4.3 Hasil Pencarian	62
Gambar 4.4 Form Input user	63
Gambar 4.5 Form Gejala dalam Diagnosa	63
Gambar 4.6 Input Intensitas Gejala Penyakit	64
Gambar 4.7 Form Hasil Diagnosa	64
Gambar 4.8 Proses Login	65
Gambar 4.9 Form Tabel Gejala	66
Gambar 4.10 Form Edit Gejala	66
Gambar 4.11 Form Tabel Penyakit	67
Gambar 4.12 Form Edit Penyakit	67
Gambar 4.13 Form edit rule	68
Gambar 4.14 Form edit hak akses	68
Gambar 4.15 Form Edit data pasien	69



BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan yang pesat yang terjadi pada teknologi komputer mengakibatkan komputer yang awalnya hanya digunakan sebagai alat penghitung dan penyimpan data, maka di era informasi ini komputer dapat menyelesaikan tugas-tugas yang rumit dilakukan oleh manusia, serta dapat menirukan proses biologis manusia dalam mengambil keputusan.

Perkembangan teknologi komputer ini juga mempengaruhi dunia medis. Bentuk dari pemanfaatan teknologi komputer dalam dunia medis adalah pemanfaatan fuzzy inference system. Fuzzy inference system (FIS) merupakan salah satu jenis dari fuzzy logic. Fuzzy Inference system adalah sistem yang dapat melakukan penalaran dengan prinsip serupa seperti manusia melakukan penalaran dengan nalurinya.

Fuzzy inference system dapat diaplikasikan untuk mendeteksi penyakit infeksi pada manusia. Dalam pengertian medis, penyakit menular atau penyakit infeksi adalah penyakit yang disebabkan oleh gen biologi (seperti virus, bakteri, atau parasit), bukan disebabkan faktor fisik (seperti luka bakar) atau kimia (seperti keracunan).

Penyakit infeksi memiliki gejala awal yang hampir serupa sehingga seorang tidak dapat dengan pasti mengetahui penyakit yang di derita. Yang membedakan suatu penyakit adalah intensitas dan frekuensi gejala serta gejala-gejala susulan yang menyerang.

Untuk itu diperlukan suatu metode pendekatan yang dapat memperhitungkan faktor-faktor ketidakpastian dalam diagnosa penyakit infeksi. Metode yang digunakan adalah metode tsukamoto.

Metode tsukamoto merupakan salah satu metode penalaran pada fuzzy inference system. Dimana hasil akhir yang diperoleh dari metode tsukamoto menggunakan rata-rata terbobot. Hasil dari rata-rata terbobot merupakan output penyakit infeksi yang dialami.

1.2 Rumusan Masalah

Dari latar belakang di atas maka diperlukan program yang membantu masyarakat umum dan dapat dirumuskan masalah sebagai berikut:

1. Bagaimana merancang dan mengimplementasikan aplikasi dalam menentukan prosentase terjadinya penyakit berdasarkan input gejala-gejala yang diberikan user menggunakan logika fuzzy inference system?
2. Bagaimana memberikan informasi tambahan seperti solusi awal mengobati dan penyebab penyakit yang diderita user?

1.3 Tujuan dan Manfaat Penelitian

Tujuan dan manfaat penelitian yang ingin dicapai adalah:

1. Untuk memberikan informasi kepada masyarakat tentang penyakit infeksi pada manusia.
2. Untuk memberikan informasi kepada masyarakat tentang gejala awal penyakit infeksi pada manusia.

1.4 Batasan Masalah

Agar permasalahan mengarah sesuai dengan tujuan maka pembahasan dibatasi hal-hal sebagai berikut:

- 1) Jenis penyakit infeksi yang digunakan sebagai studi kasus adalah penyakit infeksi yang sering mendapat perhatian utama dari masyarakat, baik karena tingkat penularan maupun bahayanya yaitu:
 - a) Cacar
 - b) Campak
 - c) Demam Berdarah Dengue
 - d) Demam Chikungunya
 - e) Disentri
 - f) Influenza
 - g) Malaria
 - h) TBC
- 2) Data gejala yang digunakan adalah gejala yang umum bukan hasil laboratorium.
- 3) Program yang dipakai adalah Delphi 7.0 sebagai pembuatan program aplikasi dan MySQL sebagai database-nya, serta Microsoft Windows XP sebagai sistem operasinya.

1.5 Metode Penelitian

Metode yang digunakan dalam penyusunan skripsi ini adalah :

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 Sistem

Data dan informasi yang telah diperoleh akan dianalisa agar didapatkan kerangka global yang bertujuan untuk mendefinisikan kebutuhan sistem baik hardware maupun software, di mana nantinya akan digunakan sebagai acuan perancangan sistem.

3) Perancangan sistem

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

4) Implementasi Sistem

Tahapan ini semua rancangan yang sudah dihasilkan diterjemahkan dengan menggunakan bahasa pemrograman Borland Delphi 7.0 dan MySQL.

5) Eksperimen dan Evaluasi

Pada tahap ini, sistem yang telah selesai dibuat akan diuji coba, yaitu pengujian sistem dilakukan untuk menguji dan mengetahui apakah sistem berjalan dengan baik dan benar sesuai dengan fuzzy inference system.

1.6. Sistematika Pembahasan

Penyusunan laporan Tugas Akhir ini menggunakan kerangka pembahasan yang terbentuk dalam susunan bab, yang dapat dijelaskan sebagai berikut :

Bab I : Pendahuluan

Bab ini merupakan dasar penyusunan laporan Tugas Akhir yang di dalamnya berisi tentang latar belakang masalah, rumusan masalah, tujuan Tugas Akhir, batasan masalah, metodologi pengembangan sistem, dan sistematika pembahasan Tugas Akhir.

Bab II : Dasar Teori

Bab ini berisi tentang permasalahan yang berhubungan dengan penelitian yang dilakukan pada Tugas Akhir ini.

Bab III : Desain Sistem

Berisi perancangan aplikasi sesuai dengan data yang didapatkan.

Bab IV: Implementasi dan Pengujian

Bab ini berisi tentang tahap implementasi yaitu identifikasi untuk menyelesaikan masalah yang dihadapi berdasarkan teori pada Bab II dan Bab III. Bab ini juga berisi hasil pengujian sistem.

Bab V : Penutup

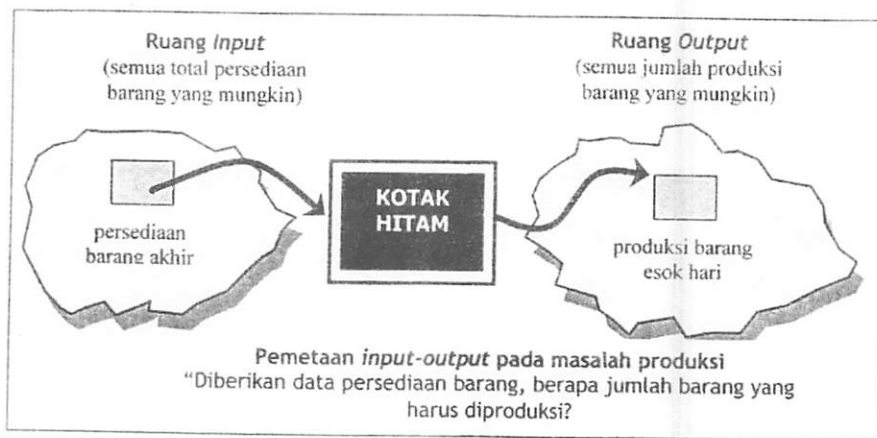
Bab ini berisi tentang kesimpulan dan saran dari hasil penyusunan laporan Tugas Akhir yang telah disusun.

BAB II

TINJAUAN PUSTAKA

2.1 Logika Fuzzy ^[4]

Logika fuzzy merupakan salah satu komponen pembentuk *soft computing*. Logika fuzzy pertama kali diperkenalkan oleh Prof. Lofti A. Zaedah pada tahun 1965. Dasar logika fuzzy adalah teori himpunan fuzzy. Pada teori himpunan fuzzy, peranan derajat keanggotaan sebagai penentu keberadaan elemen Nilai keanggotaan atau *membership function* menjadi ciri utama dari penalaran dengan logika tersebut.



Gambar 2.1 Contoh pemetaan input output

2.1.1 Himpunan Fuzzy ^[4]

Pada himpunan tegas (crisp), nilai keanggotaan suatu item x dalam suatu himpunan A , yang sering ditulis dengan $\mu_A(x)$, memiliki dua kemungkinan, yaitu :

- Satu (1), yang berarti bahwa suatu item menjadi anggota dalam suatu himpunan, atau
- Nol (0), yang berarti bahwa suatu item tidak menjadi anggota dalam suatu himpunan.

2.1.2 Notasi – Notasi Himpunan Fuzzy ^[4]

Himpunan fuzzy memiliki 2 atribut, yaitu :

- 1) Linguistik, yaitu penamaan suatu grup yang mewakili suatu keadaan atau kondisi tertentu dengan menggunakan bahasa alami, seperti : MUDA, PAROBAYA, TUA.

- 2) Numeris, yaitu suatu nilai (angka) yang menunjukkan ukuran dari suatu variabel seperti : 40, 25, 50, dsb.

Ada beberapa hal yang perlu diketahui dalam memahami sistem fuzzy, yaitu :

1) Variabel fuzzy

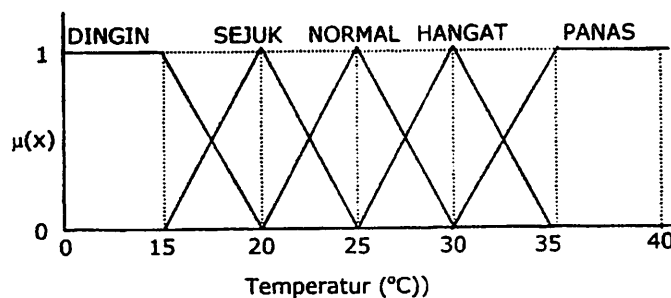
Variabel fuzzy merupakan variabel yang hendak dibahas dalam suatu sistem fuzzy. Contoh : umur, temperatur, permintaan, dsb.

2) Himpunan fuzzy

Himpunan fuzzy merupakan suatu grup yang mewakili suatu kondisi atau keadaan tertentu dalam suatu variabel fuzzy.

Contoh:

Variabel umur, terbagi menjadi 5 himpunan fuzzy, yaitu : DINGIN, SEJUK, NORMAL, HANGAT, dan PANAS.



Gambar 2.2 Himpunan Fuzzy pada Variabel Temperatur

3) Semesta Pembicaraan

Semesta pembicaraan adalah keseluruhan nilai yang diperbolehkan untuk dioperasikan dalam suatu variabel fuzzy. Semesta pembicaraan merupakan himpunan bilangan real yang senantiasa naik (bertambah) secara monoton dari kiri ke kanan. Nilai semesta pembicaraan dapat berupa bilangan positif maupun negatif. Adakalanya nilai semesta pembicaraan ini tidak dibatasi batas atasnya.

Contoh :

- Semesta pembicaraan untuk variabel umur : $[0 +\infty]$
(berada pada *range* 0 sampai dengan tak terhingga)
- Semesta pembicaraan untuk variabel temperatur: $[0 40]$
(berada pada *range* 0°C sampai dengan 40°C)

4) Domain

Domain himpunan fuzzy adalah keseluruhan nilai yang diizinkan dalam semesta pembicaraan dan boleh dioperasikan dalam suatu himpunan fuzzy. Seperti halnya semesta pembicaraan, domain merupakan bilangan real yang senantiasa naik (bertambah) secara monoton dari kiri ke kanan. Nilai dominan dapat berupa bilangan positif maupun negatif.

Contoh domain himpunan fuzzy :

- MUDA = $[0 \quad 45]$
- PAROBAYA = $[35 \quad 55]$
- TUA = $[45 \quad +\infty]$
- DINGIN = $[0 \quad 20]$

2.1.3 Fungsi Keanggotaan ^[4]

Fungsi Keanggotaan (*membership function*) adalah suatu kurva yang menunjukkan pemetaan titik – titik *input* data ke dalam nilai keanggotaannya (sering juga disebut dengan derajat keanggotaan) yang memiliki interval antara 0 sampai 1.

Salah satu cara yang dapat digunakan untuk mendapatkan nilai keanggotaan adalah dengan melalui pendekatan fungsi. Ada beberapa fungsi yang bisa digunakan:

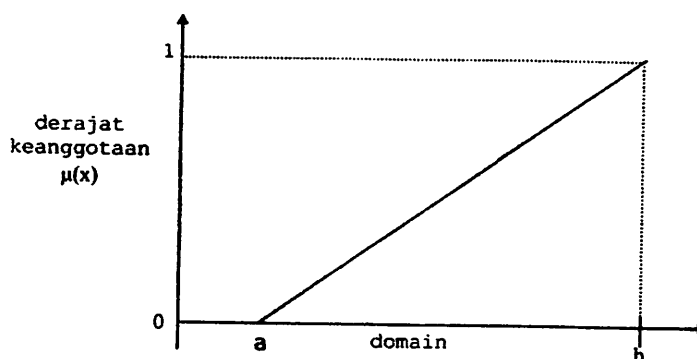
1) Representasi Linear

Pada representasi linear, pemetaan *input* ke derajat keanggotaannya digambarkan sebagai suatu garis lurus.

Ada dua keadaan himpunan fuzzy yang linear :

- Representasi Linear Naik

Kenaikkan himpunan dimulai pada nilai dominan yang memiliki derajat keanggotaan nol (0) bergerak ke kanan menuju ke nilai domain yang memiliki derajat keanggotaan lebih tinggi.



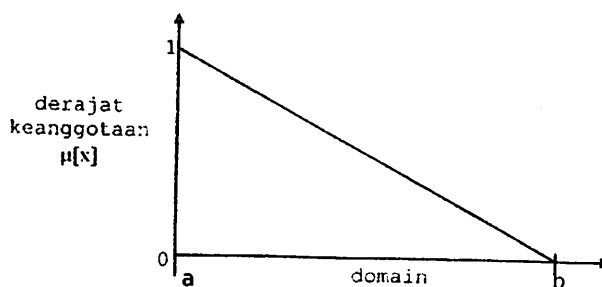
Gambar 2.3 Representasi Linear Naik

Fungsi Keanggotaan :

$$\mu[x] = \begin{cases} 0; & x \leq a \\ (x - a)/(b - a); & a \leq x \leq b \\ 1; & x \geq b \end{cases} \dots\dots\dots (2.1)$$

- Representasi Linear Turun

Merupakan kebalikan dari Linear naik. Garis lurus dimulai dari nilai dominan dengan derajat keanggotaan tertinggi pada sisi kiri, kemudian bergerak menurun ke nilai domain yang memiliki derajat keanggotaan lebih rendah.



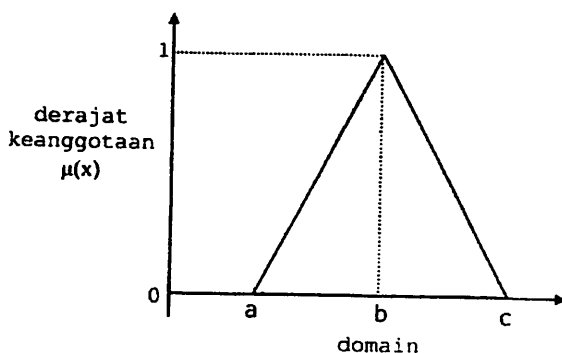
Gambar 2.4 Representasi Linear Turun

Fungsi Keanggotaan:

$$\mu[x] = \begin{cases} (b - x) / (b - a); & a \leq x \leq b \\ 0; & x \geq b \end{cases} \dots\dots\dots (2.2)$$

2) Representasi Kurva Segitiga

Kurva segitiga pada dasarnya merupakan gabungan antara 2 garis (linear).



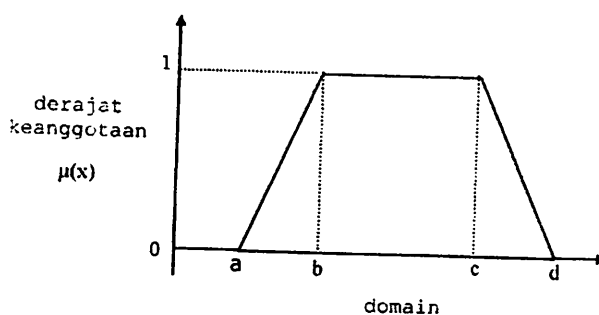
Gambar 2.5 Kurva Segitiga

Fungsi Keanggotaan :

$$\mu[x] = \begin{cases} 0; & x \leq a \text{ atau } x \geq c \\ (x - a) / (b - a); & a \leq x \leq b \\ (b - x) / (c - b); & b \leq x \leq c \end{cases} \dots\dots\dots(2.3)$$

3) Representasi Kurva Trapesium

Kurva trapesium pada dasarnya seperti bentuk segitiga, hanya saja ada beberapa titik yang memiliki nilai keanggotaan 1.



Gambar 2.6 Kurva Trapesium

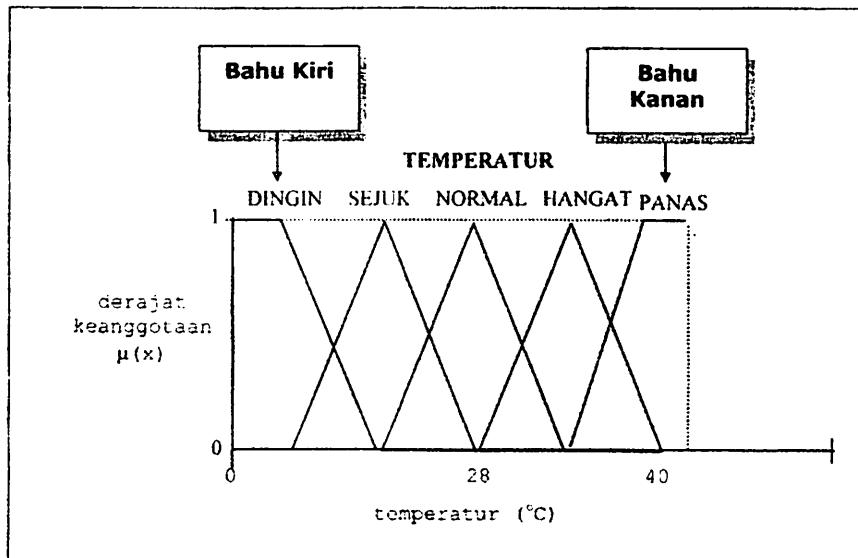
Fungsi keanggotaan:

$$\mu[x] = \begin{cases} 0; & x < a \text{ atau } x \geq c \\ (x - a) / (b - a); & a \leq x \leq b \\ (b - x) / (c - b); & b \leq x \leq c \end{cases} \dots\dots\dots(2.4)$$

4) Representasi Kurva Bentuk Bahu

Daerah yang terletak ditengah-tengah suatu variabel yang dipresentasikan dalam bentuk segitiga, pada sisi kanan dan kirinya akan naik dan turun. Tetapi terkadang salah satu sisi dari variabel tersebut tidak mengalami perubahan.

Himpunan fuzzy 'bahu' bukan segitiga, digunakan untuk mengakhiri variabel suatu daerah fuzzy. Bahu kiri bergerak dari benar ke salah, demikian juga bahu kanan bergerak dari salah ke benar.

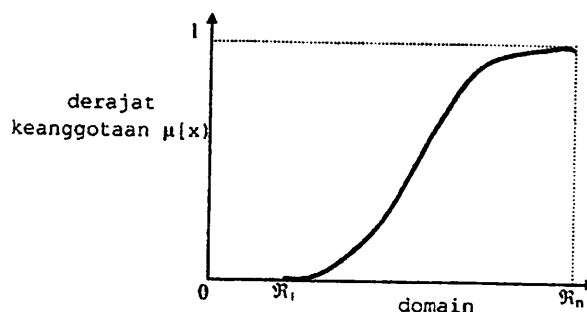


Gambar 2.7 Kurva 'Bahu'

5) Representasi Kurva-S

Kurva PERTUMBUHAN dan PENYUSUTAN merupakan kurva-S atau *sigmoid* yang berhubungan dengan kenaikan dan penurunan permukaan secara tak linear.

- Kurva-S untuk PERTUMBUHAN akan bergerak dari sisi paling kiri (nilai keanggotaan = 0) ke sisi paling kanan (nilai keanggotaannya = 1). Fungsi keanggotaan akan tertumpu pada 50% nilai keanggotaannya yang sering disebut dengan titik infleksi.

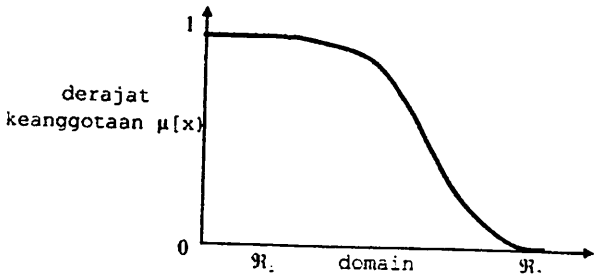


Gambar 2.8 Himpunan Fuzzy dengan Kurva-S: PERTUMBUHAN

Fungsi keanggotaan:

$$S(x; \alpha, \beta, \gamma) = \begin{cases} 0 & \rightarrow x \leq \alpha \\ 2((x - \alpha) / (\gamma - \alpha))^2 & \rightarrow \alpha \leq x \leq \beta \\ 1 - 2((\gamma - x) / (\gamma - \alpha))^2 & \rightarrow \beta \leq x \leq \gamma \\ 1 & \rightarrow x \geq \gamma \end{cases} \dots\dots\dots (2.5)$$

- Kurva-S untuk PENYUSUTAN akan bergerak dari sisi paling kanan (nilai keanggotannya = 1) ke sisi paling kiri (nilai keanggotannya = 0).

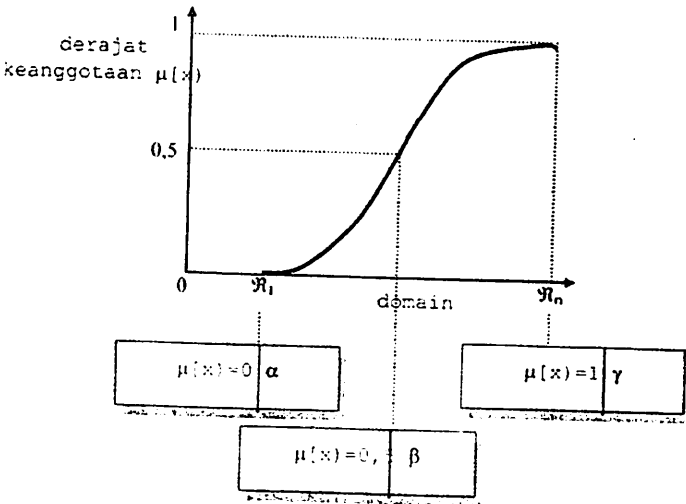


Gambar 2.9 Himpunan Fuzzy dengan Kurva-S: PENYUSUTAN

Fungsi keanggotaan:

$$S(x; \alpha, \beta, \gamma) = \begin{cases} 1 & \rightarrow x \leq \alpha \\ 2((x - \alpha) / (\gamma - \alpha))^2 & \rightarrow \alpha \leq x \leq \beta \\ 1 - 2((\gamma - x) / (\gamma - \alpha))^2 & \rightarrow \beta \leq x \leq \gamma \\ 0 & \rightarrow x \geq \gamma \end{cases} \dots\dots\dots (2.6)$$

Kurva-S didefinisikan dengan menggunakan 3 parameter, yaitu: nilai keanggotaan nol (α), nilai keanggotaan lengkap (γ), dan titik infleksi atau crossover (β) yaitu titik yang memiliki domain 50% benar.



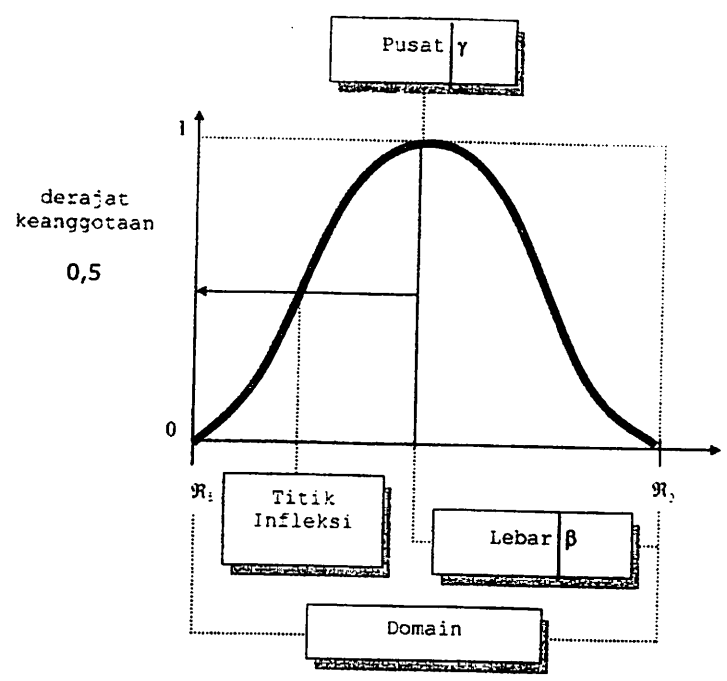
Gambar 2.10 Karakteristik Fungsi Kurva-S Dalam Bentuk Skema

6) Representasi Kurva Bentuk Lonceng (*Bell Curve*)

Untuk merepresentasikan bilangan fuzzy, biasanya digunakan kurva berbentuk lonceng. Kurva berbentuk lonceng ini terbagi atas 3 kelas yaitu: himpunan fuzzy PI, beta, dan Gauss. Perbedaan ketiga kurva ini terletak pada gradiennya.

- Kurva PI

Kurva PI berbentuk lonceng dengan derajat keanggotaannya 1 terletak pada pusat dengan domain (γ), dan lebar kurva (β).



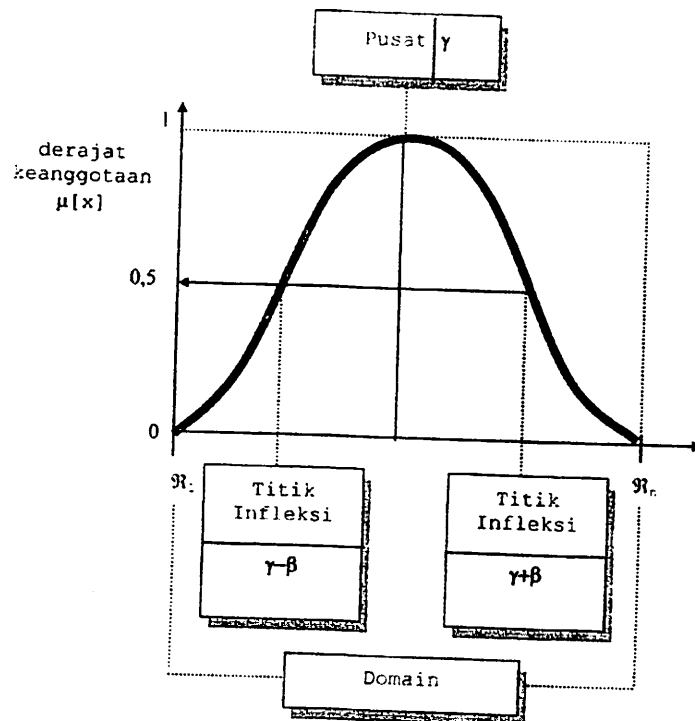
Gambar 2.11 Karakteristik Fungsional Kurva PI

Fungsi keanggotaan:

$$\Pi(x, \beta, \gamma) = \begin{cases} S\left(x; \gamma - \beta, \gamma - \frac{\beta}{2}, \gamma\right) & \rightarrow x \leq \gamma \\ 1 - S\left(x; \gamma, \gamma + \frac{\beta}{2}, \gamma + \beta\right) & \rightarrow x > \gamma \end{cases} \dots\dots\dots (2.7)$$

- Kurva BETA

Seperti halnya kurva PI, kurva BETA juga berbentuk lonceng namun lebih rapat. Kurva ini juga didefinisikan dengan 2 parameter yaitu nilai pada domain yang menunjukkan pusat kurva (γ), dan setengah lebar kurva (β).



Gambar 2.12 Karakteristik Fungsional Kurva BETA

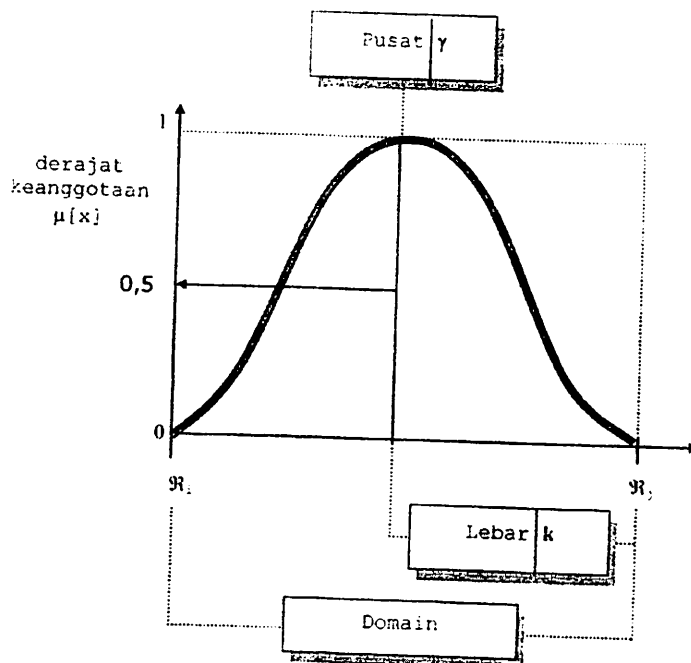
Fungsi keanggotaan:

$$B(x; \gamma, \beta) = \frac{1}{1 + \left(\frac{x - \gamma}{\beta}\right)^2} \dots \dots \dots (2.8)$$

Salah satu perbedaan mencolok kurva BETA dari kurva PI adalah fungsi keanggotaannya akan mendekati nol hanya jika nilai (β) sangat besar.

- Kurva GAUSS

Jika kurva PI dan kurva BETA menggunakan 2 parameter yaitu (γ) dan (β), kurva GAUSS juga menggunakan (γ) untuk menunjukkan nilai domain pada pusat kurva, dan (k) yang menunjukkan lebar kurva.



Gambar 2.13 Karakteristik Fungsional GAUSS

Fungsi keanggotaan:

$$G(x; k, \gamma) = e^{-k(\gamma-x)^2} \dots\dots\dots (2.9)$$

2.1.4 Operator Dasar Zadeh Untuk Operasi Himpunan Fuzzy ^[4]

Seperti halnya himpunan konvensional, ada beberapa operasi yang didefinisikan secara khusus untuk mengkombinasikan dan memodifikasi himpunan fuzzy. Nilai keanggotaan sebagai hasil dari operasi 2 himpunan sering dikenal dengan nama *fire strength* atau α -predikat. Ada 3 operator dasar yang diciptakan oleh Zadeh, yaitu :

1) Operator AND

Operator ini berhubungan dengan operasi interseksi pada himpunan. α -predikat sebagai hasil operasi dengan operator AND diperoleh dengan mengambil nilai keanggotaan terkecil antar elemen pada himpunan-himpunan yang bersangkutan.

$$\mu_{A \cap B} = \min(\mu_A(x), \mu_B(y)) \dots\dots\dots (2.10)$$

2) Operator OR

Operator ini berhubungan dengan operasi union pada himpunan. α -predikat sebagai hasil operasi dengan operator OR diperoleh dengan mengambil nilai keanggotaan terbesar antar elemen pada himpunan-himpunan bersangkutan.

$$\mu_{A \cup B} = \max(\mu_A(x), \mu_B(y)) \dots\dots\dots (2.11)$$

3) Operator NOT

Operator ini berhubungan dengan operasi komplemen pada himpunan. α -predikat sebagai hasil operasi dengan operator NOT diperoleh dengan mengurangkan nilai keanggotaan elemen pada himpunan yang bersangkutan dari 1.

$$\mu_{A'} = 1 - \mu_A(x) \dots\dots\dots (2.12)$$

2.1.5 Fungsi Implikasi ^[4]

Tiap – tiap aturan (proposisi) pada basis pengetahuan fuzzy akan berhubungan dengan suatu relasi fuzzy. Bentuk umum dari aturan yang digunakan dalam fungsi implikasi adalah:

$$\text{IF } x \text{ is } A \text{ THEN } y \text{ is } B \dots\dots\dots (2.13)$$

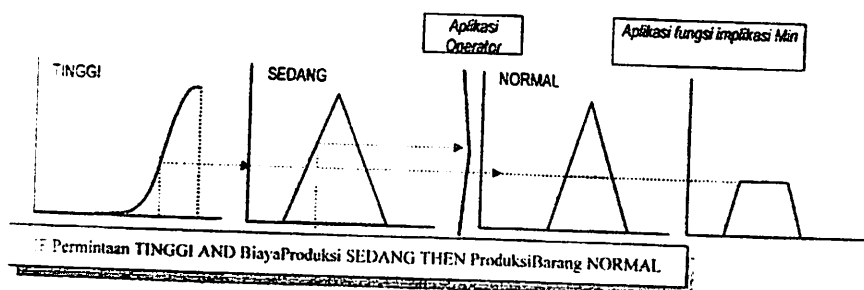
Dengan x dan y adalah saklar, dan A dan B adalah himpunan fuzzy. Proposisi yang mengikuti IF disebut sebagai anteseden, sedangkan proposisi yang mengikuti THEN disebut sebagai konsekuensi. Proposisi ini dapat diperluas dengan menggunakan operator fuzzy, seperti:

$$\text{IF } (x_1 \text{ is } A_1) \text{ o } (x_2 \text{ is } A_2) \text{ o } (x_3 \text{ is } A_3) \text{ o } \dots\dots \text{ o } (x_N \text{ is } A_N) \text{ THEN } y \text{ is } B \dots\dots\dots (2.14)$$

dengan o adalah operator (misal: OR atau AND)

Secara umum ada 2 fungsi implikasi yang dapat digunakan, yaitu:

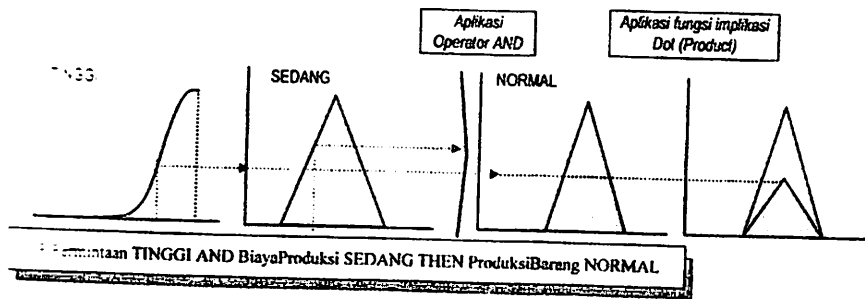
- 1) Min (*minimum*). Fungsi ini akan memotong *output* himpunan fuzzy. Gambar 2.14 menunjukkan salah satu contoh penggunaan fungsi min.



Gambar 1.26 Fungsi Implikasi: MIN

Gambar 2.14 Fungsi Implikasi: MIN

- 2) Dot (*product*). Fungsi ini akan menskala *output* himpunan fuzzy. Gambar 2.15 menunjukkan salah satu contoh penggunaan fungsi dot.



Gambar 2.15 Fungsi Implikasi: DOT

2.1.6 Alasan Digunakannya Logika Fuzzy ^[4]

Menurut Cox (1994), ada beberapa alasan mengapa orang menggunakan logika fuzzy, antara lain:

- 1) Konsep logika fuzzy mudah dimengerti. Karena logika fuzzy menggunakan dasar teori himpunan, maka konsep matematis yang mendasari penalaran fuzzy tersebut cukup mudah untuk dimengerti.
- 2) Logika fuzzy sangat fleksibel, artinya mampu beradaptasi dengan perubahan-perubahan, dan ketidakpastian yang menyertai permasalahan.
- 3) Logika fuzzy memiliki toleransi terhadap data yang tidak tepat. Jika diberikan sekelompok data yang cukup homogen, dan kemudian ada beberapa data yang "eksklusif", maka logika fuzzy memiliki kemampuan untuk menangani data eksklusif tersebut.
- 4) Logika fuzzy mampu memodelkan fungsi-fungsi nonlinear yang sangat kompleks.
- 5) Logika fuzzy dapat membangun dan mengaplikasikan pengalaman – pengalaman pakar secara langsung tanpa harus melalui proses pelatihan. Dalam hal ini, sering dikenal dengan nama *Fuzzy Expert Systems* menjadi bagian terpenting.
- 6) Logika fuzzy dapat bekerjasama dengan teknik-teknik kendali secara konvensional. Hal ini umumnya terjadi pada aplikasi di bidang teknik mesin maupun teknik elektro.
- 7) Logika fuzzy didasarkan pada bahasa alami. Logika fuzzy menggunakan bahasa sehari-hari sehingga mudah dimengerti.

2.1.7 Jenis – Jenis dari Fuzzy logic ^[4]

Fuzzy Logic dapat dikelompokkan menjadi beberapa jenis, yaitu:

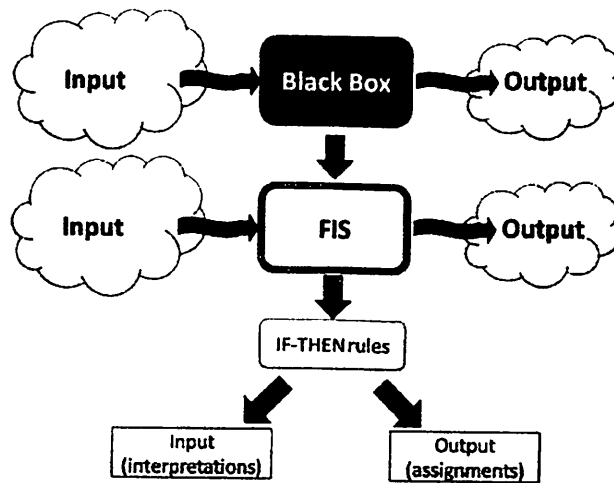
- 1) Fuzzy Inference System
Berisi metode - metode untuk melakukan inferensi fuzzy, antara lain Metode Tsukamoto, Metode Mamdani, dan Metode Sugeno.
- 2) Fuzzy Clustering
Berisi pengclusteran dengan konsep himpunan fuzzy dengan menggunakan Metode Fuzzy C-Means, Metode *Subtractive Clustering*, serta bagaimana membentuk *Fuzzy Inference System* melalui *Fuzzy Subtractive Clustering*.
- 3) Fuzzy Database
Berisi bagaimana membangun basis data dengan *query fuzzy* dengan Model Tahani, dan bagaimana membangun basis data dengan data-data fuzzy menggunakan Model Umano.
- 4) Fuzzy Quantification Theory
Berisi bagaimana mengendalikan data – data kualitatif dengan menggunakan teori himpunan fuzzy, meliputi *Fuzzy Quantification Theory I & II*.
- 5) Fuzzy Associative Memory (FAM)
Berisi bagaimana memetakan beberapa pasangan himpunan fuzzy yang membentuk aturan dalam *fuzzy associative memory*, yang meliputi FAM sederhana dengan *Hebb FAM*, dan *superimposing FAM rules*.
- 6) Fuzzy Linear Programming
Berisi bagaimana menyelesaikan *linear programming* dengan batasan fuzzy, dan *fuzzy multiobjective linear programming*.
- 7) Fuzzy Integer Transportation Problem
Berisi bagaimana menyelesaikan *integer transportation problem* dengan batasan fuzzy.

2.2 Fuzzy Inference System ^[3]

Biasanya seorang operator atau pakar memiliki pengetahuan tentang cara kerja dari sistem yang bisa dinyatakan dalam sekumpulan IF-THEN rule. Dengan melakukan fuzzy inference, pengetahuan tersebut bisa ditransfer ke dalam perangkat lunak yang selanjutnya memetakan suatu input menjadi output berdasarkan IF-THEN rule yang diberikan. Sistem fuzzy yang dihasilkan disebut *Fuzzy Inference System* (FIS). Dalam

FIS terdapat tiga metode yaitu Metode Tsukamoto, Metode Mamdani, dan Metode Sugeno.

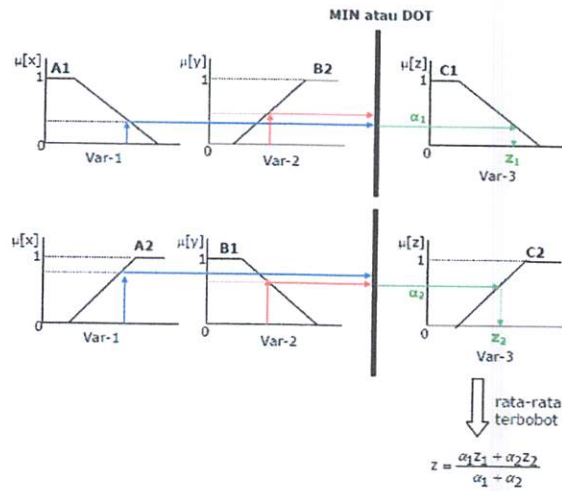
FIS telah berhasil diaplikasikan dalam berbagai bidang, seperti kontrol otomatis, klasifikasi data, analisis keputusan, dan sistem pakar. Karena kemampuannya yang fleksibel untuk bisa diterapkan di berbagai bidang, FIS sering disebut dengan nama lain, seperti *fuzzy-rule-based system*, *fuzzy expert system*, *fuzzy modeling*, *fuzzy logic controller*, dan tidak jarang cukup dengan *fuzzy system*.



Gambar 2.16 Konsep Umum Kronologi Proses Pembangunan FIS

2.2.1 Metode Tsukamoto ^[4]

Metode tsukamoto merupakan perluasan dari penalaran monoton. Pada metode tsukamoto, setiap konsekuensi pada aturan yang berbentuk IF-Then harus direpresentasikan dengan suatu himpunan fuzzy dengan fungsi keanggotaan yang monoton. Sebagai hasilnya, *output* hasil inferensi dari tiap-tiap aturan diberikan secara tegas (*crisp*) berdasarkan α -predikat (*fire strength*). Hasil akhirnya diperoleh dengan menggunakan rata-rata terbobot.

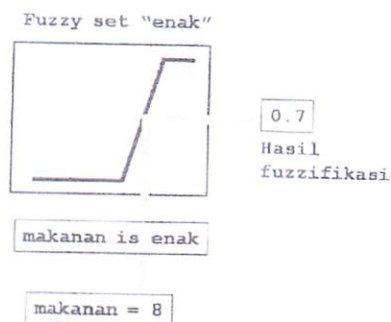


Gambar 2.17 Inferensi dengan Menggunakan Metode Tsukamoto

Fuzzy inference system terdiri dari empat proses yaitu fuzzifikasi, inferensi, komposisi dan defuzzifikasi. Pada metode tsukamoto hanya menggunakan tiga proses dimana komposisi system terdiri dari empat proses yaitu fuzzifikasi, inferensi, komposisi dan digabung menjadi satu proses.

2.2.2 Fuzzifikasi

Dalam proses ini, fungsi keanggotaan yang telah ditetapkan untuk setiap variabel input akan diterapkan untuk mendapatkan nilai aktual yang akan menentukan tingkat kebenaran tiap premis dari aturan. Tingkat kebenaran sebuah premis dari aturan sering disebut alpha predikat (α -predikat). Jika sebuah premis rule memiliki tingkat kebenaran yang tidak sama dengan 0, maka aturan akan dijalankan (Horstkotte, 2000).



Gambar 2.18 Fuzzifikasi

Gambar 2.18 merupakan contoh ketika ditanya seberapa enak makanan maka pada gambar tersebut menunjukkan rasa makanan (diberi skor dari 0 sampai 10) di

fuzzifikasi dalam fuzzy set enak menggunakan fungsi keanggotaan. Dalam kasus ini makanan diberi skor 8, dan setelah di fuzzifikasi akan didapat makanan masuk dalam fuzzy set enak dengan derajat keanggotaan $\mu = 0,7$. Dalam bentuk kalimat bisa dikatakan: “makanan anda adalah enak dengan derajat 0.7”. Dengan cara ini tiap masukan FIS difuzzifikasikan dalam semua fuzzy set yang didefinisikan dalam tiap IF-THEN rule (Naba, 2009).

2.2.3 Inferensi ^[10]

Nilai kebenaran dari premis akan dihitung, kemudian diterapkan pada konklusi pada setiap aturan menggunakan fungsi implikasi yang dipilih. Outputnya merupakan satu subset fuzzy pada tiap aturan. (Horstkotte, 2000).

2.2.4. Komposisi dan Defuzzifikasi ^[4]

Berbeda dengan metode inferensi Mamdani di mana komposisi dan defuzzifikasi dilakukan secara terpisah, pada metode Tsukamoto kedua proses ini digabung menjadi satu proses. Hasil akhirnya diperoleh dengan menggunakan rata-rata terbobot dengan rumus sebagai berikut :

$$Z = \frac{\alpha_1 z_1 + \alpha_2 z_2 + \dots + \alpha_n z_n}{\alpha_1 + \alpha_2 + \dots + \alpha_n} \dots\dots\dots (2.15)$$

2.3 Borland Delphi 7 ^[1]

Delphi adalah sebuah bahasa pemrograman visual di lingkungan windows (underwindows) yang menggunakan bahasa pascal sebagai Compiler. Keberadaan bahasa pemrograman Delphi tidak bisa dipisahkan dari bahasa Turbo Pascal yang diluncurkan pada tahun 1983 oleh Borland International Incorporation. Turbo Pascal memang dirancang untuk dijalankan pada operasi DOS (Disk Operating System) yang merupakan sistem operasi yang banyak digunakan pada saat ini. Seiring dengan perkembangan jaman, dimana sistem operasi mulai bergeser ke sistem operasi windows, maka Borland international merilis Turbo Pascal for Windows yang dijalankan dibawah sistem operasi windows 3.X.

2.3.1 Sejarah Borland Delphi ^[7]

Pada tahun 1992 muncul bahasa pemrograman baru bernama Borland Pascal 7 yang merupakan penggabungan dari Turbo Pascal dan Turbo pascal for Windows.

Namun ternyata bahasa pemrograman baru tersebut ternyata masih sulit digunakan. Trend penggunaan bahasa pemrograman visual untuk membangun sebuah aplikasi telah mendorong Borland membuat bahasa pemrograman baru pada tahun 1995 diperkenalkan kepengguna komputer sebuah bahasa pemrograman visual yang berbasis bahasa pascal. Bahasa pemrograman baru tersebut diberi nama Borland Delphi setahun kemudian versi kedua dari Delphi dilempar ke pasaran.

Penggunaan Delphi dapat mempersingkat waktu pemrograman karena anda tidak perlu lagi menuliskan kode program yang rumit dan panjang untuk menggambar, meletakkan dan mengatur komponen. Selain itu anda dapat menyusun aplikasi yang lebih interaktif. Delphi menyediakan cukup banyak pilihan komponen interface aplikasi, antara lain berupa tombol menu, drop down, ataupun menu pop up, kotak text, radio button, check box, dan sebagainya. Bahkan ada berbagai macam komponen Skin tampilan yang beragam yang disediakan oleh beberapa vendor lainnya, seperti SUIPack, Sxskincomponents, dan lain sebagainya. Anda tinggal memilih komponen yang dibutuhkan dengan klik mouse, mengatur tampilannya kemudian menuliskan sedikit kode program, maka aplikasi anda siap dijalankan.

Delphi 7.0 versi terbaru yang dikeluarkan oleh Borland, memiliki support yang sangat tinggi terhadap data base – data base yang sudah terkenal (seperti MS Access, Paradox, Foxpro, Dbase, Oracce, dan lain sebagainya) dan dilengkapi dengan objek-objek yang baru sehingga memudahkan pembuatan database maupun program lainnya (Game, Utility dan lainnya).



Gambar 2.19 Gambar Tampilan Borland Delphi 7.0

2.3.2 Kelebihan Borland Delphi 7.0 ^[7]

Borland Delphi 7.0 merupakan pilihan bagi sebagian kalangan programmer untuk membuat aplikasi. Hal ini disebabkan kelebihan yang ada pada Borland Delphi 7.0, berikut ini beberapa kelebihan Borland Delphi 7.0 antara lain:

- Berbasis Objek Oriented programming, seperti bagian yang ada pada program dipandang sebagai suatu objek yang mempunyai sifat – sifat yang dapat diubah dan diatur. Sehingga kita dapat membuat tampilan sebuah program dengan desain kita sendiri tanpa harus membuat coding yang panjang.
- Suatu file EXE. Setelah anda merancang program pada IDE Delphi akan mengkomplikasinya pada suatu file executable tunggal. Program yang anda buat dapat langsung didistribusikan dan dijalankan pada komputer lain tanpa perlu menyertakan file lain. Kecuali file yang berektensi “.exe atau aplikasi tersebut membutuhkan file lain seperti database, koneksi atau file pendukung lainnya, ini merupakan kelebihan yang sangat berarti.
- Memiliki sarana pengembangan yang bersifat grafis (Visual).
- Berorientasi Objek (object oriented)
- Dapat bekerja di dalam sistem operasi Windows.
- Mampu memanfaatkan program aplikasi berbasis Windows seperti grafis, multimedia, internet, multitasking, dan sebagainya.

Borland Delphi berbasiskan *Object Oriented Programming* (OOP) dan dikembangkan dengan basis Visual yang berarti menggunakan sarana grafis untuk mengembangkannya. Borland Delphi berorientasi pada objek – objek yang dipisah – pisah, sehingga disebut pemrograman *Object Oriented Programming* karena kode – kode program letaknya tersebar di dalam modul – modul (objek – objek) yang terpisah – pisah.

2.3.3 Kebutuhan Sistem ^[1]

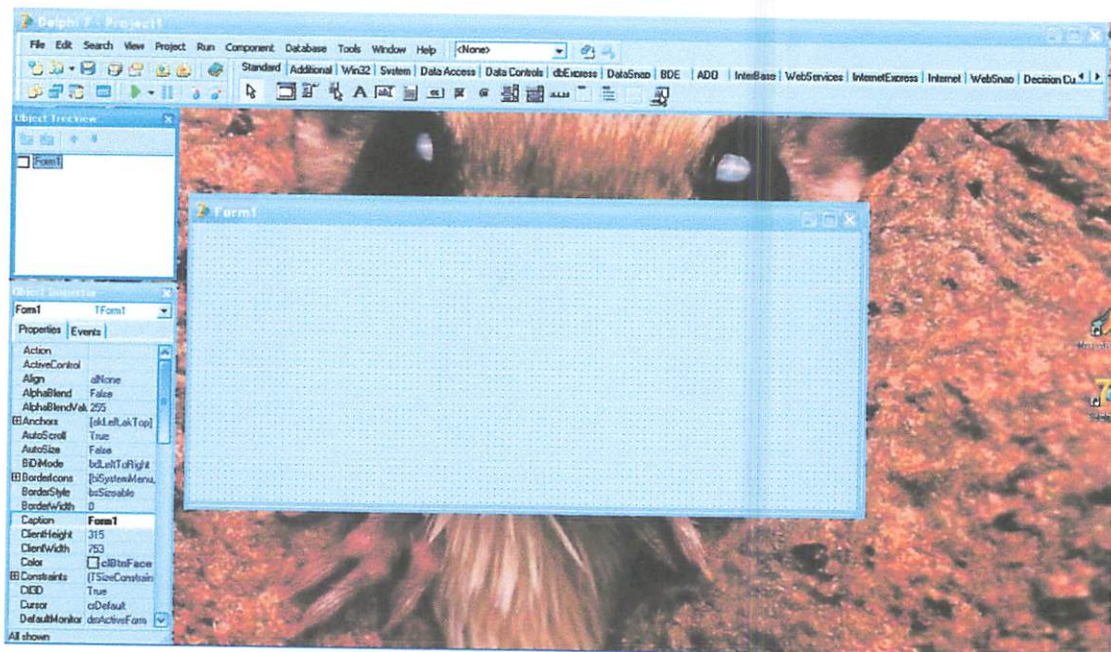
Agar delphi versi 7.0 ini dapat dioperasikan dengan baik anda membutuhkan perangkat keras dan perangkat lunak dengan spesifikasi tertentu. Delphi memerlukan perangkat keras (hardware) dengan ruang hard disk dan memori yang relatif besar untuk pembuatannya tetapi untuk pemakaiannya atau menggunakan hasil file aplikasinya tidaklah membutuhkan spesifikasi yang besar, tergantung besar

kecilnya program yang kita terapkan pada komputer tersebut. Spesifikasi minimal yang dibutuhkan antara lain:

- Processor Pentium 233 MHz atau yang lebih tinggi.
- Sistem operasi windows XP, windows 2000 atau windows 98.
- Memori membutuhkan RAM 664 MB untuk edisi Architect, Enterprise, dan propesional, kecuali untuk personal 32 MB disarankan 128 MB.
- Untuk hard disk minimal size yang kosong sebesar 500 Megabyte.
- Mouse, keyboard, dll.

2.3.4 IDE Delphi ^[7]

Delphi memiliki lingkungan pemrograman terpadu (IDE, Integrated Development Environment). Dengan IDE semua yang diperlukan dalam pengembangan dalam kondisi normal, semuanya telah tersedia. Jendela IDE Delphi 7.0 telah didesain sedemikian oleh Borland sehingga bersifat intuitif.



Gambar 2.20 Bagian-bagian IDE

Keterangan:

a) Jendela Utama

Pada jendela utama terdapat menu – menu sebagaimana menu aplikasi windows umumnya, toolbar yang merupakan langkah cepat dari beberapa menu dan Component Pallete, yaitu gudang komponen yang akan digunakan untuk membuat aplikasi.

Componen Pallette terdiri dari beberapa halaman, diantaranya adalah standart, additional dan system. Halaman tersebut berisi komponen – komponen yang sesuai dengan namanya. Misalnya halaman internet menampilkan komponen – komponen yang berkaitan dengan pengembangan aplikasi internet.

b) Object Treeview

Fasilitas ini hadir bersama Delphi 6.0 dan 7.0, jadi Delphi 5.0 kebawah belum memilikinya. Object Treeview menampilkan daftar komponen yang digunakan dalam pengembangan aplikasi sesuai dengan penempatannya.

c) Object Inspector

Object inspector digunakan untuk mengatur properti dan event suatu komponen. Semua komponen dapat diubah perilakunya menggunakan kehebatan object inspector. Akan tetapi anda tidak dapat mengubah langsung properti – properti yang tidak ditampilkan kecuali melalui penulisan kode program.

d) Form Designer

Semua program windows memiliki form dan form adalah komponen utama pengembangan aplikasi. Form designer adalah tempat melekatnya komponen yang lain. Artinya form adalah tempat komponen – komponen lain diletakkan.

2.4 My SQL

2.4.1 Sepintas Mengenai MySQL ^[8]

MySQL merupakan hasil buah pikiran dari Michael Wonty, David Axmark, dan Allan Larson dimulai tahun 1995. Mereka kemudian mendirikan perusahaan bernama MySQL-AB di Swedia. MySQL merupakan sebuah sistem manajemen database relasi (relational data system) yang bersifat terbuka (open source). Terbuka dalam artian MySQL boleh di download oleh siapa saja, baik versi asli (source code program) maupun versi binernya (program) yang bisa digunakan secara (relatif) garis.

Tujuan awal ditulisnya program MySQL adalah untuk mengembangkan aplikasi web yang akan digunakan oleh salah satu klien MySQL-AB. Memang pada saat itu, MySQL-AB adalah sebuah perusahaan konsultan database dan pengembang software (masih menggunakan nama perusahaan TeX Data Konsult AB). Situs internet MySQL-AB dapat dikunjungi di <http://www.mysql.com>

MySQL versi 1.0 dirilis pada Mei 1996 dan penggunaanya hanya terbatas di kalangan intern saja. Pada bulan Oktober 1996, MySQL versi 3.11.0 dirilis ke

masyarakat luas di bawah lisensi, artinya “Terbuka tapi Terbatas”. Dengan lisensi tersebut, maka siapapun boleh melihat program aslinya dan menggunakan server MySQL secara gratis untuk kegiatan – kegiatan non-komersial dan untuk kegiatan komersial, maka kita harus membayar lisensi tersebut.

Pada bulan Juni 2000, MySQL-AB mengumumkan bahwa MySQL versi 3.23.19 dinyatakan sebagai General Public License (GPL). Dengan lisensi GPL tersebut, maka siapa pun boleh melihat program aslinya dan menggunakan program executable-nya dengan cuma – cuma. Dan bila kita memodifikasi pada program aslinya, maka program modifikasi tersebut harus dilepas di bawah lisensi GPL juga.

Pada awalnya MySQL hanya berjalan di atas platform sistem operasi UNIX, bahkan hampir semua distro UNIX yang beredar saat ini telah memasukkan program MySQL sebagai database management system standarnya. Saat ini MySQL dapat pula berjalan di atas sistem operasi lainnya, seperti Microsoft Windows, Solaris, FreeBSD, Mac OS X, HP UX, IBM, AIX, SCO UNIX, SG1 Irix, Dec: OSIF, dan sebagainya.

MySQL memiliki kinerja, kecepatan proses, dan ketangguhan yang tidak kalah dibanding aplikasi database manajemen sistem besar lainnya yang komersial, seperti, ORACLE, Sybase, Unix dan sebagainya. Dengan Open Source, MySQL berkembang menjadi salah satu sistem manajemen database relasi terkemuka yang diakui secara internasional.

2.4.2 Alasan memilih MySQL ^[8]

Ada beberapa pertimbangan memilih MySQL sebagai program aplikasi database manajemen sistem kita diantaranya adalah:

- **Kecepatan**

Berdasarkan hasil pengujian, MySQL memiliki kecepatan paling baik dibanding database server lainnya. Hasil pengujian ini dapat kita lihat dalam <http://www.mysql.com/information/becmarks.html>. Kemudian, dengan adanya fitur tambahan query caching pada MySQL 4.0, kinerja query secara umum naik rata – rata 200% dari kinerja biasanya.

- **Mudah digunakan**

Perintah – perintah dan aturan – aturan pada MySQL dan proses instalasinya relatif mudah digunakan.

- **Open Source**

Dengan konsep ini siapa pun dapat berpartisipasi untuk mengembangkan MySQL dan hasil pengembangan itu dirilis untuk konsumsi umum atau untuk komunitas Open Source, di bawah lisensi GPL.

- **Kapabilitas.**

MySQL telah diujicobakan untuk mengelola database dengan jumlah 50 juta record, telah diujicobakan untuk mengelola 60.000 tabel dengan jumlah record 5.000.000.000. Dimana hasil uji coba tersebut berjalan dengan baik. Selain itu MySQL mendukung penggunaan index hingga 32 buah index pertabelnya, sedangkan setiap indexnya terdiri dari 1 hingga 16 kolom kriteria.

- **Replika Data**

Dengan adanya fasilitas replika data ini, anda dapat mempunyai beberapa database bayangan pada beberapa server 'anak' lainnya yang berasal dari satu database induk sehingga akan meningkatkan kinerja dan kecepatan MySQL.

- **Biaya Rendah (relatif gratis)**

Kita dapat menggunakan MySQL tanpa harus memikirkan biaya lisensi selama kita mengikuti konsep *Open Source / GNU Public Licences*.

- **Konektifitas dan keamanan.**

MySQL mendukung dan menerapkan sistem keamanan dan izin akses tingkat lanjut (*advanced permissions and security system*), termasuk dukungan pengamanan dengan cara pengacakan lapisan data (*SSL transport layer encryption*). MySQL memiliki beberapa lapisan sekuritas seperti level subnetmask, nama host, dan user dengan system perizinan yang mendetail serta password terenkripsi dan akan memberikan jaminan keamanan. MySQL dapat melakukan koneksi dengan client menggunakan protocol TCP/IP, Unix Soket (Unix), atau Named Pipes (NT). Localisation MySQL dapat mendeteksi pesan kesalahan (error code) pada client dengan menggunakan lebih dari dua puluh bahasa. Meskipun demikian, bahasa Indonesia belum termasuk didalamnya.

- **Fleksibilitas.**

MySQL mendukung perintah – perintah ANSI SQL 99 dan beberapa perintah database alternatif lainnya sehingga memudahkan untuk beralih dari dan ke MySQL.

- **Portability.**

MySQL dapat dijalankan pada beberapa sistem operasi yang berbeda, seperti Linux, Microsoft Windows, FreeBSD, Sun Solaris, IBM's AIX, Mac OS X, HP-UX, AIX, QNX, Novell NetWare, SCO OpenUnix, SGI Irix, dan masih banyak lagi.

- **Multiuser.**

MySQL dapat digunakan oleh beberapa user dalam waktu yang bersamaan tanpa mengalami masalah atau konflik. Hal ini memungkinkan sebuah database server MySQL dapat diakses client secara bersamaan.

- **Performance Tuning.**

MySQL memiliki kecepatan yang menakjubkan dalam menangani query sederhana, dengan kata lain dapat memproses lebih banyak SQL per satuan waktu.

- **Column Types**

MySQL memiliki tipe kolom yang sangat kompleks, seperti signed/unsigned integer, float, double, char, varchar, text, blob, date, time, datetime, year, set serta enum.

- **Command and Function.**

MySQL memiliki operator dan fungsi secara penuh yang mendukung perintah SELECT dan WHERE dalam query.

- **Interface**

MySQL memiliki interface (antar muka) terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API (Application Programming Interface).

- **Client and Tools.**

MySQL dilengkapi dengan berbagai tool yang dapat digunakan untuk administrasi database, dan pada setiap tool yang ada disertai petunjuk online.

- **Struktur Tabel.**

MySQL memiliki struktur tabel yang lebih fleksibel dalam menangani ALTER TABLE, dibandingkan database lainnya semacam PostgreSQL ataupun Oracle.

2.5 Perangkat Analisa dan Perancangan ^[5]

2.5.1 Pemodelan Data

Model data adalah sekumpulan cara / peralatan / *tool* untuk mendeskripsikan data – data, hubungan satu sama lain, semantiknya serta batasan konsistensi. Ada dua model data, yaitu: *Entity Relationship Diagram* (ERD) dan model relasional. Keduanya menyediakan cara untuk mendeskripsikan perancangan basis data pada perangkat logika.

Model ERD atau *Conceptual Data Model* (CDM) adalah model yang dibuat berdasarkan anggapan bahwa dunia nyata terdiri dari koleksi obyek – obyek dasar yang dinamakan entitas (*entity*) serta hubungan (*relationship*) antara entitas – entitas itu.

Model relasional atau *Physical Data Model* (PDM) adalah model yang menggunakan sejumlah tabel untuk menggambarkan data serta hubungan antara data – data tersebut. Setiap tabel mempunyai sejumlah kolom di mana setiap kolom memiliki nama yang unik.

Didalam ERD atau CDM maupun PDM, relasi (hubungan) setiap entitas mempunyai derajat hubungan (kardinalitas) yang menunjukkan jumlah maksimum entitas yang dapat berelasi dengan entitas lain. Relasi kardinalitas yang terjadi di antara dua himpunan entitas dapat berupa:

1) 1 ke 1 (*one to one*)

Setiap entitas pada suatu himpunan entitas berhubungan paling banyak satu entitas pada himpunan entitas yang lain, begitu juga sebaliknya.

2) 1 ke N (*One to many*)

Setiap entitas berhubungan dengan banyak entitas pada himpunan entitas yang lain, tetapi tidak sebaliknya.

3) N ke 1 (*many to one*)

Setiap entitas berhubungan dengan paling banyak satu entitas pada himpunan entitas yang lain, tetapi tidak sebaliknya.

4) N ke N (*many to many*)

Setiap entitas pada suatu himpunan dapat berhubungan dengan entitas pada himpunan entitas yang lain, demikian sebaliknya.

2.5.2 Metode Pengembangan Sistem

Model air terjun (*waterfall*) merupakan salah satu metode pengembangan sistem yang telah diperoleh dari proses engineering lainnya. Model ini menawarkan cara pembuatan perangkat lunak secara lebih nyata. Langkah – langkah yang penting dalam model ini adalah:

- **Penentuan dan Analisa Spesifikasi**
Jasa, kendala dan tujuan dihasilkan dari konsultasi dengan pengguna sistem. Kemudian semuanya itu dibuat dalam bentuk yang dapat dimengerti oleh user dan staf pengembang.
- **Desain sistem dan perangkat lunak**
Proses desain sistem membagi kebutuhan – kebutuhan menjadi sistem perangkat lunak atau perangkat keras. Proses tersebut menghasilkan sebuah arsitektur sistem keseluruhan. Desain perangkat lunak termasuk menghasilkan fungsi sistem perangkat lunak dalam bentuk yang mungkin ditransformasi ke dalam satu atau lebih program yang dapat dijalankan.
- **Integrasi dan uji coba sistem**
Unit program diintegrasikan dan diuji menjadi sistem yang lengkap untuk meyakinkan bahwa persyaratan perangkat lunak telah dipenuhi. Setelah diujicoba, sistem disampaikan ke customer.
- **Operasi dan pemeliharaan**
Normalnya, ini adalah fase yang terpanjang. Sistem dipasang dan digunakan. Pemeliharaan termasuk pembetulan kesalahan yang tidak ditemukan pada langkah sebelumnya. Perbaikan implementasi unit sistem dan peningkatan jasa sistem sebagai kebutuhan baru ditemukan.

2.6 Penyakit Infeksi

Penyakit infeksi adalah sebuah penyakit yang disebabkan oleh sebuah gen biologi (seperti virus, bakteri, atau parasit), bukan disebabkan faktor fisik (seperti luka bakar) atau kimia (seperti keracunan). (Wikipedia, 2011)

Setiap hari, dalam kehidupan penuh dengan bahaya penyakit infeksi potensial. Mulai dari teman yang sedang pilek sampai ayam mentah di dapur. Kuman kini semakin umum dan berada dimana – mana. Sehingga informasi bagaimana cara melindungi diri dari kuman yang menyebabkan penyakit infeksi dan bagaimana kuman menyebar

merupakan sesuatu yang sangat penting sehingga resiko infeksi bisa diminimalisir. (Medicastore, 2011)

2.6.1 Penyakit Infeksi Menyebar Melalui Kontak

Penyakit infeksi menyebar melalui dua jenis kontak, yaitu kontak langsung dan kontak tidak langsung (Medicastore, 2011)

1) Kontak Langsung

Cara mudah untuk terkena infeksi adalah melalui kontak langsung dengan seseorang yang mengidapnya. “Seseorang” ini bisa manusia, hewan bayi yang belum lahir, ibunya. Ada tiga cara penyebaran penyakit infeksi melalui kontak langsung, antara lain:

- *Manusia ke manusia.*

Penyakit infeksi paling umum menyebar melalui perpindahan langsung bakteri, virus atau kuman lain dari satu orang ke orang lainnya. Hal ini bisa muncul ketika individu dengan bakteri atau virus tersebut menyentuh, batuk atau mencium seseorang yang tidak terinfeksi. Kuman ini juga menyebar melalui pertukaran cairan tubuh melalui kontak seksual atau transfuse darah.

- *Hewan ke manusia.*

Hewan peliharaan dapat membawa banyak kuman. Digigit atau dicakar oleh hewan yang terinfeksi bisa membuat anda sakit dan dalam waktu yang ekstrem bahkan menyebabkan kematian. Menangani kotoran hewan juga bisa berbahaya. Contohnya, anda bisa terinfeksi toksoplasmosis ketika membersihkan kotoran kucing peliharaan anda, terutama jika anda sedang hamil.

- *Ibu ke janin.*

Seorang wanita hamil dapat menularkan kuman yang dapat menyebabkan infeksi penyakit pada bayi yang belum lahir. Kuman dapat melewati plasenta, seperti kasus ibu terinfeksi Streptokokus B.

2) Kontak Tidak Langsung

Organisme penyebab penyakit dapat juga ditularkan melalui kontak tidak langsung. Banyak kuman dapat menempel pada benda tak bernyawa, seperti atas meja, pegangan pintu, kran.

Ketika anda menyentuh pegangan pintu yang sama dengan yang digenggam oleh seseorang yang sakit flu atau pilek contohnya, anda mengambil kuman yang ditinggalkannya. Jika kemudian anda menyentuh mata, mulut sebelum anda mencuci tangan, anda akan terinfeksi.

Beberapa infeksi muncul dari organisme yang secara alamiah hidup di lingkungan tapi bukan ditularkan melalui orang ke orang. Contohnya termasuk infeksi jamur seperti histoplasmosis atau blastomycosis.

2.6.2 Penyakit Infeksi Menyebar melalui udara

1) Penularan Tetesan

Ketika Anda batuk atau bersin, Anda mengeluarkan tetesan ke udara sekeliling Anda. Jika sakit karena pilek atau flu, tetesan ini mengandung kuman penyebab anda sakit. Penyebaran penyakit infeksi melalui cara ini disebut penyebaran tetesan atau penularan tetesan.

Tetesannya ini mampu bertahan sejauh 3 kaki karena biasanya terlalu besar untuk bertahan di udara untuk waktu yang lama. Bagaimana pun, jika tetesan dari orang yang terinfeksi kontak dengan mata, hidung, atau mulut Anda, Anda segera merasakan gejala penyakit. Lingkungan dalam ruangan yang penuh sesak dapat membuat kesempatan penularan tetesan.

2) Penularan partikel

Beberapa penyakit yang disebabkan oleh kuman melalui udara dalam partikel berukuran lebih kecil dari tetesan. Partikel kecil ini tersuspensi di dalam udara untuk waktu yang lama dan ikut bersama angin.

Jika anda bernapas dalam udara berisi virus, bakteri atau kuman lain, Anda dapat terinfeksi dan menunjukkan tanda dan gejala penyakit. Pilek disebabkan virus, influenza dan tuberkulosis merupakan beberapa tipe penyakit infeksi yang biasanya menyebar melalui udara, baik dalam bentuk partikel dan tetesan.

2.6.3. Penyakit Infeksi Menyebar Melalui Vektor & Pembawa

1) Gigitan dan Sengatan

Beberapa kuman menempel pada serangga seperti nyamuk, lalat, kutu, untuk berpindah dari satu tempat hidup ke tempat lainnya. Nyamuk bisa

membawa parasit malaria atau virus Nil Barat dan kutu Rusa dapat membawa bakteri yang menyebabkan penyakit Lyme.

Vektor penyebaran kuman terjadi ketika seekor serangga yang membawa kuman pada tubuhnya atau dalam saluran pencernaan atau menggigit Anda. Kuman bergerak ke dalam tubuh Anda dan membuat Anda sakit.

Kadang, kuman penyebab penyakit infeksi membutuhkan serangga untuk alasan biologis tertentu. Kuman menggunakan tubuh serangga untuk membelah diri yang diperlukan sebelum kuman dapat menginfeksi yang baru.

2) Kontaminasi Makanan

Cara lain kuman penyebab penyakit dapat menginfeksi Anda adalah melalui makanan dan air yang terkontaminasi. Kadang disebut penularan pembawa umum, mekanisme ini membuat kuman menyebar ke banyak orang melalui satu sumber.

Makanan sering merupakan pembawa yang menyebarkan kuman dan menyebabkan penyakit. Sebagai contoh, kontaminasi dengan *Escherichia coli* (*E. coli*). *E. Coli* adalah bakteri yang muncul dalam atau pada makanan tertentu. Seperti hamburger yang tidak matang atau buah dan sayuran yang tidak dicuci. Ketika Anda makan makanan terkontaminasi *E. Coli*, disebut juga sebagai keracunan makanan.

2.7. Probabilitas

Probabilitas menurut Supranto (1985) adalah merupakan suatu nilai untuk mengukur besarnya tingkat kemungkinan terjadinya suatu kejadian yang acak. Kejadian acak atau “*random event*” ialah suatu kejadian yang tidak dapat ditentukan dengan pasti sebelumnya. Hal ini dapat dilihat pada rumus:

$$P(X) = \frac{\text{Jumlah hasil yang menguntungkan kejadian}}{\text{Total jumlah kejadian}} \dots\dots\dots(2.16)$$

Probabilitas kejadian X dinyatakan sebagai $P(X)$, merupakan ratio jumlah waktu X yang terjadi pada total jumlah kejadian.

Contoh: Sebanyak 10 kasus diagnosa yang diajukan, apakah hasilnya sesuai dengan basis aturan dan ternyata 8 diantaranya sesuai. Maka $P(\text{Sesuai}) = 8/10 = 0.8$

Jika diekspresikan dalam persentase, cara menghitung probabilitas $\times 100$. Dengan demikian bisa dikatakan bahwa probabilitasnya adalah 80% (Suparman, 1991)



BAB III

METODOLOGI DAN PERANCANGAN

Pada bab metodologi dan perancangan ini akan dibahas metode, rancangan yang digunakan dan langkah-langkah yang dilakukan dalam pembuatan fuzzy inference system.

Tahapan pembuatannya adalah sebagai berikut:

- 1) Mempelajari metode yang digunakan pada sistem ini (logika fuzzy) dan objek penelitian (penyakit infeksi).
- 2) Menganalisa dan merancang sistem dengan menggunakan hasil pembelajaran pada tahap sebelumnya.
- 3) Membuat sistem berdasarkan analisis dan perancangan yang dilakukan.
- 4) Uji coba sistem.
- 5) Evaluasi hasil pengecekan yang dilakukan oleh sistem.

3.1 Pengumpulan Data

Sehubungan dengan objek penelitian, pengumpulan data ini diperoleh dari dokumen-dokumen yang mendukung dan berhubungan dengan penelitian, terdiri dari dokumen-dokumen, informasi-informasi lain yang berhubungan dengan penyakit infeksi.

3.2. Deskripsi umum sistem

Sistem yang dikembangkan adalah implementasi dari metode sistem inference fuzzy dalam bentuk aplikasi untuk diagnosa penyakit infeksi dengan metode tsukamoto. Sistem akan mendiagnosa atau mengolah input berupa gejala dan intensitasnya yang diberikan oleh *user* dan akan memberikan output berupa hasil diagnosa terhadap gejala-gejala tersebut.

Sebagaimana dijelaskan pada bab sebelumnya, sistem memiliki 2 bagian pokok, yaitu bagian pengembangan dan bagian konsultasi. Pada bagian pengembangan, user yang memiliki hak khusus (pakar/pengalih pengetahuan) dapat melakukan modifikasi pada basis pengetahuan, baik proses penambahan, pengubahan, maupun penghapusan pengetahuan.

Pada bagian konsultasi, user, baik pakar atau bukan, dapat melakukan proses konsultasi kepada sistem, dalam hal ini diagnosa penyakit infeksi, dengan terlebih dahulu memberikan input berupa gejala penyakit beserta intensitasnya. Setelah proses input selesai, sistem akan memberikan output berupa hasil diagnosa berdasarkan input yang diberikan user.

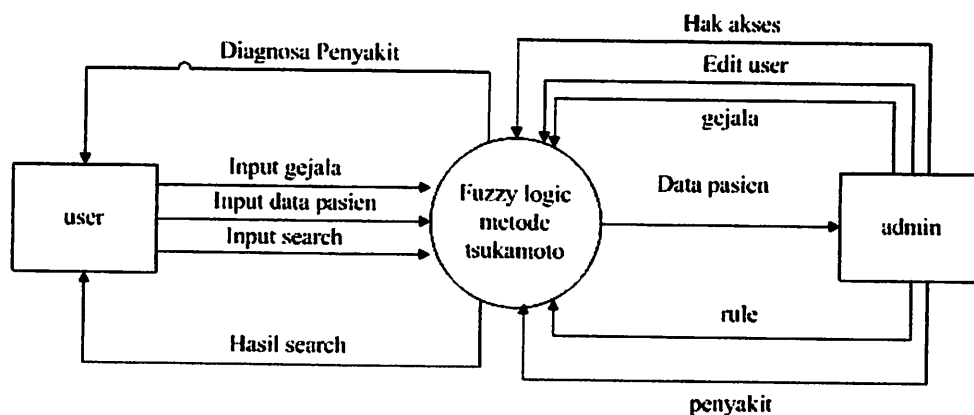
3.3. Perancangan Basis Penalaran

Pada proses pengembangan sistem Inference Fuzzy ini menggunakan basis penalaran atau inference system. Metode yang digunakan untuk membuat perancangan desain Fuzzy Inference System ialah sebagai berikut :

- 1) Pembuatan DFD
- 2) Pembuatan tabel keputusan.
- 3) Pembuatan tabel keputusan menjadi aturan dalam bentuk *if-then rule*.

3.3.1. DFD

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



Gambar 3.1 DFD level 0

Berdasarkan gambar 3.1 proses "Fuzzy Inference system untuk mendiagnosa penyakit infeksi pada manusia menggunakan metode tsukamoto" mempunyai data masukan berupa:

- Untuk User

User biasa tidak perlu memasukkan login password. User hanya memasukkan

1. Input User

Untuk memasukkan diagnosa gejala maka user akan memasukkan data user, parameter yang digunakan adalah :

a. ID

Berisi id user yang diberikan oleh admin.

b. Nama

Berisi nama user yang di inputkan oleh user.

c. Alamat

Berisi alamat user yang di inputkan oleh user.

d. Umur

Berisi umur user yang di inputkan oleh user.

2. Input Gejala

Setelah memasukkan data user maka dari sistem akan muncul gejala yang dapat di pilih oleh user.

3. Input Search

Input search diberikan oleh user.

• Untuk Admin

1. Hak Akses

Hak akses berupa login yang di berikan oleh admin untuk melakukan validasi login, parameter yang digunakan adalah:

a. Username

Berisi masukkan username yang diberikan oleh admin.

b. Password

Berisi masukkan password yang diberikan oleh admin.

2. Gejala

Berisi gejala-gejala yang diberikan oleh admin.

3. Penyakit

Berisi penyakit-penyakit dan cara pengobatannya serta pencegahan yang di berikan oleh admin.

4. Rule

Data rule yang di berikan oleh admin.

5. Edit User

Edit user di lakukan oleh admin.

Berdasarkan gambar 3.1 proses "Fuzzy Inference system untuk mendiagnosa penyakit infeksi pada manusia menggunakan metode tsukamoto" mempunyai data keluar berupa:

- Untuk user
 1. Diagnosa Prnyakit
Berisi Penyakit yang mungkin di derita user
 2. Hasil Search
Berisi informasi tentang penyakit baik berupa penyebab, pengobatan serta pencegahan.
- Untuk Admin
Data user
Berisi data-data yang di berikan oleh user.

3.3.2. Perancangan Tabel Pengetahuan

Berikut ini adalah tabel-tabel database yang digunakan dalam fuzzy inference system untuk mendiagnosa penyakit infeksi pada manusia dengan metode tsukamoto.

Tabel 3.1 Daftar Jenis Penyakit

Kode Penyakit	Nama penyakit
P01	Cacar
P02	Campak
P03	Demam Berdarah Dengue
P04	Demam Chikungunya
P05	Disentri
P06	Influenza
P07	Malaria
P08	TBC

Tabel 3.1 menunjukkan nama penyakit yang akan dimasukkan pada tabel penyakit. P menunjukkan ID penyakit pada tabel.

Tabel 3.2 Daftar Nama Gejala

Kode Gejala	Nama Gejala
G001	Sakit Kepala
G002	Nyeri Punggung
G003	Demam
G004	Lemah Badan
G005	Ruam Kulit

G006	Menggigil
G007	Mual
G008	Muntah
G009	Mata terasa panas
G010	Bersin – bersin
G011	Batuk
G012	Pilek
G013	Kejang – kejang
G014	Gangguan Perut
G015	Pendarahan
G016	Myalgia
G017	Diare
G018	Pegal-pegal
G019	Muka Merah
G020	Kulit kering
G021	Berkeringat pada malam hari
G022	Penurunan Berat Badan
G023	Nyeri sendi
G024	Sakit tenggorokan
G025	Nyeri Otot
G026	Lemas
G027	Tidak nafsu makan
G028	Panas Dingin

Tabel 3.2 menunjukkan gejala yang mungkin terjadi pada penyakit yang tercantum pada Tabel 3.1. G menunjukkan ID gejala pada tabel.

3.3.3. Tabel Keputusan

Tabel keputusan merupakan tabel yang menunjukkan semua kombinasi inputan dan hasilnya. Tabel keputusan bekerja dengan cara mengkombinasikan semua kondisi yang ada, dimana kondisi itu berisikan aturan-aturan (*rule*) yang disimpan dalam bentuk tabel pada suatu masalah sehingga dapat dipastikan tidak ada kemungkinan yang terlewat di dalam analisa logika terdapat masalah tersebut. Langkah perancangan tabel keputusan meliputi:

1) Menentukan jumlah aturan

Jumlah aturan untuk setiap rule set ditentukan oleh kondisi yang mengikuti dan jumlah kemungkinan nilai dari kondisi tersebut. Dimana pada sistem ini terdapat 66 aturan.

2) Menyusun tabel keputusan

Tabel keputusan disusun dengan mengkombinasikan setiap kemungkinan nilai untuk setiap kondisi.

3) Mereduksi tabel keputusan

Rule-rule pada tabel keputusan yang mempunyai nilai serta konklusi/kesimpulan yang sama dapat digabungkan.

Contoh 3.1 dari tabel keputusan yang diambil dari salah satu penyakit.

“Demam Berdarah”

Seorang pasien mengalami beberapa gejala sebagai berikut dengan skala 1-9 di mana nilai 1 berarti gejala muncul sesekali hingga nilai 9 yang berarti gejala sakit sekali dan muncul berulang-ulang:

- Demam (G003), skala 8
- Muntah(G008), skala 6
- Pendarahan(G015), skala 7
- Pusing (G001), skala 4
- Nyeri sendi (G023), skala 7

Dimana gejala utama demam berdarah adalah demam diikuti pusing dan muntah, bahkan dapat disertai juga dengan muntah, pendarahan dan nyeri sendi.

3.3.4. Pengalihan Tabel Keputusan ke Bentuk *If-Then Rule*

Setiap rule yang didapatkan dari tabel keputusan akan dialihkan ke dalam bentuk IF-THEN. Dari contoh 3.1 dapat ditentukan dari aturan user menderita demam bedarah, dengan aturan:

- [R1] IF (Demam TINGGI) And (Pusing SEDANG) And (Muntah SEDANG)
And (Nyeri sendi SEDANG) THEN (Peluang Demam Berdarah TINGGI)
- [R2] IF (Demam TINGGI) And (Muntah SEDANG) And (Pendarahan TINGGI)
THEN (Peluang Demam Berdarah TINGGI)
- [R3] IF (Demam TINGGI) And (Nyeri Sendi SEDANG) And (Pendarahan
TINGGI) THEN (Peluang Demam Berdarah TINGGI)
- [R4] IF (Demam TINGGI) And (Pusing SEDANG) And (Nyeri Sendi
SEDANG) And (Pendarahan SEDANG) THEN (Peluang Demam Berdarah
TINGGI)

3.3.5. Pembelajaran / *Learning*

Aturan juga dapat diperoleh dari kasus-kasus hasil diagnosa yang pernah dilakukan oleh user. Namun, untuk menentukan apakah sebuah kasus bisa

dijadikan aturan, digunakan metode *learning* di mana diperlukan campur tangan user dengan kemampuan / kepakaran pada level tertentu untuk menentukan apakah sebuah kasus layak dijadikan aturan.

3.4. Perancangan Mesin Inferensi

Inti dari inferensi yang sebenarnya adalah program komputer yang menyediakan metodologi untuk mempertimbangkan informasi dalam basis pengetahuan dan blackboard, dan merumuskan kesimpulan. Komponen ini menyediakan arahan bagaimana menggunakan pengetahuan sistem, yakni dengan mengembangkan agenda yang mengatur dan mengontrol langkah yang diambil untuk memecahkan persoalan kapan pun konsultasi berlangsung.

Sistem yang akan dibuat akan menggunakan logika fuzzy sebagai mesin inferensi. Telah dijelaskan pada bab 2 bahwa fuzzy inference system untuk mendiagnosa penyakit infeksi pada manusia menggunakan metode tsukamoto terbagi atas tiga proses yaitu : Fuzzifikasi, Inferensi, Komposisi dan Defuzzifikasi. Oleh karena itu, perlu dilakukan beberapa langkah untuk memenuhi syarat tersebut.

3.4.1. Inferensi

Telah dijelaskan di bab sebelumnya, teknik implikasi yang digunakan untuk inferensi adalah metode MIN.

Dari contoh 3.1 proses inferensinya adalah

[R1] IF (Demam TINGGI) And (Pusing SEDANG) And (Muntah SEDANG) And (Nyeri sendi SEDANG) THEN (Peluang Demam Berdarah TINGGI)
 α -predikat₁
 $= \mu_{\text{Demam.TINGGI}} \cap \mu_{\text{Pusing.SEDANG}} \cap \mu_{\text{Muntah.SEDANG}} \cap \mu_{\text{NyeriSendi.SEDANG}}$

[R2] IF (Demam TINGGI) And (Muntah TINGGI) And (Pendarahan TINGGI) THEN (Peluang Demam Berdarah TINGGI)
 α -predikat₁
 $= \mu_{\text{Demam.TINGGI}} \cap \mu_{\text{Muntah.SEDANG}} \cap \mu_{\text{Pendarahan.TINGGI}}$

[R3] IF (Demam TINGGI) And (Nyeri Sendi SEDANG) And (Pendarahan TINGGI) THEN (Peluang Demam Berdarah TINGGI)

$$\alpha\text{-predikat}_1$$
$$= \mu_{\text{Demam.TINGGI}} \cap \mu_{\text{NyeriSendi.SEDANG}} \cap \mu_{\text{Pendarahan.TINGGI}}$$

$$\begin{aligned} & \text{[R4] IF (Demam TINGGI) And (Pusing SEDANG) And (Nyeri Sendi SEDANG)} \\ & \quad \text{And (Pendarahan SEDANG) THEN (Peluang Demam Berdarah TINGGI)} \\ & \alpha\text{-predikat}_1 \\ & = \mu_{\text{Demam.TINGGI}} \cap \mu_{\text{Pusing.SEDANG}} \cap \mu_{\text{NyeriSendi.SEDANG}} \cap \\ & \quad \mu_{\text{Pendarahan.SEDANG}} \end{aligned}$$

3.4.2. Menentukan Variabel Linguistik

Seperti telah dijelaskan sebelumnya, input dari sistem adalah gejala dan intensitasnya. Intensitas merupakan variabel fuzzy yang merepresentasikan gejala, sehingga harus ditentukan variabel linguistik yang akan menentukan himpunan fuzzy dari nilai intensitas.

Sebagai output dari sistem adalah peluang dari penyakit. Maka harus ditentukan pula variabel linguistik yang akan menentukan himpunan fuzzy dari nilai peluang penyakit.

Variabel linguistik yang akan digunakan untuk mempresentasikan himpunan fuzzy untuk intensitas gejala serta domainnya ditunjukkan pada tabel 3.3.

Tabel 3.3 Variabel Linguistik dan Domain dari Intensitas

Variabel	Himpunan	Domain
Intensitas	Rendah	0 – 4
Intensitas	Sedang	2 – 8
Intensitas	Tinggi	4 – 10

Tabel 3.3 menunjukkan bahwa himpunan intensitas fuzzy RENDAH memiliki domain [0, 4], SEDANG memiliki domain [2, 8], dan TINGGI memiliki domain [4, 10].

Tabel 3.4. Variabel Linguistik dan Domain Peluang Penyakit

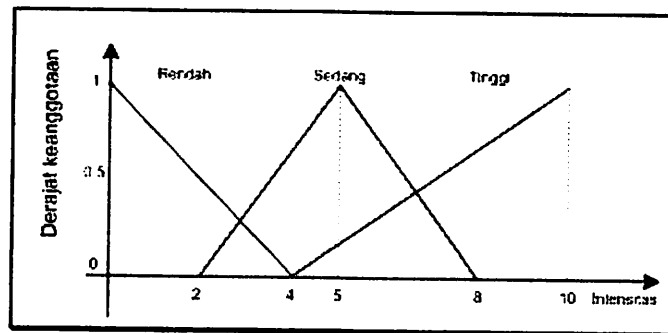
Variabel	Himpunan	Domain
Peluang	Rendah	0 – 0.4
Peluang	Tinggi	0.3 – 1.0

Tabel 3.4 menunjukkan bahwa himpunan peluang RENDAH memiliki domain $[0, 0.4]$ dan TINGGI memiliki domain $[0.3, 1.0]$.

Domain dari himpunan fuzzy akan mempengaruhi output dari sistem, maka dalam sistem dibuat sebuah prosedur untuk melakukan perubahan pada domain tersebut.

3.4.3. Menentukan Fungsi Keanggotaan (Membership Function)

Fungsi keanggotaan adalah suatu kurva yang menunjukkan pemetaan titik-titik input data ke dalam nilai keanggotaannya (sering juga disebut dengan derajat keanggotaan) yang memiliki interval antara 0 sampai 1. Gambar 3.2 akan menunjukkan fungsi keanggotaan untuk intensitas gejala. Sedangkan Gambar 3.3 akan menunjukkan fungsi keanggotaan untuk peluang penyakit.



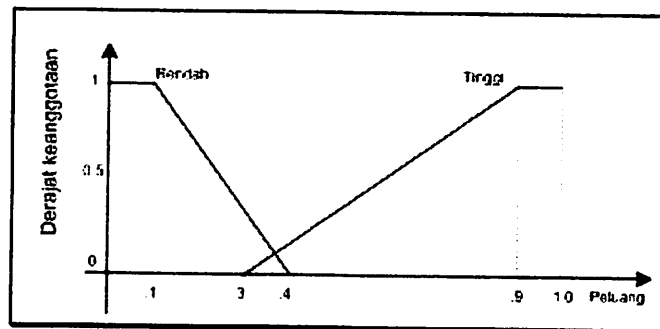
Gambar 3.2. Fungsi Keanggotaan untuk intensitas gejala.

Dari contoh 3.1 maka di dapat rumus untuk fuzzy

--

Berdasarkan contoh 3.1 maka fuzzy untuk intensitas gejala adalah dengan menggunakan 2.17, 2.18 dan 2.19 adalah:

$$\begin{aligned}
 \text{Demam.}\mu\text{GejTinggi}[8] &= (8-4)/6 = 0,667 \\
 \text{Muntah.}\mu\text{GejSedang}[6] &= (8-6)/3 = 0,667 \\
 \text{Pendarahan.}\mu\text{GejSedang}[7] &= (8-7)/3 = 0,333 \\
 \text{Pendarahan.}\mu\text{GejTinggi}[7] &= (7-4)/6 = 0,5 \\
 \text{Pusing.}\mu\text{GejSedang}[4] &= (4-2)/3 = 0,667 \\
 \text{NyeriSendi.}\mu\text{GejSedang}[7] &= (8-7)/3 = 0,333
 \end{aligned}$$



Gambar 3.3. Fungsi Keanggotaan untuk peluang penyakit

Maka di dapat rumus untuk fungsi keanggotaan peluang penyakit (z)

Karena adanya prosedur untuk melakukan perubahan domain himpunan fuzzy, maka fungsi keanggotaan di atas juga ikut berubah jika ada perubahan domain himpunan fuzzy.

Dari contoh 3.1 maka fuzzynya adalah

[R1] IF (Demam TINGGI) And (Pusing SEDANG) And (Muntah SEDANG) And (Nyeri sendi SEDANG) THEN (Peluang Demam Berdarah TINGGI)

α -predikat₁

$= \mu_{\text{Demam.TINGGI}} \cap \mu_{\text{Pusing.SEDANG}} \cap \mu_{\text{Muntah.SEDANG}} \cap \mu_{\text{NyeriSendi.SEDANG}}$

$= \min(\mu_{\text{Demam.TINGGI}}[8], \mu_{\text{Pusing.SEDANG}}[4], \mu_{\text{Muntah.SEDANG}}[6], \mu_{\text{NyeriSendi.SEDANG}}[7])$

$$= \min(0,667; 0,667; 0,667; 0,333)$$

$$= \underline{0,333}$$

Berdasarkan rumus 2.21 maka:

$$(z_1 - 0,3)/0,6 = 0,333 \rightarrow z_1 = 0,5$$

**[R2] IF (Demam TINGGI) And (Muntah TINGGI) And (Pendarahan TINGGI)
THEN (Peluang Demam Berdarah TINGGI)**

α -predikat₁

$$= \mu_{\text{Demam.TINGGI}} \cap \mu_{\text{Muntah.SEDANG}} \cap \mu_{\text{Pendarahan.TINGGI}}$$

$$= \min(\mu_{\text{Demam.TINGGI}}[8], \mu_{\text{Muntah.SEDANG}}[6], \mu_{\text{Pendarahan.TINGGI}}[7])$$

$$= \min(0,667; 0,667; 0,5)$$

$$= \underline{0,5}$$

Berdasarkan rumus 2.21 maka:

$$(z_2 - 0,3)/0,6 = 0,5 \rightarrow z_2 = 0,6$$

[R3] IF (Demam TINGGI) And (Nyeri Sendi SEDANG) And (Pendarahan TINGGI) THEN (Peluang Demam Berdarah TINGGI)

α -predikat₁

$$= \mu_{\text{Demam.TINGGI}} \cap \mu_{\text{NyeriSendi.SEDANG}} \cap \mu_{\text{Pendarahan.TINGGI}}$$

$$= \min(\mu_{\text{Demam.TINGGI}}[8], \mu_{\text{NyeriSendi.SEDANG}}[7], \mu_{\text{Pendarahan.TINGGI}}[7])$$

$$= \min(0,667; 0,333; 0,5)$$

$$= \underline{0,333}$$

Berdasarkan rumus 2.21 maka:

$$(z_3 - 0,3)/0,6 = 0,333 \rightarrow z_3 = 0,5$$

[R4] IF (Demam TINGGI) And (Pusing SEDANG) And (Nyeri Sendi SEDANG) And (Pendarahan SEDANG) THEN (Peluang Demam Berdarah TINGGI)

α -predikat₁

$$= \mu_{\text{Demam.TINGGI}} \cap \mu_{\text{Pusing.SEDANG}} \cap \mu_{\text{NyeriSendi.SEDANG}} \cap \mu_{\text{Pendarahan.SEDANG}}$$

$$= \min(\mu_{\text{Demam.TINGGI}}[8], \mu_{\text{Pusing.SEDANG}}[6], \mu_{\text{NyeriSendi.SEDANG}}[7], \mu_{\text{Pendarahan.SEDANG}}[7])$$

$$= (0,667; 0,667; 0,333; 0,333)$$

$$= \underline{0,333}$$

Berdasarkan rumus 2.21 maka:

$$(z_4 - 0,3) / 0,6 = 0,333 \rightarrow z_4 = 0,5$$

3.4.4. Penanganan Gejala yang tidak dimasukkan oleh user

Karena sistem ini bergantung pada input user, maka ada kemungkinan di antara aturan-aturan yang aktif selama diagnosa terdapat gejala yang mungkin tidak dimasukkan oleh user. Jika gejala-gejala ini diberikan nilai intensitas 0 (nol) (walau sebenarnya mungkin saja gejala tersebut dialami penderita), maka akan banyak aturan yang akan menghasilkan nilai 0 (nol) yang pada akhirnya akan sangat mempengaruhi hasil akhir diagnosa.

Untuk itu diperlukan penanganan khusus bagi gejala-gejala tersebut. Dalam hal ini, sistem akan memberikan nilai keanggotaan tertentu (yaitu 0.5) pada tiap-tiap gejala yang tidak dimasukkan oleh user, apapun domain dari gejala tersebut, baik RENDAH, SEDANG, maupun TINGGI.

3.4.5. Komposisi dan Defuzzifikasi

Karena menggunakan metode Tsukamoto, maka seperti dijelaskan pada bab sebelumnya, hasil komposisi dan defuzzifikasi diperoleh dengan menggunakan rata-rata terbobot seperti dijelaskan pada rumus 2.15.

$$Z = \frac{\alpha_1 z_1 + \alpha_2 z_2 + \dots + \alpha_n z_n}{\alpha_1 + \alpha_2 + \dots + \alpha_n} \dots \dots \dots (2.15)$$

Dari contoh 3.1 maka Z terbobotnya adalah

$$z = \frac{(\alpha\text{-pred}_1 * z_1) + (\alpha\text{-pred}_2 * z_2) + (\alpha\text{-pred}_3 * z_3) + (\alpha\text{-pred}_4 * z_4)}{\alpha\text{-pred}_1 + \alpha\text{-pred}_2 + \alpha\text{-pred}_3 + \alpha\text{-pred}_4}$$

$$z = \frac{(0,333 * 0,5) + (0,5 * 0,6) + (0,333 * 0,5) + (0,333 * 0,5)}{0,333 + 0,5 + 0,333 + 0,333}$$

$$z = 0,7995 / 1,499$$

$$z = 0,53336 = 53,336\%$$

Jadi peluang pasien tersebut menderita penyakit DEMAM BERDARAH adalah 53,336%.

3.5. Perancangan Basis Data

3.5.1. Perancangan Tabel Database

Perancangan tabel database diperlukan untuk mengetahui tabel yang digunakan. Tabel-tabel berikut adalah tabel yang diperlukan pada rancangan sistem.

Tabel 3.5. Tabel Gejala (tbGejala)

No.	Field	Type	Panjang	Keterangan
1	<u>kdGej</u>	Text	4	Kode Gejala (PK)
2	nmGej	Text	255	Nama Gejala

Tabel 3.5 digunakan untuk menyimpan gejala-gejala yang mungkin terjadi dalam suatu penyakit.

Tabel 3.6. Tabel Rule If Gejala (tbRGejala_If)

No.	Field	Type	Panjang	Keterangan
1	<u>kdRGej</u>	Text	4	Kode Rule Gejala (PK)
2	kdGej	Text	4	Kode Gejala

Tabel 3.6 digunakan untuk menyimpan bagian premis dari aturan. Field kdGejala menyatakan gejala-gejala dari rule yang aktif, sedangkan kdFuzzy merujuk pada jenis himpunan fuzzy yang digunakan.

Tabel 3.7. Tabel Rule Then Gejala (tbRGejala_Then)

No.	Field	Type	Panjang	Keterangan
1	<u>kdRGej</u>	Text	4	Kode Rule Gejala (PK)
2	kdPenyakit	Text	3	Kode Jenis Penyakit
3	ketRule	Text	255	Keterangan

Tabel 3.7 digunakan untuk menyimpan bagian konklusi dari aturan, yaitu jenis penyakit.

Tabel 3.8. Tabel Jenis Penyakit (tbPenyakit).

No.	Field	Type	Panjang	Keterangan
1	<u>kdPenyakit</u>	Text	3	Kode Jenis Penyakit (PK)
2	nmPenyakit	Text	50	Nama penyakit
3	JenisGej	Text	255	Gejala-gejala dari penyakit
4	ketPenyakit	Text	255	Detail penyakit (penyebab, pencegahan, dan lain-lain)

Tabel 3.8 digunakan untuk menyimpan jenis-jenis penyakit infeksi beserta detail pencegahan, penyebab, dan lain-lain.

Tabel 3.9. Tabel Membership Function (tbFuzzy)

No.	Field	Type	Panjang	Keterangan
1	<u>kdFuz</u>	Byte	-	Kode himpunan fuzzy
2	nmFuz	Text	10	Nama himpunan fuzzy
3	Bentuk	Byte	-	Bentuk himpunan fuzzy
4	X1	Byte	-	Koordinat x himpunan fuzzy
5	X2	Byte	-	Koordinat x himpunan fuzzy
6	X3	Byte	-	Koordinat x himpunan fuzzy

Tabel 3.9 digunakan untuk menyimpan domain dari setiap himpunan fuzzy.

Field bentuk diisi dengan kode yang menunjukkan bentuk fungsi keanggotaan (representasi linear naik/turun, segitiga, atau bentuk bahu). Misalnya himpunan fuzzy KURANG memiliki domain [4 6] dengan bentuk segitiga, maka data tersebut akan dijadikan sebuah record pada tabel tbFuzzy dengan struktur seperti berikut:

- kdFuzzy : 1
- Nama : KURANG
- Bentuk : 2 (bentuk segitiga)
- X1 : 4
- X2 : 5 (Puncak segitiga himpunan)
- X3 : 6

Contoh 3.2:

Dengan menggunakan informasi yang sama dengan Contoh 3.1, maka kita akan mendapatkan table-table seperti berikut:

Tabel 3.10. Contoh Tabel Gejala Demam berdarah (tbGejala)

<u>kdGej</u>	nmGej
G001	Pusing atau sakit kepala
G003	Demam
G008	Muntah
G015	Pendarahan
G023	Nyeri Sendi

Tabel 3.11. Contoh Tabel Penyakit (tbPenyakit)

<u>kdPenyakit</u>	<u>nmPenyakit</u>	<u>JenisGej</u>	<u>KetPenyakit</u>
P03	Demam Berdarah	Pusing, demam, ...	Pengobatan...

Tabel 3.12. Contoh Tabel Membership Function (tbFuzzy)

<u>kdFuz</u>	<u>nmFuz</u>	<u>Bentuk</u>	<u>X1</u>	<u>X2</u>	<u>X3</u>
1	RENDAH	1	0	0	4
2	SEDANG	2	2	5	8
3	TINGGI	3	4	0	10

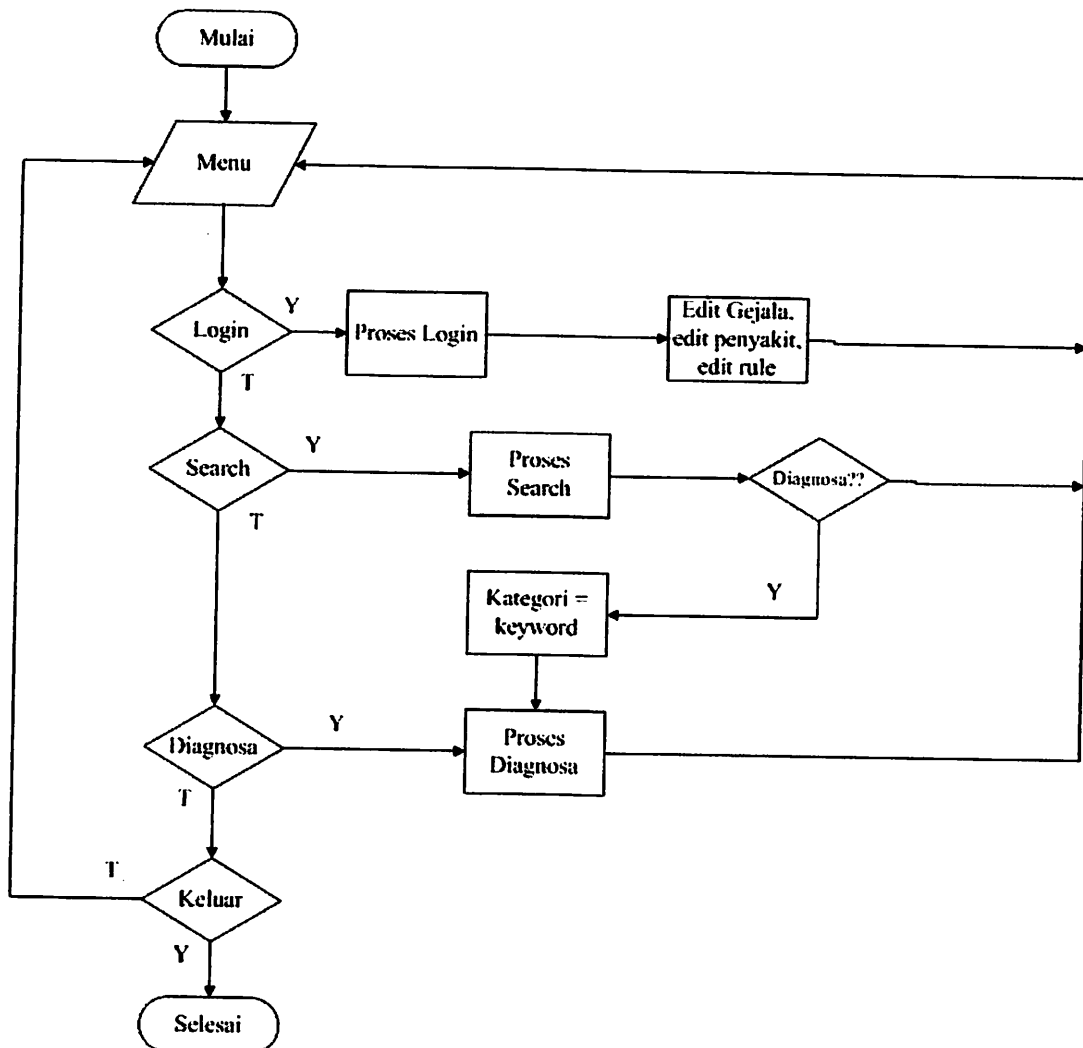
Tabel 3.13. Contoh Tabel Rule If Gejala (tbRGejala_If).

<u>kdRGej</u>	<u>kdGej</u>
R1	G003
	G002
	G001
R2	G003
	G001
	G002
R3	G003
	G004
	G005
R4	G003
	G005
	G006

Tabel 3.14. Contoh Tabel Rule Then Gejala (tbRGejala_Then).

<u>kdRGej</u>	<u>kdJenis</u>	<u>ketRule</u>
R001	P03	Infeksi.com

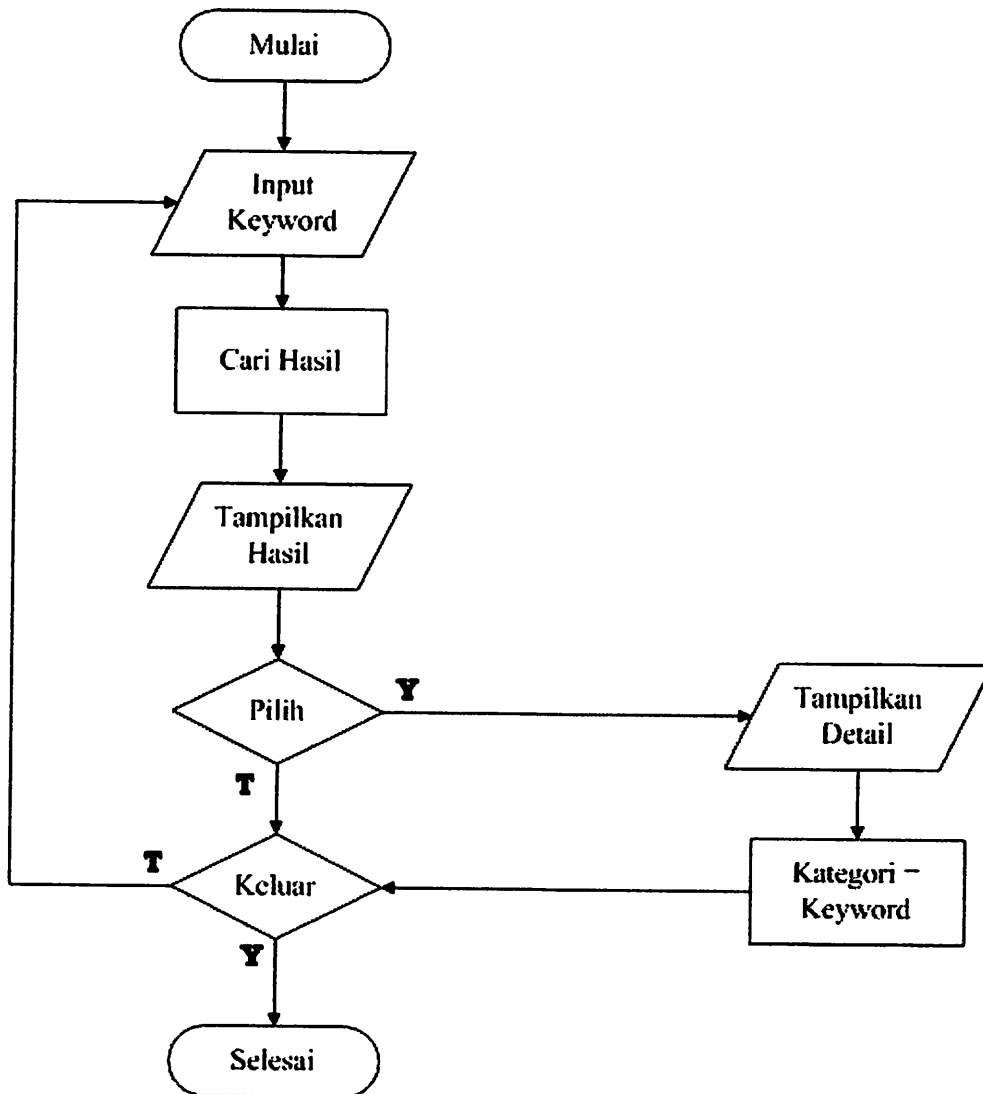
3.6. Diagram Alir Sistem



Gambar 3.4. Diagram Alir Sistem Pada User

Pada halaman awal, user dapat melakukan tiga hal, yaitu proses *login*, *search*, dan *diagnosa*. Jika user memilih *login*, maka proses *login* akan dilakukan, demikian

pula dengan *search* dan diagnosa. Dari proses *search* dapat pula dilakukan proses diagnosa sesuai jenis penyakit yang dipilih oleh user pada hasil *search*.



Gambar 3.5. Diagram Alir “Proses Search”.

Proses *Search* pada Gambar 3.5 dimulai dengan user memasukkan sebuah input berupa kata kunci / *keyword* yang akan menjadi acuan dalam pencarian. Setelah pencarian dilakukan, hasil pencarian akan ditampilkan dan user dapat memilih salah satu dari hasil tersebut. Dari pilihan user tersebut akan ditampilkan informasi dari pilihan jenis penyakit tersebut. User dapat pula melakukan proses diagnosa berdasarkan pilihan tersebut.

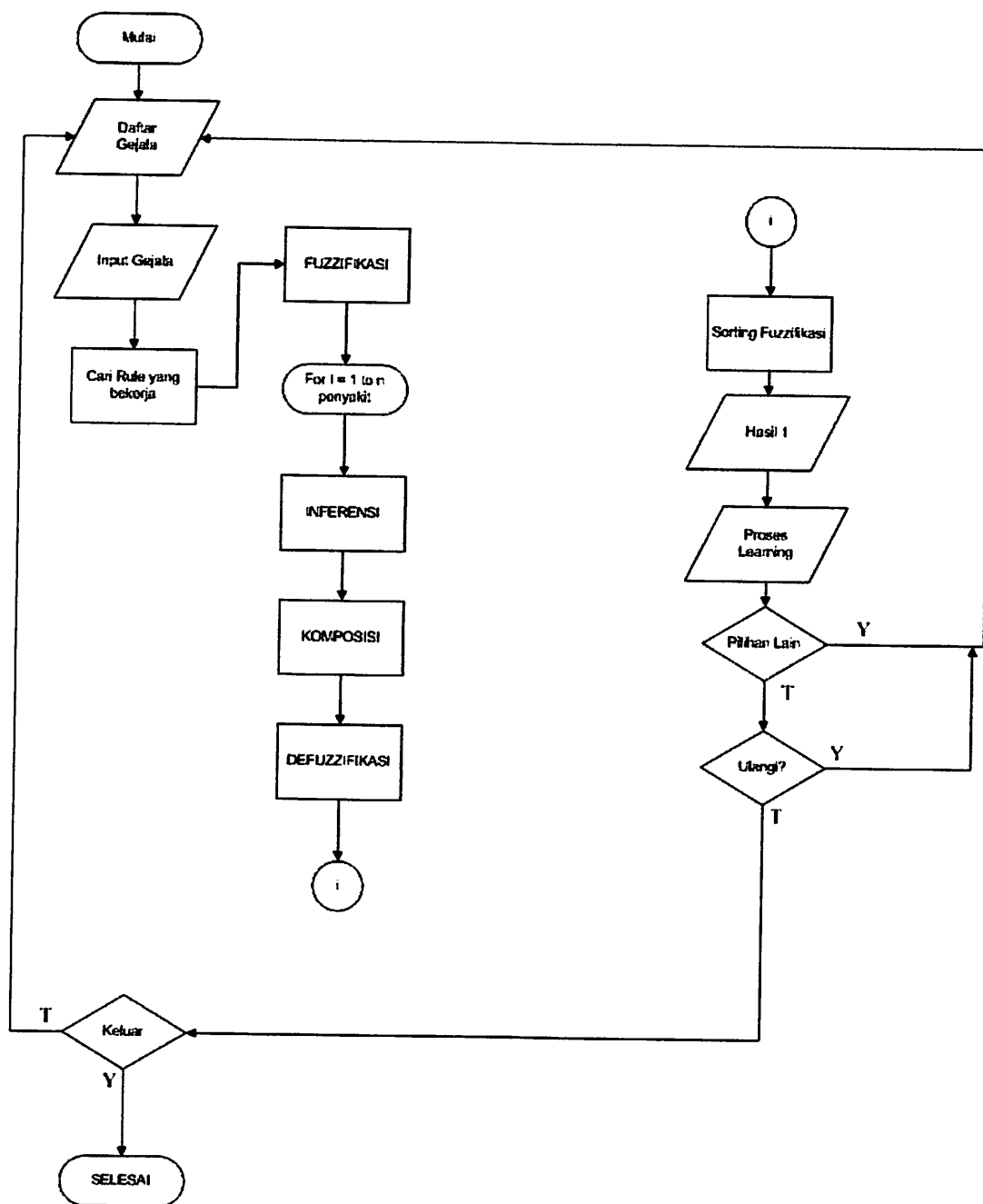
Gambar 3.6 merupakan diagram alir proses diagnosa yang dimulai dengan menampilkan semua gejala yang ada. Jika pada proses *search*, user memilih untuk

melakukan proses diagnosa berdasarkan suatu penyakit, maka pada proses diagnosa ‘hanya’ akan ditampilkan gejala-gejala yang ada pada penyakit tersebut.

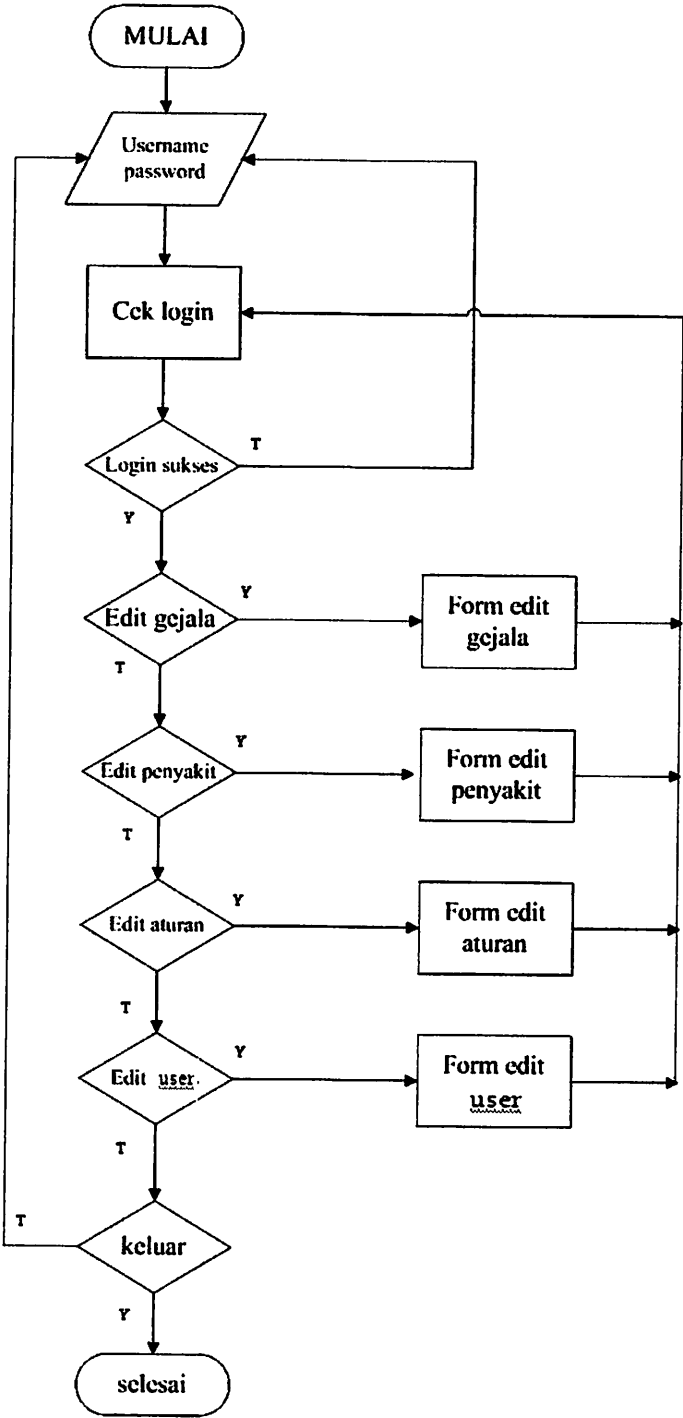
Setelah user menentukan gejala yang dialami penderita pada daftar gejala yang ada, user juga harus memberikan input berupa intensitas dari gejala tersebut.

Dari input-input tersebut, kemudian akan dicari aturan-aturan yang bekerja sesuai dengan input-input tersebut. Setelah dilakukan proses fuzzifikasi, inferensi, komposisi, dan defuzzifikasi, kemudian hasilnya akan ditampilkan kepada user. Jika user kurang puas dengan hasil diagnosa, maka user dapat melakukan proses diagnosa lanjutan dengan langkah-langkah yang sama.

Proses *login* pada Gambar 3.7 dilakukan untuk mendapatkan hak akses untuk melakukan perubahan pada basis aturan. Seperti proses *login* yang lain, proses dimulai dengan melakukan input berupa *username* dan *password*. Jika sesuai, maka user akan mendapatkan hak akses tersebut.



Gambar 3.6 Diagram Alir “Proses Diagnosa”.



Gambar 3.7. Diagram Alir Untuk admin.

3.7. Perancangan Antarmuka Pemakai (*User Interface*)

Terdiri dari beberapa pilihan, yaitu :

1. Menu *Search / Detail*

Pada bagian ini, user dapat melakukan pencarian informasi yang diinginkannya. Setelah informasi ditemukan, user dapat melakukan proses diagnosa sesuai informasi yang dilihatnya.

2. Menu *Diagnosa*

Pada bagian ini, user dapat melakukan proses diagnosa secara keseluruhan menggunakan semua pengetahuan yang dimiliki sistem.

3. Menu *Login*.

User dengan hak akses khusus dapat melakukan login di menu ini. Jika login berhasil, maka user dapat melakukan perubahan pada sistem. Adapun perubahan yang dapat dilakukan yaitu:

- Edit Gejala
- Edit Penyakit
- Edit Aturan (Rule)
- Edit Kasus

User yang tidak melakukan login atau tidak berhasil login akan dianggap sebagai user biasa yang hanya mampu melakukan konsultasi kepada sistem.

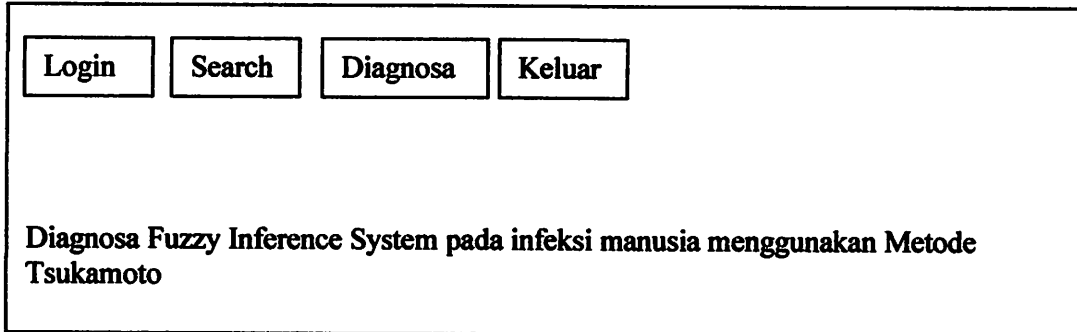
4. Menu *Exit*.

Menu ini digunakan untuk keluar dari aplikasi.

3.8. Pembuatan Desain Interface

Pembuatan form merupakan suatu bentuk desain interface yang sangat menunjang dalam pembuatan program yang mudah digunakan oleh user. Sehingga program aplikasi ini dapat digunakan oleh banyak orang tanpa perlu kesulitan mengoperasikan.

- Bentuk Form Menu

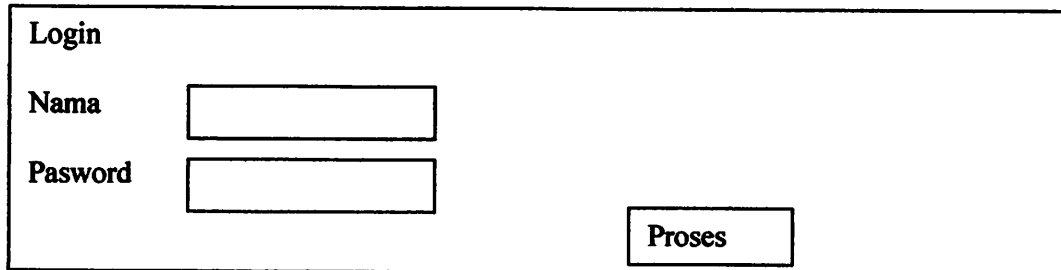


The screenshot shows a rectangular window with a title bar. Inside, there are four buttons arranged horizontally at the top: "Login", "Search", "Diagnosa", and "Keluar". Below these buttons, the text "Diagnosa Fuzzy Inference System pada infeksi manusia menggunakan Metode Tsukamoto" is displayed.

Gambar 3.8. Form Menu

Desain gambar 3.8 ini menunjukkan menu pada tampilan awal program.

- Bentuk Form Untuk Login

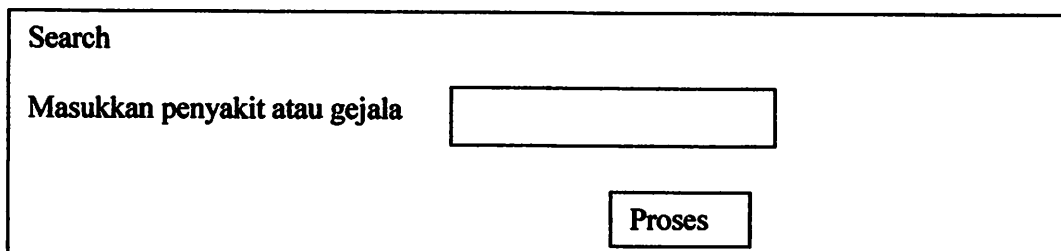


The screenshot shows a login form with a title bar. The title "Login" is at the top left. Below it, there are two labels: "Nama" and "Pasword". Each label is followed by a rectangular input field. To the right of these fields, there is a button labeled "Proses".

Gambar 3.9. Form Login

Desain gambar 3.9 ini ditujukan untuk admin ataupun dokter ahli untuk mengedit gejala, penyakit, aturan dan kasus.

- Bentuk Form Search

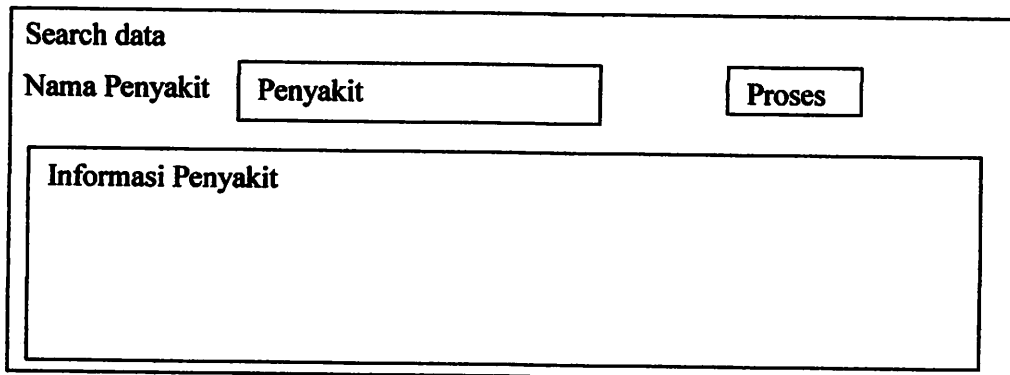


The screenshot shows a search form with a title bar. The title "Search" is at the top left. Below it, there is a label "Masukkan penyakit atau gejala" followed by a rectangular input field. To the right of this field, there is a button labeled "Proses".

Gambar 3.10 Form Search

Desain gambar 3.10 merupakan pencarian suatu penyakit atau gejala tanpa login dan tanpa proses perhitungan.

- Bentuk Form Hasil Search

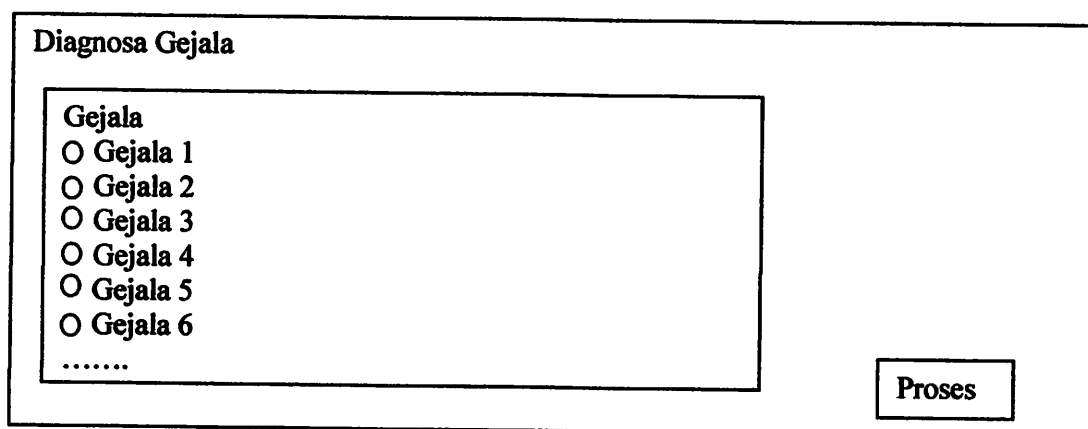


The form is titled "Search data". It contains a label "Nama Penyakit" followed by a text input field with the placeholder text "Penyakit". To the right of the input field is a button labeled "Proses". Below the input field is a large rectangular area labeled "Informasi Penyakit".

Gambar 3.11 Form Hasil Search

Desain gambar 3.11 merupakan desain hasil pencarian suatu penyakit setelah dilakukan proses pencarian.

- Bentuk Form Diagnosa



The form is titled "Diagnosa Gejala". It contains a list of symptoms with radio button options: "Gejala 1", "Gejala 2", "Gejala 3", "Gejala 4", "Gejala 5", "Gejala 6", and ".....". To the right of the list is a button labeled "Proses".

Gambar 3.12 Form Diagnosa

Desain Form 3.12 ini user dapat memilih gejala yang di rasakan dengan mencetang gejala yang sudah di sediakan.

• Bentuk Form Intensitas

Id_gejala | nama gejala

☐ Rendah

☐ Sedang

☐ Tinggi

Id_gejala | nama gejala

☐ Rendah

☐ Sedang

☐ Tinggi

Id_gejala | nama gejala

☐ Rendah

☐ Sedang

☐ Tinggi

Proses

Gambar 3.13 Form Intensitas

Setelah memilih gejala maka user dapat memilih intensitas dari gejala tersebut, cukup mencetang apakah gejala yang di rasakan rendah, sedang, atau tinggi.

• Bentuk Form Diagnosa

Proses

Hasil Diagnosa Penyakit

Gambar 3.14 Form Diagnosa

Desain gambar 3.14 merupakan desain hasil diagnosa suatu penyakit setelah dilakukan proses pencarian

• Bentuk Form Edit Gejala

Edit Gejala

ID

Gejala

simpan

Clear

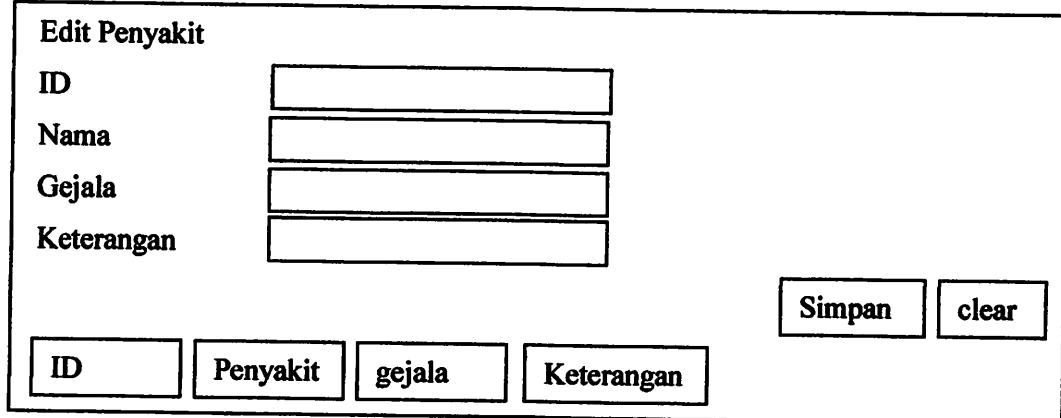
ID

Gejala

Gambar 3.15 Form Edit Gejala

Desain gambar 3.15 merupakan desain untuk edit gejala ada button simpan dan button clear.

- **Button Form Edit Penyakit**

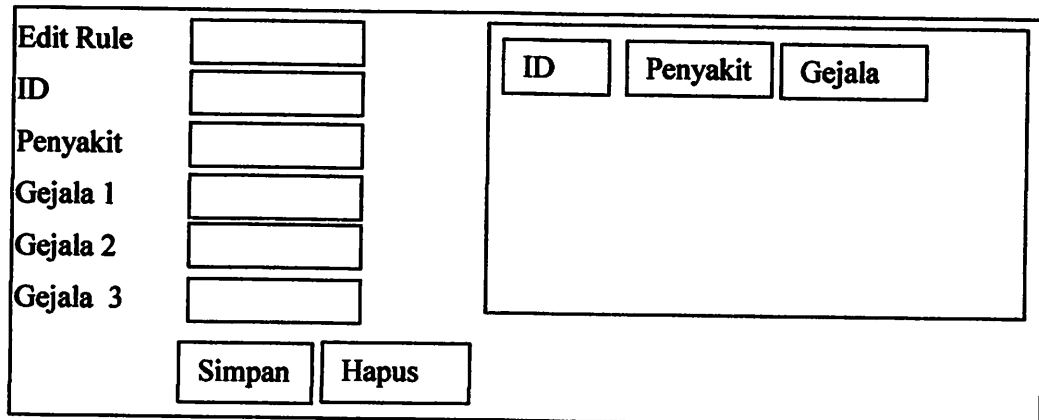


The form titled "Edit Penyakit" contains four input fields stacked vertically, labeled "ID", "Nama", "Gejala", and "Keterangan". To the right of these fields are two buttons labeled "Simpan" and "clear". Below the input fields is a horizontal row of four buttons labeled "ID", "Penyakit", "gejala", and "Keterangan".

Gambar 3.16 form edit penyakit

Desain gambar 3.16 merupakan desain form untuk mengedit suatu penyakit, baik gejala yang menyertai maupun keterangan (pencegahan, dan pengobatan awal) serta ada button untuk menyimpan dan menghapus.

- **Button Edit Rule**



The form titled "Edit Rule" contains six input fields stacked vertically, labeled "ID", "Penyakit", "Gejala 1", "Gejala 2", and "Gejala 3". To the right of these fields is a box containing three buttons labeled "ID", "Penyakit", and "Gejala". Below the input fields are two buttons labeled "Simpan" and "Hapus".

Gambar 3.17 form edit rule

Desain gambar 3.17 merupakan desain form untuk mengedit aturan dengan memasukkan gejala-gejala baru.

3.9. Struktur Basis Pengetahuan (*Knowledge Base*)

Basis pengetahuan yang digunakan dalam sistem ini adalah tentang aturan nama penyakit dan gejala penyakit. Representasi penyakit dan gejala penyakit dinyatakan Dalam Tabel 3.15.

Tabel 3.15 Penyakit dan gejala

No	Penyakit	Gejala
1.	Cacar Air	Sakit kepala
		Nyeri Punggung
		Demam
		Kelemahan Badan
		Ruam Kulit
		Menggigil
2.	Campak	Demam
		Muntah
		Myalgia
		Sakit Kepala
		Mata terasa panas
		Bersin-bersin
		Pilek
		Ruam Kulit
3.	Demam Berdarah Dague	Demam
		Sakit kepala
		Muntah
		Nyeri sendi
		Pendarahan
		Kejang-kejang
		Ruam Kulit
4.	Demam Chikungunya	Demam
		Linu Persendian
		Pegal-pegal
5.	Disentri	Demam
		Gangguan Perut
		Mual
		Menggigil
		Muntah
		Diare
6.	Influenza	Demam
		Menggigil
		Sakit Kepala
		Muntah

		Nyeri Otot
		Kelemahan Badan
		Batuk
7.	Malaria	Demam
		Menggigil
		Sakit Kepala
		Mual
		Muntah
		Kelemahan Badan
		Muka Merah
		Kulit Kering
		Kejang-kejang
8.	TBC	Lemah Badan
		Batuk
		Penurunan Berat Badan
		Demam
		Berkeringat pada malam hari

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 untuk pengembangan sistem Fuzzy Inference System untuk mendiagnosa jenis penyakit infeksi pada manusia menggunakan Metode Tsukamoto.

4.1.1 Lingkungan Perangkat Keras

Perangkat keras yang digunakan dalam pengembangan sistem ini adalah sebagai berikut :

1. Prosesor Intel Core 2 Duo Processor T5800
2. Memori 4 GB
3. Harddisk dengan kapasitas 250 GB
4. Monitor 15"
5. Keyboard
6. Mouse

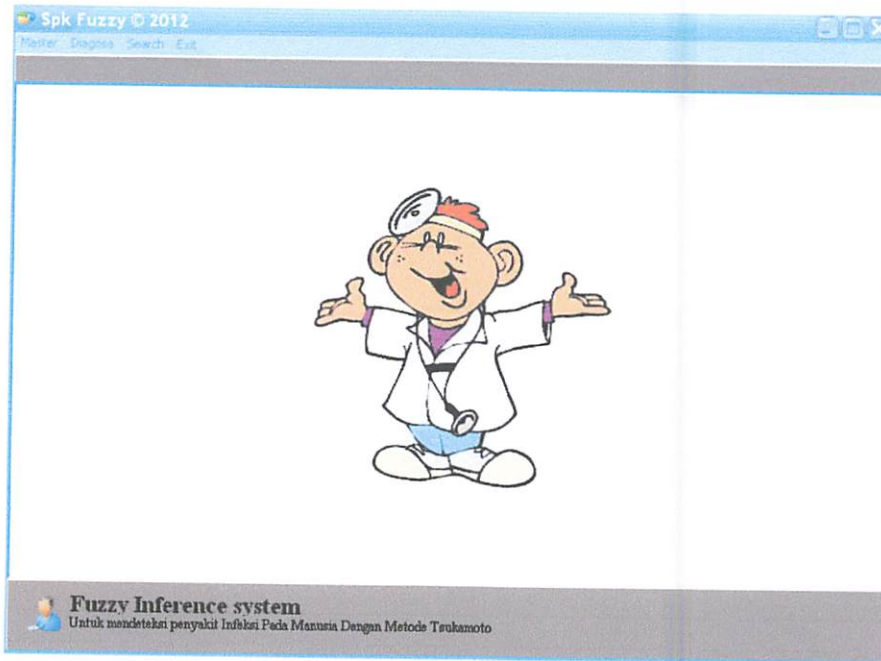
4.1.2. Lingkungan Perangkat Lunak

Perangkat lunak yang digunakan dalam pengembangan sistem ini adalah :

1. Sistem Operasi Windows XP Professional SP2
2. Borland Delphi 7
3. My SQL

4.2 Implementasi Program

Pada sub bab ini akan dibahas mengenai implementasi dari sistem diagnosa infeksi pada manusia menggunakan fuzzy inference system dengan metode tsukamoto.

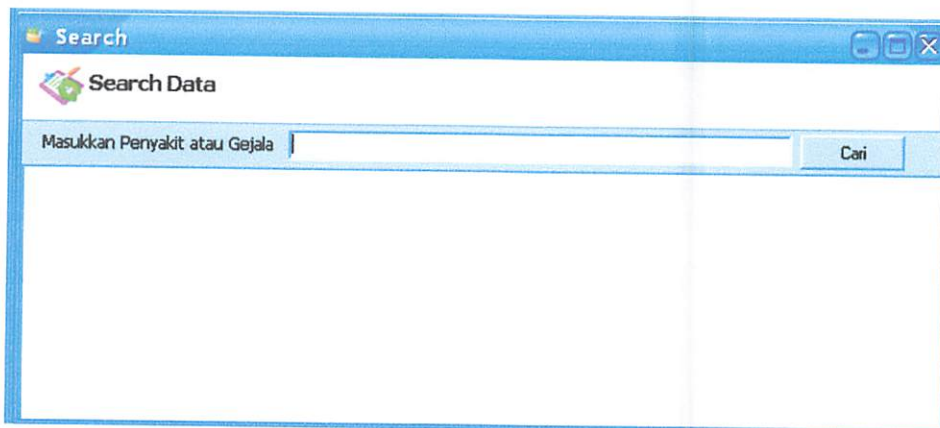


Gambar 4.1 Tampilan menu

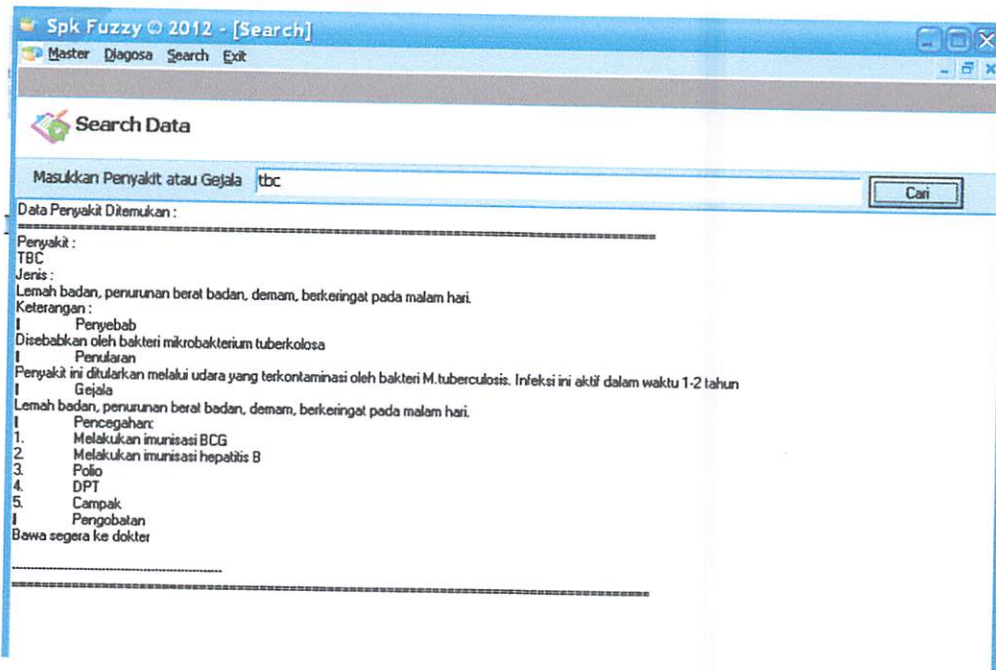
4.2.1 Pada User

4.2.1.1 Proses search

Pada bagian ini, user dapat melakukan pencarian penyakit berdasarkan kata kunci (keyword) berupa penyakit untuk di masukkan dalam search. Setelah memasukkan kata kunci dan menekan tombol search maka akan di tampilkan seperti gambar 4.3



Gambar 4.2 Proses Search



Gambar 4.3 Hasil Pencarian

Untuk menjalankan pertama-tama harus di koneksikan dengan "db_spkfuzzy". Setelah didapatkan koneksi ke "db_spkfuzzy", maka dilakukan *query* berdasarkan input user dari halaman sebelumnya. Adapun sistem akan mencari hasil dengan menggunakan *field* 'nmPenyakit' dan 'ketPenyakit' pada tabel 'tbPenyakit' sebagai pembanding. Jika *query* berhasil mendapatkan penyakit yang di masukkan kuncinya akan menampilkan detail penyebab, penyebaran, gejala, pencegahan serta pengobatannya, jika gejala yang di masukkan maka akan keluar id gejala.

4.2.1.2 Proses Diagnosa

Pada bagian ini, user dapat melakukan diagnosa secara umum maupun diagnosa berdasarkan suatu penyakit tertentu. Urutan untuk mendapatkan hasil diagnosa yaitu input data user, memilih gejala, dan memilih intensitas gejala.

4.2.1.2.1 Proses Input User

Pada tahap awal, user memasukkan nama, alamat serta umur user

Gambar 4.4 Form Input user

User dapat memasukkan identitasnya sebelum melakukan diagnosa. Hal ini memudahkan dalam pendataan user. Jika identitas tidak di simpan maka user tidak bisa melanjutkan untuk melakukan diagnosa.

4.2.1.2.2 Proses Input Gejala

Setelah user memasukkan identitas, user memasukkan atau memilih gejala-gejala yang terjadi pada penderita. Gejala yang dipilih dibatasi hanya 3 gejala untuk memudahkan dalam pengolahan data.

Gambar 4.5 Form Gejala dalam Diagnosa

Setelah terciptanya koneksi, maka user akan diberi pilihan untuk melakukan input gejala yang dialami penderita. Pilihan akan ditampilkan sebagai *check box*. .

4.2.1.2.3 Input Intensitas Gejala

Setelah memilih gejala-gejala yang dialami penderita, user masih harus melakukan proses input intensitas gejala yaitu tingkatan rasa sakit dari gejala tersebut. Input berupa intensitas rendah, sedang dan tinggi.

Gambar 4.6 Input Intensitas Gejala Penyakit

Setelah user melakukan proses input gejala pada halaman sebelumnya, maka akan di pilih intensitasnya seperti pada form di atas.

4.2.1.2.4 Hasil Diagnosa

Setelah melalui beberapa proses diagnosa, maka hasil akhir akan ditampilkan pada bagian ini. Dari hasil diagnosa tersebut, user dapat mengetahui penyakit apakah yang dialami dan gejala apa saja yang dialami

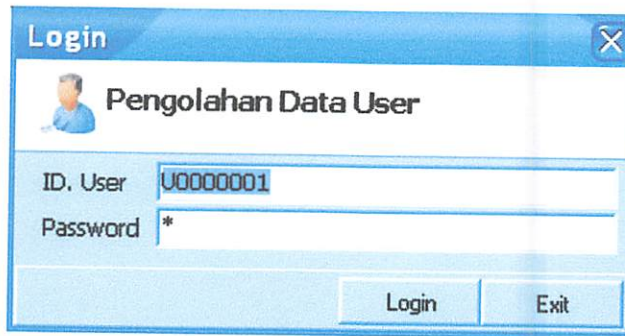
Gambar 4.7 Form Hasil Diagnosa

4.2.2 Untuk Admin

4.2.2.1 Proses Login

Pada halaman ini, user dapat melakukan login untuk mendapatkan akses ke database. Setelah mendapatkan akses, maka user dapat melakukan proses edit gejala, penyakit, kasus, maupun domain fuzzy.

Karena hanya ada satu user utama yang memiliki akses administrator untuk dapat melakukan perubahan pada sistem, maka sistem login dibuat sesederhana mungkin.



The image shows a login window with a blue title bar. The title bar contains the word 'Login' on the left and a close button (X) on the right. Below the title bar, there is a header area with a small user icon and the text 'Pengolahan Data User'. The main area of the window contains two input fields. The first field is labeled 'ID, User' and contains the text 'U00000001'. The second field is labeled 'Password' and contains a single asterisk '*'. At the bottom of the window, there are two buttons: 'Login' and 'Exit'.

Gambar 4.8 Proses Login

Jika user telah memasukkan username dan password, kemudian menekan tombol login, maka kemudian diperiksa apakah username dan password sudah cocok. Jika cocok, maka sebuah session disimpan untuk menyatakan bahwa user sudah login dan berhasil. Namun jika tidak cocok, maka diberikan sebuah variabel yang menandakan login gagal.

4.2.2.2 Edit Gejala

Pada halaman ini, user dapat melakukan perubahan pada tabel gejala, seperti edit gejala, hapus gejala, dan menambah gejala.

ID.	Gejala
G001 ☐	Sakit Kepala ☐
G002	Nyeri Punggung
G003	Demam
G004	Lemah Badan
G005	Ruam Kulit
G006	Menggigil
G007	Mual
G008	Muntah
G009	Mata berair

Gambar 4.9 Form Tabel Gejala

The screenshot shows a window titled "Data Gejala" with a sub-header "Pengolahan Data Gejala". It contains a form for editing a specific symptom record. The "ID." field is populated with "G001". Below it, the "Gejala" field is empty. To the right of the form are three buttons: "Hapus", "Edit", and "Clear". Below the form is a table of symptoms, identical to the one in Gambar 4.9, with "G002 Nyeri Punggung" highlighted.

Gambar 4.10 Form Edit Gejala

Karena memerlukan data dari database, maka diperlukan sebuah koneksi ke database db_spkfuzzy. Pada saat pertama kali memasuki halaman ini, bagian yang akan dikerjakan adalah menampilkan keseluruhan data tabel gejala. User dapat melihat detail dari gejala dengan menekan *link* pada kode gejala. Di awal halaman terdapat tombol untuk menambah data gejala.

Pada bagian detail terdapat pilihan edit dan hapus. Jika user telah melakukan proses editing dan menekan tombol Save, maka akan melakukan query untuk melakukan editing sesuai input user.

Jika user menekan link hapus dari sebuah record, maka akan melakukan query untuk menghapus data gejala yang dipilih.

Jika user ingin menambah data gejala, maka sebuah form tambah gejala akan ditampilkan. Jika user telah memasukkan data gejala yang baru dan menekan tombol Save, maka akan melakukan query untuk melakukan penambahan gejala yang baru ke tabel gejala.

4.2.2.3 Edit Penyakit

Seperti halnya edit gejala, pada bagian ini user dapat melakukan proses edit penyakit, hapus penyakit, serta menambahkan data penyakit baru.

ID.	Nama	Gejala	Keterangan
P001	Cacar Air	salit kepala, nyeri punggung, demam, letargi	Penyakit yang disebabkan oleh virus Varicella Zoster yang bersifat menular. Penyakit ini ditularkan melalui percikan ludah penderita atau melalui benda-benda yang terkontaminasi.
P002	Campak	Demam, muntah, myalgia, sakit kepala, malaise	Penyakit yang disebabkan oleh paramyxovirus (virus campak). Orang dewasa yang terkena campak ada kemungkinan saat anak-anak lolos dari vaksin campak atau saat ia terinfeksi.
P003	Demam Berdarah	Demam, sakit kepala, muntah, nyeri sendi	Demam berdarah (DB) atau Demam Berdarah Dengue (DBD) adalah penyakit demam akut yang ditemukan di daerah tropis, dengan penyebaran geografis yang luas.
P004	Demam Chikungunya	Demam, linu persendian, pegal-pegal	Penyakit ini disebabkan oleh virus Chikungunya (CHIKV) yang termasuk keluarga Togaviridae, Genus Alphavirus. Penyakit ini ditularkan melalui nyamuk Aedes.
P005	Difteri	Demam, gangguan pernafas, mulut berbusa, menggigil	Umumnya disebabkan infeksi Entamoeba histolytica yang menyebabkan disentri amuba dan infeksi bakteri golongan Shigella yang menjadi penyebab disentri basiler.
P006	Influenza	Demam, menggigil, sakit kepala, muntah, letargi	Penyakit ini disebabkan oleh virus RNA dari famili Orthomyxoviridae (virus influenza). Penyakit ini ditularkan dengan menggunakan media udara.
P007	Malaria	Demam, menggigil, sakit kepala, letargi	Penyakit ini disebabkan oleh parasit nyamuk malaria yaitu plasmodium vivax, plasmodium ovale, plasmodium falciparum, plasmodium malariae. Plasmodium malariae menyebabkan demam berkejang.
P008	TBC	Letargi badan, penurunan berat badan, batuk, demam	Penyakit ini disebabkan oleh bakteri mikrobakterium tuberkulosa. Penyakit ini ditularkan melalui udara yang terkontaminasi oleh bakteri M.tuberculosis.

Gambar 4.11 Form Tabel Penyakit

Data Penyakit

Pengolahan Data Penyakit

ID:

Nama:

Gejala:

Keterangan:

Hapus Edit Clear

ID.	Nama	Gejala	Keterangan
P001	Cacar Air	salit kepala, nyeri punggung, demam, letargi	Penyakit yang disebabkan oleh virus Varicella Zoster yang bersifat menular. Penyakit ini ditularkan melalui percikan ludah penderita atau melalui benda-benda yang terkontaminasi.
P002	Campak	Demam, muntah, myalgia, sakit kepala, malaise	Penyakit yang disebabkan oleh paramyxovirus (virus campak). Orang dewasa yang terkena campak ada kemungkinan saat anak-anak lolos dari vaksin campak atau saat ia terinfeksi.
P003	Demam Berdarah	Demam, sakit kepala, muntah, nyeri sendi	Demam berdarah (DB) atau Demam Berdarah Dengue (DBD) adalah penyakit demam akut yang ditemukan di daerah tropis, dengan penyebaran geografis yang luas.
P004	Demam Chikungunya	Demam, linu persendian, pegal-pegal	Penyakit ini disebabkan oleh virus Chikungunya (CHIKV) yang termasuk keluarga Togaviridae, Genus Alphavirus. Penyakit ini ditularkan melalui nyamuk Aedes.
P005	Difteri	Demam, gangguan pernafas, mulut berbusa, menggigil	Umumnya disebabkan infeksi Entamoeba histolytica yang menyebabkan disentri amuba dan infeksi bakteri golongan Shigella yang menjadi penyebab disentri basiler.
P006	Influenza	Demam, menggigil, sakit kepala, muntah, letargi	Penyakit ini disebabkan oleh virus RNA dari famili Orthomyxoviridae (virus influenza). Penyakit ini ditularkan dengan menggunakan media udara.
P007	Malaria	Demam, menggigil, sakit kepala, letargi	Penyakit ini disebabkan oleh parasit nyamuk malaria yaitu plasmodium vivax, plasmodium ovale, plasmodium falciparum, plasmodium malariae. Plasmodium malariae menyebabkan demam berkejang.
P008	TBC	Letargi badan, penurunan berat badan, batuk, demam	Penyakit ini disebabkan oleh bakteri mikrobakterium tuberkulosa. Penyakit ini ditularkan melalui udara yang terkontaminasi oleh bakteri M.tuberculosis.

Gambar 4.12 Form Edit Penyakit

4.2.2.4 Edit Rule

Seperti halnya edit gejala, pada bagian ini user dapat melakukan proses edit aturan, hapus aturan, serta menambahkan data aturan baru.

Data Gejala Penyakit

Pengolahan Data Gejala Penyakit

ID.

Penyakit

Gejala I

Gejala II

Gejala III

ID.	Penyakit	Gejala
R01	Cacar Air	G003 □ □ G002 □ □ G00
R02	Cacar Air	G003 □ □ G001 □ □ G00
R03	Cacar Air	G003 □ □ G004 □ □ G00
R04	Cacar Air	G003 □ □ G005 □ □ G00
R05	Cacar Air	G003 □ □ G005 □ □ G00
R06	Cacar Air	G003 □ □ G005 □ □ G00
R07	Cacar Air	G003 □ □ G001 □ □ G00
R08	Cacar Air	G003 □ □ G005 □ □ G00
R09	Cacar Air	G003 □ □ G004 □ □ G00
R10	Cacar Air	G003 □ □ G004 □ □ G00

Gambar 4.13 Form edit rule

Jika terdapat tambahan gejala yang mungkin di derita user pada suatu penyakit maka rulepun berubah. Admin bisa merubah aturan rule juga dapat menghapus aturan yang ada.

4.2.2.5 Edit Data Hak akses

Pada form ini admin dapat menambah user yang bisa login. Akan memndapat hak yang sama untuk mengedit gejala, penyakit, aturan, penambahan user serta dapat mengedit user.

Data User

Pengolahan Data User

ID. User

Nama

Password

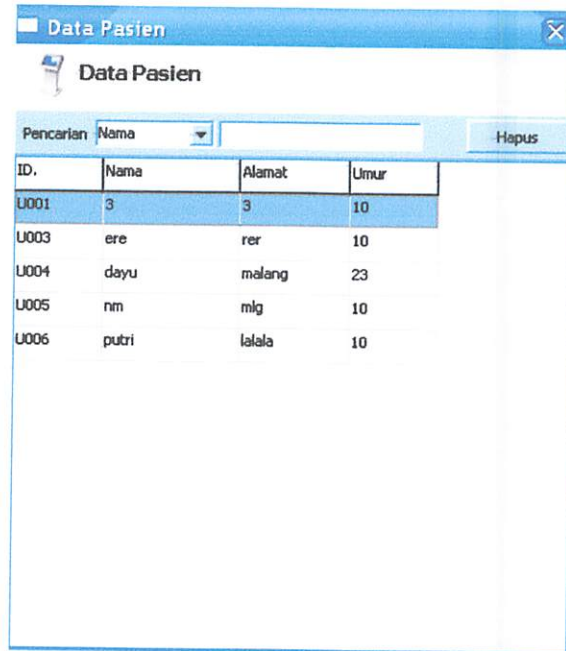
Akses

ID.	Nama	Akses
U0000001	nama	Pakar
U0000002	putri	Admin

Gambar 4.14 Form edit hak akses

4.2.2.6 Edit user

Pada form ini dapat menghapus data pasien jika dianggap data terlalu banyak.



The screenshot shows a window titled "Data Pasien" with a close button (X) in the top right corner. Below the title bar, there is a label "Data Pasien" and a small icon. The main area contains a search bar labeled "Pencarian" with a dropdown menu set to "Nama" and a text input field. To the right of the search bar is a button labeled "Hapus". Below the search bar is a table with four columns: "ID.", "Nama", "Alamat", and "Umur". The table contains six rows of data.

ID.	Nama	Alamat	Umur
U001	3	3	10
U003	ere	rer	10
U004	dayu	malang	23
U005	nm	mlg	10
U006	putri	lalala	10

Gambar 4.15 Form Edit data pasien

4.3 Analisa Data

Setelah dilakukan pengujian dengan 66 rule, maka di dapatkan tabel berikut. Hasil lengkap pada lampiran. Untuk pengujian ke dua domainnya di rubah menjadi RENDAH : [0, 6]; SEDANG : [3, 9]; TINGGI : [6, 10].

Tabel 4.1 Tabel Pengujian

kdR	#00	#01	#02	#03	#04
R01	P01	68 %	33,33 %	70 %	65%
R02	P01	68%	33,33 %	70 %	65%
R03	P01	68%	43,33%	70%	65%
R04	P01	68%	33,33 %	33,33 %	65%
R05	P01	68%	33,33 %	33,33 %	65%
R06	P01	68%	33,33 %	33,33 %	65%
R07	P01	68%	90%	43,33%	65%
R08	P01	33,33%	90%	70%	33,33%
R09	P01	33,33%	33,33%	33,33%	33,33%
R10	P01	33,33%	90%	70%	33,33%
R11	P02	68%	33,33 %	33,33 %	33,33 %
R12	P02	68%	33,33 %	33,33 %	33,33 %
R13	P02	68%	33,33 %	33,33 %	33,33 %
R14	P02	68%	33,33 %	43,33 %	33,33 %
R15	P02	68%	33,33 %	33,33 %	33,33 %
R16	P02	33,33%	90%	70%	61%
R17	P02	33,33%	90%	70%	61%
...	...	...	...	...	...
...	...	...	...	...	...
R61	P08	68%	33,33 %	43,33 %	33,33 %
R62	P08	68%	43,33 %	43,33 %	43,33 %
R63	P08	68%	33,33 %	43,33 %	33,33 %
R64	P08	61%	33,33 %	43,33 %	33,33 %
R65	P08	33,33%	90%	70%	33,33 %
R66	P08	33,33%	90%	70%	33,33 %
Rata – rata		71%	50%	63%	53 %

Keterangan:

kdR : Kode Aturan / Rule

#00 : Penyakit

#01 : Hasil Pengujian 1 (RENDAH : [0, 4]; SEDANG : [2, 8]; TINGGI : [4, 10].)

#02 : Hasil Pengujian 2 (RENDAH : [0, 6]; SEDANG : [3, 9]; TINGGI : [6, 10].)

#03 : Hasil Pengujian 3 (RENDAH : [0, 5]; SEDANG : [2.5, 8]; TINGGI : [5, 10].)

#04 : Hasil Pengujian 4 (RENDAH : [0, 3]; SEDANG : [1.5, 7]; TINGGI : [3, 10].)

Contoh Perhitungan manual

1) R01 pada #01 (Hasil Pengujian 1)

If (demam tinggi) and (sakit kepala tinggi) and (nyeri punggung sedang) Then
(Pelunag Cacar air Tinggi)

α -predikat₁

$$\begin{aligned}
 &= \mu_{\text{Demam.TINGGI}} \cap \mu_{\text{Sakit kepala.TINGGI}} \cap \mu_{\text{Nyeri Punggung.SEDANG}} \\
 &= \min(0.8 \cap 0.8 \cap 0.708) \\
 &= \underline{\mathbf{0.708}}
 \end{aligned}$$

Berdasarkan rumus 2.21 maka:

$$(0.708 - 0,3)/0,6 = z_1 \rightarrow z_1 = \mathbf{68 \%}$$

2) R02 pada #01 (Hasil Pengujian 1)

If (demam tinggi) and (sakit kepala sedang) and (nyeri punggung sedang) Then
(Pelunag Cacar air Tinggi)

α -predikat₁

$$\begin{aligned}
 &= \mu_{\text{Demam.TINGGI}} \cap \mu_{\text{Sakit kepala.sedang}} \cap \mu_{\text{Nyeri Punggung.SEDANG}} \\
 &= \min(0.8 \cap 0.708 \cap 0.708) \\
 &= \underline{\mathbf{0.708}}
 \end{aligned}$$

Berdasarkan rumus 2.21 maka:

$$(0.708 - 0,3)/0,6 = z_2 \rightarrow z_2 = \mathbf{68 \%}$$

Setelah dilakukan pengujian maka domain yang mendekati hasil diagnosa dokter adalah intensitas fuzzy RENDAH memiliki domain [0, 4], SEDANG memiliki domain [2, 8], dan TINGGI memiliki domain [4, 10], sehingga di dapatkan rata-rata tingkat keakuratannya dalam sistem fuzzy inference system untuk mendiagnosa penyakit infeksi pada manusia menggunakan metode tsukamoto adalah sebesar 71%.

BAB V

PENUTUP

5.1. Kesimpulan

Adapun kesimpulan yang dapat diambil dari penelitian ini antara lain:

1. Domain yang mendekati hasil diagnosa dokter adalah intensitas fuzzy RENDAH memiliki domain $[0, 4]$, SEDANG memiliki domain $[2, 8]$, dan TINGGI memiliki domain $[4, 10]$ sehingga di hasilkan rata-rata adalah 71%.
2. Aplikasi sistem fuzzy inference system dapat digunakan user untuk membantu mengetahui gejala-gejala yang menyertai suatu penyakit setelah diprediksi penyakitnya diketahui serta dapat mengetahui penyebab, cara pencegahan serta cara pengobatan.
3. Dengan sistem fuzzy inference system yang menggunakan metode tsukamoto dapat digunakan untuk membantu user dalam menentukan terjadinya suatu penyakit infeksi berdasarkan input gejala-gejala beserta intensitasnya yang telah di berikan oleh user.

5.2. Saran

Dari sistem yang telah dibuat ini, dapat dikemukakan beberapa saran untuk pengembangan di masa depan, antara lain:

1. Diperlukan penanganan yang lebih baik terhadap gejala-gejala yang mungkin tidak diisi oleh user, namun terdapat pada aturan.
2. Penggunaan sistem fuzzy inference system untuk kasus dengan data yang lebih kualitatif (data lebih mudah dan pasti dalam penghitungannya, misalnya usia, suhu, tekanan darah, dan lain-lain), karena sistem yang dibuat ini tergantung dengan persepsi manusia (yang tidak sama ukurannya antar manusia dalam mengintrepresikan intensitas gejala).
3. Penggunaan metode lain seperti Mamdani dan Sugeno juga dapat diaplikasikan sehingga bisa didapatkan metode yang paling sesuai.

DAFTAR PUSTAKA

- [1]. Ichwan, M. 2011. *Pemrograman Basis data Delphi 7 dan My SQL*. Informatika. Bandung.
- [2]. Huda, Miftakhul. Dkk. 2008. *Membuat aplikasi penjualan dengan java dan My SQL*. PT elex media komputindo. Jakarta.
- [3]. Naba, Agus. 2009. *Belajar Cepat Fuzzy Logic Menggunakan Matlab*. Andi Offset. Yogyakarta.
- [4]. Kusumadewi, Sri. 2010. *Aplikasi Logika Fuzzy untuk Pendukung Keputusan*. Graha Ilmu. Yogyakarta.
- [5]. Ladjamudin, Al Bahra Bin. 2006. *Rekayasa Perangkat Lunak*. Graha Ilmu. Yogyakarta.
- [6]. Soedarto DTMH. 1990. *Penyakit – penyakit infeksi di Indonesia*. Widya Medika. Jakarta.
- [7]. Stimata. 2007. *Modul Pemrograman Borland Delphi*. Stimata. Malang.
- [8]. Stimata. 2007. *Modul Pemrograman Data Base My SQL*. Stimata. Malang.
- [9]. Apotik online dan media informasi obat – penyakit
[Www.medicastore.com](http://www.medicastore.com), “ Informasi Jenis Penyakit infeksi”.
Diakses tanggal 23 Maret 2011
- [10]. Horstkotte, Erik. 2000. Fuzzy Expert Systems.
<http://www.austinlinks.com/Fuzzy/expert-systems.html>
diakses tanggal 19 April 2012
- [11]. Sugianto, Feky. 2005. *perencanaan dan pembuatan fuzzy expert system untuk analisa penyakit dalam pada manusia*. Universitas Kristen Petra. Surabaya.
http://dewey.petra.ac.id/jiunkpe_dg_2765.html
Diakses tanggal 21 agustus 2011
- [12]. Wikipedia Indonesia. *Penyakit Menular*
http://id.wikipedia.org/wiki/Penyakit_infeksi
Diakses pada tanggal 8 Desember 2011

LAMPIRAN

LAMPYRIN



PT. BNI (PERSERO) MALANG
BANK NIAGA MALANG

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

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

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

**BERITA ACARA UJIAN SKRIPSI
FAKULTAS TEKNOLOGI INDUSTRI**

NAMA : PUTRI SARI SISWANTININGRUM
NIM : 06.12.588
PROGRAM STUDI : Teknik Elektro S-1
KONSENTRASI : Teknik Komputer dan Informatika
MASA BIMBINGAN : Semester Genap Tahun Akademik 2011 - 2012
JUDUL : **FUZZY INFERENCE SYSTEM UNTUK
MENDIAGNOSA PENYAKIT INFEKSI PADA
MANUSIA DENGAN METODE TSUKAMOTO**

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

Hari : Rabu
Tanggal : 8 Agustus 2012
Dengan Nilai : 86,25 (A) *or*

PANITIA UJIAN SKRIPSI

Ketua Majelis Penguji,

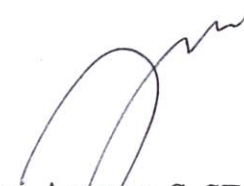

Ir. Yusuf Ismail Nakhoda, MT
NIP.Y.1018800189

Sekretaris Majelis Penguji,


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

ANGGOTA PENGUJI

Dosen Penguji I


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

Dosen Penguji II


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



INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
PROGRAM STUDI TEKNIK ELEKTRO S-1
Jl. Karanglo KM. 2 Malang

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 : PUTRI SARI S.
NIM : 06.12.588
PROGRAM STUDI : Teknik Komputer dan Informatika S-1
JUDUL : FUZZY INFERENCE SYSTEM UNTUK MENDIAGNOSA
PENYAKIT INFEKSI PADA MANUSIA
DENGAN METODE TSUKAMOTO

No	Penguji	Tanggal	Uraian	Paraf
1.	Penguji I	8 Agustus 2012	1. Software disempurnakan (inputan gejala lebih dari 3) 2. Pengujian diperbanyak. 3. Perbandingan hitung manual dengan software yang dibuat (satu/dua kasus)	
2.	Penguji II	8 Agustus 2012	1. Tabel di pengujian di tambahkan contoh cara perhitungannya.	

Disetujui :

Dosen Penguji I

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

Dosen Penguji II

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

Mengetahui :

Dosen Pembimbing I

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

Dosen Pembimbing II

Yuli Wahyuni, ST, MT
NIP.P. 1031200456



INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK ELEKTRO

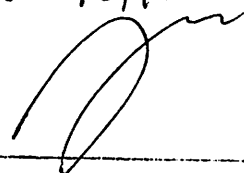
Formulir Perbaikan Ujian Skripsi

Dalam pelaksanaan Ujian Skripsi Janjang Strata 1 Jurusan Teknik Elektro Konsentrasi T. Energi Listrik / T. Elektronika / T. Infokom, maka perlu adanya perbaikan skripsi untuk mahasiswa :

NAMA : Putri Sari S.
NIM : 0612588
Perbaikan meliputi :

- Software disempurnakan (inputan gejala)
- Pengujian diperbanyak.
- Perbandingan hitung manual dg software dg dibuat (satu / dua kasus)

Malang, 8/8/12

()
Aryuanto



INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK ELEKTRO

Formulir Perbaikan Ujian Skripsi

Dalam pelaksanaan Ujian Skripsi Janjang Strata 1 Jurusan Teknik Elektro Konsentrasi T. Energi Listrik / T. Elektronika / T. Infokom, maka perlu adanya perbaikan skripsi untuk mahasiswa :

NAMA
NIM
Perbaikan meliputi

Putri Surti Siswaningrum.
0612588

tabel di bagian dan tambahan control cara
perhitungannya.

Malang,

8 agustus 2012


M. Ibrahim A. Syahki



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

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

BNI (PERSERO) MALANG
BANK NIAGA MALANG

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

Nomor Surat : ITN-205/EL-FTI/2012
Lampiran : -
Perihal : BIMBINGAN SKRIPSI

Kepada : Yth. Bapak/Ibu Yuli Wahyuni, ST, MT
Dosen Teknik Elektro S-1
ITN MALANG

Dengan Hormat

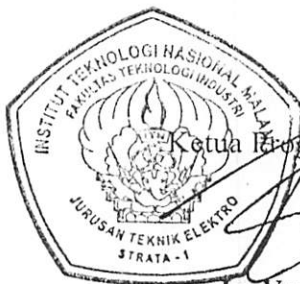
Sesuai dengan permohonan dan persetujuan dalam Proposal Skripsi untuk mahasiswa :

Nama : PUTRI SARI SISWANTININGRUM
Nim : 0612588
Fakultas : Teknologi Industri
Program Studi : Teknik Elektro S-1
Konsentrasi : Teknik Komputer & Informatika

Maka dengan ini pembimbingan tersebut kami serahkan sepenuhnya kepada Saudara/i selâma masa waktu :

" Semester Genap Tahun Akademik 2011-2012 "

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



Mengetahui

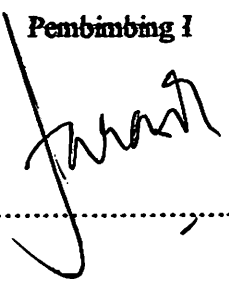
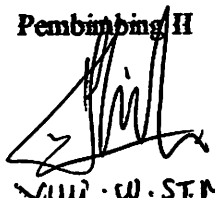
Ketua Program Studi Teknik Elektro S-1

Ir. Yusuf Ismail Nakhoda, MT

NIP.Y. 1018800189



BERITA ACARA SEMINAR PROPOSAL SKRIPSI
PROGRAM STUDI TEKNIK ELEKTRO S-1
Konsentrasi : Teknik Komputer & Informatika

1.	Nim	: 0612588		
2.	Nama	: PUTRI SARISISWANTININGRUM		
3.	Konsentrasi Jurusan	: Teknik Komputer & Informatika		
4.	Jadwal Pelaksanaan:	Waktu	Tempat	
	28 April 2012	09:00	III.1.3	
5.	Judul proposal yang diseminarkan Mahasiswa	FUZZY INFERENCE SYSTEM UNTUK MENDIAGNOSA PENYAKIT INFEKSI PADA MANUSIA MENGGUNAKAN METODE TSUKAMOTO		
6.	Perubahan judul yang dusulkan oleh Kelompok Dosen Keahlian			
7.	Catatan : flowchar 2.5 . direvisi . OFD, salah → flow chart . gambar → utk masy umum .			
	Catatan :			
	Persetujuan judul Skripsi			
	Disetujui, Dosen Keahlian I	Disetujui, Dosen Keahlian II	Disetujui, Dosen Keahlian III	
8.	 (.....)	(.....)	(.....)	
	Mengetahui, Ketua Program Studi Teknik Elektro S-1 <u>Ir. Yusuf Ismail Nakhoda, MT</u> NIP. Y. 1018800189	Disetujui, Calon Dosen Pembimbing ybs		
		Pembimbing I  (.....)	Pembimbing II  (..... Yuni W. S.F.M.T.)	



PROGRAM STUDI TEKNIK ELEKTRO S-1
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
Kampus II : Jl. Raya Karanglo Km. 2 Telp. (0341) 417636 Malang

FORMULIR BIMBINGAN SKRIPSI

n : 0612588
na : Putri Sari Siswantiningrum
sa : Semester Genap 2011-2012
bimbingan :
ul : **FUZZY INFERENCE SYSTEM UNTUK MENDIAGNOSA PENYAKIT INFEKSI
PADA MANUSIA MENGGUNAKAN METODE TSUKAMOTO**

Tanggal	Uraian	Paraf Pembimbing
5 Juli 2012	Blok diagram diganti DFD. Revisi BAB III	
6 Juli 2012	Ace BAB III	
3 Juli 2012	Revisi BAB IV ttg pengujian keseluruhan	
5 Juli 2012	Ace seminar kecil.	
5 Juli 2012	Revisi bab IV	
Agustus 2012	Ace koreksi	
	Ace revisi	

Malang,
Dosen Pembimbing

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



PROGRAM STUDI TEKNIK ELEKTRO S-1

FAKULTAS TEKNOLOGI INDUSTRI

INSTITUT TEKNOLOGI NASIONAL MALANG

Kampus II : Jl. Raya Karanglo Km. 2 Telp. (0341) 417636 Malang

FORMULIR BIMBINGAN SKRIPSI

Nim : 0612588
Nama : Putri Sari Siswantiningrum
Masa Bimbingan : Semester Genap 2011-2012
Judul : FUZZY INFERENCE SYSTEM UNTUK MENDIAGNOSA PENYAKIT INFEKSI PADA MANUSIA MENGGUNAKAN METODE TSUKAMOTO

Tanggal	Uraian	Paraf Pembimbing
16 juli 2012	Bab I →hidakusah menakai pengertian dtd. Bab III → DFD, contoh.	
28 juli 2012	Revisi abstraksi	
30 juli 2012	ACC seminar hasil	
4 agustus 2012	Demo program dan acc kompre	

Malang,

Dosen Pembimbing

Yuli Wahyuni, ST, MT

NIP.P. 1031200456

Form uji

```
unit Unit_uji;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, DB, ADODB, StdCtrls, Buttons, ExtCtrls, CheckLst;

type
  TForm_Uji = class(TForm)
    sPanel1: TPanel;
    Image1: TImage;
    sLabelFX1: TLabel;
    sLabelFX2: TLabel;
    sPanel2: TPanel;
    BtnNext: TBitBtn;
    btnDiag: TBitBtn;
    Panel1: TPanel;
    con: TADOQuery;
    g1: TCheckListBox;
    Panel2: TPanel;
    RadioGroup1: TRadioGroup;
    RadioGroup2: TRadioGroup;
    RadioGroup3: TRadioGroup;
    Panel3: TPanel;
    Memo1: TMemo;
    procedure FormCreate(Sender: TObject);
    procedure g1Click(Sender: TObject);
    procedure g1ClickCheck(Sender: TObject);
    procedure btnDiagClick(Sender: TObject);
    procedure FormClose(Sender: TObject; var Action: TCloseAction);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form_Uji: TForm_Uji;
  mode :integer;
implementation

uses Unit_Main;

{$R *.dfm}

procedure TForm_Uji.FormCreate(Sender: TObject);
var i :integer;
begin
```

```
con.close;
con.sql.clear;
con.sql.add('SELECT * FROM `tb_gejala` ');
con.Prepared;
con.open;
g1.Items.Clear;
```

```
if con.recordcount > 0 then
begin
  for i := 1 to con.RecordCount do
  begin
    g1.Items.Add(con.fieldByName('id_gejala').AsString+' |'
      + con.fieldByName('gejala').AsString);

    con.Next;
  end;
end;

mode := 0;
end;
```

```
procedure TForm_Uji.g1Click(Sender: TObject);
var i : integer;
    jml : integer;
```

```
begin
  jml := 0;
  for i:=0 to g1.Items.Count-1 do
  begin

    if (g1.Checked[i]) then
    begin
      inc(jml);
      if (jml=1) then RadioGroup1.Caption := g1.Items.Strings[i];
      if (jml=2) then RadioGroup2.Caption := g1.Items.Strings[i];
      if (jml=3) then RadioGroup3.Caption := g1.Items.Strings[i];
    end;

    end;
    if jml = 3 then btnDiag.Visible := true else btnDiag.Visible :=false;
    mode := 0;
  end;
```

```
procedure TForm_Uji.g1ClickCheck(Sender: TObject);
begin
  g1Click(self);
end;
```

```
procedure TForm_Uji.btnDiagClick(Sender: TObject);
```

```

var s1,s2,s3,tmp : string;
i:integer;
begin

if mode = 0 then
begin
    Panel2.Show;
    g1.Hide;
    mode := 1;
end else if mode = 1 then
begin
    Panel2.Hide;
    Panel3.Show;
    mode := 2;
    s1 := trim(copy(RadioGroup1.Caption ,0,pos(' ',RadioGroup1.Caption)));
    s2 := trim(copy(RadioGroup2.Caption ,0,pos(' ',RadioGroup2.Caption)));
    s3 := trim(copy(RadioGroup3.Caption ,0,pos(' ',RadioGroup3.Caption)));
    Memo1.Lines.Clear;
    con.close;
    con.sql.clear;
    con.sql.add('SELECT id_gp, penyakit, jenis, solusi FROM tb_gejala_penyakit
INNER JOIN tb_rule ON tb_gejala_penyakit.id_gp = tb_rule.id_rule INNER JOIN
tb_penyakit ON tb_rule.id_penyakit = tb_penyakit.id_penyakit and (id_gejala like
"%'+trim(s1)+'%" and id_gejala like "%'+trim(s2)+'%" and id_gejala like
"%'+trim(s3)+'%" ) GROUP BY tb_rule.id_penyakit');
    con.Prepared;
    con.open;

    if con.recordcount >0 then
    begin
        Memo1.Lines.Add('Data Aturan Penyakit Ditemukan :');

```

```

Memo1.Lines.Add('=====');
Memo1.Lines.Add('=====');
for i := 1 to con.RecordCount do
begin
    begin
        Memo1.Lines.Add('Penyakit : '+ con.fieldByName('penyakit').AsString);
        Memo1.Lines.Add('Jenis : '+ con.fieldByName('jenis').AsString);

        Memo1.Lines.Add("");
        con.Next;
    end;
    Memo1.Lines.Add('-----');
end;

```

```

Memo1.Lines.Add('=====');
Memo1.Lines.Add('=====');

```

```
end else  
begin
```

```
Memo1.Lines.Add('=====');  
=====
```

```
    Memo1.Lines.Add('Tidak Ada Penyakit yang Ditemukan berdasarkan gejala yang  
dipilih :');
```

```
Memo1.Lines.Add('=====');  
=====
```

```
    Memo1.Lines.Add('Gejala 1 : ' +s1);
```

```
    Memo1.Lines.Add('Gejala 2 : ' +s2);
```

```
    Memo1.Lines.Add('Gejala 3 : ' +s3);
```

```
end;
```

```
    Memo1.Lines.Add("");
```

```
    Memo1.Lines.Add("");
```

```
end else if mode = 2 then
```

```
begin
```

```
end;
```

```
end;
```

```
procedure TForm_Uji.FormClose(Sender: TObject; var Action: TCloseAction);
```

```
begin
```

```
    FreeAndNil(FormMain.fUji);
```

```
end;
```

```
end.
```

Form Search

unit UnitCari;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, StdCtrls, Buttons, DB,
ADODB, Grids, ExtCtrls ;

type

TFormCari = class(TForm)

sLabel4: TLabel;

kode: TLabel;

sPanel2: TPanel;

sLabelFX1: TLabel;

Image1: TImage;

Nama: TEdit;

con: TADOQuery;

sBitBtn1: TBitBtn;

Memo1: TMemo;

Panel1: TPanel;

procedure FormClose(Sender: TObject; var Action: TCloseAction);

procedure sBitBtn1Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

FormCari: TFormCari;

implementation

uses Unit_Main;

{ \$R *.dfm }

procedure TFormCari.FormClose(Sender: TObject; var Action: TCloseAction);

begin

FormMain.fcari := nil;

Action := caFree;

end;

procedure TFormCari.sBitBtn1Click(Sender: TObject);

var i: integer;

begin

Memo1.Lines.Clear;

```

con.close;
con.sql.clear;
con.sql.add('SELECT      id_penyakit,      tb_penyakit.penyakit,      tb_penyakit.jenis,
tb_penyakit.solusi FROM tb_penyakit where penyakit like "%'+trim(nama.Text)+'%" ');
con.Prepared;
con.open;

```

```

if con.recordcount >0 then
begin

```

```

    Memo1.Lines.Add('Data Penyakit Ditemukan :');

```

```

Memo1.Lines.Add('=====');

```

```

    for i := 1 to con.RecordCount do
begin

```

```

        begin

```

```

            Memo1.Lines.Add('Penyakit
'+#13#10+con.fieldByName('penyakit').AsString);

```

```

            Memo1.Lines.Add('Jenis : '+#13#10+con.fieldByName('jenis').AsString);

```

```

            Memo1.Lines.Add('Keterangan
'+#13#10+con.fieldByName('solusi').AsString);

```

```

            Memo1.Lines.Add("");

```

```

            con.Next;

```

```

        end;

```

```

        Memo1.Lines.Add('-----');
end;

```

```

Memo1.Lines.Add('=====');

```

```

end;

```

```

Memo1.Lines.Add("");

```

```

Memo1.Lines.Add("");

```

```

con.close;

```

```

con.sql.clear;

```

```

con.sql.add('select * from tb_gejala where gejala like "%'+trim(nama.Text)+'%" ');

```

```

con.Prepared;

```

```

con.open;

```

```

if con.recordcount >0 then

```

```

begin

```

```

    Memo1.Lines.Add('Data Gejala Ditemukan :');

```

```

Memo1.Lines.Add('=====');

```

```

    for i := 1 to con.RecordCount do
begin

```

```

        begin

```

```
        Memo1.Lines.Add('Gejala  
'+#13#10+con.fieldByName('id_gejala').AsString);  
        Memo1.Lines.Add("");  
        con.Next;
```

```
    end;  
    Memo1.Lines.Add('-----');  
end;
```

```
Memo1.Lines.Add('=====');  
=====
```

```
    end;  
end;  
  
end.
```

Form Diagnosa

```
unit Unit_diagnosa;
```

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
Dialogs, ExtCtrls, StdCtrls, Buttons,  
DB, ADODB, math, jpeg;
```

```
type
```

```
THasil = record
```

```
kode : string;
```

```
MB : Double;
```

```
MD : Double;
```

```
CF : Double;
```

```
end;
```

```
type
```

```
TFormDiagnosa = class(TForm)
```

```
sPanel1: TPanel;
```

```
Image1: TImage;
```

```
sPanel2: TPanel;
```

```
Panel1: TPanel;
```

```
BtnNext: TBitBtn;
```

```
jawabya: TRadioButton;
```

```
jawabtidak: TRadioButton;
```

```
lb2: TLabel;
```

```
lb1: TLabel;
```

```
tanya: TLabel;
```

```
con: TADOQuery;
```

```
pnlHasil: TPanel;
```

```
lbketsakit: TLabel;
```

```
lbSakit: TLabel;
```

```
lbket: TLabel;
```

```
btnDiag: TBitBtn;
```

```
btnNew: TBitBtn;
```

```
Imagehasil: TImage;
```

```
Image2: TImage;
```

```
sLabelFX1: TLabel;
```

```
sLabelFX2: TLabel;
```

```
procedure jawabyaClick(Sender: TObject);
```

```
procedure FormCreate(Sender: TObject);
```

```
procedure BtnNextClick(Sender: TObject);
```

```
procedure btnNewClick(Sender: TObject);
```

```
procedure btnDiagClick(Sender: TObject);
```

```
procedure FormClose(Sender: TObject; var Action: TCloseAction);
```

```
procedure FormCloseQuery(Sender: TObject; var CanClose: Boolean);
```

```
procedure FormShow(Sender: TObject);
```

```
private
```

```

LPenyakit : TStringList;
currInde : integer;
currGp : String;
maxInde : integer;
curRule : string;
countTanya : integer;
kodePenyakit : string;
TResult : array of THasil;
procedure getData(kode :String);
public

end;

var
  FormDiagnosa: TFormDiagnosa;

implementation

uses Unit_uji , Unit_Main;

{$R *.dfm}

procedure TFormDiagnosa.jawabyaClick(Sender: TObject);
begin
  if jawabya.Checked <> jawabtidak.Checked then
    BtnNext.Visible := true;
end;

procedure TFormDiagnosa.FormCreate(Sender: TObject);
var i:integer;
begin
  LPenyakit := TStringList.Create;
  con.close;
  con.sql.clear;
  con.sql.add('SELECT tb_gejala_penyakit.id_penyakit as idp,MD,MB ' +
    ' FROM tb_rule, tb_gejala_penyakit ' +
    ' WHERE tb_rule.id_gp = tb_gejala_penyakit.id_gp AND ' +
    ' (tb_gejala_penyakit.id_gejala = "G00")');
  con.Prepared;
  con.open;
  if con.recordcount >0 then
    begin
      SetLength(TResult,con.RecordCount);
      for i := 1 to con.RecordCount do
        begin
          LPenyakit.Add(' '+ con.fieldByName('idp').AsString);
          TResult[i-1].kode := con.fieldByName('idp').AsString;

```

```

        TResult[i-1].MD := 0;
        TResult[i-1].MB := 0;
        con.Next;
    end;
    currInde := 0;
    maxInde := con.RecordCount;
    countTanya := 1;
end;
getData(LPenyakit[currinde]);

end;

procedure TFormDiagnosa.getData(kode:String);
var i:integer;
begin
    con.close;
    con.sql.clear;
    con.sql.add('SELECT
gejalas,tb_gejala_penyakit.id_gp idgp
                                MD,MB,id_rule,
                                tb_gejala.gejala
        'FROM tb_rule, tb_gejala_penyakit, tb_gejala, ' +
        ' tb_gejala_penyakit tb_gejala_penyakit1 ' +
        ' WHERE tb_rule.id_gp = tb_gejala_penyakit.id_gp AND ' +
        ' tb_gejala_penyakit.id_gejala = tb_gejala.id_gejala AND ' +
        ' tb_rule.id_rule sebelum = tb_gejala_penyakit1.id_gp AND ' +
        ' (tb_gejala_penyakit.id_penyakit = '"+trim(kode)+"') AND ' +
        ' (tb_gejala_penyakit.id_gejala <> "G00") AND ' +
        ' (tb_gejala_penyakit1.id_gejala = "G00")');
    con.Prepared;
    con.open;
    if con.recordcount >0 then
    begin
        for i := 1 to con.RecordCount do
        begin
            begin
                tanya.Caption:= con.fieldByName('gejalas').AsString;
                curRule := con.fieldByName('id_rule').AsString;
                currGp := con.fieldByName('idgp').AsString;
                con.Next;
            end;
        end
    end;
end;

end;
procedure TFormDiagnosa.BtnNextClick(Sender: TObject);
var i,tempIdx:integer;
    tempIdY,tempIdSblm : string;
    tempMax : Double;
begin

```

```

if (countTanya>3) or (currInde>= maxInde) then
begin
  for i:= 0 To maxInde-1 do
  begin
    Tresult[i].CF := Tresult[i].MB-Tresult[i].MD/(1-min(Tresult[i].MB,Tresult[i].MD));
    end;
  tempMax :=Tresult[i].CF;
  tempIdx :=0;
  for i:= 0 To maxInde-1 do
  begin
    if Tresult[i].CF > tempMax then
    begin
      tempMax := Tresult[i].CF;
      tempIdx := i;
    end;
  end;
  con.close;
  con.sql.clear;
  con.sql.add('SELECT nama, keterangan FROM tb_penyakit WHERE (id_penyakit =
'+Tresult[tempIdx].kode+'" );

  con.Prepared;
  con.open;

  if con.recordcount >0 then
  begin
    lbketsakit.Caption :='Hasil Diagnosa Awal, Penyakit Anda: '// [CF:
+FloatToStr(Tresult[tempIdx].CF)+']';
    lbSakit.Caption :=con.fieldByName('nama').AsString;
    lbket.Caption :=con.fieldByName('keterangan').AsString;
    Imagehasil.Picture.LoadFromFile(ExtractFilePath(Application.ExeName)
    +'images\' + Tresult[tempIdx].kode+'.jpg' );
    kodePenyakit := Tresult[tempIdx].kode;

    btnDiag.Show;
    BtnNext.Hide;

  end
  else
  begin
    lbSakit.Caption :='Tidak Sakit';
    lbket.Caption :='Lakukan diagnosa ulang untuk memastikan';
    Imagehasil.Picture.LoadFromFile(ExtractFilePath(Application.ExeName)
    +'images\2.jpg' );
  end;

  pnlHasil.Show;
  exit;
end;
if jawabya.Checked=true then

```

```

begin
    con.close;
    con.sql.clear;
    con.sql.add('SELECT id_rule_ya, id_rule,MD,MB, ' +
        ' gejala, tb_rule.id_gp AS idp ' +
        ' FROM tb_rule, tb_gejala_penyakit, tb_gejala ' +
        ' WHERE tb_rule.id_rule_ya = tb_gejala_penyakit.id_gp AND ' +
        ' tb_gejala_penyakit.id_gejala = tb_gejala.id_gejala AND ' +
        ' (tb_rule.id_rule = '"+curRule+"'"));

    con.Prepared;
    con.open;
    if con.recordcount > 0 then
        begin
            for i := 1 to con.RecordCount do
                begin
                    tanya.Caption:= con.fieldByName('gejala').AsString;
                    tempIdY := con.fieldByName('id_rule_ya').AsString;
                    currGp := con.fieldByName('idp').AsString;
                    con.Next;

                    end;
                end
            end;

        con.close;
        con.sql.clear;
        con.sql.add('SELECT id_rule, id_rule_sebelum ,MD,MB FROM tb_rule ' +
            ' WHERE (id_gp = '"+tempIdY+"' ) and (id_rule_sebelum = '"+currGp+"' ) );

        con.Prepared;
        con.open;
        if con.recordcount > 0 then
            begin
                TResult[currInde].MB:= TResult[currInde].MB+con.fieldByName('MB').AsFloat;
                curRule := con.fieldByName('id_rule').AsString;
            end;

            inc(countTanya);
        end
    else
        begin
            countTanya := 1;
            con.close;
            con.sql.clear;
            con.sql.add('SELECT id_rule_ya, id_rule,MD,MB, ' +
                ' gejala, tb_rule.id_gp AS idp ' +
                ' FROM tb_rule, tb_gejala_penyakit, tb_gejala ' +
                ' WHERE tb_rule.id_rule_ya = tb_gejala_penyakit.id_gp AND ' +

```

```

'   tb_gejala_penyakit.id_gejala = tb_gejala.id_gejala AND ' +
'   (tb_rule.id_rule = ""+curRule+"");

con.Prepared;
con.open;
TResult[currInde].MD:= TResult[currInde].MD+con.fieldByName('MD').AsFloat;
inc(currInde);
if (currInde>= maxInde) then
begin
for i:= 0 To maxInde-1 do
begin
Tresult[i].CF := Tresult[i].MB-Tresult[i].MD/(1-min(Tresult[i].MB,Tresult[i].MD));
end;
tempMax :=Tresult[i].CF;
tempIdx :=0;
for i:= 0 To maxInde-1 do
begin
if Tresult[i].CF > tempMax then
begin
tempMax := Tresult[i].CF;
tempIdx := i;
end;
end;
con.close;
con.sql.clear;
con.sql.add('SELECT nama, keterangan FROM tb_penyakit WHERE (id_penyakit =
""+Tresult[tempIdx].kode+"") ');

con.Prepared;
con.open;

if con.recordcount >0 then
begin
lbketsakit.Caption :='Hasil Diagnosa Awal, Penyakit Anda [CF:
'+FloatToStr(Tresult[tempIdx].CF)+']';
lbSakit.Caption :=con.fieldByName('nama').AsString;
lbket.Caption :=con.fieldByName('keterangan').AsString;
Imagehasil.Picture.LoadFromFile(ExtractFilePath(Application.ExeName)
+'images\' + Tresult[tempIdx].kode+'.jpg' );
end
else
begin
lbSakit.Caption :='Tidak Sakit';
lbket.Caption :='Lakukan diagnosa ulang untuk memastikan';
Imagehasil.Picture.LoadFromFile(ExtractFilePath(Application.ExeName)
+'images\2.jpg' );
end;

pnlHasil.Show;
exit;

```

```

    end;
    getData(LPenyakit.ValueFromIndex[currinde]);
end;
begin

end;
jawabya.Checked := false;
jawabtidak.Checked := false;
BtnNext.Hide;
end;

procedure TFormDiagnosa.btnNewClick(Sender: TObject);
var i:integer;
begin
    kodePenyakit := '';
    jawabya.Checked := false;
    jawabtidak.Checked := false;
    btnDiag.Hide;
    BtnNext.Hide;
    pnlHasil.Hide;
    LPenyakit := TStringList.Create;
    con.close;
    con.sql.clear;
    con.sql.add('SELECT tb_gejala_penyakit.id_penyakit as idp,MD,MB ' +
        ' FROM tb_rule, tb_gejala_penyakit ' +
        ' WHERE tb_rule.id_gp = tb_gejala_penyakit.id_gp AND ' +
        ' (tb_gejala_penyakit.id_gejala = "G00")');
    con.Prepared;
    con.open;
    if con.recordcount > 0 then
    begin
        SetLength(TResult,con.RecordCount);
        for i := 1 to con.RecordCount do
        begin
            LPenyakit.Add(' + con.fieldByName('idp').AsString);
            TResult[i-1].kode := con.fieldByName('idp').AsString;
            TResult[i-1].MD := 0;
            TResult[i-1].MB := 0;
            con.Next;
        end;
        currInde := 0;
        maxInde := con.RecordCount;
        countTanya := 1;
    end;
    getData(LPenyakit[currinde]);
end;

procedure TFormDiagnosa.btnDiagClick(Sender: TObject);
begin
    if (FormMain.fuji=nil) then

```

```

FormMain.fuji := TForm_Uji.Create(Application);

FormMain.fuji.Show;
Formmain.fdiag.WindowState := wsMinimized;

end;

procedure TFormDiagnosa.FormClose(Sender: TObject;
  var Action: TCloseAction);
begin
  FormMain.freg.WindowState := wsNormal;
  FormMain.fdiag := nil ;
  Action := caFree;
end;

procedure TFormDiagnosa.FormCloseQuery(Sender: TObject;
  var CanClose: Boolean);
begin
  if (MessageDlg('Yakin Untuk Mengakhiri Proses?',mtConfirmation,[mbYes,mbNo],0))=
mrNo then
    CanClose := False;
end;

procedure TFormDiagnosa.FormShow(Sender: TObject);
begin
  BtnNextClick(self);
end;

end.

```

Form Data Gejala

unit Unit_gejala;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, DB, ADODB, StdCtrls, Buttons,
Grids, ExtCtrls ;

type

TFormGejala = class(TForm)

sLabel3: TLabel;

sLabel1: TLabel;

kode: TLabel;

sPanel2: TPanel;

sLabelFX1: TLabel;

Image1: TImage;

idgejala: TEdit;

Data: TStringGrid;

sPanel1: TPanel;

BtnDel: TBitBtn;

BtnClear: TBitBtn;

BtnSave: TBitBtn;

ket: TMemo;

con: TADOQuery;

procedure FormCreate(Sender: TObject);

procedure BtnClearClick(Sender: TObject);

procedure idgejalaChange(Sender: TObject);

procedure DataSelectCell(Sender: TObject; ACol, ARow: Integer;
var CanSelect: Boolean);

procedure FormClose(Sender: TObject; var Action: TCloseAction);

procedure BtnDelClick(Sender: TObject);

procedure BtnSaveClick(Sender: TObject);

private

procedure getCode();

procedure getData();

public

{ Public declarations }

end;

var

FormGejala: TFormGejala;

implementation

uses Unit_Main;

{\$R *.dfm}

procedure TFormGejala.getCode();

```

var
  Kodes : String;
  i : Integer;
  found : boolean;

begin

  i := 1;
  Kodes := FormatCurr('G000',1);
  idgejala.Text := Kodes;
  found := false;
  While Not found do
  begin
    Kodes := FormatCurr('G000',i);
    con.close;
    con.sql.clear;
    con.sql.add('SELECT * FROM tb_gejala WHERE id_gejala="'+kodes+'" ');
    con.Prepared;
    con.open;
    if con.recordcount >0 then
      found := False
    Else
      begin
        idgejala.Text := Kodes;
        found := True;
      end;
    i := i + 1;
  end;

end;

procedure TFormGejala.FormCreate(Sender: TObject);
begin
  with data do
    begin
      ColCount := 2;
      Cells[0,0] := 'ID.';
      Cells[1,0] := 'Gejala';
      ColWidths[1] := 500;

    end;
    getCode;
    BtnClearClick(sender);
  end;

procedure TFormGejala.getData();
var i:integer;
begin
  con.close;
  con.sql.clear;
  con.sql.add('select * from tb_gejala ');
  con.Prepared;

```

```

con.open;
Data.RowCount := 1;
Data.RowCount := 2;
Data.FixedRows := 1;
if con.recordcount > 0 then
begin
  for i := 1 to con.RecordCount do
  begin
    with data do
    begin
      Cells[0,i] := con.fieldByName('id_gejala').AsString;
      Cells[1,i] := con.fieldByName('gejala').AsString;
      RowCount := i + 1;
      con.Next;
    end;
  end
end;
end;
procedure TFormGejala.BtnClearClick(Sender: TObject);
begin
  getData;
  idgejala.Text := '';
  BtnDel.Visible := False;
  BtnSave.Visible := false;
  idgejala.Text := '';
  BtnSave.Caption := 'Simpan';
  getCode;

  ket.Text := '';
  kode.Caption := '';
end;

procedure TFormGejala.idgejalaChange(Sender: TObject);
begin
  If (trim(ket.Text) <> '') then
    BtnSave.Visible := true
  else
    BtnSave.Visible := false;
  if (BtnSave.Caption = 'Edit') then
  begin
    BtnDel.Visible := false;
    BtnSave.Caption := 'Simpan';
  end
end;

procedure TFormGejala.DataSelectCell(Sender: TObject; ACol, ARow: Integer;
var CanSelect: Boolean);
begin
  kode.Caption := IntToStr(ARow);
  BtnSave.Caption := 'Edit';

```

```

    BtnSave.Visible := True;
    BtnDel.Visible := true
end;

procedure TFormGejala.FormClose(Sender: TObject; var Action: TCloseAction);
begin
    FormMain.fgejala:= nil;
    Action := caFree;
end;

procedure TFormGejala.BtnDelClick(Sender: TObject);
begin
    if (MessageDlg('Yakin Untuk Menghapus data?',mtConfirmation,[mbYes,mbNo],0))=
mrYes then
        begin
            try
                con.close;
                con.sql.clear;
                con.sql.add('Delete from tb_gejala where id_gejala= ""'+
data.Cells[0,strtoint(kode.Caption)] +' " ');
                con.ExecSQL;

                MessageDlg('Proses Penghapusan Berhasil',mtInformation,[ mbOK],0 );
                BtnClearClick(self);

            except
                MessageDlg('Proses Penghapusan Gagal',mtError,[ mbOK],0 );
            exit;
            end;
        end
end;

procedure TFormGejala.BtnSaveClick(Sender: TObject);
var aRow : integer;

begin
    if (MessageDlg('Yakin Untuk '+ BtnSave.Caption +'
data?',mtConfirmation,[mbYes,mbNo],0))= mrYes then
        begin
            try
                if BtnSave.Caption='Simpan' then
                    begin
                        con.close;
                        con.sql.clear;
                        con.sql.add('INSERT INTO tb_gejala ( `id_gejala`, `gejala`) VALUES
('"+trim(idgejala.Text )+"', '"+trim(ket.Text )+" '));
                        con.ExecSQL;

```

```

end
else If BtnSave.Caption ='Edit' then
begin
    aRow := StrToInt(kode.Caption);
    with data do
        begin
            idgejala.Text := Cells[0,arow] ;
            ket.Text := Cells[1,arow] ;

            end;

        BtnSave.Caption := 'Update';
        BtnDel.Visible := False;
        exit;
    end
    else if BtnSave.Caption='Update' then
    begin
        con.close;
        con.sql.clear;
        con.sql.add('UPDATE tb_gejala SET  gejala = '"+trim(ket.Text )+"'
WHERE ('id_gejala'='"+idgejala.Text +'")');
        con.ExecSQL;
        end;
        MsgBox('Proses Penyimpanan Berhasil',mtInformation,[ mbOK],0 );
        BtnClearClick(self);

    except
        MsgBox('Proses Penyimpanan Gagal',mtError,[ mbOK],0 );
        exit;
        end;
    end
    else
    begin
        BtnClearClick(self);
    end;
end;

end.

```

Form Rule

unit Unit_Gejala_Penyakit;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, DB, ADODB, StdCtrls, Buttons , ExtCtrls, sLabel, CheckLst, Grids;

type

TFormGejalaPenyakit = class(TForm)

sLabel4: TLabel;

kode: TLabel;

sPanel2: TPanel;

sLabelFX1: TLabel;

Image1: TImage;

Data: TStringGrid;

sPanel1: TPanel;

BtnDel: TBitBtn;

BtnClear: TBitBtn;

BtnSave: TBitBtn;

sLabel1: TLabel;

penyakit: TComboBox;

con: TADOQuery;

g1: TComboBox;

g2: TComboBox;

Label1: TLabel;

g3: TComboBox;

Label2: TLabel;

sLabel3: TLabel;

idgejala: TEdit;

procedure FormCreate(Sender: TObject);

procedure BtnClearClick(Sender: TObject);

procedure BtnSaveClick(Sender: TObject);

procedure penyakitChange(Sender: TObject);

procedure DataSelectCell(Sender: TObject; ACol, ARow: Integer;
var CanSelect: Boolean);

procedure BtnDelClick(Sender: TObject);

procedure FormClose(Sender: TObject; var Action: TCloseAction);

private

procedure getData();

procedure getCode();

public

{ Public declarations }

end;

var

FormGejalaPenyakit: TFormGejalaPenyakit;

implementation

```
uses Unit_Main;
```

```
{ $R *.dfm }
```

```
procedure TFormGejalaPenyakit.getData();
```

```
var i:integer;
```

```
begin
```

```
    con.close;
```

```
    con.sql.clear;
```

```
    con.sql.add('SELECT      tb_rule.id_rule      id,      tb_rule.keterangan      p,  
tb_gejala_penyakit.id_gejala idg FROM tb_rule INNER JOIN tb_gejala_penyakit ON  
tb_rule.id_rule = tb_gejala_penyakit.id_gp ');
```

```
    con.Prepared;
```

```
    con.open;
```

```
    Data.RowCount := 1;
```

```
    Data.RowCount := 2;
```

```
    Data.FixedRows := 1;
```

```
    if con.recordcount > 0 then
```

```
    begin
```

```
        for i := 1 to con.RecordCount do
```

```
        begin
```

```
            with data do
```

```
            begin
```

```
                Cells[0,i] := con.fieldByName('id').AsString;
```

```
                Cells[1,i] := con.fieldByName('p').AsString;
```

```
                Cells[2,i] := con.fieldByName('idg').AsString;
```

```
                RowCount := i+1;
```

```
                con.Next;
```

```
            end;
```

```
        end
```

```
    end;
```

```
con.close;
```

```
con.sql.clear;
```

```
con.sql.add('SELECT * FROM `tb_gejala` ');
```

```
con.Prepared;
```

```
con.open;
```

```
g1.Items.Clear;
```

```
g2.Items.Clear;
```

```
g3.Items.Clear;
```

```
if con.recordcount > 0 then
```

```
begin
```

```
    for i := 1 to con.RecordCount do
```

```
    begin
```

```
        g1.Items.Add(con.fieldByName('id_gejala').AsString+' |'  
+ con.fieldByName('gejala').AsString);
```

```
        g2.Items.Add(con.fieldByName('id_gejala').AsString+' |'
```

```

        + con.fieldByName('gejala').AsString);
    g3.Items.Add(con.fieldByName('id_gejala').AsString+' |'
        + con.fieldByName('gejala').AsString);
    con.Next;
end;
g1.ItemIndex := 0;
g2.ItemIndex := 0;
g3.ItemIndex := 0;
end;

con.close;
con.sql.clear;
con.sql.add('SELECT * FROM `tb_penyakit` ');
con.Prepared;
con.open;
penyakit.items.Clear;
if con.recordcount > 0 then
begin
    for i := 1 to con.RecordCount do
    begin
        penyakit.Items.Add(con.fieldByName('id_penyakit').AsString+' |'
            + con.fieldByName('penyakit').AsString);
        con.Next;
    end;
    penyakit.ItemIndex := 0;
end;

end;
procedure TFormGejalaPenyakit.getCode();
var
    Kodes : String;
    i : Integer;
    found : boolean;

begin
    i := 1;
    Kodes := FormatCurr('R00',1);
    idgejala.Text := Kodes;
    found := false;
    While Not found do
    begin
        Kodes := FormatCurr('R000',i);
        con.close;
        con.sql.clear;
        con.sql.add('SELECT      `id_gp`      FROM      `tb_gejala_penyakit`      WHERE
`id_gp`='''+kodes+''' ');
        con.Prepared;
        con.open;

```

```

    if con.recordcount > 0 then
        found := False
    Else
        begin
            idgejala.Text := Kodes;
            found := True;
        end;
        i := i + 1;
    end;

end;

procedure TFormGejalaPenyakit.FormCreate(Sender: TObject);
begin
    with data do
        begin
            ColCount := 4;
            Cells[0,0] := 'ID.';
            Cells[1,0] := 'Penyakit';
            Cells[2,0] := 'Gejala';

            ColWidths[1] := 120;
            ColWidths[2] := 500;
            ColWidths[2] := 1500;

            end;
        getData;
        BtnClearClick(self);
    end;

procedure TFormGejalaPenyakit.BtnClearClick(Sender: TObject);
begin
    getData;

    BtnDel.Visible := False;
    BtnSave.Visible := false;
    BtnSave.Caption := 'Simpan';
    kode.Caption := '';
    penyakit.Enabled := true;
    getCode();
end;

procedure TFormGejalaPenyakit.BtnSaveClick(Sender: TObject);
var i,j : integer;
    tempA,tempB :string;
begin
    if      (MessageDlg('Yakin      Untuk      '+'      BtnSave.Caption      '+'
data?',mtConfirmation,[mbYes,mbNo],0))= mrYes then
        begin
            try

```

```

if BtnSave.Caption='Simpan' then
begin

    con.close;
    con.sql.clear;
    tempA := trim(copy(penyakit.Text,0,pos(' ',penyakit.Text)));
    tempB := trim(copy(g1.Text,0,pos(' ',g1.Text)));
    tempB := tempB+' '+trim(copy(g2.Text,0,pos(' ',g2.Text)));
    tempB := tempB+' '+trim(copy(g3.Text,0,pos(' ',g3.Text)));
    { Memo1.Lines.Add('INSERT INTO `tb_gejala_penyakit` (
'id_gejala', `id_penyakit`) VALUES ("'+trim( tempB )+'" , "'+trim(tempA )+'" )');

    for i:=0 to gejala.Items.Count -1 do
    begin
        if (gejala.Checked[i]) then
        begin
            tempB :='G' + trim( copy(gejala.Items.ValueFromIndex[i],0,pos(' ',gejala.Items.ValueFromIndex[i])));
            con.SQL.Clear;
            con.sql.add('INSERT INTO `tb_gejala_penyakit` ( `id_gejala`,
'id_penyakit`) VALUES ("'+trim( tempB )+'" , "'+trim(tempA )+'" )');
            con.ExecSQL;
        end;
    end;
}
end
else If BtnSave.Caption ='Edit' then
begin

    { con.close;
    con.sql.clear;
    con.sql.add('SELECT * FROM `tb_gejala` ');
    con.Prepared;
    con.open;
    gejala.Clear;
    if con.recordcount >0 then
    begin
        for i := 1 to con.RecordCount do
        begin
            gejala.Items.Add(con.fieldByName('id_gejala').AsString+' '
            + con.fieldByName('gejala').AsString);
            con.Next;
        end

    end;

    con.close;
    con.sql.clear;
    con.sql.add('SELECT `id_gejala` FROM `tb_gejala_penyakit` WHERE
'id_penyakit'="'+ data.Cells[0,strtoint(kode.Caption)] +' "');

```

```

con.Prepared;
con.open;
if con.recordcount >0 then
begin
  for i := 1 to con.RecordCount do
  begin
    for j := 0 to gejala.Items.Count-1 do
    begin
      if con.fieldByName('id_gejala').AsString = 'G' + trim(
copy(gejala.Items.ValueFromIndex[j],0,pos(' ',gejala.Items.ValueFromIndex[j]))) then
      begin
        gejala.Checked[j] := True;
      end;
    end;
    con.Next;
  end;

  end;
}
penyakit.Enabled :=False;
BtnSave.Caption := 'Update';
BtnDel.Visible := False;
exit;
end
else if BtnSave.Caption='Update' then
begin
  {con.close;
  con.sql.clear;
  con.sql.add('Delete from `tb_gejala_penyakit` where id_penyakit= "'+
data.Cells[0,strtoint(kode.Caption)] +' " ');
  con.ExecSQL;
  for i:=0 to gejala.Items.Count -1 do
  begin
    if (gejala.Checked[i]) then
    begin
      tempB :='G' + trim( copy(gejala.Items.ValueFromIndex[i],0,pos(
',gejala.Items.ValueFromIndex[i])));
      con.SQL.Clear;
      con.sql.add('INSERT INTO `tb_gejala_penyakit` ( `id_gejala`,
`id_penyakit`) VALUES ("'+trim( tempB )+'", "'+trim(data.Cells[0,strtoint(kode.Caption)]
)+'" ) ');
      con.ExecSQL;
    end;
  end;
  }

  con.ExecSQL;
end;
MessageDlg('Proses Penyimpanan Berhasil',mtInformation,[ mbOK],0 );

```

```

        BtnClearClick(self);

    except
        MessageDlg('Proses Penyimpanan Gagal',mtError,[ mbOK],0 );
    exit;
    end;

end
else
begin
    BtnClearClick(self);
end;
end;

procedure TFormGejalaPenyakit.penyakitChange(Sender: TObject);
begin
    If (trim(penyakit.Text) <> "") then
        BtnSave.Visible := true
    else
        BtnSave.Visible := false;
    if (BtnSave.Caption = 'Edit') then
        begin
            BtnDel.Visible := false;
            BtnSave.Caption := 'Simpan';
        end
    end;
end;

procedure TFormGejalaPenyakit.DataSelectCell(Sender: TObject; ACol,
    ARow: Integer; var CanSelect: Boolean);
begin
    kode.Caption := IntToStr(ARow);
    BtnSave.Caption := 'Edit';
    BtnSave.Visible := True;
    BtnDel.Visible := true;

end;

procedure TFormGejalaPenyakit.BtnDelClick(Sender: TObject);
begin
    if (MessageDlg('Yakin Untuk Menghapus data?',mtConfirmation,[mbYes,mbNo],0))=
mrYes then
        begin
            try

                con.close;
                con.sql.clear;
                con.sql.add('Delete from `tb_gejala_penyakit` where id_penyakit= "' +
data.Cells[0,strtoint(kode.Caption)] +' " ');
                con.ExecSQL;
            except
            end;
        end;
    end;
end;

```

```
MessageDlg('Proses Penghapusan Berhasil',mtInformation,[ mbOK],0 );  
BtnClearClick(self);
```

```
except
```

```
    MessageDlg('Proses Penghapusan Gagal',mtError,[ mbOK],0 );
```

```
exit;
```

```
end;
```

```
end
```

```
end;
```

```
procedure TFormGejalaPenyakit.FormClose(Sender: TObject;  
    var Action: TCloseAction);
```

```
begin
```

```
    FormMain.fgejalPenyakit := nil;
```

```
    Action := caFree;
```

```
end;
```

```
end.
```

Form Login

```
unit Unit_Login;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, DB, ADODB, StdCtrls, Buttons, ExtCtrls;

type
  TFormLogin = class(TForm)
    sLabel3: TLabel;
    sLabel1: TLabel;
    kode: TLabel;
    sPanel2: TPanel;
    sLabelFX1: TLabel;
    Image1: TImage;
    IdUser: TEdit;
    Pass: TEdit;
    sPanel1: TPanel;
    BtnClear: TBitBtn;
    BtnSave: TBitBtn;
    con: TADOQuery;
    Timer1: TTimer;
    procedure BtnClearClick(Sender: TObject);
    procedure BtnSaveClick(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  FormLogin: TFormLogin;

implementation

uses Unit_Main;

{$R *.dfm}

procedure TFormLogin.BtnClearClick(Sender: TObject);
begin
  Application.Terminate;
end;

procedure TFormLogin.BtnSaveClick(Sender: TObject);
begin
  con.close;
  con.sql.clear;
```

```
con.sql.add('select * from tb_user where id_user="'+IdUser.Text+'" and  
pass="'+Pass.Text+'" ');  
con.Prepared;  
con.open;  
if con.recordcount >0 then  
begin  
hide;  
FormMain.isLogin := true;  
FormMain.Show;  
  
end else  
begin  
Application.MessageBox('Login Gagal.', 'Error', 0);  
end;  
end;  
  
end.
```

Form Main Menu

unit Unit_Main;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, ExtCtrls, Menus, ImgList, StdCtrls, jpeg,
ComCtrls,
ToolWin, Unit_Pasien, Unit_gejala, Unit_Gejala_Penyakit,
Unit_Penyakit, Unit_User, Unit_Ru, Unit_diagnosa, Unit_reg, UnitCari,
Unit_uji, acAlphaImageList;

type

TFormMain = class(TForm)

sPanel1: TPanel;

sLabelFX1: TLabel;

sLabelFX2: TLabel;

sPanel2: TPanel;

ImageHome: TImage;

MainMenu1: TMainMenu;

Master1: TMenuItem;

Penyakit1: TMenuItem;

User1: TMenuItem;

RuleBase1: TMenuItem;

Pasien1: TMenuItem;

Image1: TImage;

Gejala1: TMenuItem;

Search1: TMenuItem;

Exit2: TMenuItem;

Diagosa1: TMenuItem;

procedure User1Click(Sender: TObject);

procedure FormClose(Sender: TObject; var Action: TCloseAction);

procedure Penyakit1Click(Sender: TObject);

procedure Pasien1Click(Sender: TObject);

procedure Gejala1Click(Sender: TObject);

procedure Pertanyaan1Click(Sender: TObject);

procedure FormCreate(Sender: TObject);

procedure btnNewClick(Sender: TObject);

procedure About1Click(Sender: TObject);

procedure Logout1Click(Sender: TObject);

procedure Exit1Click(Sender: TObject);

procedure Search1Click(Sender: TObject);

procedure Exit2Click(Sender: TObject);

procedure Master1Click(Sender: TObject);

private

maxP, progP: Integer;

public

fuser : TFormUser;

fpenyakit : TFormPenyakit;

```
    fpasien : TFormPasien;  
    fgejala : TFormGejala;  
    fgejalPenyakit : TFormGejalaPenyakit;  
    fdiag   : TFormDiagnosa;  
    fuji    : TForm_Uji;  
    freg     : TFormReg;  
    fcari    : TFormCari;  
    mode : integer;  
    isLogin : boolean;  
end;
```

```
var  
    FormMain: TFormMain;  
    durasi, curIndex : integer;
```

```
implementation
```

```
uses Unit_About , Unit_Login;
```

```
{ $R *.dfm }
```

```
procedure TFormMain.User1Click(Sender: TObject);  
begin  
    if ( fuser=nil) then  
        fuser := TFormUser.Create(self);  
        fuser.Show;  
    end;
```

```
procedure TFormMain.FormClose(Sender: TObject; var Action: TCloseAction);  
begin  
    if (fpenyakit<>nil) then  
        fpenyakit := nil;  
  
    if (fpasien<>nil) then  
        fpasien := nil;  
  
    if (fgejala<>nil) then  
        fgejala := nil;  
  
    if (fgejalPenyakit<>nil) then  
        fgejalPenyakit := nil;
```

```
Application.Terminate
```

```
end;
```

```
procedure TFormMain.Penyakit1Click(Sender: TObject);  
begin
```

```

    if (fpenyakit=nil) then
        fpenyakit := TFormPenyakit.Create(self);
        fpenyakit.Show;
    end;

    procedure TFormMain.Pasien1Click(Sender: TObject);
    begin
        if (fpasien=nil) then
            fpasien := TFormPasien.Create(self);
            fpasien.Show;
        end;

        procedure TFormMain.Gejala1Click(Sender: TObject);
        begin
            if (fgejala=nil) then
                fgejala := TFormGejala.Create(self);
                fgejala.Show;
            end;

            procedure TFormMain.Pertanyaan1Click(Sender: TObject);
            begin
                if (fgejalPenyakit=nil) then
                    fgejalPenyakit := TFormGejalapenyakit.Create(self);
                    fgejalPenyakit.Show;
                end;

                procedure TFormMain.FormCreate(Sender: TObject);
                begin
                    durasi := 0;
                    curIndex := 1;
                    isLogin := false;
                    maxP := 100;
                    progP := 0;
                    mode := 0;
                end;

                procedure TFormMain.btnNewClick(Sender: TObject);
                begin
                    if (freg=nil) then
                        freg := TFormReg.Create(FormMain);
                        freg.Show;
                        mode := 1;
                    end;

                    procedure TFormMain.About1Click(Sender: TObject);
                    begin
                        formabout.ShowModal;
                    end;

                    procedure TFormMain.Logout1Click(Sender: TObject);

```

```

begin
    mode :=0;

    if (fuser<>nil) then
    begin
        fuser.Close;
        fuser := nil;
    end;

    if (fpenyakit<>nil) then
    begin
        fpenyakit.Close;
        fpenyakit := nil;
    end;
    if (fpasien<>nil) then
    begin
        fpasien.Close;
        fpasien := nil;
    end;

    if (fgejala<>nil) then
    begin
        fgejala.Close;
        fgejala := nil;
    end;

    if (fgejalPenyakit<>nil) then
    begin
        fgejalPenyakit.Close;
        fgejalPenyakit := nil;
    end;
    if (fcari<>nil) then
    begin
        fcari.Close;
        fcari := nil;
    end;

end;

procedure TFormMain.Exit1Click(Sender: TObject);
begin
    Application.Terminate
end;

procedure TFormMain.Search1Click(Sender: TObject);

```

```
begin
  if (fcari=nil) then
    fcari := TFormCari.Create(self);
    fcari.Show;
  end;
```

```
procedure TFormMain.Exit2Click(Sender: TObject);
begin
```

```
  if (fuser<>nil) then
  begin
    fuser.Close;
    fuser := nil;
  end;
```

```
  if (fpenyakit<>nil) then
  begin
    fpenyakit.Close;
    fpenyakit := nil;
  end;
  if (fpasien<>nil) then
  begin
    fpasien.Close;
    fpasien := nil;
  end;
```

```
  if (fgejala<>nil) then
  begin
    fgejala.Close;
    fgejala := nil;
  end;
```

```
  if (fgejalPenyakit<>nil) then
  begin
    fgejalPenyakit.Close;
    fgejalPenyakit := nil;
  end;
  if (fcari<>nil) then
  begin
    fcari.Close;
    fcari := nil;
  end;
  if (freg<>nil) then
  begin
    freg.Close;
    freg := nil;
```

```
end;  
end;
```

```
procedure TFormMain.Master1Click(Sender: TObject);  
begin
```

```
if isLogin=false then begin  
formLogin.Show ;  
hide;  
end;  
end;
```

```
end.
```

Data User

unit Unit_Pasien;

interface

uses

**Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, DB, ADODB, StdCtrls, Buttons,
Grids, ExtCtrls ;**

type

TFormPasien = class(TForm)

kode: TLabel;

sPanel2: TPanel;

sLabelFX1: TLabel;

Image1: TImage;

Data: TStringGrid;

sPanel1: TPanel;

BtnDel: TBitBtn;

con: TADOQuery;

sLabel1: TLabel;

jenis: TComboBox;

keyword: TEdit;

procedure FormCreate(Sender: TObject);

procedure keywordChange(Sender: TObject);

**procedure DataSelectCell(Sender: TObject; ACol, ARow: Integer;
var CanSelect: Boolean);**

procedure BtnDelClick(Sender: TObject);

procedure FormClose(Sender: TObject; var Action: TCloseAction);

private

{ Private declarations }

public

{ Public declarations }

end;

var

FormPasien: TFormPasien;

implementation

uses Unit_Main;

{ \$R *.dfm }

procedure TFormPasien.FormCreate(Sender: TObject);

begin

Data.RowCount := 2;

with data do

```

begin
    ColCount := 4;
    Cells[0,0] := 'ID.';
    Cells[1,0] := 'Nama';
    Cells[2,0] := 'Alamat';
    Cells[3,0] := 'Umur';
    ColWidths[1] := 100;
    ColWidths[2] := 280;
    ColWidths[2] := 80;

    end;
    jenis.Items.Add('Nama');
    jenis.Items.Add('ID Pasien');
    jenis.Items.Add('Alamat');
    jenis.ItemIndex :=0;
    keywordChange(self);
end;

procedure TFormPasien.keywordChange(Sender: TObject);
var i:integer;
begin
    con.close;
    con.sql.clear;
    con.sql.add('select * from tb_pasien where ` ` + StringReplace(jenis.Text,
'',_',[rfIgnoreCase]) + ` ` like "%'+ trim(keyword.Text) +'%` ` ');
    con.Prepared;
    con.open;
    Data.RowCount := 1;
    Data.RowCount := 2;
    Data.FixedRows := 1;
    if con.recordcount >0 then
    begin
        for i := 1 to con.RecordCount do
        begin
            with data do
            begin
                Cells[0,i] := con.fieldByName('id_pasien').AsString;
                Cells[1,i] := con.fieldByName('nama').AsString;
                Cells[2,i] := con.fieldByName('alamat').AsString;
                Cells[3,i] := con.fieldByName('umur').AsString;
                RowCount :=i+1;
                con.Next;
            end;
        end
    end
end;

end;

procedure TFormPasien.DataSelectCell(Sender: TObject; ACol, ARow: Integer;

```

```

    var CanSelect: Boolean);
begin
    kode.Caption := IntToStr(ARow);
    BtnDel.Visible := true
end;

procedure TFormPasien.BtnDelClick(Sender: TObject);
begin
    if (MessageDlg('Yakin Untuk Menghapus data?',mtConfirmation,[mbYes,mbNo],0))=
mrYes then
        begin
            try
                con.close;
                con.sql.clear;
                con.sql.add('Delete    from    tb_pasien    where    id_pasien=    '+
data.Cells[0,strtoint(kode.Caption)] +' ');
                con.ExecSQL;

                MessageDlg('Proses Penghapusan Berhasil',mtInformation,[ mbOK],0 );

            except
                MessageDlg('Proses Penghapusan Gagal',mtError,[ mbOK],0 );
            exit;
            end;
        end ;
        Data.RowCount := 1;
        Data.RowCount := 2;
        Data.FixedRows := 1;
        keyword.Text := ' ';
        keywordChange(self);

        BtnDel.Visible :=false;
end;

procedure TFormPasien.FormClose(Sender: TObject; var Action: TCloseAction);
begin
    FormMain.fpasien := nil;
    Action := caFree;
end;

end.

```

Form Data Penyakit

unit Unit_Penyakit;

interface

uses

**Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, DB, ADODB, StdCtrls, Grids, ExtCtrls, Buttons, sLabel;**

type

TFormPenyakit = class(TForm)

sPanel2: TPanel;

sLabelFX1: TLabel;

Image1: TImage;

idpeny: TEdit;

sLabel3: TLabel;

Nama: TEdit;

sLabel4: TLabel;

sLabel1: TLabel;

Data: TStringGrid;

sPanel1: TPanel;

BtnDel: TBitBtn;

BtnClear: TBitBtn;

BtnSave: TBitBtn;

con: TADOQuery;

ket: TMemo;

solusi: TMemo;

kode: TLabel;

Label1: TLabel;

procedure FormCreate(Sender: TObject);

procedure BtnClearClick(Sender: TObject);

procedure idpenyChange(Sender: TObject);

procedure FormClose(Sender: TObject; var Action: TCloseAction);

**procedure DataSelectCell(Sender: TObject; ACol, ARow: Integer;
var CanSelect: Boolean);**

procedure BtnDelClick(Sender: TObject);

procedure BtnSaveClick(Sender: TObject);

private

procedure getCode();

procedure getData();

public

{ Public declarations }

end;

var

FormPenyakit: TFormPenyakit;

implementation

uses Unit_Main;

```

{$R *.dfm}
procedure TFormPenyakit.getCode();
var
  Kodes : String;
  i : Integer;
  found : boolean;

begin
  i := 1;
  Kodes := FormatCurr('P00',1);
  idpeny.Text := Kodes;
  found := false;
  While Not found do
  begin
    Kodes := FormatCurr('P000',i);
    con.close;
    con.sql.clear;
    con.sql.add('SELECT * FROM `tb_penyakit` WHERE id_penyakit="'+kodes+'" ');
    con.Prepared;
    con.open;
    if con.recordcount >0 then
      found := False
    Else
      begin
        idpeny.Text := Kodes;
        found := True;
      end;
    i := i + 1;
  end;

end;

procedure TFormPenyakit.FormCreate(Sender: TObject);
begin
  with data do
  begin
    ColCount := 4;
    Cells[0,0] := 'ID.';
    Cells[1,0] := 'Nama';
    Cells[2,0] := 'Gejala';
    Cells[3,0] := 'Keterangan';
    ColWidths[1] := 120;
    ColWidths[2] := 200;
    ColWidths[3] := 1500;

    end;
  getCode;
  BtnClearClick(sender);
end;

```

```

procedure TFormPenyakit.getData();
var i:integer;
begin
    con.close;
    con.sql.clear;
    con.sql.add('select * from tb_penyakit ');
    con.Prepared;
    con.open;
    Data.RowCount := 1;
    Data.RowCount := 2;
    Data.FixedRows := 1;
    if con.recordcount >0 then
    begin
        for i := 1 to con.RecordCount do
        begin
            with data do
            begin
                Cells[0,i] := con.fieldByName('id_penyakit').AsString;
                Cells[1,i] := con.fieldByName('penyakit').AsString;
                Cells[2,i] := con.fieldByName('jenis').AsString;
                Cells[3,i] := con.fieldByName('solusi').AsString;
                RowCount :=i+1;
                con.Next;
            end;
        end
    end;
end;
procedure TFormPenyakit.BtnClearClick(Sender: TObject);
begin
    getData;
    idpeny.Text := "";
    BtnDel.Visible :=False;
    BtnSave.Visible := false;
    idpeny.Text := "";
    BtnSave.Caption :='Simpan';
    getCode;

    nama.Text := "";
    ket.Clear;
    solusi.Clear;
    kode.Caption := "";
end;

procedure TFormPenyakit.idpenyChange(Sender: TObject);
begin
    if (trim(nama.Text) <>"") and (trim(ket.Text) <>"") then
        BtnSave.Visible := true
    else
        BtnSave.Visible := false;

```

```

    if (BtnSave.Caption = 'Edit') then
    begin
        BtnDel.Visible := false;
        BtnSave.Caption := 'Simpan';
    end
end;

procedure TFormPenyakit.FormClose(Sender: TObject;
var Action: TCloseAction);
begin
    formmain.fpenyakit := nil;
    Action := caFree;
end;

procedure TFormPenyakit.DataSelectCell(Sender: TObject; ACol,
ARow: Integer; var CanSelect: Boolean);
begin
    kode.Caption := IntToStr(ARow);
    BtnSave.Caption := 'Edit';
    BtnSave.Visible := True;
    BtnDel.Visible := true
end;

procedure TFormPenyakit.BtnDelClick(Sender: TObject);
begin
    if (MessageDlg('Yakin Untuk Menghapus data?',mtConfirmation,[mbYes,mbNo],0))=
mrYes then
        begin
            try

                con.close;
                con.sql.clear;
                con.sql.add('Delete    from    tb_penyakit    where    id_penyakit=    '+
data.Cells[0,strtoint(kode.Caption)] +' ');
                con.ExecSQL;

                MessageDlg('Proses Penghapusan Berhasil',mtInformation,[ mbOK],0 );
                BtnClearClick(self);

            except
                MessageDlg('Proses Penghapusan Gagal',mtError,[ mbOK],0 );
            exit;
            end;
        end
end;

procedure TFormPenyakit.BtnSaveClick(Sender: TObject);
var aRow : integer;

begin

```

```

        if (MessageDlg('Yakin Untuk ' + BtnSave.Caption +
data?',mtConfirmation,[mbYes,mbNo],0))= mrYes then
        begin
            try

                if BtnSave.Caption='Simpan' then
                begin

                    con.close;
                    con.sql.clear;
                    con.sql.add('INSERT INTO tb_penyakit ( `id_penyakit`, `penyakit`,
`jenis`, `solusi`) VALUES ("'+trim(idpeny.Text )+'" , "'+trim(nama.Text )+'" ,
"'"+trim(ket.Text )+"", "'"+trim(solusi.Text )+"")');
                    con.ExecSQL;
                end
                else If BtnSave.Caption ='Edit' then
                begin
                    aRow := StrToInt(kode.Caption);
                    with data do
                    begin
                        idpeny.Text := Cells[0,arow] ;
                        Nama.Text := Cells[1,arow] ;
                        ket.Text := Cells[2,arow] ;
                        solusi.Text := Cells[3,arow] ;

                    end;

                    BtnSave.Caption := 'Update';
                    BtnDel.Visible := False;
                    exit;
                end
                else if BtnSave.Caption='Update' then
                begin
                    con.close;
                    con.sql.clear;
                    con.sql.add('UPDATE tb_penyakit SET nama = "'+trim(nama.Text )+"",
jenis = "'+trim(ket.Text )+"", solusi = "'+trim(solusi.Text )+" WHERE (`id_penyakit`=
"'"+idpeny.Text +'"')');
                    con.ExecSQL;
                end;
                MessageDlg('Proses Penyimpanan Berhasil',mtInformation,[ mbOK],0 );
                BtnClearClick(self);

            except
                MessageDlg('Proses Penyimpanan Gagal',mtError,[ mbOK],0 );
            exit;
        end;
    end
else

```

```
begin
  BtnClearClick(self);
end;
end;
end.
```

Form Registrasi

unit Unit_reg;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, DB, ADODB, Spin, StdCtrls, Buttons, ExtCtrls ;

type

TFormReg = class(TForm)

sPanel1: TPanel;

Image1: TImage;

sPanel2: TPanel;

btnDiag: TBitBtn;

con: TADOQuery;

sLabelFX3: TLabel;

Nama: TEdit;

sLabel4: TLabel;

sLabel1: TLabel;

alamat: TEdit;

sLabel2: TLabel;

umur: TSpinEdit;

sLabel3: TLabel;

IdUser: TEdit;

BtnClear: TBitBtn;

sLabelFX2: TLabel;

sLabelFX1: TLabel;

procedure IdUserChange(Sender: TObject);

procedure btnDiagClick(Sender: TObject);

procedure BtnClearClick(Sender: TObject);

procedure FormClose(Sender: TObject; var Action: TCloseAction);

procedure FormCreate(Sender: TObject);

procedure BitBtn1Click(Sender: TObject);

private

procedure getCode();

public

kodePasien :string;{ Public declarations }

end;

var

FormReg: TFormReg;

implementation

uses Unit_diagnosa , Unit_Main, Unit_uji;

{ \$R *.dfm }

```

procedure TFormReg.getCode();
var
  Kodes : String;
  i : Integer;
  found : boolean;

begin
  i := 1;
  Kodes := FormatCurr('U000',1);
  IdUser.Text := Kodes;
  found := false;
  While Not found do
  begin
    Kodes := FormatCurr('U000',i);
    con.close;
    con.sql.clear;
    con.sql.add('SELECT * FROM `tb_pasien` WHERE id_pasien="'+kodes+'" ');
    con.Prepared;
    con.open;
    if con.recordcount > 0 then
      found := False
    Else
      begin
        iduser.Text := Kodes;
        found := True;
      end;
    i := i + 1;
  end;

end;

procedure TFormReg.IdUserChange(Sender: TObject);
begin
  If (trim(nama.Text) <> "") and (trim(alamat.Text) <> "") then
    btnDiag.Visible := true
  else
    btnDiag.Visible := false;
end;

procedure TFormReg.btnDiagClick(Sender: TObject);
begin
  if (MessageDlg('Yakin Untuk Simpam data?',mtConfirmation,[mbYes,mbNo],0))=
mrYes then
    begin
      try

        con.close;
        con.sql.clear;
        con.sql.add('INSERT INTO tb_pasien '+

```

```

        '(id_pasien, nama, alamat, umur)' +
        ' VALUES ("'+trim(IdUser.Text)+'", "' +trim>Nama.Text)+'
        "', "' +trim(alamat.Text)+'", "' +trim(IntToStr(umur.Value))+ "')');
con.ExecSQL;

```

```

// MessageDlg('Proses Penyimpanan Berhasil',mtInformation,[ mbOK],0 );
kodePasien := IdUser.Text;
if (Formmain.fuji=nil) then
    Formmain.fuji := TForm_Uji.Create(Application);
Formmain.fuji.Show;
Formmain.fuji.WindowState := wsNormal;

```

```

    BtnClearClick(self);

```

```

except
    MessageDlg('Proses Penyimpanan Gagal',mtError,[ mbOK],0 );
exit;
end;

```

```

end
else
begin
    BtnClearClick(self);
end;

```

```

end;

```

```

procedure TFormReg.BtnClearClick(Sender: TObject);
begin

```

```

    IdUser.Text := "";
    btnDiag.Visible := False;
    getCode;
    nama.Text := "";
    alamat.Text := "";

```

```

end;

```

```

procedure TFormReg.FormClose(Sender: TObject; var Action: TCloseAction);
begin

```

```

    Formmain.freg:= nil;
    if (Formmain.fuji<>nil) then Formmain.fuji.Close;
    FormMain.mode := 0;
    Action := caFree;
end;

```

```

procedure TFormReg.FormCreate(Sender: TObject);
begin

```

```

    BtnClearClick(sender);
end;

```

```

procedure TFormReg.BitBtn1Click(Sender: TObject);

```

```
begin
kodePasien := IdUser.Text;
    if (Formmain.fuji=nil) then
        Formmain.fuji := TForm_Uji.Create(Application);
        Formmain.fuji.Show;
        Formmain.fuji.WindowState := wsNormal;
end;

end.
```

Form Data User

unit Unit_User;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, ExtCtrls, StdCtrls, Grids,
Buttons, ImgList, DB, ADODB ;

type

TFormUser = class(TForm)

sPanel2: TPanel;

sLabelFX1: TLabel;

Image1: TImage;

IdUser: TEdit;

sLabel3: TLabel;

Nama: TEdit;

sLabel4: TLabel;

Pass: TEdit;

sLabel1: TLabel;

sLabel2: TLabel;

Akses: TComboBox;

Data: TStringGrid;

sPanel1: TPanel;

BtnDel: TBitBtn;

BtnClear: TBitBtn;

BtnSave: TBitBtn;

con: TADOQuery;

kode: TLabel;

procedure FormCreate(Sender: TObject);

procedure BtnClearClick(Sender: TObject);

procedure IdUserChange(Sender: TObject);

**procedure>NamaKeyUp(Sender: TObject; var Key: Word;
Shift: TShiftState);**

procedure FormClose(Sender: TObject; var Action: TCloseAction);

**procedure>DataSelectCell(Sender: TObject; ACol, ARow: Integer;
var CanSelect: Boolean);**

procedure BtnDelClick(Sender: TObject);

procedure BtnSaveClick(Sender: TObject);

private

{ Private declarations }

public

procedure getCode();

procedure getData();

end;

var

FormUser: TFormUser;

implementation

uses Unit_Main;

{ \$R *.dfm }

procedure TFormUser.getCode();

var

Kodes : String;

i : Integer;

found : boolean;

begin

i := 1;

Kodes := FormatCurr('U0000000',1);

IdUser.Text := Kodes;

found := false;

While Not found do

begin

Kodes := FormatCurr('U0000000',i);

con.close;

con.sql.clear;

con.sql.add('SELECT * FROM `tb\_user` WHERE id_user="'+kodes+'" ');

con.Prepared;

con.open;

if con.recordcount > 0 then

found := False

Else

begin

iduser.Text := Kodes;

found := True;

end;

i := i + 1;

end;

end;

procedure TFormUser.FormCreate(Sender: TObject);

begin

with data do

begin

ColCount := 3;

Cells[0,0] := 'ID.';

Cells[1,0] := 'Nama';

Cells[2,0] := 'Akses';

ColWidths[1] := 280;

ColWidths[2] := 80;

end;

getCode;

Akses.Items.Add('Pakar');

```

    Akses.Items.Add('Admin');
    akses.ItemIndex :=0;
    BtnClearClick(sender);
end;

procedure TFormUser.getData();
var i:integer;
begin
    con.close;
    con.sql.clear;
    con.sql.add('select * from tb_user ');
    con.Prepared;
    con.open;
    if con.recordcount >0 then
    begin
        for i := 1 to con.RecordCount do
        begin
            with data do
            begin
                Cells[0,i] := con.fieldByName('id_user').AsString;
                Cells[1,i] := con.fieldByName('nama').AsString;
                Cells[2,i] := con.fieldByName('jabatan').AsString;
                RowCount :=i+1;
                con.Next;
            end;
        end
    end;
end;

procedure TFormUser.BtnClearClick(Sender: TObject);
begin
    getData;
    IdUser.Text := "";
    BtnDel.Visible :=False;
    BtnSave.Visible := false;
    IdUser.Text := "";
    BtnSave.Caption :='Simpan';
    getCode;
    pass.Enabled := true;
    nama.Text := "";
    Pass.Text := "";
    kode.Caption := "";
end;

procedure TFormUser.IdUserChange(Sender: TObject);
begin
    If (trim(nama.Text) <>") and (trim(Pass.Text) <>") then
        BtnSave.Visible := true
    else
        BtnSave.Visible := false;
    if (BtnSave.Caption = 'Edit') then

```

```

begin
    BtnDel.Visible := false;
    BtnSave.Caption := 'Simpan';
end
end;

procedure TFormUser>NamaKeyUp(Sender: TObject; var Key: Word;
Shift: TShiftState);
begin
    if not (char(key) in ['a'..'z',' ','A'..'Z' ]) then
        nama.Clear;
    end;

procedure TFormUser.FormClose(Sender: TObject; var Action: TCloseAction);
begin
    FormMain.fuser := nil;
    Action := caFree;
end;

procedure TFormUser.DataSelectCell(Sender: TObject; ACol, ARow: Integer;
var CanSelect: Boolean);
begin
    kode.Caption := IntToStr(ARow);
    BtnSave.Caption := 'Edit';
    BtnSave.Visible := True;
    BtnDel.Visible := true
end;

procedure TFormUser.BtnDelClick(Sender: TObject);
begin
    if (MessageDlg('Yakin Untuk Menghapus data?',mtConfirmation,[mbYes,mbNo],0))=
mrYes then
        begin
            try

                con.close;
                con.sql.clear;
                con.sql.add('Delete      from      tb_user      where      id_user=      '+
data.Cells[0,strtoint(kode.Caption)] +' ' ');
                con.ExecSQL;

                MessageDlg('Proses Penghapusan Berhasil',mtInformation,[ mbOK],0 );
                BtnClearClick(self);

            except
                MessageDlg('Proses Penghapusan Gagal',mtError,[ mbOK],0 );
            exit;
        end;
    end
end;
end;

```

```

procedure TFormUser.BtnSaveClick(Sender: TObject);
var aRow : integer;

begin
  if (MessageDlg('Yakin Untuk '+ BtnSave.Caption +
data?',mtConfirmation,[mbYes,mbNo],0))= mrYes then
    begin
      try

        if BtnSave.Caption='Simpan' then
          begin

            con.close;
            con.sql.clear;
            con.sql.add('INSERT INTO tb_user ( `id_user`, `pass`, `nama`,
`jabatan`) VALUES ("'+trim(IdUser.Text )+'",md5(" + trim(pass.Text) + " ),
"'+trim(nama.Text )+'", "' +trim(Akses.Text )+'")');
            con.ExecSQL;
          end
        else If BtnSave.Caption ='Edit' then
          begin
            aRow := StrToInt(kode.Caption);
            with data do
              begin
                IdUser.Text := Cells[0,arow] ;
                Nama.Text := Cells[1,arow] ;
                Akses.Text := Cells[2,arow] ;

                end;
            pass.Text := 'xxx';
            pass.Enabled := false;
            BtnSave.Caption := 'Update';
            BtnDel.Visible := False;
            exit;
          end
        else if BtnSave.Caption='Update' then
          begin
            con.close;
            con.sql.clear;
            con.sql.add('UPDATE tb_user SET nama = "' +trim(nama.Text )+'",
jabatan = "' +trim(akses.Text )+' WHERE (id_user= "' +IdUser.Text +'"');
            con.ExecSQL;
          end;
          MessageDlg('Proses Penyimpanan Berhasil',mtInformation,[ mbOK],0 );
          BtnClearClick(self);

        except
          MessageDlg('Proses Penyimpanan Gagal',mtError,[ mbOK],0 );
        exit;
      end;
    end;
  end;
end;

```

```
        end;  
    end  
    else  
    begin  
        BtnClearClick(self);  
    end;  
end;  
end.
```

Aturan

- P001

1. If (demam tinggi) and (sakit kepala tinggi) and (nyeri punggung sedang) Then (Pelunag Cacar air Tinggi)
2. If (demam tinggi) and (sakit kepala sedang) and (nyeri punggung sedang) Then (Pelunag Cacar air Tinggi)
3. If (demam tinggi) and (Kelemahan badan tinggi) and (ruam kulit tinggi) then (Peluang Cacar air Tinggi)
4. If (demam tinggi) and (ruam kulit tinggi) and (mengigil sedang) then (Peluang Cacar air tinggi)
5. If (demam sedang) and (ruam kulit rendah) and (sakit kepala rendah) then (peluang cacar air rendah)
6. If (demam sedang) and (ruam kulit rendah) and (sakit kepala sedang) then (peluang cacar air rendah)
7. If (demam rendah) and (sakit kepala rendah) and (nyeri punggung rendah) then (peluang cacar air rendah)
8. If (demam rendah) and (ruam kulit rendah) and (mengigil rendah) then (peluan cacar air rendah)
9. If (demam rendah) and (kelemahan badan sedang) and (ruam kulit rendah) then (peluang cacar air rendah)
10. If (demam rendah) and (kelemahan badan rendah) and (ruam kulit rendah) then (peluang cacar air rendah)

- P002

11. If (demam tinggi) and (malaise sedang) and (myalgia sedang) then (peluang Campak tinggi)
12. If (demam tinggi) and (sakit kepala sedang) and (mata terasa panas sedang) then (Peluang campak tinggi)
13. If (demam tinggi) and (ruam kulit sedang) and (bersin-bersin sedang) then (Peluang Campak Tinggi)
14. If (demam sedang) and (malaise tinggi) and (myalgia rendah) then (peluang campak rendah)
15. If (demam sedang) and (sakit kepala rendah) and (mata terasa panas rendah) then (peluang campak rendah)
16. If (demam rendah) and (ruam kulit rendah) and (bersin-bersin rendah) then (then peluang campak rendah)
17. If (demam rendah) and (malaise rendah) and (myalgia rendah) then (peluang campak rendah)

- P003

18. If (demam tinggi) and (sakit kepala sedang) and (muntah sedang) then (peluang demam berdarah tinggi)
19. If (demam tinggi) and (nyeri sendi sedang) and (pendarahan sedang) then (peluang demam berdarah tinggi)
20. If (demam tinggi) and (kejang-kejang sedang) and (ruam kulit sedang) then (peluang Demam berdarah tinggi)
21. If (demam sedang) and (kejang-kejang rendah) and (ruam kulit rendah) then (peluang demam berdarah rendah)
22. If (demam sedang) and (nyeri sendi sedang) and (pendarahan rendah) then (peluang demam berdarah rendah)

23. If (demam rendah) and (kejang-kejang rendah) and (ruam kulit rendah) then (peluang demam berdarah rendah)
24. If (demam rendah) and (nyeri sendi rendah) and (pendarahan badan sedang) then (peluang demam berdarah rendah)
25. If (demam rendah) and (sakit kepala rendah) and (muntah rendah) then (peluang peluang demam berdarah rendah)

- P004

26. If (demam tinggi) and (linu persendian tinggi) and (pegal-pegal sedang) then (peluang Demam Chikungunya tinggi)
27. If (demam sedang) and (linu persendian sedang) and (pegal-pegal sedang) then (peluang demam chikungunya tinggi)
28. If (demam sedang) and (linu persendian rendah) and (pegal-pegal sedang) then (peluang demam chikungunya rendah)
29. If (demam rendah) and (linu persendian sedang) and (pegal-pegal rendah) then (peluang demam chikungunya rendah)
30. If (demam rendah) and (linu persendian rendah) and (pegal-pegal rendah) then (peluang demam chikungunya rendah)

- P005

31. If (demam tinggi) and (gangguan perut tinggi) and (mual tinggi) then (peluang disentri tinggi)
32. If (demam tinggi) and (diare tinggi) and (muntah tinggi) then (peluang disentri tinggi)
33. If (demam tinggi) and (menggigil sedang) and (muntah tinggi) then (peluang disentri tinggi)
34. If (demam sedang) and (gangguan perut sedang) and (mual sedang) then (peluang disentri tinggi)
35. If (demam sedang) and (gangguan perut sedang) and (mual sedang) then (peluang disentri tinggi)
36. If (demam sedang) and (mengigil sedang) and (muntah sedang) then (peluang disentri tinggi)
37. If (demam sedang) and (gangguan perut rendah) and (mual rendah) then (peluang disentri rendah)
38. if(demam sedang) and (diare rendah) and (muntah rendah) then (peluang disentri rendah)
39. if (demam rendah) and (mengigil rendah) and (muntah rendah) then (peluang disentri rendah)
40. if (demam rendah) and (ganguan perut rendah) and (mual rendah) then (peluang disentri rendah)
41. if (demam rendah) and (diare rendah) and (muntah rendah) then (peluang disentri rendah)

- P006

42. If (demam tinggi) and (mengigil sedang) and (sakit kepala tinggi) then (peluang influenza tinggi)
43. If (demam tinggi) and (malaise sedang) and (nyeri otot sedang) then (peluang influenza tinggi)
44. If (demam tinggi) and (kelemahan badan sednag) and (batuk tinggi) then (peluang influenza tinggi)

45. If (demam sedang) and (mengigil tinggi) and (sakit kepala tinggi) then (peluang influenza tinggi)
46. If (demam sedang) and (kelemahan badan tinggi) and (batuk tinggi) then (peluang influenza tinggi)
47. If (demam sedang) and (kelemahan badan sedang) and (batuk sedang) then (peluang influenza tinggi)
48. If (demam rendah) and (mengigil rendah) and (sakit kepala rendah) then (peluang influenza rendah)
49. If (demam rendah) and (malaise rendah) and (nyeri otot rendah) then (peluang influenza rendah)
50. If (demam rendah) and (kelemahan badan rendah) and (batuk rendah) then (peluang influenza rendah)

- P007

51. If (demam tinggi) and (mengigil sedang) and (sakit kepala sedang) then (peluang Malaria tinggi)
52. If (demam tinggi) and (mual sedang) and (kulit kering sedang) then (peluang malaria tinggi)
53. If (demam tinggi) and (kelemahan badan tinggi) and (kejang-kejang sedang) then (peluang malaria tinggi)
54. If (demam tinggi) and (muka merah tinggi) and (mual sedang) then (peluang malaria tinggi)
55. If (demam tinggi) and (muntah sedang) and (sakit kepala sedang) then (peluang malaria tinggi)
56. If (demam rendah) and (mengigil rendah) and (sakit kepala rendah) then (peluang malaria rendah)
57. If (demam rendah) and (mual rendah) and (kulit kering rendah) then (peluang malaria rendah)
58. If (demam rendah) and (kelemahan badan rendah) and (kejang-kejang rendah) then (peluang malaria rendah)
59. If (demam rendah) and (muka merah rendah) and (mual rendah) then (peluang malaria rendah)
60. If (demam rendah) and (muntah rendah) and (sakit kepala rendah) then (peluang malaria rendah)

- P008

61. If (demam sedang) and (penurunan berat badan tinggi) and (lemah badan tinggi) then (peluang Tbc tinggi)
62. If (batuk tinggi) and (penurunan berat badan tinggi) and (berkeringat pada malam hari tinggi) then (peluang tbc tinggi)
63. If (demam tinggi) and (penurunan badan sedang) and (lemah badan sedang) then (peluang tbc tinggi)
64. If (batuk sedang) and (penurunan badan sedang) and (berkeringat pada malam hari sedang) then (peluang tbc tinggi)
65. If (demam rendah) and (penurunan badan rendah) and (lemah badan rendah) then (peluang tbc rendah)
66. If (batuk rendah) and (penurunan badan rendah) and (berkeringat pada malam hari rendah) then (peluang tbc rendah)

Tabel Hasil Pengujian

kdR	#00	#01	#02	#03	#04
R01	P01	68 %	33,33 %	70 %	65%
R02	P01	68%	33,33 %	70 %	65%
R03	P01	68%	43,33%	70%	65%
R04	P01	68%	33,33 %	33,33 %	65%
R05	P01	68%	33,33 %	33,33 %	65%
R06	P01	68%	33,33 %	33,33 %	65%
R07	P01	68%	90%	43,33%	65%
R08	P01	33,33%	90%	70%	33,33%
R09	P01	33,33%	33,33%	33,33%	33,33%
R10	P01	33,33%	90%	70%	33,33%
R11	P02	68%	33,33 %	33,33 %	33,33 %
R12	P02	68%	33,33 %	33,33 %	33,33 %
R13	P02	68%	33,33 %	33,33 %	33,33 %
R14	P02	68%	33,33 %	43,33 %	33,33 %
R15	P02	68%	33,33 %	33,33 %	33,33 %
R16	P02	33,33%	90%	70%	61%
R17	P02	33,33%	90%	70%	61%
R18	P03	68%	33,33 %	33,33 %	61%
R19	P03	68%	33,33 %	33,33 %	33,33 %
R20	P03	68%	33,33 %	33,33 %	33,33 %
R21	P03	33,33%	33,33 %	43,33 %	33,33 %
R22	P03	33,33%	33,33 %	33,33 %	33,33 %
R23	P03	33,33%	90%	70%	61%
R24	P03	33,33%	33,33 %	33,33 %	61%
R25	P03	33,33%	90%	70%	61%
R26	P04	68%	33,33 %	33,33 %	33,33 %
R27	P04	61%	33,33 %	33,33 %	33,33 %
R28	P04	33,33%	33,33 %	33,33 %	33,33 %
R29	P04	33,33%	33,33 %	70 %	33,33 %
R30	P04	33,33%	90%	43.33%	61%
R31	P05	68%	43,33%	43,33%	33,33%
R32	P05	68%	43,33%	43,33%	33,33%
R33	P05	68%	33,33 %	43,33 %	33,33 %
R34	P05	61%	33,33 %	43,33 %	33,33 %
R35	P05	61%	33,33 %	43,33 %	61%
R36	P05	61%	33,33 %	43,33 %	61%
R37	P05	68%	33,33 %	43,33 %	65%
R38	P05	68%	33,33 %	43,33 %	65%
R39	P05	68%	90%	70%	65%
R40	P05	68%	90%	70%	65%
R41	P05	68%	90%	70%	65%
R42	P06	68%	33,33 %	70 %	65%
R43	P06	68%	33,33 %	43,33 %	65%
R44	P06	68%	33,33 %	43,33 %	65%
R45	P06	68%	33,33 %	33,33 %	65%
R46	P06	68%	33,33 %	33,33 %	33,33 %

R47	P06	61%	33,33 %	33,33 %	33,33 %
R48	P06	68%	90%	70%	61%
R49	P06	68%	90%	70%	61%
R50	P06	33,33%	90%	70%	33,33 %
R51	P07	68%	33,33 %	43,33 %	65%
R52	P07	68%	33,33 %	43,33 %	65%
R53	P07	68%	33,33 %	43,33 %	65%
R54	P07	68%	33,33 %	43,33 %	65%
R55	P07	68%	33,33 %	43,33 %	65%
R56	P07	68%	90%	70%	65%
R57	P07	68%	90%	70%	65%
R58	P07	68%	90%	70%	61%
R59	P07	33,33%	90%	70%	33,33 %
R60	P07	33,33%	90%	70%	33,33 %
R61	P08	68%	33,33 %	43,33 %	33,33 %
R62	P08	68%	43,33 %	43,33 %	43,33 %
R63	P08	68%	33,33 %	43,33 %	33,33 %
R64	P08	61%	33,33 %	43,33 %	33,33 %
R65	P08	33,33%	90%	70%	33,33 %
R66	P08	33,33%	90%	70%	33,33 %
Rata - rata		71%	50%	63%	53 %

Keterangan:

kdR : Kode Aturan / Rule

#00 : Penyakit

#01 : Hasil Pengujian 1 (RENDAH : [0, 4]; SEDANG : [2, 8]; TINGGI : [4, 10].)

#02 : Hasil Pengujian 2 (RENDAH : [0, 6]; SEDANG : [3, 9]; TINGGI : [6, 10].)

#03 : Hasil Pengujian 3 (RENDAH : [0, 5]; SEDANG : [2.5, 8]; TINGGI : [5, 10].)

#04 : Hasil Pengujian 4 (RENDAH : [0, 3]; SEDANG : [1.5, 7]; TINGGI : [3, 10].)