

repository.ub.ac.id

**PEMODELAN SISTEM PAKAR UNTUK MENDETEKSI PENYAKIT  
DIABETES MELITUS MENGGUNAKAN METODE K-NEAREST  
NEIGHBOUR - CERTAINTY FACTOR**

**SKRIPSI**

Untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Komputer

Disusun oleh:

M Fadjrin Hidayah Tulloh

NIM: 115060807111031



PROGRAM STUDI INFORMATIKA/ILMU KOMPUTER  
PROGRAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER  
UNIVERSITAS BRAWIJAYA  
MALANG  
2015

UNIVERSITAS BRAWIJAYA

## PENGESAHAN

PEMODELAN SISTEM PAKAR UNTUK MENDETEKSI PENYAKIT *DIABETES MELITUS*  
MENGUNAKAN METODE *K-NEAREST NEIGHBOUR - CERTAINTY FACTOR*

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Komputer

Disusun Oleh :

M Fadjrin Hidayah Tulloh

NIM: 115060807111031

Skripsi ini telah diuji dan dinyatakan lulus pada  
17 Desember 2015

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II

Nurul Hidayat S.Pd., M.Sc  
NIP: 19680430 200212 1 001

Edy Santoso S.Si., M.Kom  
NIP. 19740414 200312 1 004

Mengetahui  
Ketua Program Studi Informatika/Ilmu Komputer

Drs. Marji, M.T.

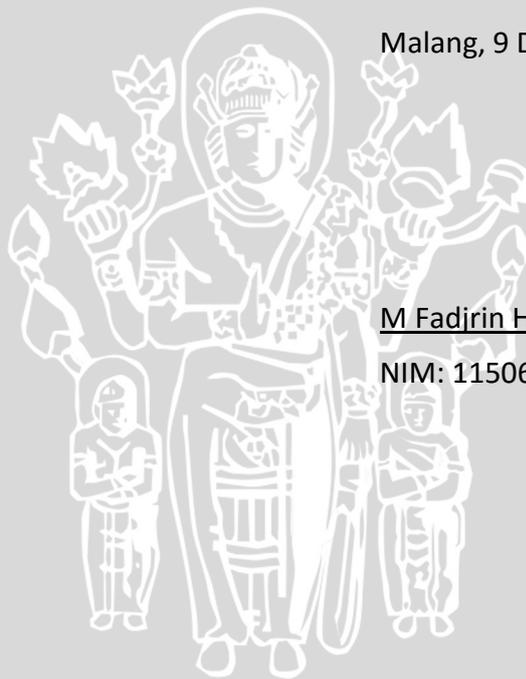
NIP: 19670801 199203 1 001

## PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 9 Desember 2015



M Fadrin Hidayah Tulloh

NIM: 115060807111031

## KATA PENGANTAR

Puji syukur kehadirat Allah SWT yang telah mencurahkan segala rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan skripsi ini. Tidak lupa shalawat dan salam kepada junjungan besar nabi Muhammad SAW beserta keluarga dan para sahabat. Judul Skripsi ini adalah **“Pemodelan Sistem Pakar Untuk Mendeteksi Penyakit *Diabetes Melitus* Menggunakan Metode *K-Nearest Neighbour - Certainty Factor*”**.

Skripsi ini merupakan bagian dari tugas akhir penulis selama mengikuti perkuliahan dan sebagai salah satu syarat untuk mencapai gelar Sarjana Komputer di Fakultas Ilmu Komputer, Universitas Brawijaya. Melalui kesempatan ini, Penulis ingin menyampaikan rasa hormat dan terima kasih yang sebesar-besarnya kepada semua pihak yang telah memberikan bantuan dan dukungan selama pengerjaan skripsi, diantaranya:

1. Nurul Hidayat, S.Pd., M.Sc. Selaku dosen pembimbing pertama yang telah meluangkan waktunya untuk membantu dan membimbing penulis dalam pengerjaan tugas akhir ini.
2. Edy Santoso S.Si., M.Kom. Selaku dosen pembimbing kedua yang telah meluangkan waktunya untuk membantu dan membimbing penulis dalam pengerjaan tugas akhir ini.
3. Ir. Sutrisno, MT. Selaku Ketua Fakultas Ilmu Komputer Universitas Brawijaya
4. Drs. Marji, MT Selaku Ketua Program Studi Informatika/Illmu Komputer Fakultas Ilmu Komputer Universitas Brawijaya.
5. Seluruh dosen Informatika/Illmu Komputer Universitas Brawijaya atas kesediaan membagi ilmunya kepada penulis.
6. Seluruh civitas akademika informatika/ilmu komputer Universitas Brawijaya yang telah banyak membantu dan memberi dukungan selama penulisan skripsi ini.
7. Kedua orang tua Susanto Narwito dan Atik Kusbiati, beserta keluarga yang selalu memberikan doa dan dorongan semangat hingga skripsi ini dapat terselesaikan.
8. Keluarga besar SCANF dan seluruh sahabat Informatika 2011. Terimakasih atas bantuannya selama menempuh studi.
9. Semua pihak yang telah membantu penulis dalam pengerjaan skripsi yang tidak bisa disebutkan satu persatu.

Penulis menyadari bahwa tugas akhir ini tidak lepas dari kekurangan dan kesalahan. Penulis mengharapkan kritik dan saran dari berbagai pihak demi penyempurnaan tugas akhir ini. Akhir kata, semoga skripsi ini dapat bermanfaat

dan berguna bagi pembaca terutama mahasiswa FILKOM Universitas Brawijaya.  
Terima kasih.

Malang, 9 Desember 2015

Penulis

[fadjrinfaa@gmail.com](mailto:fadjrinfaa@gmail.com)



## ABSTRAK

Penyakit merupakan kelainan pada tubuh manusia yang disebabkan oleh kelainan biomedis atau lingkungan manusia. Seiring dengan berjalannya waktu semakin banyak penyakit yang memiliki potensi buruk terhadap kesehatan maupun keselamatan manusia. Diabetes melitus adalah salah satu penyakit yang dianggap berbahaya bahkan bisa sampai mengakibatkan kematian. Umumnya penyakit ini terdiri dari 2 yaitu diabetes melitus tipe I dan diabetes melitus tipe II. Terkadang seseorang yang terkena penyakit ini tidak mengetahui bahwa dirinya ternyata menderita diabetes melitus. Hal ini dikarenakan tidak tahunya masyarakat mengenai gejala-gejala dari penyakit tersebut sehingga menganggap dirinya hanya sakit biasa, padahal apabila terlalu lama dibiarkan bisa mengakibatkan kematian. Maka diperlukan sistem pakar untuk membantu dalam mendiagnosa gejala-gejala yang dialami. Sistem pakar ini menerapkan metode *k-nearest neighbour-certanty factor*. Proses pertama gejala akan dihitung dengan menggunakan metode *k-nearest neighbour* dengan data latih, kemudian akan dilanjutkan dengan menghitung prosentase kemungkinan menggunakan metode *certainty factor*. Hasil pengujian dari metode yang digunakan menghasilkan nilai rata-rata akurasi sebesar 93,5%, dengan pengujian variasi yang terdiri dari 75 data latih dan 40 data uji yang berbeda-beda.

Kata kunci: Penyakit, *Diabetes Melitus*, Sistem Pakar, *K-Nearest Neighbour*, *Certainty Factor*



## ABSTRACT

*Disease is disorder of human body that caused by biomedical abnormalities or environment. As time passed more disease has bad potetial for health and human safety. Diabetes melitus is one of many disease that dangerous and can even caused death. Usually this disease consist of two type, that is Diabetes mellitus type I and Diabetes Mellitus type II. Sometimes a person who affected by this disease do not know that he was suffering from diabetes mellitus. This is because many people doesnt know about symptomp of the disease and consider themself just a common sick, meanwhile if left it too long can caused death. Thats why a expert system needed to help diagnose sympmtomps that happen. This expert system implementing K- nearest neighbour – Certanty factor method. First, symptoms will be calculated using K-nearest neighbour method, and will then be followed by calculating the percentage of the possibility using certainty factor method. The test results from the method that used, resulted in the average value of an accuracy of 93.5%, with a variation test consisting of 75 training data and test data 40 that are different.*

**Keyword:** *Disease, Diabetes Melitus, Expert System, K-Nearest Neighbour, Certainty Factor*

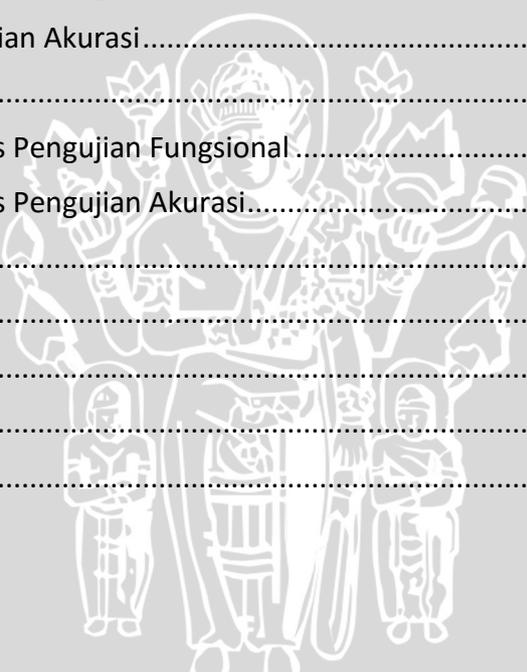


## DAFTAR ISI

PENGESAHAN .....	ii
PERNYATAAN ORISINALITAS .....	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	vi
ABSTRACT.....	vii
DAFTAR ISI.....	viii
DAFTAR TABEL.....	xi
DAFTAR GAMBAR.....	xiii
DAFTAR PERSAMAAN.....	xv
<b>BAB 1 PENDAHULUAN.....</b>	<b>1</b>
1.1 Latar belakang.....	1
1.2 Rumusan masalah.....	2
1.3 Tujuan .....	2
1.4 Manfaat.....	2
1.5 Batasan masalah.....	2
1.6 Sistematika pembahasan.....	3
<b>BAB 2 LANDASAN KEPUSTAKAAN .....</b>	<b>5</b>
2.1 Kajian Pustaka.....	5
2.2 Sistem Pakar.....	8
2.2.1 Ciri-Ciri Sistem Pakar .....	9
2.2.2 Keuntungan Sistem Pakar .....	9
2.2.3 Kelemahan Sistem Pakar.....	10
2.2.4 Struktur Sistem Pakar.....	10
2.2.5 Konsep Dasar Sistem Pakar.....	13
2.2.6 Tujuan Sistem Pakar.....	13
2.2.7 Bentuk Sistem Pakar .....	14
2.3 Representasi Pengetahuan.....	14
2.4 Basis Pengetahuan.....	15
2.5 Metode Inferensi .....	15

2.6 K-Nearest Neighbour .....	17
2.7 Faktor Kepastian ( <i>Certainty Factor</i> ).....	18
2.8 Pengertian Pemodelan dan Simulasi Sistem .....	20
2.9 <i>Blackbox Testing</i> .....	21
2.10 <i>Diabetes Melitus</i> .....	24
2.11 Pengujian Sistem.....	26
2.11.1 Pengujian Validasi .....	26
2.11.2 Pengujian Akurasi.....	26
<b>BAB 3 METODOLOGI .....</b>	<b>27</b>
3.1 Studi Literatur .....	28
3.2 Analisis Kebutuhan .....	28
3.3 Pengumpulan Data .....	29
3.4 Perancangan .....	29
3.5 Implementasi Sistem .....	31
3.6 Pengujian Sistem.....	31
3.7 Kesimpulan.....	32
<b>BAB 4 PERANCANGAN.....</b>	<b>33</b>
4.1 Perancangan Sistem Pakar.....	34
4.1.1 Akuisisi Pengetahuan .....	34
4.1.2 Subsistem Basis Pengetahuan.....	35
4.1.3 Representasi Pengetahuan .....	37
4.1.4 Mesin Inferensi.....	37
4.1.5 Perhitungan Manual.....	41
4.1.6 Antarmuka.....	50
4.1.7 Tabel Penyimpanan Data .....	54
4.1.8 Skenario Pengujian.....	55
<b>BAB 5 IMPLEMENTASI .....</b>	<b>57</b>
5.1 Spesifikasi Sistem .....	58
5.1.1 Spesifikasi Perangkat Keras.....	58
5.1.2 Spesifikasi Perangkat Lunak .....	58
5.2 Batasan Implementasi .....	58
5.3 Implementasi Algoritma .....	59

5.3.1 Implementasi Algoritma Metode KNN.....	59
5.3.2 Implementasi Algoritma Metode CF.....	62
5.4 Implementasi Antarmuka .....	64
5.4.1 Implementasi Antarmuka Homepage.....	64
5.4.2 Implementasi Antarmuka Form Gejala.....	64
5.4.3 Implementasi Antarmuka List Data.....	65
5.4.4 Implementasi Antarmuka KNN .....	66
5.4.5 Implementasi Antarmuka CF.....	67
<b>BAB 6 PENGUJIAN DAN ANALISIS.....</b>	<b>69</b>
6.1 Pengujian .....	70
6.1.1 Pengujian Fungsional .....	70
6.1.2 Pengujian Akurasi.....	72
6.2 Analisis .....	73
6.2.1 Analisis Pengujian Fungsional.....	73
6.2.2 Analisis Pengujian Akurasi.....	74
<b>BAB 7 Penutup .....</b>	<b>76</b>
7.1 Kesimpulan.....	76
7.2 Saran .....	76
<b>DAFTAR PUSTAKA.....</b>	<b>77</b>
<b>LAMPIRAN .....</b>	<b>79</b>



## DAFTAR TABEL

Tabel 2.1 Referensi Paper Terkait .....	6
Tabel 2.2 Referensi Paper Terkait (lanjutan) .....	7
Tabel 2.3 Nilai Interpretasi Pakar .....	19
Tabel 3.1 Tabel Pengumpulan Data .....	29
Tabel 3.2 Akurasi Perbandingan Hasil Pengujian .....	32
Tabel 4.1 Gejala Diabetes Tipe 1 .....	34
Tabel 4.2 Gejala Diabetes Tipe 2 .....	35
Tabel 4.3 Data Latih .....	41
Tabel 4.4 Data Latih (lanjutan) .....	42
Tabel 4.5 Data Latih (lanjutan) .....	43
Tabel 4.6 Data Latih (lanjutan) .....	44
Tabel 4.7 Data Latih (lanjutan) .....	45
Tabel 4.8 Contoh Kasus .....	45
Tabel 4.9 Jarak .....	46
Tabel 4.10 Jarak (lanjutan) .....	47
Tabel 4.11 Sorting .....	47
Tabel 4.12 Sorting (lanjutan) .....	48
Tabel 4.13 CF User .....	48
Tabel 4.14 CF Pakar DM1 dan DM2 .....	49
Tabel 4.15 Data Uji .....	54
Tabel 4.16 Data Training .....	54
Tabel 4.17 Data Jarak .....	54
Tabel 4.18 Data Count .....	54
Tabel 4.19 Data Uji_CF .....	54
Tabel 4.20 Data CF DM1 .....	55
Tabel 4.21 Data CF DM2 .....	55
Tabel 4.22 CF Hasil .....	55
Tabel 4.23 Skenario Pengujian Fungsional .....	55
Tabel 4.24 Skenario Pengujian Fungsional (lanjutan) .....	56
Tabel 4.25 Skenario Pengujian Akurasi .....	56

Tabel 5.1 Spesifikasi Perangkat Keras .....	58
Tabel 5.2 Spesifikasi Perangkat Lunak .....	58
Tabel 5.3 Source code Menghitung jarak antara data uji dengan data latih.....	59
Tabel 5.4 Source code Sorting data jarak menurut nilai terkecil.....	60
Tabel 5.5 Source code Klasifikasi berdasarkan nilai k inputan .....	61
Tabel 5.6 Source code Menghitung nilai CF user .....	62
Tabel 5.7 Source code Menghitung nilai CF user dan pakar.....	62
Tabel 5.8 Source code Menghitung CF Combine .....	63
Tabel 6.1 Kasus Tombol Mulai .....	70
Tabel 6.2 Kasus Isi Form Gejala.....	70
Tabel 6.3 Kasus Tombol Proses.....	71
Tabel 6.4 Kasus Tampil List Data.....	71
Tabel 6.5 Kasus Tombol Hitung KNN.....	71
Tabel 6.6 Kasus Tombol Hitung CF.....	72
Tabel 6.7 Kasus Tombol Reset.....	72
Tabel 6.8 Pengujian Akurasi .....	72
Tabel 6.9 Rata-Rata Pengujian Akurasi .....	73
Tabel 6.10 Hasil Pengujian Validasi.....	73
Tabel 6.11 Hasil Pengujian Validasi (lanjutan).....	74



## DAFTAR GAMBAR

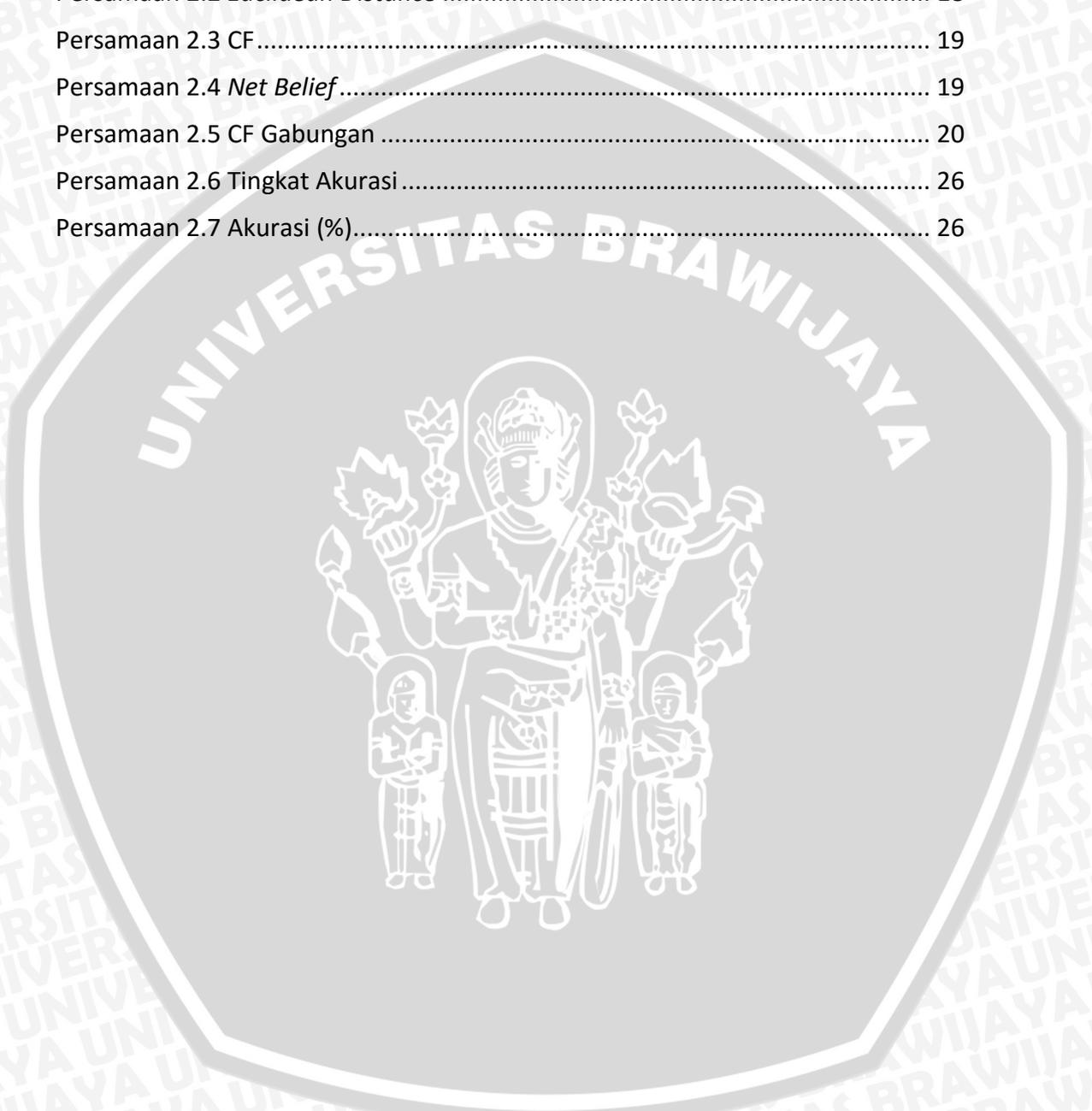
Gambar 2.1 Arsitektur Sistem Pakar .....	11
Gambar 2.2 Alur Metode Forward Chaining.....	16
Gambar 2.3 Alur Metode Backward Chaining .....	17
Gambar 3.1 Diagram Alir Metodologi Penelitian.....	27
Gambar 3.2 Blok Diagram Perancangan Sistem .....	30
Gambar 3.3 Diagram alir perancangan sistem.....	31
Gambar 3.4 Blok Diagram Pengujian Sistem.....	32
Gambar 4.1 Pohon Perancangan .....	33
Gambar 4.2 Bagan Paramater Perhitungan Hasil .....	36
Gambar 4.3 Flowchart <i>K-Nearest Neighbour</i> .....	38
Gambar 4.4 Flowchart <i>Certainty Factor</i> .....	39
Gambar 4.5 Flowchart Sub Proses <i>Certainty Factor</i> .....	40
Gambar 4.6 Sitemap .....	50
Gambar 4.7 Homepage .....	51
Gambar 4.8 Form Gejala .....	51
Gambar 4.9 List Data.....	52
Gambar 4.10 KNN.....	53
Gambar 4.11 CF.....	53
Gambar 5.1 Pohon Implementasi .....	57
Gambar 5.2 Antarmuka Homepage .....	64
Gambar 5.3 Antarmuka Form Gejala .....	65
Gambar 5.4 Antarmuka List Data Uji .....	65
Gambar 5.5 Antarmuka List Data Training.....	66
Gambar 5.6 Antarmuka List Data Inputan Nilai k .....	66
Gambar 5.7 Antarmuka KNN Data Jarak dan Sorting .....	67
Gambar 5.8 Antarmuka KNN Data Sorting Menurut k Inputan .....	67
Gambar 5.9 Antarmuka Tabel CF .....	68
Gambar 5.10 Antarmuka Perhitungan CF .....	68
Gambar 6.1 Pohon Pengujian dan Analisis .....	69
Gambar 6.2 Grafik Pengujian Akurasi .....	74

Gambar 6.3 Grafik Rata-Rata Pengujian ..... 75



## DAFTAR PERSAMAAN

Persamaan 2.1 Normalisasi.....	17
Persamaan 2.2 <i>Euclidean Distance</i> .....	18
Persamaan 2.3 CF .....	19
Persamaan 2.4 <i>Net Belief</i> .....	19
Persamaan 2.5 CF Gabungan .....	20
Persamaan 2.6 Tingkat Akurasi.....	26
Persamaan 2.7 Akurasi (%).....	26



## BAB 1 PENDAHULUAN

### 1.1 Latar belakang

Penyakit merupakan suatu fenomena kompleks yang berpengaruh negatif pada kehidupan manusia. Kadang kala istilah ini digunakan secara umum untuk menerangkan kecederaan, kecacatan, sindrom, simptom, keserongan tingkah laku, dan variasi biasa sesuatu struktur atau fungsi, sementara dalam konteks lain boleh dianggap sebagai kategori yang boleh dibedakan. Perilaku dan cara hidup manusia merupakan penyebab bermacam-macam penyakit baik di jaman dahulu maupun jaman sekarang. Ditinjau dari segi biologis penyakit merupakan kelainan berbagai organ tubuh manusia. Kelainan pada tubuh manusia tersebut disebabkan oleh kelainan biomedis atau lingkungan manusia (Soejoeti, 2008). Pada jaman sekarang ini, terdapat berbagai macam penyakit yang dianggap memiliki dampak yang sangat buruk bagi tubuh atau bahkan kematian. Penyebab dari penyakit itu sendiri adalah kuman, bakteri, virus, racun, kegagalan organ berfungsi, dan juga oleh penyakit baka/keturunan.

Salah satu penyakit yang dianggap menyebabkan kematian dan juga merupakan salah satu jenis penyakit keturunan adalah *diabetes melitus*. *Diabetes melitus* merupakan suatu kelompok metabolik dengan karakteristik hiperglikemia yang terjadi kelainan sekresi insulin, kerja insulin atau kedua-duanya (Soegondo, 2005). Pada umumnya penyakit *diabetes* terdiri menjadi 2 tipe yaitu *diabetes melitus* tipe I dan *diabetes melitus* tipe II. *Diabetes melitus* tipe I ditandai oleh penghancuran sel-sel beta pankreas. Sedangkan *diabetes melitus* tipe II, mekanisme yang tepat mengakibatkan resistensi insulin dan gangguan sekresi insulin dimana faktor genetik diperkirakan memegang peranan dalam proses terjadinya resistensi insulin (Smeltzer, 2001). Masalahnya masih banyak orang yang tidak tahu mengenai penyakit diabetes melitus beserta dengan gejalanya, sehingga tidak mengetahui bahwa dirinya terkena penyakit diabetes melitus yang bisa sampai mengakibatkan kematian apabila tidak ditangani dengan cepat. Oleh karena itu diperlukan sebuah sistem yang dapat membantu dalam mendeteksi lebih dini agar dalam penanganannya dapat dilakukan dengan tepat.

Pada penelitian sebelumnya yang berjudul “Perbandingan *K-Nearest Neighbor* dan *Fuzzy K-Nearest Neighbor* pada Diagnosis Penyakit *Diabetes Melitus*”, pengujian dilakukan terhadap 4 jumlah data latih yang berbeda yaitu 80, 130, 180, dan 230 dengan menggunakan jumlah data uji yang sama yaitu 50 data. Hasil metode tersebut menunjukkan tingkat akurasi tertinggi terdapat pada FK-NN yakni mencapai 98%, sedangkan K-NN akurasi tertingginya hanya mencapai 96% (Meristika, 2013). Dan pada penelitian lain yang berjudul “Sistem Pakar Pencegahan Dini Kanker Serviks Berdasarkan Faktor Resiko Menggunakan Metode *Certainty Factor* Berbasis Mobile Web”, perbandingan hasil output tingkat resiko

sistem dengan tingkat resiko dari pakar dengan akurasi output tingkat resiko adalah sebesar 100% (Krisnawati, 2014).

Berdasarkan masalah dan penelitian tersebut, maka penulis melakukan penelitian yang berjudul “Pemodelan Sistem Pakar Untuk Mendeteksi Penyakit *Diabetes Melitus* Menggunakan Metode *K-Nearest Neighbour - Certainty Factor*”. Dimana kriteria gejala yang diberikan akan digunakan dalam perhitungan *K-Nearest Neighbor* yang selanjutnya hasil dari perhitungan tersebut akan dilanjutkan ke metode *Certainty Factor*. Metode tersebut dianggap memiliki akurasi yang cukup baik dalam penelitian sebelumnya. Oleh karena itu, dengan menggunakan sistem pakar dan kedua metode tersebut diharapkan dapat membantu masyarakat dan para pakar dalam mengenali penyakit *diabetes melitus* dengan lebih dini.

## 1.2 Rumusan masalah

Berdasarkan latar belakang masalah tersebut, maka penulis memiliki beberapa rumusan masalah dalam sistem pakar yang akan dibuat yaitu:

1. Bagaimana merancang pemodelan sistem pakar dengan metode *K-Nearest Neighbour - Certainty Factor* untuk mendeteksi penyakit *diabetes melitus*.
2. Bagaimana hasil pengujian sistem dalam pemodelan sistem pakar dengan metode *K-Nearest Neighbour - Certainty Factor* untuk mendeteksi penyakit *diabetes melitus*.

## 1.3 Tujuan

Penelitian ini memiliki tujuan tertentu agar dapat mengetahui hasil yang ingin dicapai. Tujuan dari penelitian ini adalah sebagai berikut:

1. Merancang pemodelan sistem pakar deteksi penyakit *diabetes melitus* untuk membantu masyarakat dalam melakukan deteksi penyakit *diabetes melitus*.
2. Menguji pemodelan sistem pakar untuk mendeteksi penyakit *diabetes melitus* dengan menggunakan metode *K-Nearest Neighbour - Certainty Factor*.

## 1.4 Manfaat

Manfaat dari penelitian ini diharapkan dapat membantu masyarakat dan dokter dalam melakukan deteksi lebih dini pada penyakit *diabetes melitus*.

## 1.5 Batasan masalah

Untuk menghindari meluasnya masalah yang akan diteliti, maka diberikan batasan masalah sebagai berikut:

1. Jenis penyakit *diabetes melitus* yang diteliti yaitu diabetes tipe 1 dan tipe 2.
2. Data diperoleh dari Puskesmas Poncokusumo Tumpang Kabupaten Malang.
3. Perancangan aplikasi menggunakan database MySQL dan bahasa pemrograman PHP.

4. Gejala penyakit meliputi *polifagia*, *polyuria*, *polodipsia*, turunnya berat badan tanpa alasan yang jelas, umur (15-20), faktor keturunan, mulut kering, mudah kelelahan, sering mengantuk, timbulnya rasa kesemutan (mati rasa) atau sakit pada tangan atau kaki, timbulnya borok (luka) pada kaki yang tak kunjung sembuh, gatal-gatal, disfungsi ereksi pada pria, *obesitas* 20% dari berat badan normal, dan kadar glukosa darah lebih dari 140 mg/dl (sewaktu).
5. Pengujian yang dilakukan adalah pengujian sistem terhadap fungsi-fungsi fitur dan pengujian akurasi.
6. Pemodelan yang dimaksud hanya sebatas pada perancangan metode yang digunakan.

## 1.6 Sistematika pembahasan

Untuk mencapai tujuan yang ada, maka diberikan sistematika penulisan sebagai berikut:

### BAB I PENDAHULUAN

Pada bab ini, berisi latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, dan sistematika penulisan.

### BAB II TINJAUAN PUSTAKA

Pada bab dua berisi tentang dasar-dasar teori yang mendukung penelitian ini. Dasar teori berisi tentang kajian pustaka yang berisi paper-paper yang dijadikan sebagai referensi, sistem pakar, pengertian sistem pakar, konsep dari sistem pakar, ciri sistem pakar, tujuan sistem pakar, keuntungan pemakaian sistem pakar, kelemahan sistem pakar, struktur sistem pakar, komponen sistem pakar, *k-nearest neighbour*, *certainty factor*, pengertian pemodelan dan simulasi sistem, *blackbox testing*, *diabetes melitus*, dan akurasi hasil pengujian.

### BAB III METODOLOGI

Pada bab ini berisi studi literatur, analisis kebutuhan, pengumpulan data, perancangan, implementasi sistem, pengujian sistem, dan kesimpulan.

### BAB IV PERANCANGAN

Membahas tentang analisa kebutuhan dari sistem yang akan dibangun dan kemudian merancang hal-hal yang berhubungan dengan analisa tersebut.

### BAB V IMPLEMENTASI DAN PEMBAHASAN

Membahas tentang hasil perancangan dari analisis kebutuhan dan implementasi dari sistem yang akan dibangun.

### BAB VI PENGUJIAN DAN ANALISIS

Memuat tentang hasil pengujian dan analisis terhadap sistem yang telah direalisasikan.

## BAB VII PENUTUP

Memuat kesimpulan yang diperoleh dari pembuatan dan pengujian perangkat lunak yang dikembangkan dalam skripsi ini serta saran-saran untuk pengembangan lebih lanjut.



## BAB 2 LANDASAN KEPUSTAKAAN

Pada bab dua berisi tentang kajian pustaka yang berisi paper-paper yang dijadikan sebagai referensi, sistem pakar, ciri-ciri sistem pakar, keuntungan sistem pakar, kelemahan sistem pakar, struktur sistem pakar, konsep dasar sistem pakar, tujuan sistem pakar, bentuk sistem pakar, representasi pengetahuan, basis pengetahuan, metode inferensi, *k-nearest neighbour*, *certainty factor*, pengertian pemodelan dan simulasi sistem, *blackbox testing*, *diabetes melitus*, dan akurasi hasil pengujian.

### 2.1 Kajian Pustaka

Kajian pustaka ini terdiri dari beberapa referensi mengenai skripsi yang dikerjakan. Berdasarkan judul skripsi yang dibahas, penulis menemukan beberapa hasil penelitian yang relevan dan metode yang terkait untuk mendukung skripsi ini.

Penelitian pertama adalah penelitian yang dilakukan oleh Yanita Selly Meristika, Achmad Ridhok, dan Lailil Muflikhah. Perbandingan *K-Nearest Neighbor* dan *Fuzzy K-Nearest Neighbor pada Diagnosis Penyakit Diabetes Melitus* ini bertujuan untuk membandingkan tingkat akurasi dari kedua metode yang digunakan. Pengujian ini dilakukan terhadap 4 jumlah data latih yang berbeda yaitu 80, 130, 180, dan 230 dengan menggunakan jumlah data uji yang sama yaitu 50 data. Hasil pengujian yang dilakukan menunjukkan tingkat akurasi tertinggi terdapat pada FK-NN yakni mencapai 98%. Sedangkan K-NN akurasi tertingginya hanya mencapai 96%. Ini berarti *Fuzzy K-Nearest Neighbor* memberikan prediksi yang lebih baik dibandingkan *K-Nearest Neighbor* (Meristika, 2013).

Penelitian selanjutnya dilakukan oleh Suci Krisnawati<sup>1</sup>, Rekyan Regasari Mardi Putri, ST., MT2, Dr.dr.Tatit Nurseta<sup>3</sup>, Sp.OG(K). Penelitian yang berjudul Sistem Pakar Pencegahan Dini Kanker *Serviks* Berdasarkan Faktor Resiko Menggunakan Metode *Certainty Factor* Berbasis *Mobile Web* ini bertujuan untuk pencegahan dini kanker *serviks*. Pada penelitian ini terdapat 23 kriteria dan 30 data testing dari pakar. Proses pengujian menggunakan 30 data testing dari pakar/dokter dan menghasilkan tingkat akurasi sebesar 100% (Krisnawati, 2014). Pada tabel 2.1 akan ditunjukkan mengenai penelitian yang terkait dan relevan dengan judul skripsi ini.

Tabel 2.1 Referensi Paper Terkait

No	Judul	Obyek (Input)	Metode (Proses)	Hasil (Output)
1	Perbandingan <i>K-Nearest Neighbor</i> dan <i>Fuzzy K-Nearest Neighbor</i> pada Diagnosis Penyakit <i>Diabetes Melitus</i> (Meristika,2013)	Jumlah hamil, 2 jam PP (OGTT), tekanan diastolik, tebal kulit trisep (TSFT), 2 jam serum insulin (INS), indeks massa badan (IMB), riwayat diabetes keluarga (DPF) dan usia.	<ul style="list-style-type: none"> <li>• Proses input data uji dan data latih dari <i>Microsoft excel</i>.</li> <li>• perhitungan normalisasi atribut menggunakan <i>min-max normalization</i>.</li> <li>• Menghitung <i>euclidean distance</i>.</li> <li>• Proses KNN yaitu mengambil mayoritas kelas pada K yang telah ditentukan sebagai kelas target pada data yang baru.</li> <li>• Proses FKNN yaitu menghitung nilai derajat keanggotaan dan mengambil nilai terbesar dari proses tersebut dan ditentukan kelas targetnya.</li> <li>• Perbandingan akurasi antara KNN dan FK-NN.</li> </ul>	Output dari diagnosa resiko DM pada sistem yaitu positif DM (1) dan negatif DM (0).
2	Sistem Pakar Pencegahan Dini Kanker Serviks Berdasarkan Faktor Resiko Menggunakan Metode <i>Certainty</i>	Umur $\leq 40$ tahun, umur $\geq 40$ tahun, pernah mengidap kanker, tidak pernah mengidap kanker, pernah <i>hystrectomy</i> , tidak pernah <i>hystrectomy</i> , merokok, tidak pernah merokok, belum pernah berhubungan seksual, jumlah partner seksual 1, jumlah patrner seksual 2 atau	<ul style="list-style-type: none"> <li>• Melakukan wawancara dengan pakar</li> <li>• Menentukan nilai MB dan MD</li> <li>• Menghitung nilai CF dari setiap gejala</li> </ul>	Keluaran terdiri dari 3 output yaitu resiko rendah, sedang, tinggi

Tabel 2.2 Referensi Paper Terkait (lanjutan)

No	Judul	Obyek (Input)	Metode (Proses)	Hasil (Output)
	<i>Factor Berbasis Mobile Web</i> (Krisnawati, 2014)	lebih, melakukan hubungan seksual pertama kali saat lebih muda dari 16 tahun, melakukan hubungan seksual pertama kali saat 16 tahun atau lebih tua, menggunakan alat kontrasepsi, tidak menggunakan alat kontrasepsi, belum punya anak, jumlah anak 1, jumlah anak 2, jumlah anak 3 atau lebih, pernah mengidap penyakit menular seksual, tidak pernah mengidap penyakit menular seksual, hasil screening normal, dan hasil screening abnormal.		
3	Pemodelan Sistem Pakar Untuk Mendeteksi Penyakit <i>Diabetes Melitus</i> Menggunakan Metode <i>K-Nearest Neighbour - Certainty Factor</i>	( <i>polifagia</i> ), ( <i>poliuria</i> ), ( <i>polodipsia</i> ), turunnya berat badan tanpa alasan yang jelas, umur (15-20), faktor keturunan, mulut kering, mudah kelelahan, sering mengantuk, timbulnya rasa kesemutan (mati rasa) atau sakit pada tangan atau kaki, timbulnya rasa borok (luka) pada kaki yang tak kunjung sembuh, gatal-gatal, disfungsi ereksi pada pria, obesitas 20% dari berat badan normal, berumur 20 tahun atau lebih, dan kadar glukosa darah lebih dari 140 mg/dl (sewaktu)	<ul style="list-style-type: none"> <li>• <i>K-Nearest Neighbour</i> diawali dengan menghitung euclidian distance terhadap data uji</li> <li>• <i>Sorting</i> hasil perhitungan <i>euclidian distance</i></li> <li>• Klasifikasi dari hasil sorting</li> <li>• Hasil klasifikasi <i>K-Nearest Neighbour</i> dilanjutkan untuk perhitungan CF</li> <li>• Menghitung CF dari setiap gejala</li> <li>• Nilai kepastian dari perhitungan CF</li> </ul>	Keluaran berupa diabetes tipe 1 dan 2 beserta nilai kepastian

Sumber: Meristika (2013) Krisnawati (2014)

Penelitian terakhir adalah penelitian yang diusulkan oleh penulis. Penelitian ini berjudul *Pemodelan Sistem Pakar Untuk Mendeteksi Penyakit Diabetes Melitus Menggunakan Metode K-Nearest Neighbour - Certainty Factor*. Terdapat 18 kriteria pada penelitian yang akan diusulkan yang terdiri dari gejala-gejala dari *diabetes melitus*. Proses perhitungan menggunakan metode *K-Nearest Neighbour* yang selanjutnya akan dihitung dengan *Certainty Factor* untuk nilai kepastian. Dari hasil kedua perhitungan tersebut maka akan diketahui apakah masuk kedalam *diabetes* tipe 1 atau 2 serta nilai kepastian.

## 2.2 Sistem Pakar

Sistem pakar adalah sistem berbasis komputer yang menggunakan pengetahuan, fakta dan teknik penalaran dalam memecahkan masalah yang biasanya hanya dapat dipecahkan oleh seorang pakar dalam bidang tersebut. Dengan adanya suatu sistem pakar ini orang awam dapat menyelesaikan suatu permasalahan sebenarnya hanya dapat diperoleh dengan bantuan para ahli di bidangnya (Kusumadewi, 2003).

Pengertian sistem pakar menurut para ahli, yaitu (Kusumadewi, 2003):

- Menurut *Jackson*

Sistem pakar adalah sebuah program komputer yang merepresentasikan dan mempertimbangkan dengan pengetahuan dari beberapa subjek spesial dengan sebuah pandangan untuk menyelesaikan masalah-masalah atau memberikan nasihat.

- Menurut *Giarrantano* dan *Riley*

Sistem pakar adalah suatu sistem komputer yang menyamai atau meniru kemampuan seorang pakar.

- Menurut *Ignizio*

Sistem pakar adalah sebuah model atau prosedur yang berkaitan dalam suatu domain tertentu, yang mana tingkat keahliannya dapat dibandingkan dengan keahlian seorang pakar.

- Menurut *Durkin*

Sistem pakar merupakan sebuah program komputer yang dirancang untuk memodelkan kemampuan dalam menyelesaikan suatu permasalahan yang dilakukan oleh seorang pakar.

Sistem pakar mempunyai beberapa persoalan mendasar, menurut *Turban* dan *Frenzel* konsep dasar sistem pakar terdiri atas (Kusumadewi, 2003):

1. Kesepakatan (*Expertise*)

Kesepakatan merupakan penguasaan pengetahuan dibidang tertentu yang diperoleh dari serangkaian pelatihan, membaca, atau pengalaman. Bentuk pengetahuan meliputi fakta dalam lingkup permasalahan, teori lingkup

permasalahan, aturan dan prosedur buku berkenaan lingkup permasalahan, strategi dan *meta knowledge*.

## 2. Pakar (*Expert*)

Seorang pakar adalah orang yang memiliki pengetahuan, penilaian, pengalaman, metode khusus, serta kemampuan menerapkan bakat dalam member nasihat dan memecahkan masalah.

## 3. Pengalihan kepakaran

Tujuan dari sistem pakar adalah mengalihkan kepakaran seorang pakar ke dalam komputer yang akan digunakan oleh pihak lain yang bukan pakar, untuk menemukan solusi atas permasalahan.

## 4. Penalaran (*Interference*)

Salah satu fitur yang harus dimiliki adalah kemampuan menalar.

## 5. Aturan-aturan (*Rules*)

Sebagian besar sistem pakar adalah berbasis aturan. Aturan tersebut biasanya berbentuk *if-then*. Aturan digunakan sebagai prosedur untuk memecahkan permasalahan.

## 6. Kemampuan penjelasan (*Explanation capability*)

Kemampuan penjelasan merupakan komponen tambahan yang berfungsi untuk memberikan penjelasan kepada pengguna mengapa suatu pertanyaan dinyatakan oleh sistem pakar.

### 2.2.1 Ciri-Ciri Sistem Pakar

Ciri-ciri dari sebuah sistem pakar yaitu (Kusumadewi, 2003):

1. Memiliki informasi yang handal
2. Terbatas pada bidang yang spesifik
3. Dapat memberikan suatu penalaran untuk data-data yang tidak lengkap atau tidak pasti
4. Mudah dimodifikasi
5. Berdasarkan *rule* atau kaidah tertentu
6. Dirancang untuk dapat dikembangkan secara bertahap
7. *Knowledge base* dan *interface engine* terpisah

### 2.2.2 Keuntungan Sistem Pakar

Kelebihan/keunggulan yang dapat diperoleh dengan mengembangkan sistem pakar, antara lain (Muhammad, 2005):

1. Memungkinkan orang awam bisa mengerjakan pekerjaan para ahli

2. Bisa melakukan proses secara berulang secara otomatis
3. Menyimpan pengetahuan dan keahlian para pakar
4. Meningkatkan *output* dan produktivitas
5. Meningkatkan kualitas
6. Mampu mengambil dan melestarikan keahlian para pakar
7. Mampu beroperasi dalam lingkungan yang berbahaya
8. Memiliki kemampuan untuk mengakses pengetahuan
9. Memiliki realibilitas
10. Meningkatkan kapabilitas sistem komputer
11. Memiliki kemampuan untuk bekerja dengan informasi yang tidak lengkap dan mengandung ketidakpastian
12. Sebagai media pelengkap dalam pelatihan
13. Menghemat waktu dalam pengambilan keputusan

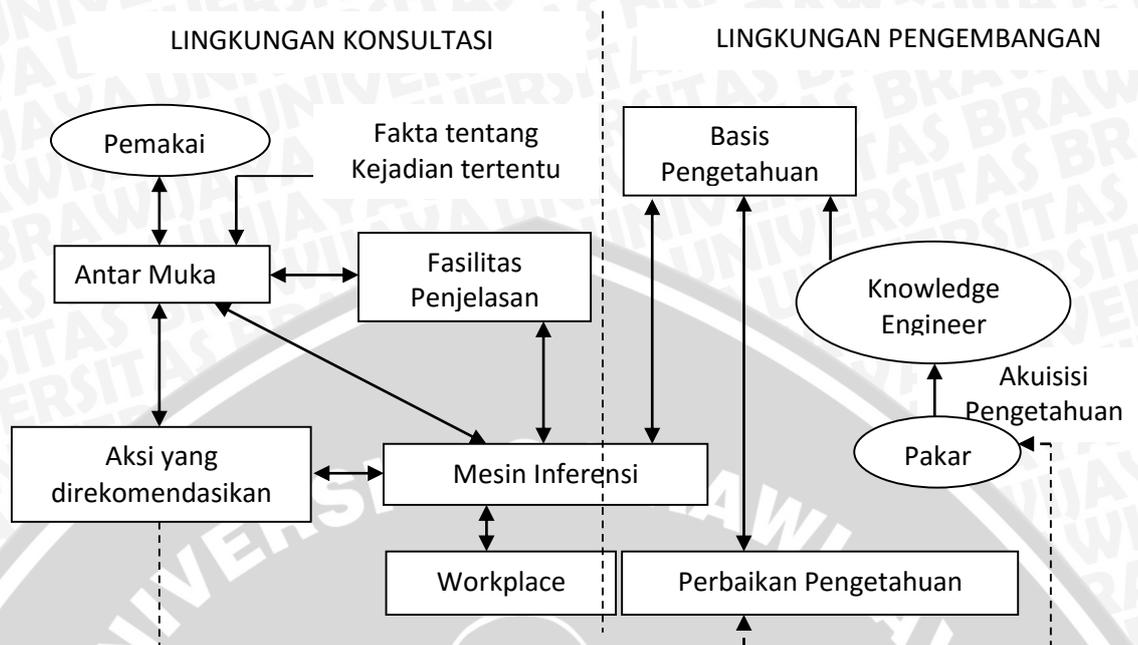
### 2.2.3 Kelemahan Sistem Pakar

Disamping memiliki beberapa keuntungan, sistem pakar juga memiliki beberapa kelemahan, yaitu (Kusumadewi, 2003):

1. Biaya pembuatan dan pemeliharaan sistem yang cukup (*relative*) mahal
2. Sulit dikembangkan untuk ketersediaan pakar yang berkompeten
3. Sistem pakar tidak bisa mencapai 100% nilai kebenaran karena tergantung dari perkembangan pengetahuan. Sehingga perlu adanya pengujian ulang secara teliti, dalam hal ini peran manusia tetap mempunyai peran yang sangat dominan.

### 2.2.4 Struktur Sistem Pakar

Sistem pakar disusun dari 2 bagian utama, yaitu lingkungan pengembangan (*development environment*). Lingkungan ini berfungsi sebagai pembangun sistem pakar baik pada segi pembangun komponen maupun dari segi basis pengetahuan. Lingkungan konsultasi (*consultation environment*). Lingkungan konsultasi digunakan oleh seseorang yang bukan pakar untuk memperoleh pengetahuan pakar (Kusumadewi, 2003). Arsitektur dari sistem pakar terdapat pada gambar 2.1:



**Gambar 2.1** Arsitektur Sistem Pakar  
Sumber: Kusumadewi (2003)

1. Antarmuka (*Interface*)

Merupakan mekanisme yang digunakan oleh pengguna dan sistem pakar untuk berkomunikasi. Antarmuka akan menerima informasi dari pengguna dan mengubahnya dalam bentuk yang dapat diterima oleh sistem. Selain itu antarmuka menerima informasi dari sistem dan menyajikannya ke dalam bentuk yang dapat dimengerti oleh pengguna.

2. Basis Pengetahuan (*Knowledge Base*)

Basis pengetahuan merupakan inti dari suatu sistem pakar, yaitu berupa representasi pengetahuan dari pakar. Basis pengetahuan tersusun atas fakta dan kaidah. Fakta adalah informasi tentang objek, peristiwa, atau situasi. Kaidah adalah cara untuk membangkitkan suatu fakta baru dari fakta yang sudah diketahui. Ada 2 bentuk pendekatan basis pengetahuan:

- o Penalaran berbasis aturan (*rule-based reasoning*)

Pada penalaran ini direpresentasikan dengan menggunakan aturan berbentuk *IF-THEN*. Bentuk ini digunakan apabila memiliki sejumlah pengetahuan pakar pada suatu permasalahan tertentu dan pakar dapat menyelesaikan masalah tersebut secara berurutan.

- o Penalaran berbasis kasus (*case-based reasoning*)

Pada penalaran ini akan berisi solusi-solusi yang telah dicapai sebelumnya, kemudian akan diturunkan suatu solusi untuk keadaan yang terjadi sekarang (fakta yang ada). Bentuk ini digunakan apabila user menginginkan untuk tahu lebih banyak lagi pada kasus-kasus yang hampir sama (mirip).

### 3. Akuisisi Pengetahuan

Akuisisi pengetahuan adalah akumulasi, transfer dan transformasi keahlian dalam menyelesaikan masalah dari sumber pengetahuan kedalam program komputer. Dalam tahap ini *knowledge engineer* berusaha menyerap pengetahuan untuk selanjutnya ditransfer ke dalam basis pengetahuan. Terdapat 3 metode utama dalam akuisisi pengetahuan, yaitu:

- *Knowledge capture*/wawancara

Metode ini melibatkan pembicaraan dengan pakar secara langsung dalam suatu wawancara

- *Knowledge analysis*/analisis protokol

Dalam metode ini pakar diminta melakukan sebuah pekerjaan dan mengungkapkan proses pemikirannya menggunakan kata-kata

- *Knowledge modelling*

Metode ini menciptakan sebuah cara yang berbeda dalam mengubah dan menampilkan basis pengetahuan. Terdiri dari bagian-bagian seperti pohon (*trees*), *matriks*, *map*, *timeline*, *frame*, dan *knowledge page*.

### 4. Representasi Pengetahuan

Representasi pengetahuan merupakan sebuah metode yang digunakan untuk mengkodekan pengetahuan dalam sebuah sistem pakar yang berbasis pengetahuan yang dimaksudkan untuk menangkap sifat-sifat penting masalah dan membuat informasi sehingga dapat diakses oleh prosedur pemecahan masalah.

### 5. Mesin Inferensi (*Inference Engine*)

Mesin inferensi berperan sebagai otak dari sistem pakar. Mesin inferensi berfungsi untuk memandu proses penalaran terhadap suatu kondisi, berdasarkan pada basis pengetahuan yang tersedia. Didalam mesin inferensi terjadi proses untuk memanipulasi dan mengarahkan kaidah, model, dan fakta yang disimpan dalam basis pengetahuan dalam rangka mencapai solusi atau kesimpulan. Dalam prosesnya, mesin inferensi menggunakan strategi penalaran pasti (*Exact Reasoning*) dan strategi penalaran tak pasti (*Inexact Reasoning*). *Exact reasoning* akan dilakukan jika semua data yang dibutuhkan untuk menarik suatu kesimpulan tersedia, sedangkan *inexact reasoning* dilakukan pada keadaan sebaliknya. Strategi pengendalian berfungsi sebagai panduan arah dalam melakukan proses penalaran.

### 6. Lingkungan Kerja (*Workplace*)

*Workplace* merupakan area dari sekumpulan memori kerja (*working memory*). *Workplace* digunakan untuk merekam hasil-hasil dan kesimpulan yang dicapai.

## 7. Fasilitas Penjelas

Fasilitas penjelasan adalah komponen tambahan yang akan meningkatkan kemampuan sistem pakar. Komponen ini menggambarkan penalaran sistem kepada pemakai.

## 8. Perbaikan Pengetahuan

Pakar memiliki kemampuan menganalisa dan meningkatkan kinerja serta kemampuan untuk belajar dari kinerjanya. Kemampuan berperan penting dalam pemberlajaran komputerisasi, sehingga program mampu menganalisa penyebab kesuksesan dan kegagalan yang dialaminya.

### 2.2.5 Konsep Dasar Sistem Pakar

Konsep dasar dari sistem pakar yaitu meliputi keahlian (*expertise*), ahli (*experts*), pemindahan keahlian (*transferring expertise*), inferensi (*inferencing*), aturan (*rules*), dan kemampuan memberikan penjelasan (*explanation capability*) (Hidayat, 2010).

Keahlian adalah pengetahuan yang mendalam tentang suatu masalah tertentu, dimana keahlian bisa diperoleh dari pelatihan/pendidikan, membaca, dan pengalaman dunia nyata. Ada dua macam pengetahuan yaitu pengetahuan dari sumber yang ahli dan pengetahuan dari sumber yang tidak ahli. Pengetahuan dari sumber yang ahli dapat digunakan untuk mengambil keputusan dengan cepat dan tepat (Hidayat, 2010).

Ahli adalah seirang yang memiliki keahlian tentang suatu hal dalam tingkatan tertentu. Ahli dapat menggunakan suatu permasalahan yang ditetapkan dengan beberapa cara yang berubah-ubah dan merubahnya ke dalam bentuk yang dapat dipergunakan oleh dirinya sendiri dengan cepat dan cara pemecahan yang mengesankan (Hidayat, 2010).

### 2.2.6 Tujuan Sistem Pakar

Tujuan dari sistem pakar adalah untuk memindahkan kemampuan (*transferring expertise*) dari seorang ahli atau sumber keahlian yang lain ke dalam komputer dan kemudian memindahkannya dari komputer kepada pemakai yang tidak ahli (bukan pakar). Proses ini meliputi empat aktivitas yaitu (Hidayat, 2010):

1. Akuisisi pengetahuan (*knowledge acquisition*) yaitu kegiatan mencari dan mengumpulkan pengetahuan dari para ahli atau sumber keahlian yang lain.
2. Representasi pengetahuan (*knowledge representation*) adalah kegiatan menyimpan dan mengatur penyimpanan pengetahuan yang diperoleh dalam komputer. Pengetahuan berupa fakta dan aturan disimpan dalam komputer sebagai sebuah komponen yang disebut basis pengetahuan.
3. Inferensi pengetahuan (*knowledge inferencing*) adalah kegiatan melakukan inferensi berdasarkan pengetahuan yang telah disimpan didalam komputer.

4. Pemindahan pengetahuan (*knowledge transfer*) adalah kegiatan pemindahan pengetahuan dari komputer ke pemakai yang tidak ahli.

### 2.2.7 Bentuk Sistem Pakar

Sistem pakar dikelompokkan ke dalam empat bentuk yaitu (Hidayat, 2010):

1. Mandiri merupakan sistem pakar yang murni berdiri sendiri, tidak digabung dengan perangkat lunak lain, bisa dijalankan pada komputer pribadi dan *mainframe*.
2. Terkait atau tergabung merupakan sistem pakar hanya bagian dari program yang lebih besar. Program tersebut biasanya menggunakan teknik algoritma konvensional tapi bisa mengakses sistem pakar yang ditempatkan sebagai subrutin, yang bisa dimanfaatkan setiap kali dibutuhkan.
3. Terhubung adalah sistem pakar yang berhubungan dengan software lain. Misalnya *spreadsheet*, DBMS, program grafik. Pada saat proses inferensi, sistem pakar bisa mengakses data alam *spreadsheet* atau DBMS atau program grafik bisa dipanggil untuk menayangkan output visual.
4. Sistem mengabdikan merupakan bagian dari komputer khusus yang diabdikan kepada fungsi tunggal. Sistem tersebut bisa membantu menganalisa data radar pesawat tempur atau membuat keputusan intelejen tentang bagaimana memodifikasi pembangunan kimiawi.

### 2.3 Representasi Pengetahuan

Representasi pengetahuan merupakan metode yang digunakan untuk mengkodekan pengetahuan dalam sebuah sistem pakar (Arif, 2009). Metode representasi yang cocok untuk pengetahuan bersifat deklaratif adalah:

1. Logika (*Logic*)

Logika merupakan suatu pengkajian ilmiah tentang serangkaian penalaran, sistem kaidah, dan prosedur yang membantu proses penalaran. PROLOG merupakan bahasa pemrograman yang ideal dalam mengimplementasikan representasi logika (Arif, 2009).

2. Prosedural

Representasi prosedural, representasi ini menggambarkan pengetahuan sebagai sekumpulan instruksi untuk memecahkan suatu problem. Dalam sistem berbasis kaidah, aturan *IF-THEN* dapat ditafsirkan sebagai sebuah prosedur untuk mencapai tujuan pemecahan problema (Arif, 2009).

3. *Network*

Representasi ini menangkap pengetahuan sebagai sebuah grap dimana node-nodenya menggambarkan objek atau konsep dalam problema yang dihadapi, sedangkan lengkungan-lengkungan menggambarkan hubungan atau asosiasi

antara keduanya. Contoh representasi *network* antara lain jaringan semantik dan graf (Arif, 2009).

#### 4. Terstruktur

Bahasa-bahasa representasi terstruktur memperluas jaringan dengan membuat setiap nodenya menjadi sebuah struktur data kompleks yang berisi tempat-tempat bernama dengan nilai tertentu. Nilai-nilai ini dapat berupa data numeric atau simbolik sederhana, pointer ke *frame* (bingkai) lain, atau bahkan merupakan prosedur untuk mengerjakan tugas tertentu. Contoh dari representasi ini adalah skrip (*script*), *frame*, dan objek (*object*) (Arif, 2009).

Sedangkan representasi yang cocok untuk pengetahuan prosedural (ada aksi dan reaksi) adalah kaidah produksi (*production rule*) (Hidayati, 2012). Kaidah produksi adalah kaidah yang menyediakan cara formal untuk merepresentasikan rekomendasi, arahan, atau strategi. Kaidah produksi dituliskan dalam bentuk jika-maka (*if-then*). Kaidah *if-then* menghubungkan anteseden (*antecedent*) dengan konsekuensi yang diakibatkannya (Arif, 2009).

## 2.4 Basis Pengetahuan

Basis pengetahuan berisi pengetahuan-pengetahuan dalam penyelesaian masalah didalam domain tertentu. Ada dua bentuk pendekatan basis pengetahuan yang sangat umum digunakan, yaitu (Arif, 2009):

#### 1. Penalaran berbasis aturan (*Rule-Based Reasoning*)

Pada penalaran berbasis aturan, pengetahuan direpresentasikan dengan menggunakan aturan berbentuk *if-then*. Bentuk ini digunakan apabila memiliki sejumlah pengetahuan pakar pada suatu permasalahan tertentu, dan pakar dapat menyelesaikan masalah tersebut secara berurutan. Disamping itu, juga digunakan apabila dibutuhkan penjelasan tentang jejak (langkah-langkah) pencapaian solusi.

#### 2. Penalaran berbasis kasus (*Case-Based Reasoning*)

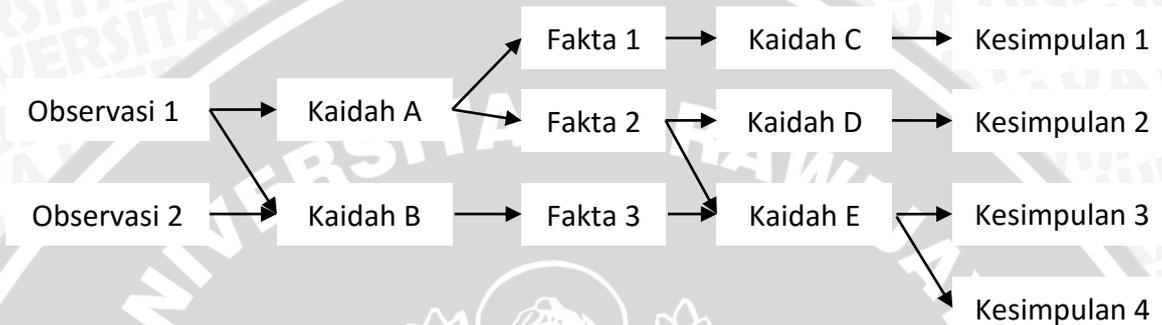
Pada penalaran berbasis kasus, basis pengetahuan akan berisi solusi-solusi yang telah dicapai sebelumnya, kemudian akan diturunkan suatu solusi untuk keadaan yang terjadi sekarang (fakta yang ada). Bentuk ini akan digunakan apabila *user* menginginkan untuk tahu lebih banyak lagi pada kasus-kasus yang hampir sama (mirip). Selain itu, bentuk ini juga digunakan apabila telah memiliki sejumlah situasi atau kasus tertentu dalam basis pengetahuan.

## 2.5 Metode Inferensi

Inferensi merupakan proses untuk menghasilkan informasi dari fakta yang diketahui atau diasumsikan. Proses inferensi dalam sistem pakar disebut mesin inferensi. Berikut adalah dua jenis metode inferensi (Hidayati, 2012).

1. *Forward Chaining*

Teknik pencarian yang dimulai dengan fakta yang diketahui, kemudian dicocokkan fakta-fakta tersebut dengan bagian *IF* dari aturan *IF-THEN*. Bila ada aturan yang cocok dengan bagian *IF*, maka aturan tersebut dieksekusi. Bila aturan dieksekusi maka sebuah fakta baru (bagian *THEN*) ditambahkan ke dalam basis data. Pencocokan dimulai dari aturan teratas dan setiap aturan hanya boleh dieksekusi sekali. Alur metode inferensi *Forward Chaining* terlihat seperti gambar 2.2.



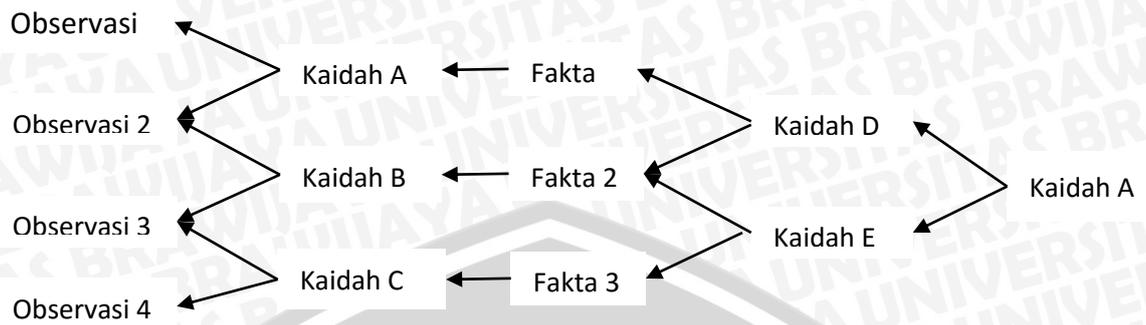
**Gambar 2.2** Alur Metode Forward Chaining  
 Sumber : Hidayati (2012)

Berdasarkan gambar 2.4 observasi adalah kegiatan yang para dokter lakukan untuk melakukan rekaman jantung pada pasien, lalu melihat pola yang dihasilkan dari rekaman tersebut (kaidah). Sehingga menghasilkan bahwa pasien tersebut mengalami penyakit jantung (fakta). Para dokter akan menyempurnakan observasi dengan melakukan pengolahan data rekaman jantung agar dapat membuat kesimpulan seberapa besar seseorang mempunyai resiko mengalami SCD (Kesimpulan).

2. *Backward Chaining*

Metode inferensi yang bekerja mundur kearah kondisi awal. Proses diawali dari goal (yang berada pada bagian *THEN* dari aturan *IF-THEN*), kemudian pencarian mulai dijalankan untuk mencocokkan apakah fakta-fakta yang ada cocok dengan premis-premis di bagian *IF*. Jika cocok, maka aturan dieksekusi, kemudian hipotesis di bagian *THEN* ditempatkan di basis data sebagai fakta baru. Jika tidak cocok simpan premis di bagian *IF* ke dalam subgoal. Proses berakhir jika goal ditemukan atau tidak ada aturan yang bisa membuktikan kebenaran subgoal atau goal. Alur metode inferensi *Backward Chaining* terlihat seperti Gambar 2.3.





**Gambar 2.3** Alur Metode Backward Chaining  
 Sumber : Hidayati (2012)

Berdasarkan gambar 2.5 maka didapat penjelasan bahwa ketika para dokter mengetahui (kesimpulan) seberapa besar pasien mengalami SCD, dokter akan melakukan penelitian terhadap rekaman jantung dari pasien (kaidah) lalu akan mendapatkan fakta bahwa seseorang yang memiliki resiko besar seperti pasien mempunyai pola rekaman jantung seperti yang ditunjukkan. Dari fakta ini para dokter akan menyarankan pada pasien yang mempunyai resiko mengalami SCD yang besar agar bersedia untuk ditanamkan ICD.

## 2.6 K-Nearest Neighbour

Algoritma *K-Nearest Neighbour* adalah sebuah metode yang melakukan klasifikasi terhadap objek yang memiliki jarak paling dekat dengan objek tersebut. Jarak didapatkan dari data pembelajaran yang diproyeksikan ke ruang berdimensi banyak yang mana setiap dimensi merepresentasikan fitur dari data. Menurut (J. Nilsson, 2009), algoritma ini mengklasifikasikan data baru yang belum diketahui kelasnya dengan memilih data sejumlah  $k$  yang letaknya terdekat dari data baru tersebut. Class terbanyak dari data terdekat sejumlah  $k$  tersebut dipilih sebagai class yang diprediksikan untuk data yang baru. Nilai  $k$  umumnya ditentukan dalam jumlah ganjil untuk menghindari munculnya jarak yang sama dalam proses klasifikasi. Dekat atau jauhnya tetangga biasanya dihitung berdasarkan jarak *euclidean*.

Ketepatan algoritma KNN sangat dipengaruhi oleh ada atau tidaknya fitur-fitur yang tidak relevan atau bobot tersebut tidak setara dengan relevansi terhadap klasifikasinya. Algoritma KNN memiliki konsistensi yang kuat. Selain itu algoritma KNN sangatlah sederhana karena bekerja berdasarkan jarak terpendek dari sampel uji ke sampel latih untuk menentukan KNN nya. Setelah mengumpulkan KNN, lalu diambil mayoritas dari KNN untuk dijadikan prediksi dari sampel data uji. Algoritma ini mencari  $k$  *training record* (tetangga) yang memiliki jarak terdekat dari *record* baru tersebut, sehingga KNN sangat mudah diimplementasikan.

Pada proses KNN, langkah pertama yang dilakukan adalah normalisasi dari setiap data baik itu data baru maupun data latih. Langkah ini dilakukan karena atribut cenderung memiliki nilai dengan rentang yang bervariasi. Misalnya, dalam menentukan jarak antara dua *record*, atribut dengan nilai rentang yang besar

memiliki banyak pengaruh dalam menentukan jarak daripada atribut yang memiliki nilai rentang kecil. Oleh karena itu, perlu dilakukan nilai untuk membakukan skala pengaruh yang ada pada atribut terhadap hasil (Moradian, 2009). Atribut yang telah dinormalisasi dapat digunakan pada proses klasifikasi. Persamaan untuk menghitung normalisasi ditunjukkan pada persamaan 2.1.

$$V' = \frac{v - \min A}{\max A - \min A} \quad (2.1)$$

Dimana:

$V'$  : hasil normalisasi yang nilainya berkisar antara 0 dan 1

$V$  : nilai atribut A yang akan dinormalisasi

$\min A$  : nilai minimum dari suatu atribut A

$\max A$  : nilai maksimum dari suatu atribut A

Selanjutnya, hitung jarak terdekat antara data uji dengan data latih. Perhitungan untuk mencari jarak menggunakan *euclidean* ditunjukkan pada persamaan 2.2.

$$\begin{aligned} x_1 &= (x_{11}, x_{12}, \dots, x_{1n}) \\ x_2 &= (x_{21}, x_{22}, \dots, x_{2n}) \\ d(x_1, x_2) &= \sqrt{\sum_r^n (a_r(x_1) - a_r(x_2))^2} \end{aligned} \quad (2.2)$$

Dimana  $x_1$  dan  $x_2$  merupakan dua *record* dengan  $n$  atribut. Persamaan menghitung jarak antara  $x_1$  dan  $x_2$ , dengan tujuan untuk menentukan perbedaan antara nilai-nilai atribut pada *record*  $x_1$  dan  $x_2$ . Setelah diketahui jarak antar *record*, kemudian diambil sebanyak  $k$  tetangga terdekat untuk memprediksi label kelas dari *record* baru menggunakan label kelas tetangga. Diambil kelas mayoritas sebagai kelas target *output* data yang baru.

## 2.7 Faktor Kepastian (*Certainty Factor*)

Sistem pakar harus mampu bekerja dalam ketidakpastian (Giarratano, 1994). Sejumlah teori yang ditemukan untuk menyelesaikan masalah ketidakpastian antara lain: probabilitas klasik (*classical probability*), probabilitas Bayes (*Bayesian probability*), teori *Hartley* berdasarkan himpunan klasik (*Hartley theory based on classical sets*), teori *Shannon* berdasarkan probabilitas (*Shannon theory based on probability*), teori *Dempster-Shafer* (*Dempster-Shafer theory*), teori *fuzzy Zadeh* (*Zadeh's fuzzy theory*) dan faktor kepastian (*Certainty Factor*) (Kusumadewi, 2003).

Shortliffe dan Buchanan mengembangkan *Certainty Factor* di pertengahan 1970-an untuk MYCIN, sebuah sistem pakar untuk diagnosa dan penyembuhan meningitis dan infeksi darah. Sejak itu CF menjadi standar pendekatan untuk mengatasi ketidakpastian dalam *rule base system*.

Aturan dalam *Certainty Factor* diekspresikan dalam bentuk pernyataan *IF-THEN*, seperti:

$$IF H[HIPOTESIS] THEN E[Evidence]$$

Hal ini sama halnya dengan arti logika  $H \rightarrow E$ . Sebuah aturan pada  $H \rightarrow E$  berarti bahwa jika  $H$  terjadi maka  $E$  akan mengambil tempat. Kelemahan pada logika biasa adalah ketidaktepatannya mengatasi masalah ketidakpastian.

Sebuah aturan  $H \rightarrow E$  berasosiasi dengan ukuran dari *Certainty Factor* (CF) dalam persamaan 2.3.

$$R: IF[evidence E] THEN H[hipotesis H] \text{ dengan } CF(R)$$

$$\text{Dimana } CF(R) = MB(R) - MD(R) \tag{2.3}$$

CF(R) mengekspresikan konsep observasi dari sebuah *evidence* E, yang memberikan pengaruh terhadap hipotesis H dalam aturan R.

Secara garis besar ada dua cara dalam mendapatkan tingkat keyakinan (CF) dari sebuah aturan, yaitu (Sutojo T, 2011):

1. Metode '*Net Belief*' yang diusulkan oleh E.H. *Shotliffe* dan B.G. *Buchanan*, yang dirumuskan pada persamaan 2.4.

$$CF(\text{Rule}) = MB(H,E) - MD(H,E) \tag{2.4}$$

Keterangan:

CF(Rule) = faktor kepastian

MB(H,E) = *measure of belief* (ukuran kepercayaan) terhadap hipotesis H, jika diberikan evidence E (antara 0 dan 1)

MD(H,E) = *measure of disbelief* (ukuran kepercayaan) terhadap hipotesis H, jika diberikan evidence E (antara 0 dan 1)

2. Dengan cara mewawancarai seorang pakar

Nilai CF (*rule*) didapat dari interpretasi *term* dari pakar, yang diubah menjadi nilai CF tertentu, yang ditunjukkan pada tabel 2.2.

**Tabel 2.3** Nilai Interpretasi Pakar

<i>Definitely not</i> (pasti tidak)	-1.0
<i>Almost certainly not</i> (hampir pasti tidak)	-0.8
<i>Probability</i> (kemungkinan besar tidak)	-0.6
<i>Maybe not</i> (mungkin tidak)	-0.4
<i>Unknown</i> (tidak tahu)	-0.2 hingga 0.2
<i>Maybe</i> (mungkin)	0.4
<i>Probably</i> (kemungkinan besar)	0.6
<i>Almost certainly</i> (hampir pasti)	0.8
<i>Definetely</i> (pasti)	1.0

Sumber: Sutojo T (2011)

CF gabungan merupakan CF akhir dari sebuah calon konklusi. CF gabungan diperlukan jika suatu konklusi diperoleh dari aturan sekaligus. CF akhir dari suatu aturan dengan aturan yang lain digabungkan untuk mendapatkan nilai CF akhir bagi calon konklusi tersebut. Adapun rumus untuk melakukan perhitungan CF gabungan ditunjukkan pada persamaan 2.5.

$$\begin{aligned}
 &CF(x) + CF(y) - (CF(x) \times CF(y)), CF(x), CF(y) > 0 \\
 &\frac{CF(x)+CF(y)}{1-\min(|CF(x)||CF(y)|)} \text{ salah satu } CF(x), CF(y) < 0 \\
 &CF(x) + CF(y) + (CF(x) \times CF(y)), CF(x), CF(y) > 0 \quad (2.5)
 \end{aligned}$$

## 2.8 Pengertian Pemodelan dan Simulasi Sistem

Pemodelan adalah proses untuk membuat sebuah model. Model adalah representasi dari sebuah bentuk nyata, sedangkan sistem adalah saling keterhubungan dan ketergantungan antar elemen yang membangun sebuah kesatuan, biasanya dibangun untuk mencapai tujuan tertentu. Sebuah pemodelan sistem, dengan demikian, merupakan gambaran bentuk nyata yang dimodelkan secara sederhana, menggambarkan konstruksi integrasi hubungan dan ketergantungan elemen, fitur-fitur dan bagaimana sistem tersebut bekerja (Marlissa, 2013).

Dilakukannya sebuah modelling system bertujuan untuk menganalisa dan memberi prediksi yang sangat mendekati kenyataan sebelum sebuah sistem nantinya diimplementasikan. Pemodelan secara umum merupakan pengembangan model matematika dengan bantuan *software* komputer. Simulasi pemodelan sistem diperlukan sebelum sistem yang ada diubah, bertujuan untuk meminimalkan terjadinya kesalahan atau ketidaksesuaian yang bakal terjadi. Pengembangan simulasi pemodelan system mempertimbangkan komponen-komponen seperti entitas yang terlibat dalam sistem, *variable input*, pengukuran kinerja dan hubungan fungsional. Validitas merupakan isu utama dari sebuah pemodelan sistem. Teknik validasi sebuah model dilakukan dengan cara mensimulasikan sebuah model menurut *input* yang diketahui dan kemudian membandingkan *output* yang dihasilkan model dengan *output* sistem sebenarnya (Marlissa, 2013).

Kesulitan untuk memprediksi dan mengamati proses tertentu pada *real world*, persoalan dimana model bisa saja memformulasikan sebuah proses tertentu namun tidak memungkinkan untuk melakukan analisa demi sebuah solusi, kurangnya data yang tidak memungkinkan dilakukannya validasi atas model matematik, secara umum merupakan alasan dibutuhkannya *modelling system* dan simulasi *modelling system* sebagai tool yang membantu dalam membuat keputusan pilihan. Namun, sebuah formulasi pemodelan sistem dan simulasinya sekaligus juga akan memakan banyak waktu bila berhadapan dengan sejumlah hambatan seperti ketidakjelasan sasaran, model yang diformulasikan dengan

tidak tepat (terlalu kompleks atau terlalu sederhana), menggunakan ekemen dan pengukuran kinerja yang tidak sesuai (Marlissa, 2013).

Penerapan pemodelan sistem dan simulasinya secara meluas dipergunakan pada bidang pemerintahan, pertahanan dan keamanan, sistem komunikasi, manufaktur, transportasi, kesehatan, lingkungan dan analisa bisnis. Kemampuan untuk mempelajari pengaruh informasi tertentu dan pengaruh dinamika lingkungan terhadap sebuah sistem operasi melalui sebuah pemodelan sistem dan simulasi pemodelan sistem tanpa mengganggu atau membebani sistem yang sedang berjalan, merupakan salah satu manfaat dari pemodelan sistem (Marlissa, 2013).

## 2.9 Blackbox Testing

Metode ujicoba *blackbox* memfokuskan pada keperluan fungsional dari software. Karena itu ujicoba *blackbox* memungkinkan pengembang *software* untuk membuat himpunan kondisi *input* yang akan melatih seluruh syarat-syarat fungsional suatu program. Ujicoba *blackbox* bukan merupakan alternatif dari ujicoba *whitebox*, tetapi merupakan pendekatan yang melengkapi untuk menemukan kesalahan lainnya, selain menggunakan metode *whitebox* (Ayuliana, 2009).

Ujicoba *blackbox* berusaha untuk menemukan kesalahan dalam beberapa kategori, diantaranya :

- a) Fungsi-fungsi yang salah atau hilang
- b) Kesalahan *interface*
- c) Kesalahan dalam struktur data atau akses *database* eksternal
- d) Kesalahan performa
- e) kesalahan inisialisasi dan terminasi

Tidak seperti metode *whitebox* yang dilaksanakan diawal proses, ujicoba *blackbox* diaplikasikan pada beberapa tahapan berikutnya. Karena ujicoba *blackbox* dengan sengaja mengabaikan struktur kontrol, sehingga perhatiannya difokuskan pada informasi *domain*. Uji coba didesain untuk dapat menjawab pertanyaan-pertanyaan berikut :

1. Bagaimana validitas fungsionalnya diuji.
2. Jenis input seperti apa yang akan menghasilkan kasus uji yang baik.
3. Apakah sistem secara khusus sensitif terhadap nilai input tertentu.
4. Bagaimana batasan-batasan kelas data diisolasi.
5. Berapa rasio data dan jumlah data yang dapat ditoleransi oleh sistem.
6. Apa akibat yang akan timbul dari kombinasi spesifik data pada operasi sistem.

Dengan mengaplikasikan uji coba *blackbox*, diharapkan dapat menghasilkan sekumpulan kasus uji yang memenuhi kriteria berikut :

1. Kasus uji yang berkurang, jika jumlahnya lebih dari 1, maka jumlah dari uji kasus tambahan harus didesain untuk mencapai uji coba yang cukup beralasan
2. Kasus uji yang memberitahukan sesuatu tentang keberadaan atau tidaknya suatu jenis kesalahan, daripada kesalahan yang terhubung hanya dengan suatu uji coba yang spesifik

### **Equivalence Partitioning**

*Equivalence partitioning* merupakan metode uji coba *blackbox* yang membagi *domain input* dari program menjadi beberapa kelas data dari kasus uji coba yang dihasilkan. Kasus uji penanganan *single* yang ideal menemukan sejumlah kesalahan (misalnya : kesalahan pemrosesan dari seluruh data karakter) yang merupakan syarat lain dari suatu kasus yang dieksekusi sebelum kesalahan umum diamati (Ayuliana, 2009).

*Equivalence partitioning* berusaha untuk mendefinisikan kasus uji yang menemukan sejumlah jenis kesalahan, dan mengurangi jumlah kasus uji yang harus dibuat. Kasus uji yang didesain untuk *Equivalence partitioning* berdasarkan pada evaluasi dari ekuivalensi jenis/class untuk kondisi *input*. *Class-class* yang ekuivalen merepresentasikan sekumpulan keadaan *valid* dan *invalid* untuk kondisi input. Biasanya kondisi *input* dapat berupa spesifikasi nilai numerik, kisaran nilai, kumpulan nilai yang berhubungan atau kondisi boolean. Ekuivalensi *class* dapat didefinisikan dengan panduan berikut :

- a) Jika kondisi input menspesifikasikan kisaran/*range*, maka didefinisikan 1 yang *valid* dan 2 yang *invalid* untuk *equivalence class*
- b) Jika kondisi input memerlukan nilai yang spesifik, maka didefinisikan 1 yang *valid* dan 2 yang *invalid* untuk *equivalence class*
- c) Jika kondisi input menspesifikasikan anggota dari himpunan, maka didefinisikan 1 yang *valid* dan 1 yang *invalid* untuk *equivalence class*
- d) Jika kondisi *input* adalah *boolean*, maka didefinisikan 1 yang *valid* dan 1 yang *invalid* untuk *equivalence class*

Misalkan, terdapat data terpelihara untuk sebuah aplikasi perbankan otomatis. User dapat mengaksesnya dari komputer pribadinya dengan menyediakan *password* 6 digit, dan mengikuti serangkaian perintah *keyword* yang mengakses berbagai fungsi perbankan. *Software* yang digunakan untuk aplikasi perbankan menerima data dalam bentuk :

*Area code* – *blank* atau 3 digit nomor

*Prefix* – 3 digit nomor yang tidak diawali oleh 0 atau 1

*Suffix* – 4 digit nomor

*Password* – 6 digit alphanumerik

*Commands* – “*check*”, “*deposit*”, “*bill pay*”, dsb

Kondisi *input* yang dihubungkan dengan setiap elemen data untuk aplikasi perbankan dapat dispesifikasikan sebagai :

*Area code* : kondisi *input*, *Boolean* – area code boleh ada maupun tidak

Kondisi *input*, *Range* – nilai didefinisikan antara 200 dan 999, dengan beberapa pengecualian khusus (misal : tidak ada nilai > 905) dan syarat (misal : seluruh *area code* memiliki angka 0 atau 1 pada posisi digit ke-2)

*Prefix* : kondisi *input*, *Range* – nilai yang dispesifikasikan > 200

*Suffix* : kondisi *input*, *Value* – sepanjang 4 digit

*Password* : kondisi *input*, *Boolean* – *Password* boleh ada maupun tidak  
kondisi *input*, *Value* – 6 string karakter

*Command* : kondisi *input*, *Set* – mengandung perintah-perintah yang ada diatas

Aplikasikan panduan untuk derivasi dari *class-class* yang ekuivalen, kasus uji untuk setiap *domain input* data item dapat dibentuk dan dieksekusi. Kasus uji dipilih sehingga sejumlah atribut dari *equivalence class* dieksekusi sekali saja (Ayuliana, 2009).

### **Boundary Value Analysis**

Sejumlah besar kesalahan cenderung terjadi dalam batasan domain input dari pada nilai tengah. Untuk alasan ini *boundary value analysis* (BVA) dibuat sebagai teknik ujicoba. BVA mengarahkan pada pemilihan kasus uji yang melatih nilai-nilai batas. BVA merupakan desain teknik kasus uji yang melengkapi *equivalence partitioning*. Dari pada memfokuskan hanya pada kondisi *input*, BVA juga menghasilkan kasus uji dari *domain output*. Panduan untuk BVA hampir sama pada beberapa bagian seperti yang disediakan untuk *equivalence partitioning* :

- Jika kondisi *input* menspesifikasikan kisaran yang dibatasi oleh nilai **a** dan **b**, kasus uji harus dibuat dengan nilai a dan b, sedikit diatas dan sedikit dibawah a dan b
- Jika kondisi *input* menspesifikasikan sejumlah nilai, kasus uji harus dibuat dengan melatih nilai maksimum dan minimum, juga nilai-nilai sedikit diatas dan sedikit dibawah nilai maksimum dan minimum tersebut.
- Aplikasikan panduan 1 dan 2 untuk kondisi *output*. Sebagai contoh, asumsikan tabel temperatur VS tabel tekanan sebagai *output* dari program analisis *engineering*. Kasus uji harus didesain untuk membuat laporan *output* yang menghasilkan nilai maksimum (dan minimum) yang mungkin untuk tabel masukan.

- d) Jika struktur data program *internal* telah mendeskripsikan batasan (misal : *array* ditetapkan maks. 100), maka desain kasus uji yang akan melatih struktur data pada batasan tersebut.

Kebanyakan pengembang *software* secara intuitif melakukan BVA pada beberapa tingkatan. Dengan mengaplikasikan panduan diatas, ujicoba batasan akan lebih lengkap, selain itu memiliki kemungkinan pendeteksian kesalahan yang lebih tinggi (Ayuliana, 2009).

### **Cause-Effect Graphing Techniques**

*Cause-effect graphing* merupakan desain teknik kasus ujicoba yang menyediakan representasi singkat mengenai kondisi logikal dan aksi yang berhubungan. Tekniknya mengikuti 4 tahapan berikut :

1. **Causes** (kondisi input), dan **Effects** (aksi) didaftarkan untuk modul dan *identifier* yang dtujukan untuk masing-masing
2. **Causes-effect graph** (seperti pada gambar dibawah) dibuat
3. *Graph* dikonversikan kedalam tabel keputusan
4. Aturan tabel keputusan dikonversikan kedalam kasus uji

## **2.10 Diabetes Melitus**

*Diabetes Melitus* merupakan suatu kelompok penyakit metabolik dengan karakteristik *hiperglikemia* yang terjadi karena kelainan *sekresi insulin*, kerja *insulin* atau kedua-duanya (Soegondo Sidartawan, 2005) (Tijar, 2011).

### **Diabetes Melitus tipe I**

*Diabetes Melitus tipe I* ditandai oleh penghancuran sel-sel *beta* pankreas. Kombinasi faktor genetik, imunologi dan mungkin pula lingkungan (misalnya, infeksi virus) diperkirakan turut menimbulkan destruksi sel *beta* (Tijar, 2011).

1. Faktor-faktor genetik

Penderita *diabetes* tidak mewarisi *diabetes* tipe I itu sendiri, tetapi mewarisi suatu *predisposisi* atau kecenderungan genetik ke arah terjadinya *Diabetes Melitus tipe I*. Kecendrungan genetik ini ditemukan pada individu yang memiliki tipe *antigen HLA (human leococyte antigen)* tertentu. HLA merupakan kumpulan gen yang bertanggung jawab atas *antigen trasplantasi* dan proses imun lainnya.

2. Faktor-faktor *imunologi*

Pada *diabetes* tipe I terdapat bukti adanya suatu respon *autoimun*. Respon ini merupakan respon *abnormal* dimana antibodi terarah pada jaringan normal tubuh dengan cara bereaksi terhadap jaringan tersebut yang dianggapnya seolah-olah sebagai jaringan asing (Smeltzer Suzanne C, 2001).

### 3. Virus dan bakteri

Virus penyebab DM adalah *rubela*, *mumps*, dan *human coxsackievirus B4*. Melalui mekanisme infeksi *sitolitik* dalam sel beta, virus ini mengakibatkan destruksi atau perusakan sel. Bisa juga, virus ini menyerang melalui reaksi autoimunitas yang menyebabkan hilangnya otoimun dalam sel *beta*. *Diabetes Melitus* akibat bakteri masih belum bisa dideteksi. Namun, para ahli kesehatan menduga bakteri cukup berperan menyebabkan DM.

### 4. Bahan toksik atau beracun

Bahan beracun yang mampu merusak sel *beta* secara langsung adalah *alloxan*, *pyrinuron (rodentisida)*, dan *streptozotcin* (produk dari sejenis jamur). Bahan lain adalah sianida yang berasal dari singkong (Maulana Mirza, 2009).

### **Diabetes Melitus tipe II**

Mekanisme yang tepat yang menyebabkan *resistensi insulin* dan gangguan *sekresi insulin* pada *diabetes* tipe II masih belum diketahui. Faktor genetik diperkirakan memegang peranan dalam proses terjadinya *resistensi insulin* (Smeltzer Suzanne C, 2001). Selain itu terdapat pula faktor-faktor resiko tertentu yang berhubungan dengan proses terjadinya *diabetes* tipe II. Menurut Hans Tandra (2008), faktor-faktor ini adalah (Tijar, 2011):

#### 1. Ras atau *Etnis*

Beberapa ras tertentu, seperti suku Indian di Amerika, Hispanik, dan orang Amerika di Afrika, mempunyai resiko lebih besar terkena *diabetes* tipe II. Kebanyakan orang dari ras-ras tersebut dulunya adalah pemburu dan petani dan biasanya kurus. Namun, sekarang makanan lebih banyak dan gerak badannya makin berkurang sehingga banyak mengalami *obesitas* sampai *diabetes*.

#### 2. *Obesitas*

Lebih dari 8 diantara 10 penderita *diabetes* tipe II adalah yang kelewat gemuk. Makin banyak jaringan lemak, jaringan tubuh dan otot akan makin resisten terhadap kerja insulin, terutama bila lemak tubuh atau kelebihan berat badan terkumpul di daerah sentral atau perut (*central obesity*). Lemak ini akan memblokir kerja insulin sehingga glukosa tidak dapat diangkut ke dalam sel dan menumpuk dalam peredaran darah.

#### 3. Kurang Gerak Badan

Makin kurang gerak badan, makin mudah seseorang terkena *diabetes*. Olahraga atau aktivitas fisik membantu untuk mengontrol berat badan. Glukosa darah dibakar menjadi energi. Sel-sel tubuh menjadi lebih sensitif terhadap insulin. Peredaran darah lebih baik. Dan resiko terjadinya *diabetes* tipe II akan turun sampai 50%.

#### 4. Penyakit Lain

Beberapa penyakit tertentu dalam prosesnya cenderung diikuti dengan tingginya kadar glukosa darah. Akibatnya, seseorang juga bisa terkena diabetes. Penyakit-penyakit itu antara lain hipertensi, penyakit jantung koroner, stroke, penyakit pembuluh darah perifer, atau infeksi kulit yang berlebihan.

#### 5. Usia

Resiko terkena diabetes akan meningkat dengan bertambahnya usia, terutama di atas 40 tahun. Namun, belakangan ini, dengan makin banyaknya anak yang mengalami obesitas, angka kejadian *diabetes* tipe II pada anak dan remaja pun meningkat.

## 2.11 Pengujian Sistem

### 2.11.1 Pengujian Validasi

Pengujian validasi berfokus kepada sudut pandang pengguna dan data keluaran dari sistem yang dapat dikenali oleh pengguna. Pengujian validasi berhasil ketika fungsi dari perangkat lunak dalam perilakunya dapat dijelaskan dan sesuai dengan keinginan pengguna (Pressman, 2010). Pengujian validasi merupakan pengujian untuk memenuhi kebutuhan fungsional dari sistem.

### 2.11.2 Pengujian Akurasi

Akurasi merupakan suatu perhitungan yang menyatakan seberapa dekat suatu hasil pengukuran terhadap angka sebenarnya. Semakin besar tingkat akurasi yang diperoleh oleh sistem maka semakin akurat hasil klasifikasi yang dilakukan. Pada penelitian ini tingkat akurasi dapat diperoleh dengan persamaan 2.6, sedangkan untuk mencari prosentase dalam tingkat akurasi dapat diperoleh dengan persamaan 2.7 (Puspasari, 2013).

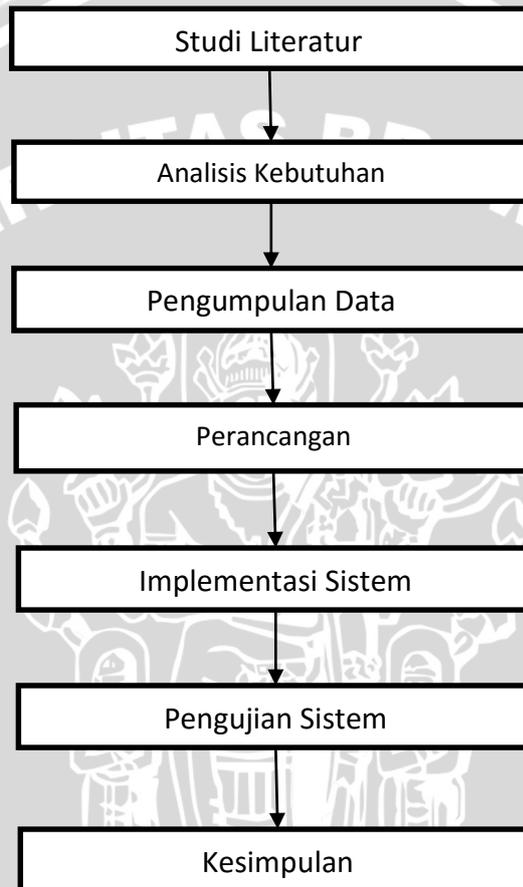
$$\text{tingkat akurasi} = \frac{\sum \text{data uji benar}}{\sum \text{total data uji}} \quad (2.6)$$

$$\text{akurasi (\%)} = \frac{\sum \text{data uji benar}}{\sum \text{total data uji}} \times 100 \quad (2.7)$$

Jumlah data uji benar merupakan jumlah *record* dari data yang diuji dalam pengklasifikasian dan hasilnya sama dengan kelas sebenarnya. Jumlah total data uji merupakan jumlah seluruh *record* yang digunakan dalam klasifikasi (semua data uji).

## BAB 3 METODOLOGI

Pada bab ini menjelaskan tahapan yang akan dilakukan dalam pembuatan sistem. Tahapan tersebut berisi studi literatur, analisis kebutuhan, pengumpulan data, perancangan, implementasi sistem, pengujian sistem, dan kesimpulan. Pada gambar 3.1 ditunjukkan gambaran diagram blok dari tahapan yang akan dilakukan dalam metodologi penelitian.



**Gambar 3.1** Diagram Alir Metodologi Penelitian

### 3.1 Studi Literatur

Studi literatur mempelajari literatur dari beberapa bidang ilmu yang berhubungan dengan pembuatan sistem pakar untuk identifikasi penyakit *diabetes melitus*, diantaranya:

- a) Sistem Pakar
- b) Algoritma *K-Nearest Neighbour* dan *Certainty Factor*
- c) *Diabetes Melitus*

Literatur tersebut diperoleh dari buku, jurnal, website, e-book dan penelitian sebelumnya.

### 3.2 Analisis Kebutuhan

Analisis kebutuhan digunakan untuk mendapatkan seluruh kebutuhan yang diperlukan dalam membangun sistem pakar sesuai dengan metode *k-nearest neighbour - certainty factor*. Analisis kebutuhan dibagi menjadi 2 yaitu kebutuhan fungsional dan kebutuhan non-fungsional. Kebutuhan fungsional merupakan kebutuhan sistem dalam hal yang bisa dilakukan oleh sistem, sedangkan kebutuhan non-fungsional merupakan kebutuhan sistem dalam hal pendukung, platform berjalannya sistem, dan spesifikasi software dan hardware yang dibutuhkan dalam menjalankan sistem tersebut. Pada tahap ini yang dilakukan analisis terhadap semua spesifikasi kebutuhan yang diinginkan oleh pengguna untuk membangun sistem. Hasil dari analisis kebutuhan tersebut berupa informasi yang akan digunakan dalam membangun sistem agar sesuai dengan harapan pengguna. Berikut ini adalah analisis kebutuhan fungsional sistem dalam implementasinya:

1. Perangkat lunak yang dikembangkan harus memiliki tampilan yang dapat memudahkan pengguna dalam mengoperasikan program tersebut.
2. Perangkat lunak dapat melakukan proses perhitungan dengan menggunakan metode yang akan digunakan.
3. Perangkat lunak dapat menentukan hasil berdasarkan perhitungan metode untuk menentukan sebuah keputusan berdasarkan pakar.

Kebutuhan non-fungsional sistem yang dibutuhkan untuk melakukan pengembangan sistem yaitu dengan menggunakan bahasa pemrograman PHP dan menggunakan MySQL sebagai penyimpanan database. Sistem dibangun mempunyai tampilan berupa web karena menggunakan bahasa pemrograman PHP.

### 3.3 Pengumpulan Data

Lokasi penelitian tugas akhir ini terletak di Puskesmas Poncokusumo Tumpang Kabupaten Malang. Variabel penelitian pada skripsi ini adalah gejala-gejala dari pasien *diabetes melitus* dan untuk menentukan masuk kedalam diabetes tipe 1 atau 2 berdasarkan hasil nilai perhitungan tiap gejala dengan menggunakan algoritma *K-Nearest Neighbour - Certainty Factor*.

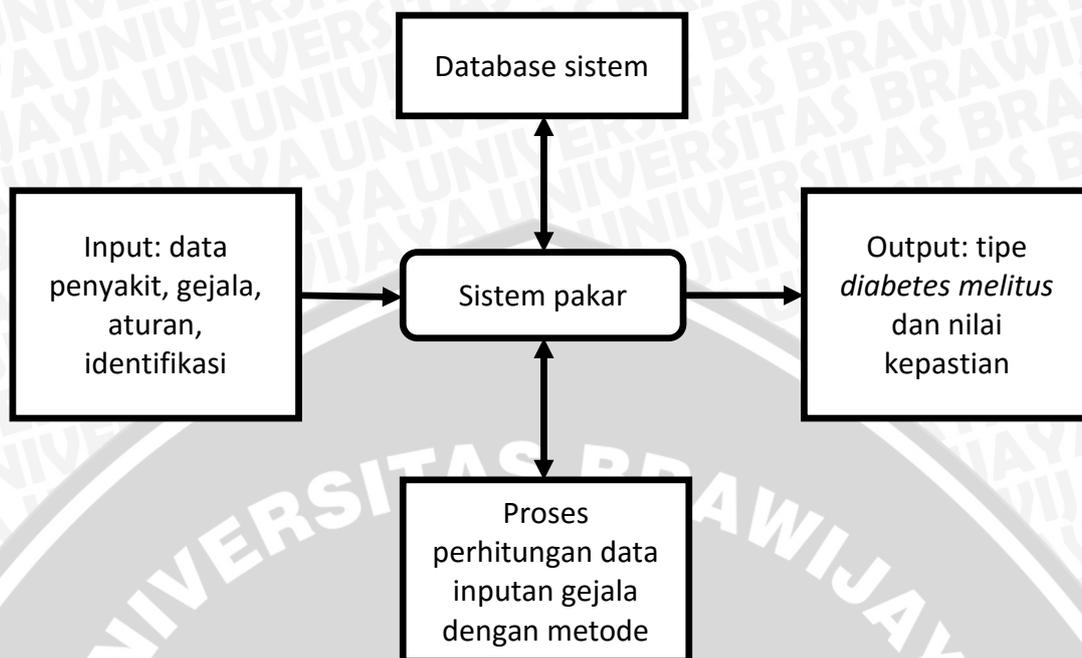
Data didapat berdasarkan konsultasi dengan pakar dan referensi lain mengenai *diabetes melitus*. Data diolah sedemikian rupa berdasarkan hubungan permasalahan yang berkaitan satu sama lain. Selain itu, data juga didapatkan langsung dari responden penelitian. Metode pengumpulan data primer yang bersifat kuantitatif dapat menggunakan instrumen kuisioner dan wawancara. Penjelasan mengenai pengumpulan data ditunjukkan pada tabel 3.1.

**Tabel 3.1** Tabel Pengumpulan Data

No.	Kebutuhan Data	Sumber Data	Metode	Kegunaan Data
1.	Data tentang <i>diabetes melitus</i>	Buku, internet, dan pakar	Mencari literatur yang relevan di buku dan internet, serta wawancara kepada pakar.	Sebagai data pengetahuan tentang <i>diabetes melitus</i>
2.	Data penyakit <i>diabetes melitus</i>	Puskesmas Poncokusumo	Observasi	Data yang diperoleh akan digunakan dalam metode

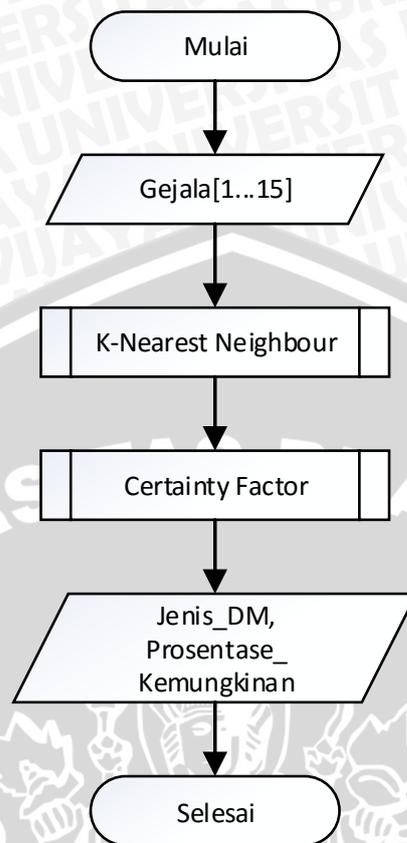
### 3.4 Perancangan

Pemodelan Sistem Pakar Untuk Mendeteksi Penyakit *Diabetes Melitus* Menggunakan Metode *K-Nearest Neighbour - Certainty Factor*, digunakan untuk klastering dan menghitung kepastian berdasarkan data sampel/hasil observasi. Nilai perhitungan dari kedua metode tersebut dapat digunakan untuk proses pengambilan keputusan dalam melakukan identifikasi. Hasil output sistem ini terdiri dari: jenis penyakit *diabetes melitus* dan nilai kepastian dari penyakit. Blok diagram dari perancangan yang akan dibuat ditunjukkan pada gambar 3.2:



**Gambar 3.2** Blok Diagram Perancangan Sistem

Pada gambar 3.3 dijelaskan bahwa diagram alir dari perancangan sistem yang akan dibuat. Pertama sistem akan memulai, kemudian user akan menginputkan data gejala yang ada. Setelah itu akan memasuki proses perhitungan KNN, dalam proses KNN dihitung jarak yang kemudian disorting menurut nilai terkecil. Hasil dari sorting tersebut akan menunjukkan masuk ke dalam diabetes tipe 1 atau 2. Dari perhitungan KNN dilanjutkan ke perhitungan CF, dalam perhitungan CF hanya melanjutkan hasil klasterisasi KNN. CF akan menghitung nilai inputan user yang kemudian menghasilkan nilai kepastian terhadap tipe diabetes yang telah diketahui. Output keseluruhan dari sistem ini adalah hasil klasterisasi dan nilai kepastian penyakit yang diketahui.



**Gambar 3.3** Diagram alir perancangan sistem

### 3.5 Implementasi Sistem

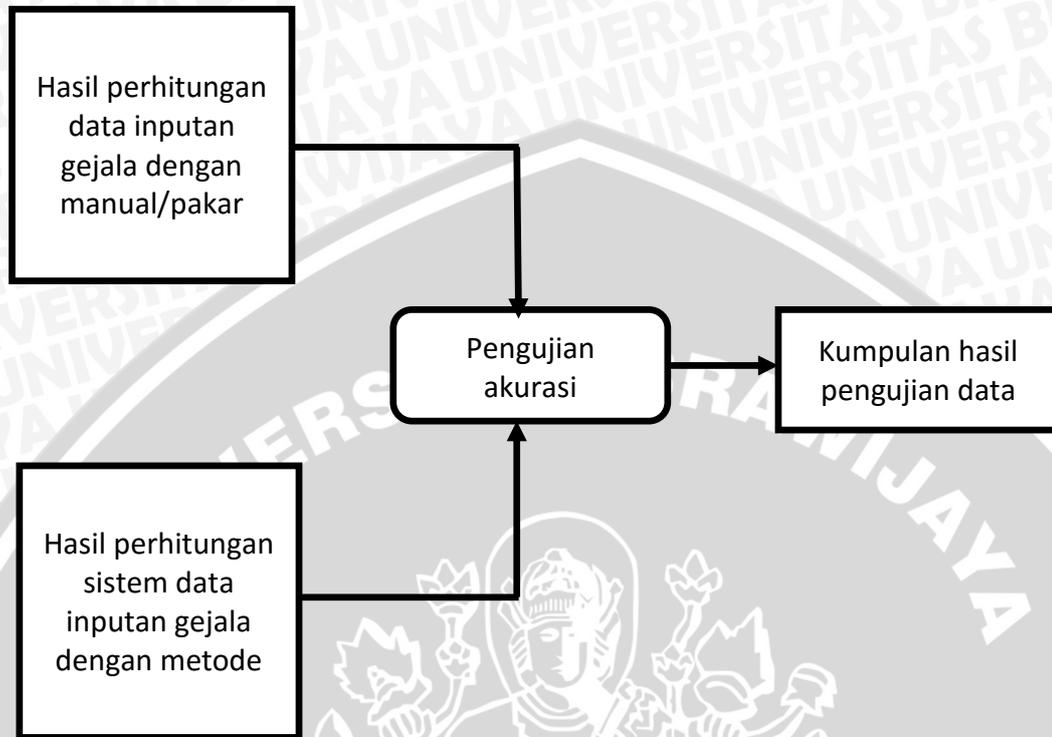
Pada implementasi sistem akan membahas mengenai bagaimana desain dari pemodelan sistem pakar menggunakan metode *K-Nearest Neighbour - Certainty Factor*. Implementasi perangkat lunak digunakan dengan menggunakan bahasa pemrograman berorientasi objek yaitu menggunakan bahasa PHP. Implementasi system meliputi:

- Pembuatan *user interface* dan penerapan algoritma *Metode K-Nearest Neighbour - Certainty Factor* dalam program yang dibuat dengan PHP.
- Memasukkan data pelatihan ke penampungan data, disini digunakan PHPMyAdmin sebagai penyimpanan data latih.
- Melakukan proses perhitungan terhadap nilai yang telah dimasukkan dengan *Metode K-Neareset Neighbour - Certainty Factor*.

### 3.6 Pengujian Sistem

Pada tahap ini dilakukan pengujian keberhasilan dan akurasi sistem yang telah dibuat pada tahap implementasi. Pengujian dilakukan dengan cara memeriksa apakah sistem sudah berjalan dengan baik dan tidak ada error yang terjadi. Selain itu, pengujian juga dilakukan dengan cara membandingkan hasil klasifikasi dari

sistem dengan hasil identifikasi yang dilakukan oleh pakar. Blok diagram pengujian sistem ditunjukkan pada Gambar 3.4.



**Gambar 3.4** Blok Diagram Pengujian Sistem

Pengujian *black box* juga akan dilakukan, hal ini sebagai indikator keberhasilan pada setiap fungsionalitas yang ada pada sistem pakar. Tabel 3.2 adalah contoh tabel hasil pengujian sistem dengan hasil pengujian pakar.

**Tabel 3.2** Akurasi Perbandingan Hasil Pengujian

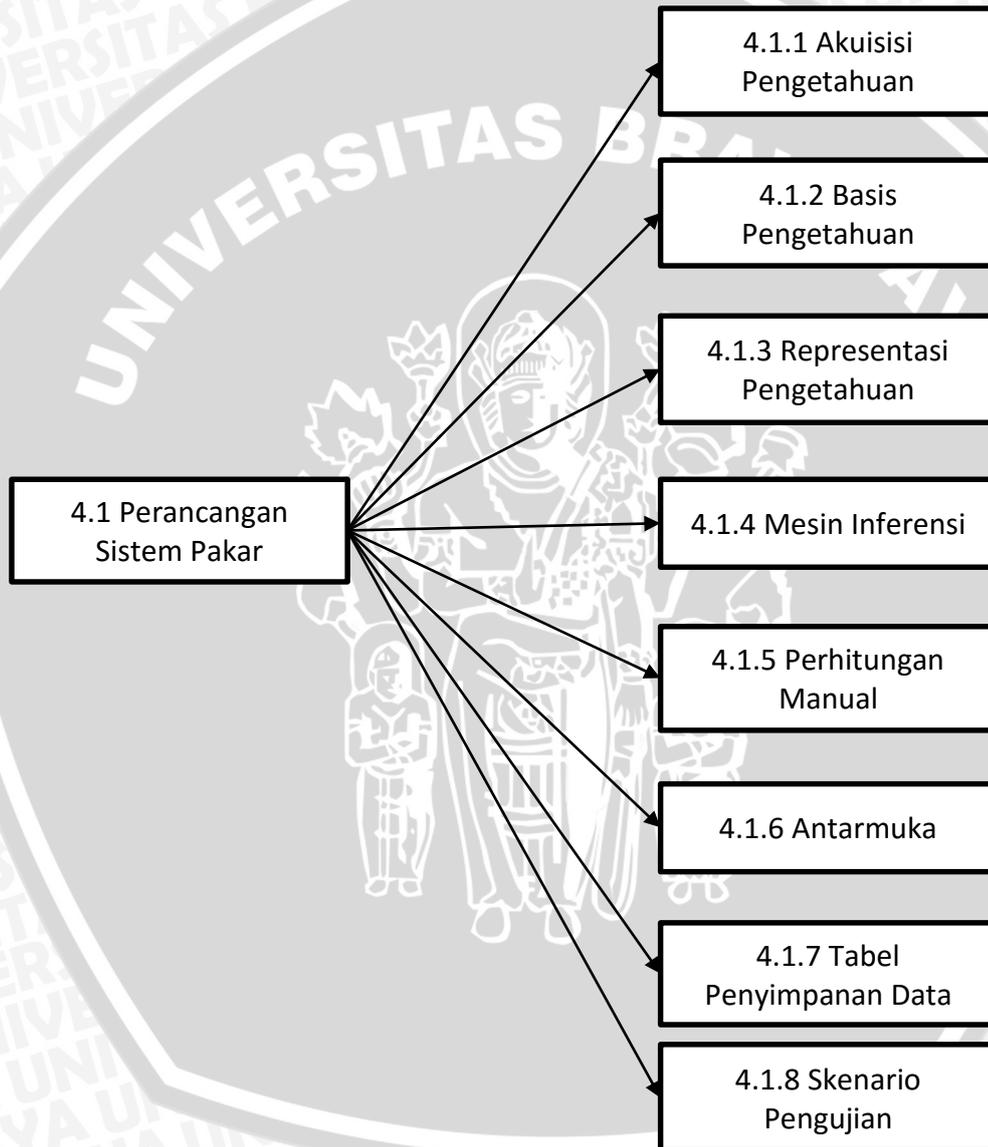
No.	Gejala	Hasil Diagnosa Sistem	Hasil Diagnosa Pakar	Akurasi Hasil Perbandingan	Prosentase Sistem

### 3.7 Kesimpulan

Pembuatan kesimpulan dilakukan setelah semua tahapan perancangan, implementasi, dan pengujian sistem aplikasi telah selesai dilakukan. Kesimpulan diambil dari hasil pengujian dan analisis terhadap sistem yang dibangun. Tahap terakhir dari penulisan adalah saran yang dimaksudkan untuk memberikan pertimbangan atas pengembangan aplikasi selanjutnya.

## BAB 4 PERANCANGAN

Bab ini akan membahas perancangan yang meliputi perancangan sistem pakar. Perancangan sistem pakar merupakan tahapan-tahapan yang terdapat pada sistem pakar. Urutan yang ada pada tahapan perancangan sistem pakar akan dijelaskan pada gambar 4.1.



Gambar 4.1 Pohon Perancangan

## 4.1 Perancangan Sistem Pakar

Tahapan ini bertujuan untuk mengubah model informasi yang telah dibuat pada tahapan sebelumnya. Perancangan yang dilakukan adalah perancangan untuk seluruh subsistem yang terdapat dalam arsitektur sistem pakar. Perancangan tersebut meliputi akuisisi pengetahuan, basis pengetahuan, representasi pengetahuan, mesin inferensi, fasilitas penjelas, perhitungan manual, dan antarmuka.

### 4.1.1 Akuisisi Pengetahuan

Akuisisi pengetahuan adalah akumulasi, transfer dan transformasi keahlian dalam menyelesaikan masalah dari sumber pengetahuan ke dalam program komputer. Dalam tahap ini *knowledge engineer* berusaha menyerap pengetahuan untuk selanjutnya ditransfer ke dalam basis pengetahuan. Terdapat tiga metode utama dalam akuisisi pengetahuan, yaitu: wawancara, analisis protokol dan observasi pada pekerjaan pakar.

Proses akuisisi pengetahuan melibatkan identifikasi pengetahuan merepresentasikan pengetahuan dalam format yang sesuai, menyusun pengetahuan, dan mentransfer pengetahuan ke mesin. Metode yang digunakan dalam proses akuisisi pengetahuan adalah metode wawancara.

Dari proses wawancara dengan pakar, diperoleh beberapa faktor yang dapat digunakan dalam menentukan jenis penyakit *diabetes melitus* tipe 1 atau 2. Pada tabel 4.1 akan ditunjukkan mengenai gejala yang ada pada diabetes tipe 1.

**Tabel 4.1** Gejala Diabetes Tipe 1

Gejala	MB	MD
Banyak makan (polifagia)	0,8	0,4
Sering buang air kecil (poliuria)	0,9	0,4
Faktor keturunan	1	0
Umur kurang atau sama dengan (15-20) tahun	1	0
Mudah kelelahan	0,5	0,2
Banyak minum (polodipsia)	0,8	0,1
Turunnya berat badan tanpa alasan yang jelas	0,1	0,4
Timbulnya rasa kesemutan (mati rasa) atau sakit pada tangan atau kaki	0,2	0,7
Timbulnya borok (luka) pada kaki yang tak kunjung sembuh	0,1	0,7
Gatal-gatal	0,1	0,5
Kadar glukosa darah lebih dari 140 mg/dl (sewaktu)	0,5	0,5

Sedangkan pada tabel 4.2 akan ditunjukkan mengenai gejala yang ada pada diabetes tipe 2.

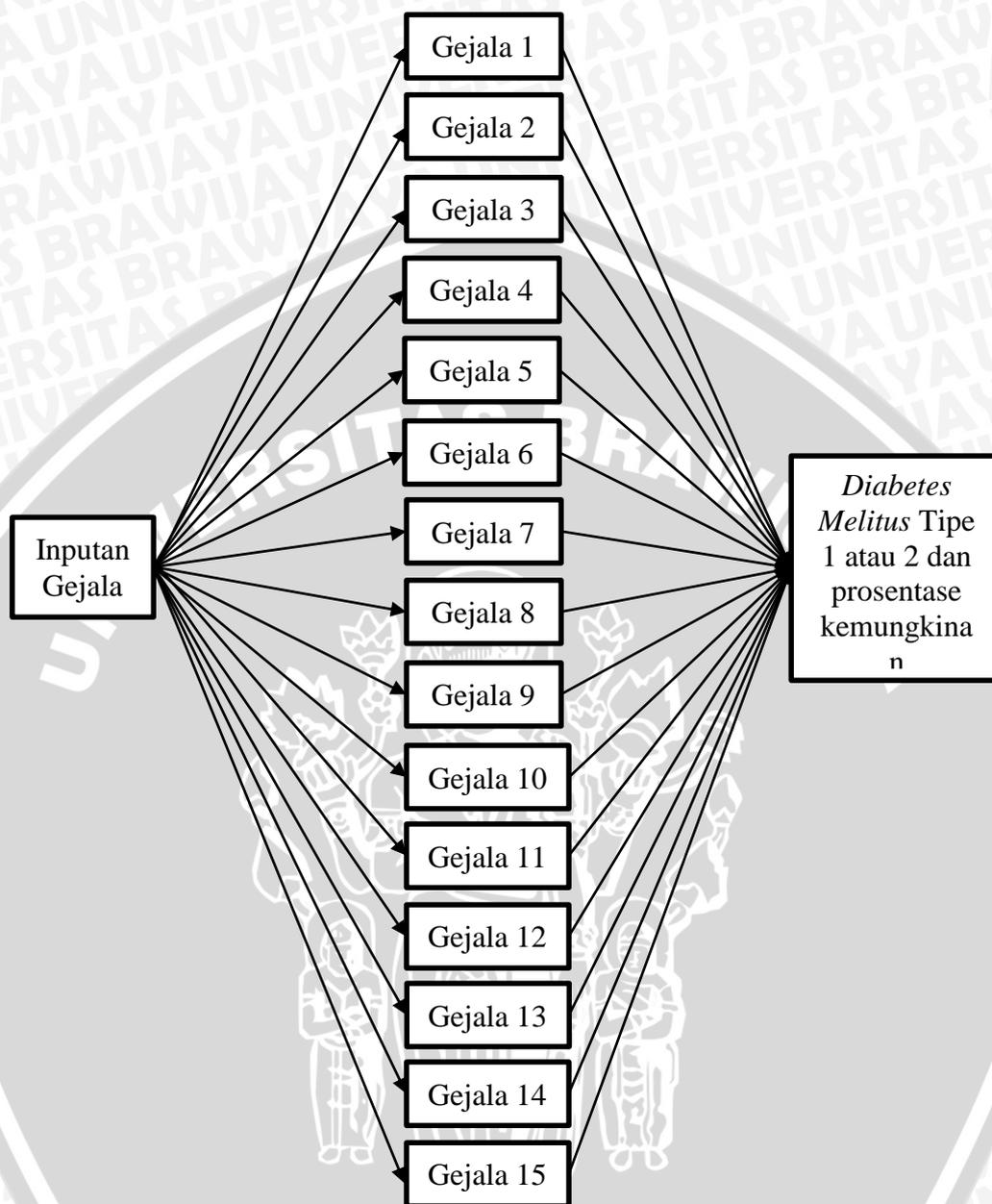
Tabel 4.2 Gejala Diabetes Tipe 2

Gejala	MB	MD
Banyak makan (polifagia)	0,8	0,4
Sering buang air kecil (poliuria)	0,9	0,4
Banyak minum (polodipsia)	0,8	0,1
Turunnya berat badan tanpa alasan yang jelas	0,5	0,5
Faktor keturunan	0,5	0,5
Mulut kering	0,2	0
Sering mengantuk	0,4	0,2
Timbulnya rasa kesemutan (mati rasa) atau sakit pada tangan atau kaki	0,2	0,5
Timbulnya rasa borok (luka) pada kaki yang tak kunjung sembuh	0,5	0,3
Gatal-gatal	0,2	0,3
Disfungsi ereksi pada pria	0,2	0,7
Obesitas 20% dari berat badan normal	0,6	0,2
Kadar glukosa darah lebih dari 140 mg/dl (sewaktu)	0,5	0,5

Nilai dari MB merupakan nilai dari kepercayaan terhadap gejala-gejala yang ada pada penyakit *diabetes melitus*. Range nilai yang diberikan antara 0 sampai 1, nilai tersebut merupakan ukuran yang diberikan dari hasil wawancara dari pakar. Apabila nilai mendekati 1 maka nilai kepercayaan semakin besar, sebaliknya bila mendekati 0 maka nilai kepercayaan semakin kecil terhadap gejala pada penyakit tersebut.

#### 4.1.2 Subsistem Basis Pengetahuan

Basis pengetahuan berisi pengetahuan relevan yang diperlukan untuk memahami, merumuskan, dan memecahkan masalah persoalan. Basis tersebut mencakup dua elemen dasar yaitu fakta dan aturan khusus yang mengarahkan penggunaan pengetahuan untuk memecahkan persoalan khusus dalam domain tertentu yang ditunjukkan pada gambar 4.2.



**Gambar 4.2** Bagan Paramater Perhitungan Hasil

Pada gambar 4.2 ditunjukkan bahwa pada sistem terdapat nilai inputan sebanyak 15 yaitu banyak makan (polifagia), sering buang air kecil (poliuria), banyak minum (polodipsia), turunnya berat badan tanpa alasan yang jelas, umur (15-20), faktor keturunan, mulut kering, mudah kelelahan, sering mengantuk, timbulnya rasa kesemutan (mati rasa) atau sakit pada tangan atau kaki, timbulnya borok (luka) pada kaki yang tak kunjung sembuh, gatal-gatal, disfungsi ereksi pada pria, obesitas 20% dari berat badan normal, dan kadar glukosa darah lebih dari 140 mg/dl (sewaktu). Dari inputan tersebut akan diperoleh output berupa *Diabetes Mellitus* Tipe 1 atau 2 beserta prosentase kemungkinan.

Seorang pakar memiliki pengetahuan tentang masalah khusus yang disebut *domain knowledge*. *Domain knowledge* mencakup pengetahuan pakar dalam menentukan faktor-faktor yang digunakan untuk mendiagnosa penyakit hepatitis. Fakta yang digunakan pada sistem ini berasal dari user kemudian pakar memberikan konklusi dengan menggabungkan fakta-fakta yang ada dengan basis pengetahuan.

#### 4.1.3 Representasi Pengetahuan

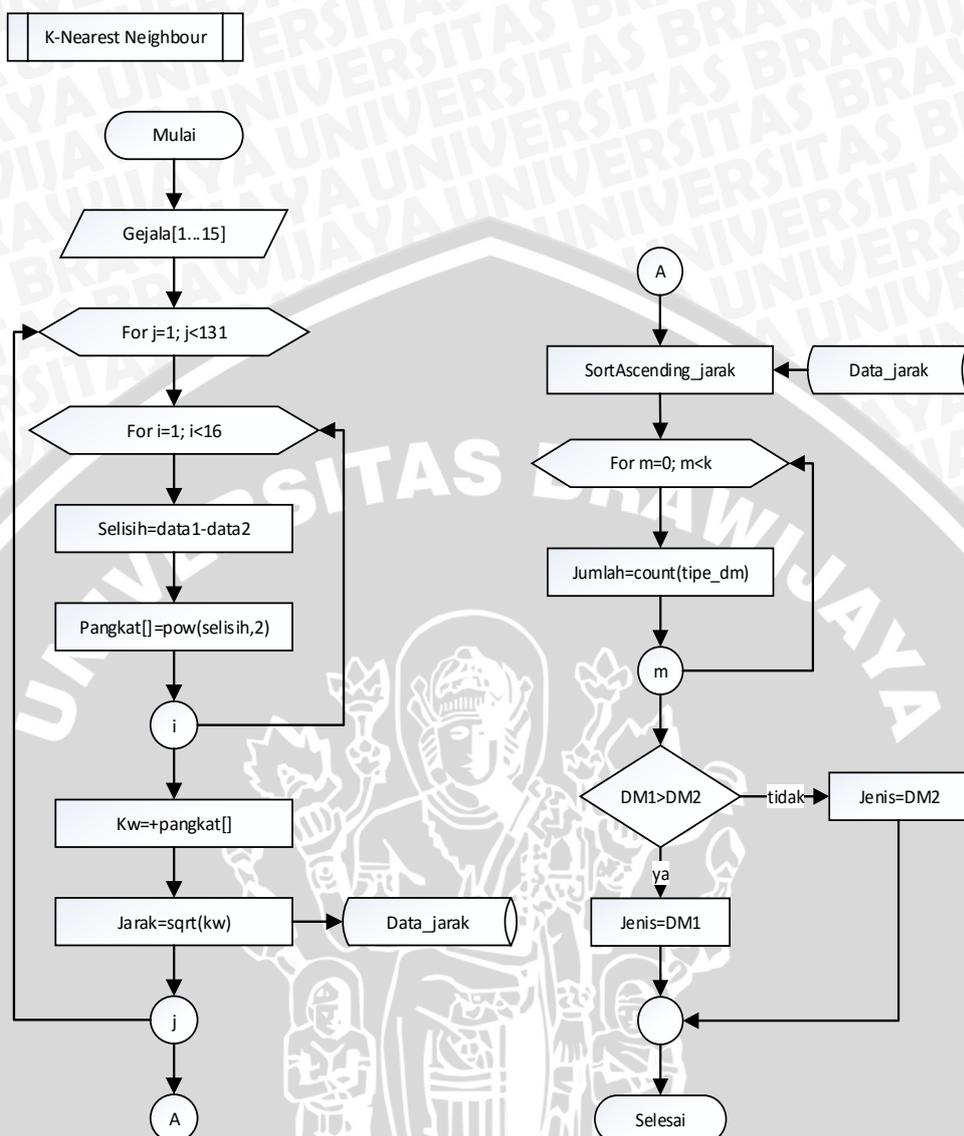
Representasi pengetahuan adalah cara untuk menyajikan pengetahuan yang diperoleh ke dalam suatu skema sehingga dapat diketahui relasi antara suatu pengetahuan dengan pengetahuan yang lain. Representasi pengetahuan disajikan dalam bentuk jika-maka (IF-THEN). Metode yang akan digunakan adalah *k-nearest neighbour – certainty factor* yang mana metode pertama akan menghasilkan klasifikasi sedangkan untuk metode kedua berupa prosentase kemungkinan yang merupakan lanjutan dari metode pertama.

- IF sorting menurut k inputan > DM1 THEN tipe DM1
- IF sorting menurut k inputan > DM2 THEN tipe DM2
- IF hasil knn DM1 THEN prosentase kemungkinan DM1
- IF hasil knn DM2 THEN prosentase kemungkinan DM2

#### 4.1.4 Mesin Inferensi

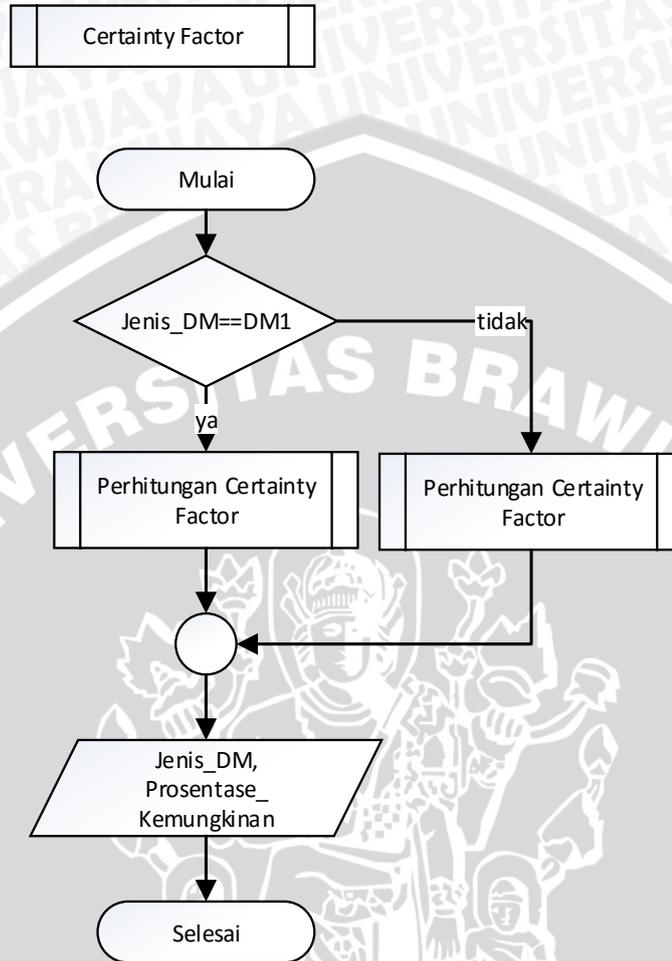
Mesin inferensi merupakan program komputer yang memberikan metodologi untuk penalaran tentang informasi yang ada dalam basis pengetahuan dan untuk mensintesis kesimpulan. Mesin inferensi berperan sebagai otak dalam sistem pakar yang berfungsi untuk memandu proses penalaran berdasarkan suatu kondisi berdasarkan pada basis pengetahuan yang ada. Komponen ini mengandung mekanisme pola pikir dan penalaran yang digunakan oleh pakar dalam menyelesaikan suatu masalah.

Dalam mesin inferensi, terdapat dua macam teknik pelacakan, yaitu *forward chaining* (runut maju) dan *backward chaining* (runut mundur). Namun apabila untuk permasalahan yang kompleks dan tidak terdapat aturan yang didefinisikan untuk permasalahan tersebut, maka akan membutuhkan aturan yang sangat banyak. Pada penelitian ini menggunakan metode *k-nearest neighbour-certainty factor*, yang salah satu metode yaitu *certainty factor* merupakan metode yang bisa digunakan untuk mengatasi ketidakpastian. Penjelasan dari mesin inferensi mengenai metode *k-nearest neighbour-certainty factor* ditunjukkan pada gambar 4.3.

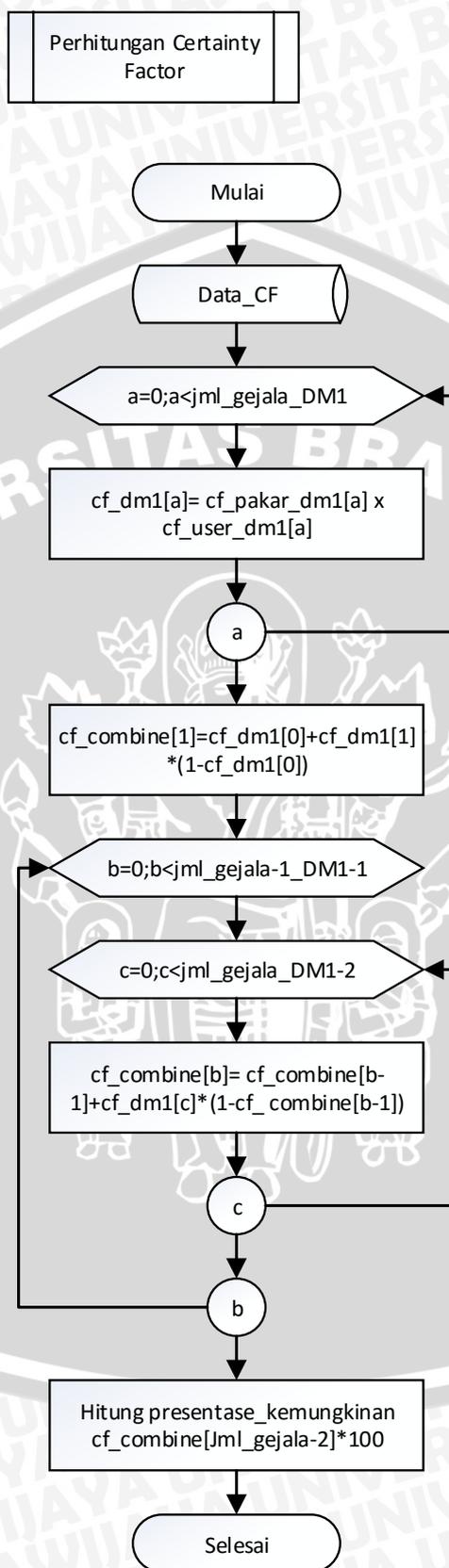


Gambar 4.3 Flowchart K-Nearest Neighbour

Sedangkan penjelasan dari mesin inferensi mengenai metode *certainty factor* tersebut ditunjukkan pada gambar 4.4 dan gambar 4.5.



**Gambar 4.4** Flowchart *Certainty Factor*



Gambar 4.5 Flowchart Sub Proses Certainty Factor

### 4.1.5 Perhitungan Manual

Perhitungan manual ini akan menjelaskan mengenai ilustrasi perhitungan yang akan diberikan dalam mendeteksi penyakit *diabetes melitus* menggunakan metode *k-nearest neighbour-certainty factor*. Proses pertama adalah perhitungan *k-nearest neighbour* yang terdiri dari perhitungan jarak, sorting, dan penentuan klas berdasarkan k inputan yang menghasilkan tipe dari *diabetes melitus*. Kemudian dilanjutkan dengan proses kedua yaitu perhitungan *certainty factor* yang merupakan lanjutan dari perhitungan sebelumnya yang sudah menghasilkan tipe dari *diabetes melitus*. Perhitungan *certainty factor* dimulai dengan CF user yang dikombinasikan dengan CF pakar dan menghasilkan CF Combine. Nilai CF combine akan menghasilkan prosentase kemungkinan dari tipe diabetes yang sudah diketahui dari perhitungan sebelumnya.

Pada proses KNN, langkah pertama yang dilakukan adalah normalisasi dari setiap data baik itu data baru maupun data latih. Dikarenakan data yang akan digunakan sudah berupa nilai 0 dan 1, maka tidak perlu adanya normalisasi pada data. Tabel 4.3 adalah tabel data latih yang akan digunakan dalam perhitungan manual.

Tabel 4.3 Data Latih

No	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14	G15	Tipe
1	1	1	0	1	1	1	0	0	1	0	1	1	0	0	1	DM1
2	0	0	1	1	1	1	0	0	0	0	0	0	0	1	1	DM1
3	1	0	1	0	1	1	1	0	1	0	0	0	0	0	1	DM1
4	1	1	1	0	1	1	0	1	1	1	1	0	1	0	1	DM1
5	0	1	1	0	1	1	0	0	0	1	1	0	1	1	1	DM1
6	1	1	0	0	1	1	0	1	0	1	0	0	1	1	0	DM1
7	1	0	1	1	1	1	0	0	1	0	0	0	1	1	0	DM1
8	1	1	1	1	1	1	1	0	0	1	0	1	1	1	1	DM1
9	1	1	1	0	1	1	1	1	0	0	0	0	0	0	0	DM1
10	1	1	0	0	1	1	1	1	1	1	1	0	0	1	0	DM1
11	1	0	0	1	1	1	0	1	0	1	0	0	1	1	1	DM1
12	1	0	0	1	1	1	1	1	1	1	1	0	1	1	1	DM1
13	0	1	1	1	1	1	1	0	0	1	1	0	0	0	1	DM1
14	1	0	0	1	1	1	0	1	1	1	0	0	0	0	1	DM1
15	1	1	0	1	1	1	1	1	0	0	0	0	1	1	1	DM1
16	0	0	1	1	1	1	1	0	1	0	1	0	0	1	1	DM1



Tabel 4.4 Data Latih (lanjutan)

No	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14	G15	Tipe
17	0	1	0	1	1	1	0	1	0	1	1	1	1	1	0	DM1
18	0	1	1	0	1	1	0	1	0	0	1	1	1	1	0	DM1
19	1	0	1	1	1	1	1	0	1	1	1	1	1	1	1	DM1
20	1	1	0	0	1	1	0	0	1	1	0	0	0	0	1	DM1
21	0	0	1	1	1	1	0	0	0	0	1	0	0	1	1	DM1
22	1	0	1	1	1	1	0	1	0	1	0	0	1	0	1	DM1
23	1	1	1	1	1	1	0	0	0	1	0	0	1	0	0	DM1
24	1	1	0	0	1	1	0	1	0	0	1	1	0	0	0	DM1
25	0	1	1	1	1	1	0	1	0	1	1	0	1	0	0	DM1
26	1	1	1	0	1	1	0	1	1	1	0	0	1	0	1	DM1
27	1	1	1	0	1	1	1	1	0	0	0	1	0	0	0	DM1
28	1	0	1	1	1	1	0	0	1	0	1	0	1	1	0	DM1
29	1	1	1	1	1	1	1	0	0	1	0	1	1	0	1	DM1
30	0	1	1	1	1	1	1	0	0	1	1	0	1	0	1	DM1
31	1	1	0	1	1	1	1	1	0	0	0	0	0	1	1	DM1
32	1	0	1	1	1	1	1	0	1	1	0	1	0	1	1	DM1
33	1	0	1	1	1	1	0	1	0	0	0	0	1	0	1	DM1
34	1	1	1	1	1	1	0	1	0	1	0	0	1	0	0	DM1
35	1	1	0	0	1	1	1	1	0	1	1	0	0	1	0	DM1
36	1	1	1	0	0	0	0	0	1	0	1	1	1	1	1	DM2
37	1	1	1	0	0	0	0	1	0	1	0	0	0	1	1	DM2
38	1	0	1	1	0	0	1	1	1	1	1	1	1	1	1	DM2
39	1	0	1	1	0	0	1	1	1	1	1	0	0	0	1	DM2
40	1	0	1	0	0	0	1	0	0	1	0	1	0	0	1	DM2
41	0	1	1	1	0	0	0	1	1	1	1	0	1	0	1	DM2
42	0	1	1	1	0	0	1	1	0	1	1	1	1	0	1	DM2
43	1	0	1	0	0	0	0	0	1	0	1	1	0	1	1	DM2
44	1	1	0	1	0	0	0	0	0	0	0	0	1	1	1	DM2
45	1	1	0	0	0	0	1	1	1	0	1	1	0	0	1	DM2
46	1	1	0	1	0	0	1	0	1	1	1	0	1	1	1	DM2
47	1	0	1	0	0	0	1	1	1	1	0	1	0	1	1	DM2



Tabel 4.5 Data Latih (lanjutan)

No	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14	G15	Tipe
48	1	1	0	1	0	0	1	1	1	1	1	0	0	0	1	DM2
49	1	1	1	1	0	0	1	1	1	1	0	1	0	1	1	DM2
50	1	1	1	0	0	0	1	0	0	0	1	0	0	0	1	DM2
51	1	1	0	1	0	0	0	0	0	0	0	0	0	0	1	DM2
52	1	1	0	1	0	0	0	1	0	0	0	0	0	0	1	DM2
53	0	1	1	0	0	0	0	1	0	0	1	0	0	0	1	DM2
54	1	0	1	1	0	0	1	1	1	1	1	0	1	0	1	DM2
55	0	0	1	1	0	0	1	0	0	0	0	0	0	1	1	DM2
56	1	0	1	1	0	0	0	1	0	0	1	0	0	0	1	DM2
57	0	1	1	1	0	0	1	1	1	1	0	0	1	0	1	DM2
58	0	1	0	1	0	0	1	1	1	0	0	1	1	1	1	DM2
59	1	0	1	1	0	0	0	1	0	0	0	0	0	0	1	DM2
60	1	0	1	1	0	0	0	0	0	0	1	0	1	0	1	DM2
61	1	0	1	1	0	0	1	1	1	1	0	0	0	0	1	DM2
62	1	0	1	1	0	0	0	1	1	1	0	0	1	0	1	DM2
63	0	1	1	1	0	0	1	0	0	0	0	1	0	0	0	DM2
64	0	0	1	1	0	0	1	0	0	1	1	1	0	0	0	DM2
65	1	0	1	0	0	0	0	0	0	1	1	1	0	1	0	DM2
66	1	1	0	1	0	0	0	0	0	1	0	1	1	1	0	DM2
67	0	1	0	1	0	0	1	0	1	1	1	0	1	0	0	DM2
68	1	1	1	0	0	0	0	1	0	0	1	0	0	0	0	DM2
69	1	0	1	1	0	0	0	0	0	0	0	0	0	0	1	DM2
70	1	0	0	1	0	0	0	0	0	0	1	0	1	1	0	DM2
71	1	1	0	1	0	0	1	0	0	1	1	1	0	1	1	DM2
72	1	1	1	0	0	0	1	1	0	0	1	0	1	0	1	DM2
73	1	1	0	1	0	0	1	1	0	0	0	1	1	1	1	DM2
74	0	1	1	1	0	0	1	0	1	0	0	1	1	1	1	DM2
75	1	0	1	0	0	0	0	1	0	1	0	0	1	0	1	DM2
76	1	1	1	0	0	0	0	1	0	1	1	0	0	1	1	DM2
77	1	1	1	1	0	0	1	0	1	0	0	0	0	1	1	DM2
78	1	1	1	1	0	0	0	0	1	0	0	1	0	0	1	DM2



Tabel 4.6 Data Latih (lanjutan)

No	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14	G15	Tipe
79	1	1	0	1	0	0	1	0	0	1	0	0	1	1	1	DM2
80	1	1	1	0	0	0	1	0	0	1	1	0	1	1	1	DM2
81	1	1	1	0	0	0	0	0	1	1	0	1	1	1	1	DM2
82	0	1	1	1	0	0	1	0	1	0	1	1	1	0	1	DM2
83	1	1	1	0	0	0	0	1	0	0	0	1	0	1	1	DM2
84	1	0	1	1	0	0	0	1	1	0	1	1	0	1	1	DM2
85	0	1	1	0	0	0	0	0	1	1	1	0	1	0	1	DM2
86	0	1	1	1	0	0	0	0	1	0	1	1	0	1	1	DM2
87	1	0	1	1	0	0	0	1	0	1	0	0	1	1	1	DM2
88	0	1	1	1	0	0	1	1	1	1	0	1	1	1	1	DM2
89	1	1	0	1	0	0	1	1	0	1	1	0	0	0	1	DM2
90	1	0	1	1	0	0	1	0	0	1	0	1	0	1	1	DM2
91	1	0	1	1	0	0	0	1	1	1	0	0	1	0	1	DM2
92	1	1	0	1	0	0	1	1	0	0	1	1	1	0	1	DM2
93	1	0	1	1	0	0	0	0	1	0	1	1	0	1	1	DM2
94	1	1	0	1	0	0	0	0	0	0	0	0	1	1	1	DM2
95	1	1	1	0	0	0	0	1	1	0	1	0	0	0	1	DM2
96	1	0	1	1	0	0	1	0	1	0	1	0	1	1	1	DM2
97	1	0	1	0	0	0	1	0	1	1	0	1	0	1	1	DM2
98	1	1	1	0	0	0	1	1	0	1	1	0	0	0	1	DM2
99	1	1	1	1	0	0	1	1	0	1	0	1	0	1	1	DM2
100	1	0	1	1	0	0	1	0	0	0	1	0	1	0	1	DM2
101	0	1	1	1	0	0	0	1	0	1	0	0	0	0	1	DM2
102	1	1	1	0	0	0	0	1	0	0	0	0	1	0	1	DM2
103	1	0	1	0	0	0	0	1	0	0	1	0	0	0	1	DM2
104	0	1	1	1	0	0	1	1	0	1	1	0	1	0	1	DM2
105	1	0	1	1	0	0	1	0	0	0	1	0	0	1	1	DM2
106	1	1	1	0	0	0	0	1	0	0	1	0	1	0	1	DM2
107	1	0	1	1	0	0	1	1	0	1	0	0	1	0	1	DM2
108	0	1	1	0	0	0	1	1	1	0	0	1	0	1	1	DM2
109	1	1	0	1	0	0	0	1	0	0	1	0	0	0	1	DM2



Tabel 4.7 Data Latih (lanjutan)

No	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14	G15	Tipe
110	1	1	1	0	0	0	0	0	1	0	1	0	1	0	1	DM2
111	1	0	1	1	0	0	1	0	1	1	0	0	0	0	1	DM2
112	1	1	0	1	0	0	0	1	1	0	0	0	1	0	1	DM2
113	1	1	1	0	0	0	1	0	0	1	0	1	0	0	0	DM2
114	0	1	1	1	0	0	1	0	0	1	0	1	0	0	0	DM2
115	1	0	1	1	0	0	0	0	0	0	1	1	0	1	0	DM2
116	1	1	1	0	0	0	0	0	0	1	0	1	0	1	0	DM2
117	0	1	1	0	0	0	1	0	1	0	1	0	1	0	0	DM2
118	1	0	1	1	0	0	0	1	0	0	1	0	1	0	0	DM2
119	1	1	0	1	0	0	0	0	1	0	0	0	0	0	1	DM2
120	0	1	1	1	0	0	0	1	0	0	1	0	1	1	0	DM2
121	1	0	1	1	0	0	1	0	0	1	0	1	0	1	1	DM2
122	0	1	1	1	0	0	0	1	0	1	0	1	1	0	1	DM2
123	1	1	0	1	0	0	0	1	0	1	0	0	0	1	1	DM2
124	1	0	1	1	0	0	1	0	1	0	0	0	0	1	1	DM2
125	1	1	1	0	0	0	0	0	1	1	0	1	0	0	1	DM2
126	0	1	1	1	0	0	0	1	1	0	1	1	0	1	1	DM2
127	1	1	1	0	0	0	0	0	1	1	0	0	1	0	1	DM2
128	1	0	1	1	0	0	0	0	1	0	0	1	0	1	1	DM2
129	0	1	1	1	0	0	0	1	0	1	0	0	1	0	1	DM2
130	1	1	0	1	0	0	1	1	1	1	0	1	0	1	1	DM2

Selanjutnya, hitung jarak terdekat antara data uji dengan data latih. Perhitungan untuk mencari jarak menggunakan *euclidean* ditunjukkan pada persamaan 2.2. Dimana  $x_1$  dan  $x_2$  merupakan dua record dengan  $n$  atribut. Persamaan menghitung jarak antara  $x_1$  dan  $x_2$ , dengan tujuan untuk menentukan perbedaan antara nilai-nilai atribut pada *record*  $x_1$  dan  $x_2$ . Setelah diketahui jarak antar *record*, kemudian diambil sebanyak  $k$  tetangga terdekat untuk memprediksi label kelas dari *record* baru menggunakan label kelas tetangga. Diambil kelas mayoritas sebagai kelas target *output* data yang baru. Tabel 4.4 adalah contoh kasus dari perhitungan manual.

Tabel 4.8 Contoh Kasus

No	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14	G15
1	0	1	0	1	1	1	0	0	1	0	0	1	1	0	1



Dari tabel diatas kemudian dihitung jarak antara data latih dengan data contoh kasus dengan menggunakan rumus *euclidean*. Hasil dari perhitungan jarak diberikan pada tabel 4.5.

Tabel 4.9 Jarak

No	Jarak	No	Jarak	No	Jarak	No	Jarak
1	1,7321	36	2,6458	71	3,0000	106	3,0000
2	2,4495	37	3,3166	72	3,1623	107	3,1623
3	2,6458	38	3,1623	73	2,6458	108	2,8284
4	2,6458	39	3,3166	74	2,2361	109	2,8284
5	2,6458	40	3,1623	75	3,1623	110	2,6458
6	2,8284	41	2,6458	76	3,4641	111	3,0000
7	2,4495	42	2,8284	77	2,8284	112	2,2361
8	2,4495	43	3,0000	78	2,2361	113	3,1623
9	3,0000	44	2,4495	79	2,8284	114	2,8284
10	3,1623	45	2,8284	80	3,3166	115	3,1623
11	2,6458	46	2,8284	81	2,6458	116	3,1623
12	2,8284	47	3,3166	82	2,2361	117	2,8284
13	2,6458	48	3,0000	83	3,0000	118	3,1623
14	2,4495	49	3,0000	84	3,0000	119	2,2361
15	2,4495	50	3,1623	85	2,6458	120	3,0000
16	2,6458	51	2,4495	86	2,4495	121	3,1623
17	2,4495	52	2,6458	87	3,1623	122	2,4495
18	2,6458	53	3,0000	88	2,6458	123	3,0000
19	2,6458	54	3,1623	89	3,1623	124	3,0000
20	2,2361	55	3,0000	90	3,1623	125	2,6458
21	2,6458	56	3,1623	91	2,8284	126	2,6458
22	2,6458	57	2,6458	92	2,6458	127	2,6458
23	2,4495	58	2,2361	93	2,8284	128	2,6458
24	2,6458	59	3,0000	94	2,4495	129	2,6458
25	2,6458	60	2,8284	95	3,0000	130	2,8284
26	2,4495	61	3,1623	96	3,0000		
27	2,8284	62	2,8284	97	3,1623		
28	2,6458	63	2,6458	98	3,4641		
29	2,2361	64	3,1623	99	3,1623		
30	2,4495	65	3,4641	100	3,0000		
31	2,6458	66	2,6458	101	2,8284		
32	2,6458	67	2,6458	102	2,8284		
33	2,4495	68	3,3166	103	3,3166		



**Tabel 4.10** Jarak (lanjutan)

No	Jarak	No	Jarak	No	Jarak
34	2,6458	69	2,8284	104	3,0000
35	3,3166	70	3,0000	105	3,3166

Setelah diketahui nilai dari jarak setiap gejala yang diketahui, selanjutnya dilakukan sorting berdasarkan nilai jarak terkecil ke terbesar. Hasil sorting dijelaskan pada tabel 4.6.

**Tabel 4.11** Sorting

No	Jarak	No	Jarak	No	Jarak	No	Jarak
1	1,7321	25	2,6458	93	2,8284	64	3,1623
20	2,2361	28	2,6458	101	2,8284	72	3,1623
29	2,2361	31	2,6458	102	2,8284	75	3,1623
58	2,2361	32	2,6458	108	2,8284	87	3,1623
74	2,2361	34	2,6458	109	2,8284	89	3,1623
78	2,2361	36	2,6458	114	2,8284	90	3,1623
82	2,2361	41	2,6458	117	2,8284	97	3,1623
112	2,2361	52	2,6458	130	2,8284	99	3,1623
119	2,2361	57	2,6458	9	3,0000	107	3,1623
2	2,4495	63	2,6458	43	3,0000	113	3,1623
7	2,4495	66	2,6458	48	3,0000	115	3,1623
8	2,4495	67	2,6458	49	3,0000	116	3,1623
14	2,4495	73	2,6458	53	3,0000	118	3,1623
15	2,4495	81	2,6458	55	3,0000	121	3,1623
17	2,4495	85	2,6458	59	3,0000	35	3,3166
23	2,4495	88	2,6458	70	3,0000	37	3,3166
26	2,4495	92	2,6458	71	3,0000	39	3,3166
30	2,4495	110	2,6458	83	3,0000	47	3,3166
33	2,4495	125	2,6458	84	3,0000	68	3,3166
44	2,4495	126	2,6458	95	3,0000	80	3,3166
51	2,4495	127	2,6458	96	3,0000	103	3,3166
86	2,4495	128	2,6458	100	3,0000	105	3,3166
94	2,4495	129	2,6458	104	3,0000	65	3,4641
122	2,4495	6	2,8284	106	3,0000	76	3,4641
3	2,6458	12	2,8284	111	3,0000	98	3,4641
4	2,6458	27	2,8284	120	3,0000		
5	2,6458	42	2,8284	123	3,0000		

**Tabel 4.12** Sorting (lanjutan)

No	Jarak	No	Jarak	No	Jarak
11	2,6458	45	2,8284	124	3,0000
13	2,6458	46	2,8284	10	3,1623
16	2,6458	60	2,8284	38	3,1623
18	2,6458	62	2,8284	40	3,1623
19	2,6458	69	2,8284	50	3,1623
21	2,6458	77	2,8284	54	3,1623
22	2,6458	79	2,8284	56	3,1623
24	2,6458	91	2,8284	61	3,1623

Kemudian dilihat nilai  $k$  inputan yang diberikan oleh user, misalkan  $k = 5$ . Maka diambil data 5 teratas dari tabel sorting, berikut ini penjelasannya.

Data pertama  $\rightarrow 1,7321$  DM1

Data kedua  $\rightarrow 2,2361$  DM1

Data ketiga  $\rightarrow 2,2361$  DM1

Data keempat  $\rightarrow 2,2361$  DM2

Data kelima  $\rightarrow 2,2361$  DM2

Setelah diketahui nilai dari sorting kelima data dan kelas *diabetes melitus*, maka contoh kasus masuk ke dalam DM1 menurut perhitungan *k-nearest neighbour*. Langkah selanjutnya adalah menghitung *certainty factor* berdasarkan output yang dihasilkan dari perhitungan *k-nearest neighbour*. Tabel 4.4 contoh kasus untuk CF User.

**Tabel 4.13** CF User

Gejala	CF User
G2	1
G4	1
G5	1
G6	1
G9	1
G12	1
G13	1
G15	1

Selain menggunakan CF User diperlukan nilai dari pakar yaitu berupa CF Pakar yang terdiri dari CF Pakar DM1 dan CF Pakar DM2. Nilai dari kedua CF Pakar akan diberikan pada table 4.8.

**Tabel 4.14** CF Pakar DM1 dan DM2

Gejala	CF Pakar DM1	CF Pakar DM2
G1	0,8	0,8
G2	0,9	0,9
G3	0,8	0,8
G4	0,1	0,5
G5	1	-
G6	1	0,5
G7	-	0,2
G8	0,5	-
Gejala	CF Pakar DM1	CF Pakar DM2
G9	-	0,4
G10	0,2	0,2
G11	0,1	0,5
G12	0,1	0,2
G13	-	0,2
G14	-	0,6
G15	0,5	0,5

Langkah selanjutnya adalah menghitung nilai CF Combine dari CF User dan CF Pakar. Disini hanya akan menggunakan CF Pakar DM1 karena dari perhitungan KNN sudah diketahui bahwa contoh kasus masuk ke DM 1, maka nilai prosentase yang dihitung hanya CF Combine DM1. Perhitungan ini akan menggunakan persamaan 2.5. Proses perhitungan akan diberikan sebagai berikut ini:

$$\begin{aligned}
 \text{CF(A)} &= \text{CF1} + [\text{CF2} * (1-\text{CF1})] \\
 &= 0,9 + [0,1 * (1-0,9)] \\
 &= 0,91
 \end{aligned}$$

$$\begin{aligned}
 \text{CF(B)} &= \text{CF3} + [\text{CF(A)} * (1-\text{CF3})] \\
 &= 1 + [0,91 * (1-1)] \\
 &= 1
 \end{aligned}$$

$$\begin{aligned}
 \text{CF(C)} &= \text{CF4} + [\text{CF(B)} * (1-\text{CF4})] \\
 &= 1 + [1 * (1-1)] \\
 &= 1
 \end{aligned}$$

$$\begin{aligned}
 \text{CF(D)} &= \text{CF5} + [\text{CF(C)} * (1-\text{CF5})] \\
 &= 0,1 + [1 * (1-0,1)] \\
 &= 1
 \end{aligned}$$

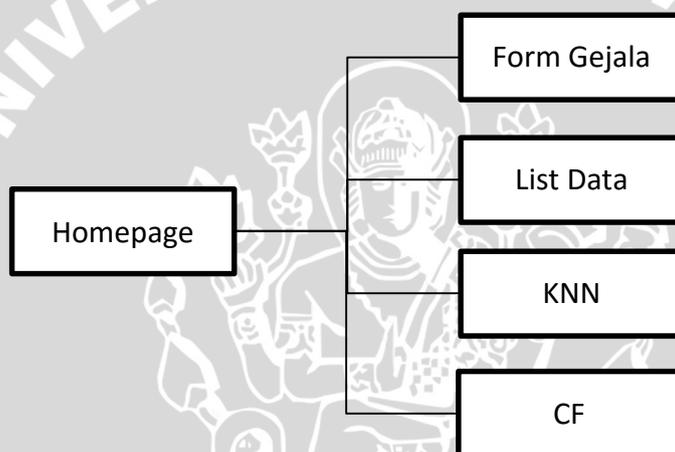
$$\begin{aligned}
 \text{CF(E)} &= \text{CF6} + [\text{CF(D)} * (1-\text{CF6})] \\
 &= 0,5 + [1 * (1-0,5)]
 \end{aligned}$$

= 1

Kemudian dibandingkan nilai dari CF Combine yang memiliki nilai paling besar. Nilai terbesar dari CF Combine DM1 adalah 1, maka nilai  $1 \times 100\%$  memiliki hasil 100%. Jadi dari perhitungan knn dan cf diperoleh hasil untuk contoh kasus masuk kedalam *diabetes melitus* tipe 1 dan memiliki nilai prosentase kemungkinan terhadap *diabetes melitus* tipe 1 sebesar 100%.

#### 4.1.6 Antarmuka

Perancangan antarmuka pada “Pemodelan Sistem Pakar Untuk Mendeteksi Penyakit *Diabetes Melitus* Menggunakan Metode *K-Nearest Neighbour - Certainty Factor*” terdiri dari homepage, form gejala, list data, knn, dan cf. Antarmuka yang dibuat pada sistem ini akan dijelaskan pada gambar 4.6 sitemap.

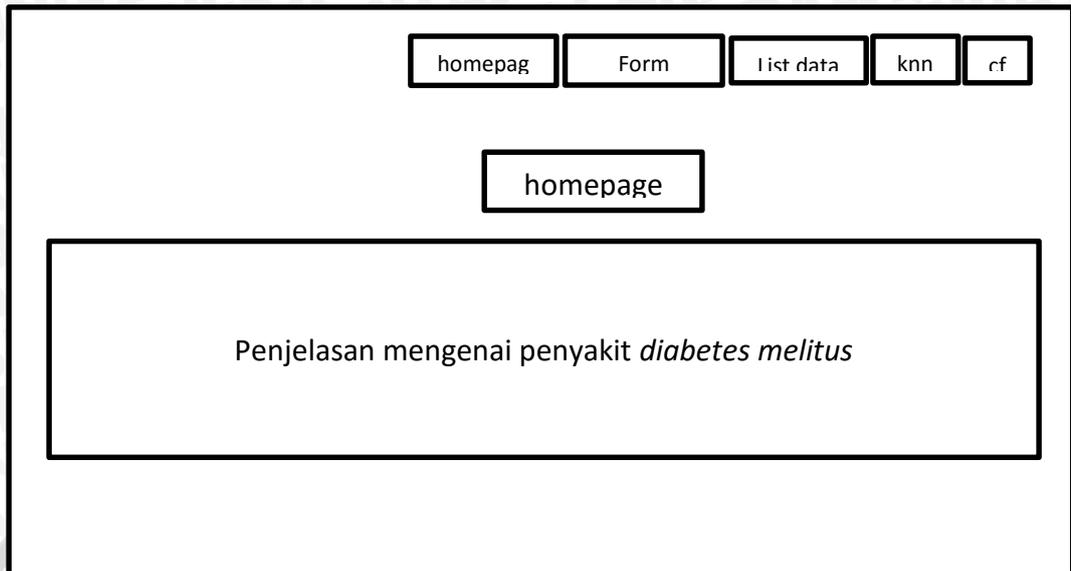


**Gambar 4.6** Sitemap

Penjelasan dari sitemap gambar 4.6 akan diberikan sebagai berikut ini:

- Homepage

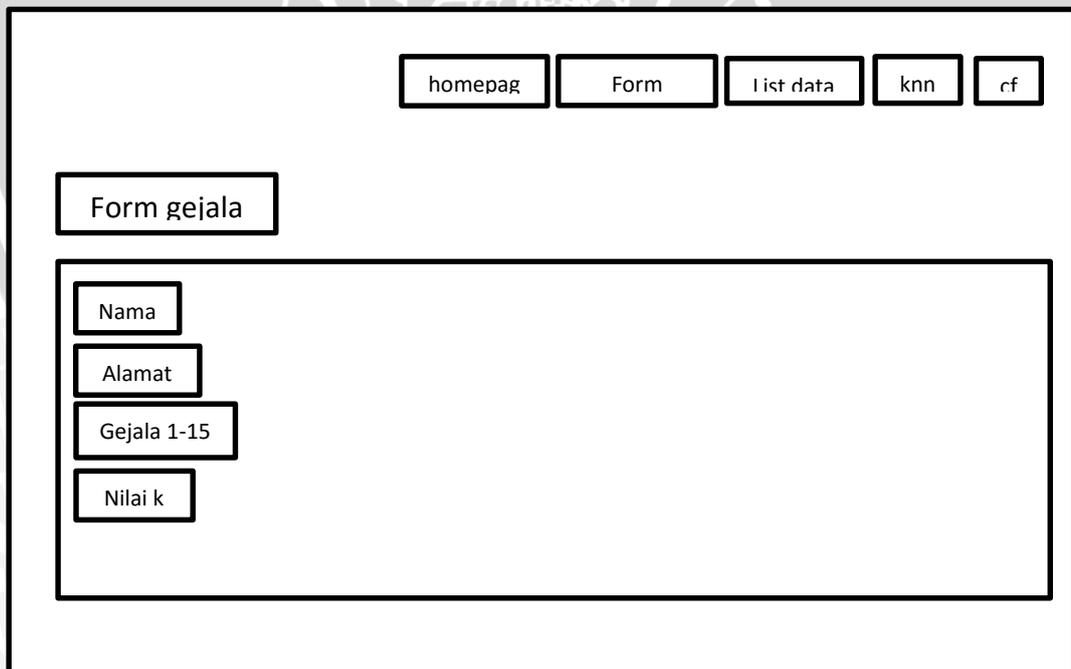
Homepage adalah halaman pertama dari sistem yang akan dibuat. Disini terdapat penjelasan mengenai penyakit *diabetes melitus* yang bertujuan untuk memberikan pengetahuan pada user mengenai penyakit *diabetes melitus*. Gambar 4.7 adalah homepage dari sistem yang akan dibuat.



Gambar 4.7 Homepage

- Form Gejala

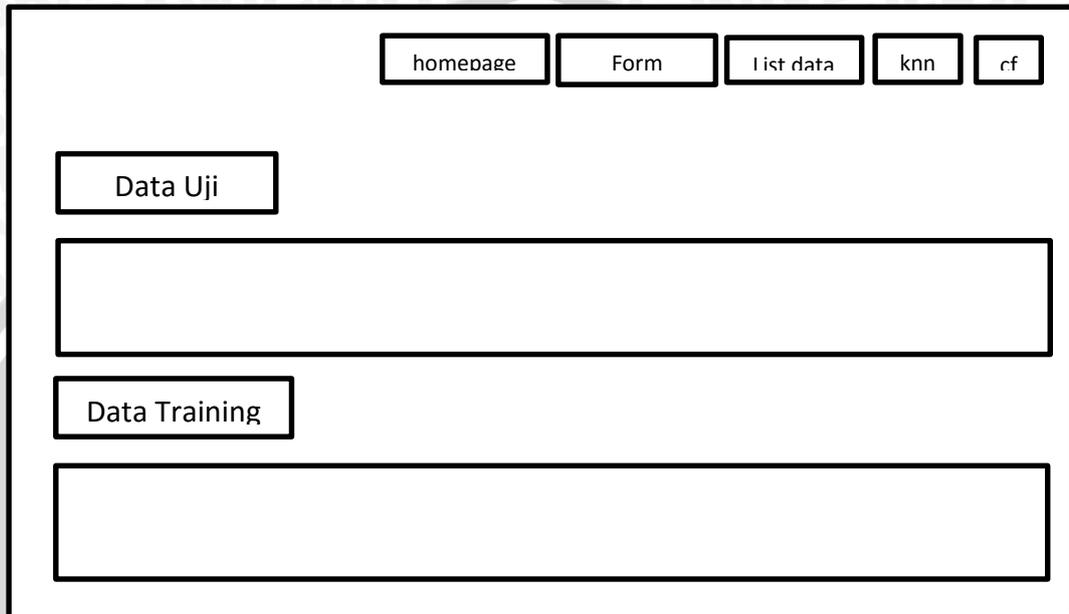
Halaman yang merupakan kelanjutan dari homepage berupa form gejala. Pada halaman ini terdapat 18 inputan dari user berupa nama, alamat, gejala penyakit dari 1-15 dan nilai k. Penjelasan mengenai form gejala ditunjukkan pada gambar 4.8.



Gambar 4.8 Form Gejala

- List Data

List data merupakan kumpulan dari data uji dan data training. Data uji didapatkan dari inputan form yang diberikan oleh user, sedangkan data training adalah data yang didapatkan dari puskesmas. Penjelasan dari list data akan diberikan pada gambar 4.9:

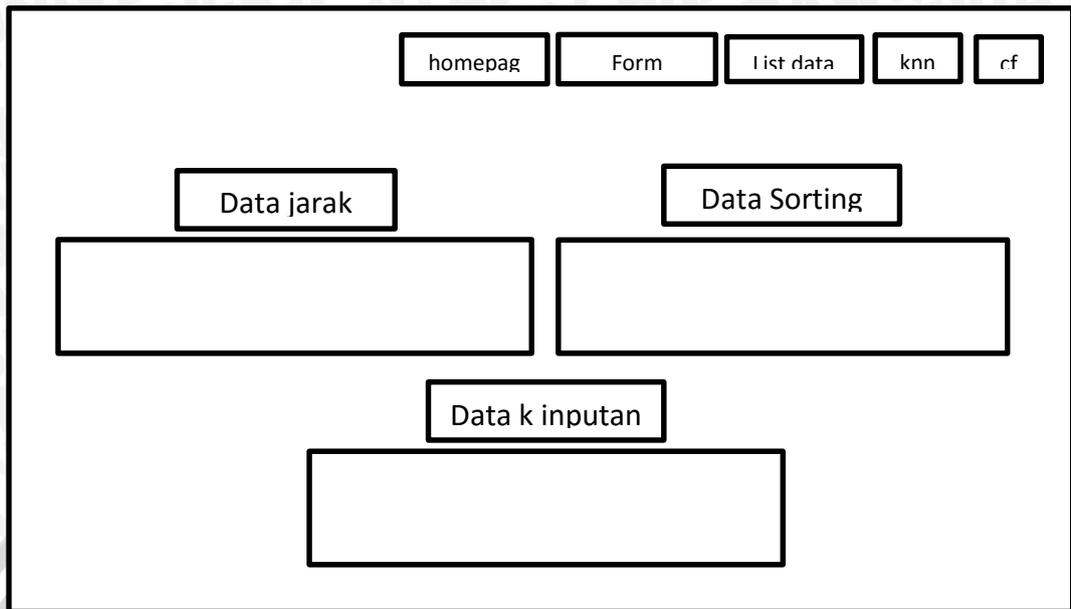


The screenshot shows a web application interface with a navigation bar at the top containing buttons for 'homepage', 'Form', 'List data', 'knn', and 'cf'. Below the navigation bar, there are two main sections: 'Data Uji' and 'Data Training'. Each section has a corresponding empty rectangular input field below its label.

**Gambar 4.9** List Data

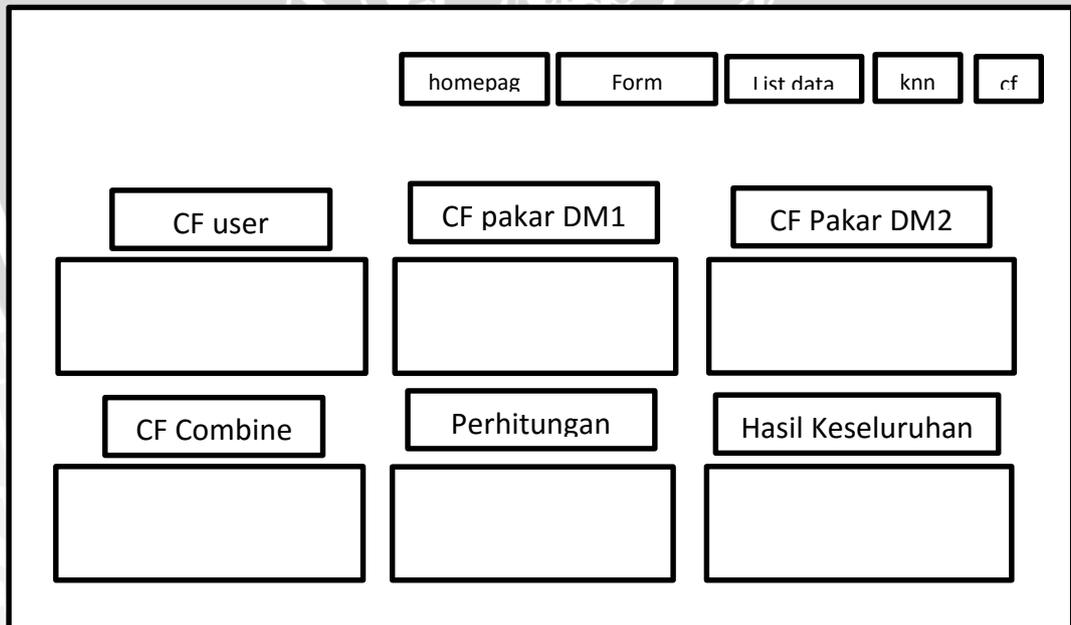
- KNN

KNN adalah halaman perhitungan *k-nearest neighbour* yang terdiri dari data jarak dan data sorting. Data jarak merupakan hasil perhitungan jarak dari data uji dengan data training sedangkan data sorting adalah hasil sorting dari data jarak. Pada gambar 4.10 akan dijelaskan mengenai KNN.



Gambar 4.10 KNN

- CF  
Halaman terakhir adalah CF yang berupa CF User, CF Pakar, CF Combine, dan hasil akhir dari perhitungan KNN dan CF. Selain itu juga terdapat fungsi untuk menghapus data uji dan data yang terkait dengan perhitungan. Pada gambar 4.11 akan dijelaskan mengenai halaman CF.



Gambar 4.11 CF

### 4.1.7 Tabel Penyimpanan Data

Table penyimpanan data merupakan gambaran dari tabel penyimpanan data yang digunakan dalam sistem. Tahapan ini digunakan untuk memperjelas data yang digunakan dalam bentuk tabel-tabel yang ada dalam database. Dalam desain database ini terdapat 8 tabel diantaranya data\_uji, data\_training, data\_jarak, data\_count, data\_uji\_cf, cf\_dm1, cf\_dm2, dan cf\_hasil. Data\_uji disini berisi data inputan dari user yang akan dijelaskan pada tabel 4.9.

**Tabel 4.15** Data Uji

nama	G1	G2	G3	G4	G5	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14	G15	k

Tabel 4.10 diberikan untuk menjelaskan mengenai data\_training. Data\_training berisi gejala dan tipe diabetes yang digunakan untuk melakukan perhitungan *k-nearest neighbour*.

**Tabel 4.16** Data Training

No	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14	G15	Tipe_DM

Pada proses pertama perhitungan *k-nearest neighbour* yaitu perhitungan jarak akan disimpan pada data\_jarak yang akan dijelaskan pada tabel 4.11.

**Tabel 4.17** Data Jarak

No	Jarak	Tipe_dm

Setelah menghitung jarak dan melakukan sorting terhadap nilai k inputan dari user. Maka data tersebut akan disimpan pada data\_count yang dijelaskan pada tabel 4.12.

**Tabel 4.18** Data Count

No	Jarak	Tipe_dm

Sedangkan untuk proses proses perhitungan *certainty factor* data inputan user disimpan dalam data\_uji\_cf yang dijelaskan pada tabel 4.13

**Tabel 4.19** Data Uji\_CF

gejala	nilai

Nilai kepastian dan ketidakpastian dari pakar disimpan dalam data\_cf\_dm1 dan data\_cf\_dm2. Kedua data tersebut akan dijelaskan pada tabel 4.14 dan 4.15.

**Tabel 4.20** Data CF DM1

gejala	mb	md

**Tabel 4.21** Data CF DM2

gejala	mb	md

Hasil dari perhitungan CF user dan CF pakar akan disimpan dalam cf\_hasil. Data cf\_hasil akan dijelaskan pada tabel 4.16.

**Tabel 4.22** CF Hasil

gejala	mb	md

#### 4.1.8 Skenario Pengujian

Pada tahap ini merupakan tahapan dimana proses rancangan uji coba dan evaluasi dari sistem. Skenario pengujian digunakan untuk mengetahui gambaran apa yang akan dilakukan dalam pengujian sistem. Pada pengujian sistem terdapat dua macam pengujian yang akan digunakan yaitu pengujian fungsional dan pengujian akurasi. Penjelasan mengenai pengujian fungsional ditunjukkan pada tabel 4.17.

**Tabel 4.23** Skenario Pengujian Fungsional

No	Nama Test	Hasil yang diharapkan	Hasil Pengujian	Nilai
1	Tombol Mulai	- Sistem dapat menampilkan halaman form gejala		
2	Isi Form Gejala	- Sistem dapat melakukan proses mengisi form gejala		
3	Tombol Proses	- Sistem dapat melakukan input dari user kedalam database		
4	Tampil List Data	- Sistem dapat menampilkan list data uji dan data latih		
5	Tombol Hitung KNN	- Sistem dapat melakukan proses hitung KNN		

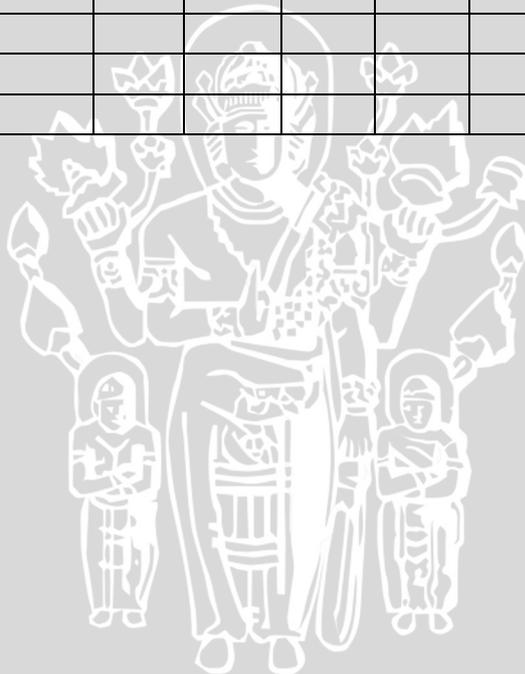
**Tabel 4.24** Skenario Pengujian Fungsional (lanjutan)

6	Tombol Hitung CF	- Sistem dapat melakukan proses hitung CF	
7	Tombol Reset	- Sistem dapat melakukan reset pada database	

Sedangkan untuk pengujian akurasi dilakukan sebanyak 4 kali dengan nilai k yang bervariasi. Penjelasan mengenai pengujian akurasi ditunjukkan pada tabel 4.18.

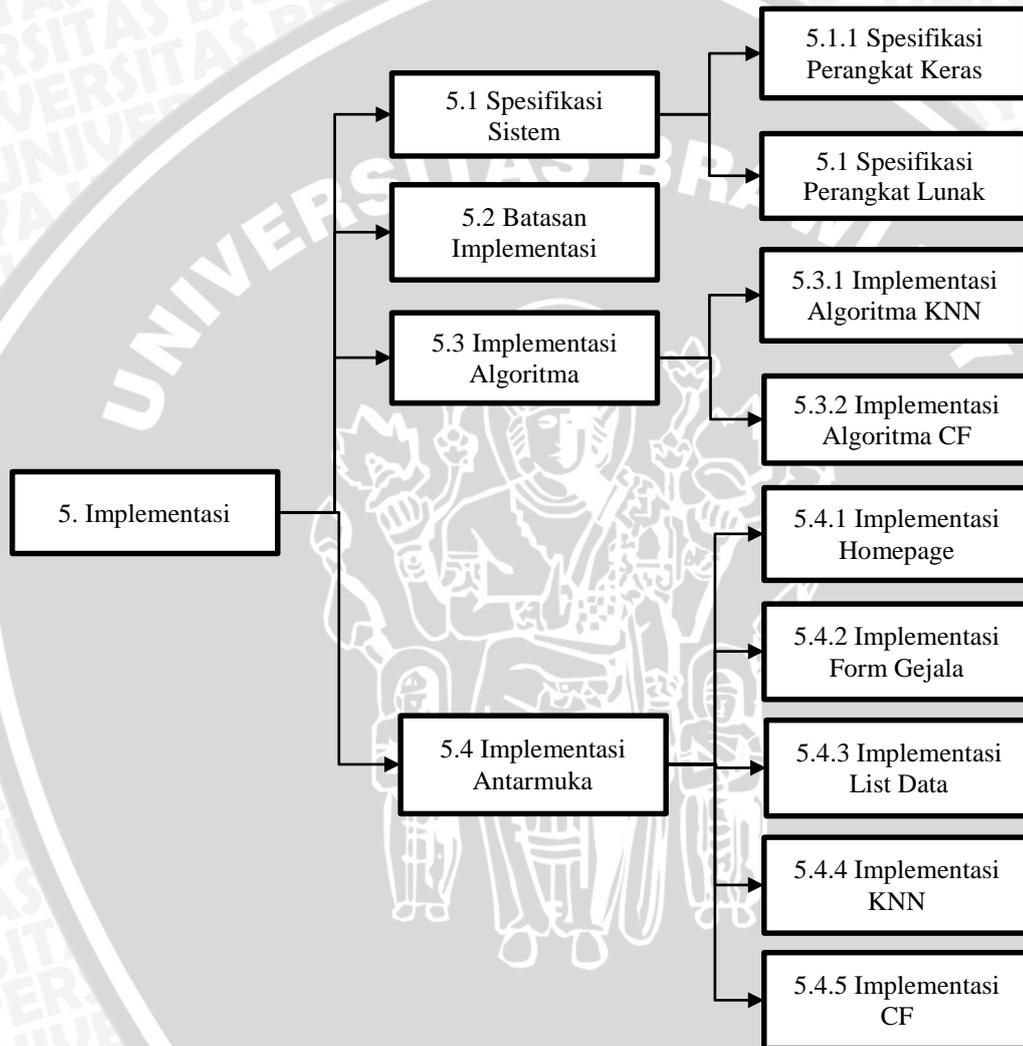
**Tabel 4.25** Skenario Pengujian Akurasi

Uji	Nilai k									
	1	3	5	7	9	11	13	17	23	37
1										
2										
3										
4										



## BAB 5 IMPLEMENTASI

Bab ini membahas tentang implementasi dari pemodelan sistem pakar untuk mendeteksi penyakit *diabetes melitus* yang sebelumnya telah dijelaskan pada bab perancangan. Pada bab ini terdiri dari spesifikasi sistem, batasan implementasi, implementasi algoritma, dan implementasi antarmuka. Alur implementasi dari sistem yang akan dibuat ditunjukkan pada gambar 5.1.



Gambar 5.1 Pohon Implementasi

## 5.1 Spesifikasi Sistem

Pada subbab ini membahas tentang spesifikasi yang harus dipenuhi oleh sistem pada saat implementasi. Spesifikasi sistem terdiri dari spesifikasi perangkat keras dan spesifikasi perangkat lunak.

### 5.1.1 Spesifikasi Perangkat Keras

Dalam mengimplementasikan pemodelan sistem pakar untuk mendeteksi penyakit *diabetes melitus* ini menggunakan spesifikasi perangkat keras yang ditunjukkan pada tabel 5.1.

**Tabel 5.1** Spesifikasi Perangkat Keras

Nama Komponen	Spesifikasi
Prosesor	Intel Core i5
Memori (RAM)	6 Gb
Hardisk	500 Gb
Kartu Grafis	Nvidia Geforce GT 540M
Monitor	14.0'

### 5.1.2 Spesifikasi Perangkat Lunak

Dalam mengimplementasikan pemodelan sistem pakar untuk mendeteksi penyakit *diabetes melitus* ini menggunakan spesifikasi perangkat lunak yang ditunjukkan pada tabel 5.2.

**Tabel 5.2** Spesifikasi Perangkat Lunak

Nama Komponen	Spesifikasi
Sistem Operasi	Microsoft Windows 10
Basis Data	MySQL
Tools Dokumentasi	Microsoft Office 2013
Tools Diagram	Microsoft Visio 2013
Bahasa Pemrograman	PHP
Tools Pemrograman	Sublime
Tools Browser	Google Chrome

## 5.2 Batasan Implementasi

Subbab ini menjelaskan tentang batasan implementasi dari sistem yang akan dibangun. Berikut merupakan batasan implementasi dari pemodelan sistem pakar untuk mendeteksi penyakit *diabetes melitus* yang akan dibangun.

1. Pemodelan Sistem Pakar untuk mendeteksi penyakit *diabetes melitus* dibangun berdasarkan ruang lingkup berbasis web dengan bahasa pemrograman PHP dan sebagai tempat penyimpanan data menggunakan MySQL.

2. Metode yang digunakan dalam Pemodelan Sistem Pakar untuk mendeteksi penyakit *diabetes melitus* ini menggunakan metode *K-Nearest Neighbour-Certainty Factor*.
3. Data yang digunakan dalam Pemodelan Sistem Pakar untuk mendeteksi penyakit *diabetes melitus* adalah data dari puskesmas Poncokusumo Tumpang Kabupaten Malang sebanyak 150 data.
4. Dalam prosesnya, metode *K-Nearest Neighbour* menggunakan masukkan nilai inputan berupa gejala-gejala yang kemudian dilanjutkan menghitung prosentase kemungkinan menggunakan *Certainty Factor* berdasarkan klasifikasi metode sebelumnya.
5. Pengguna dalam sistem ini terdiri dari masyarakat umum maupun dokter.
6. Keluaran dari sistem ini berupa diagnosa penyakit diabetes dan prosentase kemungkinan dari diagnosa tersebut.

### 5.3 Implementasi Algoritma

Pada subbab ini menjelaskan tentang implementasi koding dari Pemodelan Sistem Pakar untuk mendeteksi penyakit *diabetes melitus*. Implementasi algoritma ini terdiri dari implementasi algoritma metode KNN dan implementasi algoritma metode CF.

#### 5.3.1 Implementasi Algoritma Metode KNN

- **Langkah 1** : Menghitung jarak antara data uji dengan data latih

*Source code* hasil dari implementasi algoritma Menghitung jarak antara data uji dengan data latih ditunjukkan pada tabel 5.3.

**Tabel 5.3** Source code Menghitung jarak antara data uji dengan data latih

```

1 $k=$_POST['k'];
2 $sql0=mysql_query("UPDATE data_uji SET k=$k");
3 $sql1=mysql_query("SELECT g1, g2, g3, g4, g5, g6, g7,
4 g8, g9, g10, g11, g12, g13, g14, g15 FROM data_uji");
5 $data1=mysql_fetch_array($sql1);
6 $sql4=mysql_query("select count(no) from
7 data_training");
8 $data4=mysql_fetch_array($sql4);
9 $jumlah=$data4[0];
10 for($j=1;$j<=$jumlah;$j++)
11 {
12 $sql=mysql_query("SELECT g1, g2, g3, g4, g5, g6, g7, g8,
13 g9, g10, g11, g12, g13, g14, g15, tipe_dm
14 from data_training where no = '$j'");
15 $data=mysql_fetch_array($sql);
16 for($i=1;$i<16;$i++)
17 {
18 $g="g$i";
19 $nilai=$data[$g];
20 $nilail=$data1[$g];
21 $selisih=$nilai-$nilail;
22 $pangkat[$i]=pow($selisih, 2);
23 }

```

```

24 for ($k=1;$k<16;$k++)
25 {
26 $g1="g$k";
27 }
28 $kw=sqrt ($pangkat [1]+$pangkat [2]+$pangkat [3]+$pangkat [4]
29 +$pangkat [5]+$pangkat [6]+$pangkat [7]+$pangkat [8]+$pangka
30 t [9]+$pangkat [10]+$pangkat [11]+$pangkat [12]+$pangkat [13]
31 +$pangkat [14]+$pangkat [15]);
32 $kw_koma=round ($kw, 4);
33 $tipe=$data ['tipe_dm'];
34 $sql2=mysql_query ("insert into data_jarak
35 (jarak,tipe_dm) values ('$kw_koma','$tipe')");}

```

Penjelasan Source Code :

- Baris 1-2 menerima nilai k yang diinputkan oleh user kemudian dimasukan dalam database.
- Baris 3-7 melakukan query pada database data uji dan data training.
- Baris 9 melakukan perulangan berdasarkan jumlah data training yang ada.
- Baris 15 melakukan perulangan sejumlah gejala yang ada.
- Baris 17-30 melakukan perhitungan jarak antara data uji dengan data training.
- Baris 33 memasukkan hasil perhitungan jarak kedalam database.
- **Langkah 2** : *Sorting* data jarak menurut nilai terkecil

Source code hasil dari implementasi algoritma *Sorting* data jarak menurut nilai terkecil ditunjukkan pada tabel 5.4.

**Tabel 5.4** Source code *Sorting* data jarak menurut nilai terkecil

```

1 $sql=mysql_query ("select no, jarak, tipe_dm from data_jarak
2 order by jarak asc");
3 while ($data=mysql_fetch_array ($sql))
4 {
5 echo "<tr>
6 <td>$data [no]</td>
7 <td>$data [jarak]</td>
8 <td>$data [tipe_dm]</td>
9 </tr>";
10 }
11 ?>

```

Penjelasan Source Code :

- Baris 1-2 melakukan query pada data jarak sekaligus melakukan *sorting ascending*.
- Baris 3 melakukan perulangan sebanyak data jarak.
- Baris 5-9 menampilkan hasil *sorting* dari data jarak.
- **Langkah 3** : Klasifikasi berdasarkan nilai k inputan

Source code hasil dari implementasi algoritma Klasifikasi berdasarkan nilai k inputan ditunjukkan pada tabel 5.5.

**Tabel 5.5** Source code Klasifikasi berdasarkan nilai k inputan

```

1  $sql4=mysql_query("select k from data_uji");
2  $data4=mysql_fetch_array($sql4);
3  $k=$data4['k'];
4  $sql1=mysql_query("select no, jarak, tipe_dm from
5  data_jarak order by jarak asc");
6  for($i=0;$i<$k;$i++)
7  {
8  $data1=mysql_fetch_array($sql1);
9  $no1=$data1['no'];
10 $jarak1=$data1['jarak'];
11 $tipe_dm1=$data1['tipe_dm'];
12 echo "<tr>
13 <td>$no1</td>
14 <td>$jarak1</td>
15 <td>$tipe_dm1</td>
16 </tr>";
17 $sql2=mysql_query("insert into data_count
18 (no, jarak, tipe_dm) values ('$no1', '$jarak1', '$tipe_dm1')");
19 }
20 $sql=mysql_query("select count(*) from data_count where
21 tipe_dm = 'DM1'");
22 $data=mysql_fetch_array($sql);
23 $dm1=$data[0];
24 $sql1=mysql_query("select count(*) from data_count where
25 tipe_dm = 'DM2'");
26 $data1=mysql_fetch_array($sql1);
27 $dm2=$data1[0];
28 if($dm1>$dm2)
29 {
30 echo "<h3>Hasil Perhitungan KNN data test masuk ke dalam
31 DM1</h3>";
32 }
33 else if($dm2>$dm1)
34 {
35 echo "<h3>Hasil Perhitungan KNN data test masuk ke dalam
36 DM2</h3>";
37 }

```

**Penjelasan Source Code :**

- Baris 1-3 melakukan *query* pada data uji untuk mengambil nilai k.
- Baris 4-5 melakukan *sorting* pada data jarak.
- Baris 6-20 melakukan perulangan sebanyak nilai k dan memasukkan hasil *sorting* ke data count.
- Baris 21-38 klasifikasi berdasarkan nilai k inputan.



### 5.3.2 Implementasi Algoritma Metode CF

- **Langkah 1** : Menghitung nilai CF user

*Source code* hasil dari implementasi algoritma Menghitung nilai CF user ditunjukkan pada tabel 5.6.

**Tabel 5.6** Source code Menghitung nilai CF user

```
1 $sql=mysql_query("SELECT g1, g2, g3, g4, g5, g6, g7, g8,
2 g9, g10, g11, g12, g13, g14, g15 FROM data_uji");
3 $data=mysql_fetch_array($sql);
4 for($i=1;$i<15;$i++)
5 {
6 $g="g$i";
7 if($data[$g]==1)
8 {
9 $nilai=$data[$g];
10 $sql1=mysql_query("insert into data_uji_cf (gejala,nilai)
11 values ('$g','$nilai')");
12 }
13 }
```

Penjelasan *Source Code* :

- Baris 1-3 melakukan *query* pada data uji.
- Baris 4-13 melakukan perulangan sebanyak gejala, melakukan perhitungan cf user, dan memasukkan ke data uji cf.
- **Langkah 2** : Menghitung nilai CF user dan pakar

*Source code* hasil dari implementasi algoritma Menghitung nilai CF user dan pakar ditunjukkan pada tabel 5.7.

**Tabel 5.7** Source code Menghitung nilai CF user dan pakar

```
1 $sql2=mysql_query("select gejala, nilai from
2 data_uji_cf");
3 while($data2=mysql_fetch_array($sql2))
4 {
5 $sql3=mysql_query("select gejala, mb, md from $db");
6 while($data3=mysql_fetch_array($sql3))
7 {
8 if($data2['gejala']==$data3['gejala'])
9 {
10 $gej2=$data2['gejala'];
11 $gej3=$data3['gejala'];
12 $mb=$data3['mb'];
13 $md=$data3['md'];
14 $sql4=mysql_query("insert into cf_hasil (gejala,mb,md)
15 values ('$gej2','$mb','$md')");
16 }
17 }
18 }
```

Penjelasan *Source Code* :

- Baris 1-2 melakukan *query* pada data uji cf.
- Baris 3 melakukan perulangan sebanyak data uji cf.
- Baris 5 melakukan *query* pada data cf pakar.
- Baris 6 melakukan perulangan sebanyak data cf pakar.
- Baris 8-15 melakukan perhitungan cf gabungan antara cf user dan cf pakar yang kemudian dimasukkan dalam data cf hasil.
- **Langkah 3** : Menghitung CF Combine

*Source code* hasil dari implementasi algoritma Menghitung CF Combine ditunjukkan pada tabel 5.8.

**Tabel 5.8** Source code Menghitung CF Combine

1	<code>\$sql7=mysql_query("select mb from cf_hasil");</code>
2	<code>\$hasil = array();</code>
3	<code>while(\$data7=mysql_fetch_array(\$sql7))</code>
4	<code>{</code>
5	<code>\$hasil[] = \$data7[0];</code>
6	<code>}</code>
7	<code>\$itung = array();</code>
8	<code>\$itung[0]=\$hasil[0]+\$hasil[1]*(1-\$hasil[0]);</code>
9	<code>echo "H[0]=".\$hasil[0]."+".\$hasil[1]."x(1-</code>
10	<code>.\$hasil[0].")=".\$itung[0];</code>
11	<code>echo "&lt;/br&gt;";</code>
12	<code>for(\$i=2;\$i&lt;count(\$hasil);\$i++)</code>
13	<code>{</code>
14	<code>    \$a=2;</code>
15	<code>    \$itung[\$i-1]=\$hasil[\$i]+\$itung[\$i-\$a]*(1-</code>
16	<code>    \$hasil[\$i]);</code>
17	<code>    \$b=\$i-1;</code>
18	<code>    echo "H[".\$b."]=".\$hasil[\$i]."+".\$itung[\$i-</code>
19	<code>    \$a]. "x(1-".\$hasil[\$i].")=".\$itung[\$i-1];</code>
20	<code>    echo "&lt;/br&gt;";</code>
21	<code>}</code>
22	<code>echo "&lt;/br&gt;";</code>

Penjelasan *Source Code* :

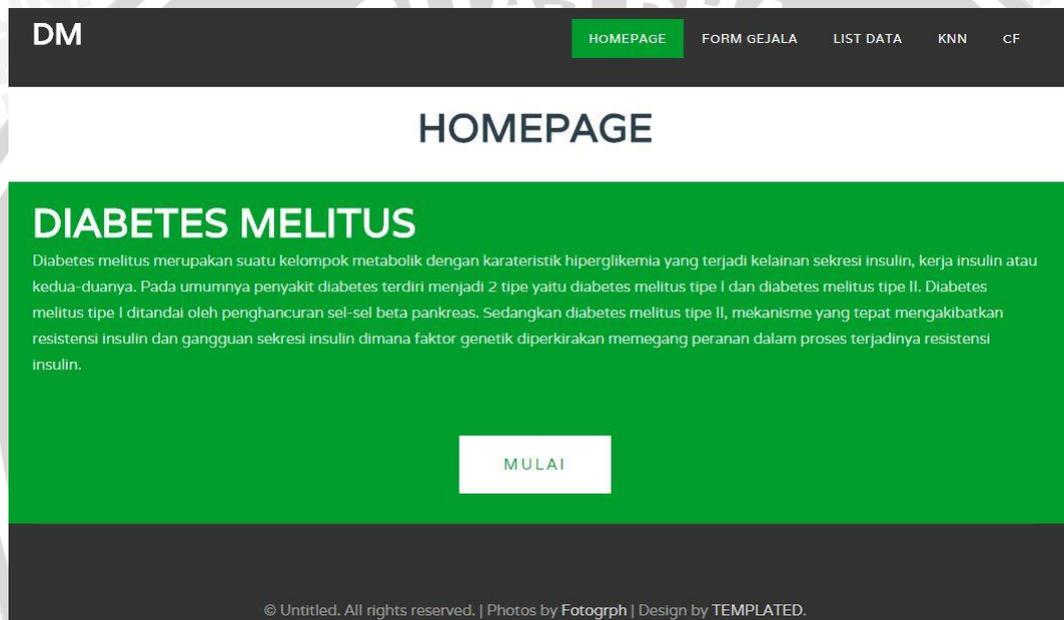
- Baris 1 melakukan *query* pada data cf hasil.
- Baris 2 inialisasi array hasil.
- Baris 3-6 perulangan sebanyak data cf hasil dan memasukkan nilainya kedalam array hasil.
- Baris 7 inialisasi array itung.
- Baris 8-22 melakukan perhitungan cf combine.

## 5.4 Implementasi Antarmuka

Pada subbab ini menggambarkan tentang implementasi antarmuka dari sistem. Implementasi antarmuka ini menjelaskan tentang tampilan antarmuka dari sistem yang dibangun. Tampilan antarmuka terdiri dari homepage, form gejala, list data, knn, dan cf.

### 5.4.1 Implementasi Antarmuka Homepage

Implementasi antarmuka login merupakan tampilan awal dari sistem berupa homepage. Homepage ini berisi deskripsi mengenai penyakit *diabetes melitus* tipe 1 dan tipe 2. Selain itu, user juga dapat memulai melakukan diagnosa dengan menekan tombol mulai. Tampilan antarmuka homepage ditunjukkan pada gambar 5.2.



Gambar 5.2 Antarmuka Homepage

### 5.4.2 Implementasi Antarmuka Form Gejala

Pada implementasi antarmuka form gejala berupa form isian. Form tersebut nantinya akan diisi oleh user sesuai dengan gejala yang sedang dialami. Isi dari form tersebut yaitu nama dan gejala-gejala dari *diabetes melitus*. Selain itu, terdapat tombol proses untuk menuju ke tahap selanjutnya. Tampilan antarmuka form gejala ditunjukkan pada gambar 5.3.

## ISI DATA GEJALA

Isikan sesuai kolom yang tersedia:

Nama

Pilih sesuai dengan gejala yang ada di bawah ini:

Banyak makan (polifagia)	<input type="radio"/> Ya	<input type="radio"/> Tidak
Sering buang air kecil (poliuria)	<input type="radio"/> Ya	<input type="radio"/> Tidak
Banyak minum (polidipsia)	<input type="radio"/> Ya	<input type="radio"/> Tidak
Turunnya berat badan tanpa alasan yang jelas	<input type="radio"/> Ya	<input type="radio"/> Tidak
Umur (15-20)	<input type="radio"/> Ya	<input type="radio"/> Tidak
Faktor keturunan	<input type="radio"/> Ya	<input type="radio"/> Tidak
Mulut kering	<input type="radio"/> Ya	<input type="radio"/> Tidak
Mudah kelelahan	<input type="radio"/> Ya	<input type="radio"/> Tidak
Sering mengantuk	<input type="radio"/> Ya	<input type="radio"/> Tidak
Timbulnya rasa kesemutan (mati rasa) atau sakit pada tangan atau kaki	<input type="radio"/> Ya	<input type="radio"/> Tidak
Timbulnya borok (luka) pada kaki yang tak kunjung sembuh	<input type="radio"/> Ya	<input type="radio"/> Tidak
Gatal-gatal	<input type="radio"/> Ya	<input type="radio"/> Tidak
Disfungsi ereksi pada pria	<input type="radio"/> Ya	<input type="radio"/> Tidak
Obesitas 20% dari berat badan normal	<input type="radio"/> Ya	<input type="radio"/> Tidak
Kadar glukosa darah lebih dari 140 mg/dl (sewaktu)	<input type="radio"/> Ya	<input type="radio"/> Tidak

Gambar 5.3 Antarmuka Form Gejala

### 5.4.3 Implementasi Antarmuka List Data

Implementasi antarmuka list data disini terdiri dari data uji, data training, dan inputan nilai k. Data uji yang ditampilkan adalah hasil dari inputan user pada form gejala. Data training merupakan data yang telah dimasukkan melalui phpmyadmin. Inputan nilai k digunakan untuk menghitung KNN. Tampilan antarmuka list data ditunjukkan pada gambar 5.4 sampai dengan 5.6.

## LIST DATA

### DATA UJI

Nama	Alamat	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14	G15	
remo		0	1	0	1	1	1	0	0	1	0	0	1	1	0	1	delete

Gambar 5.4 Antarmuka List Data Uji

### DATA TRAINING

No	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14	G15	Tipe DM
1	1	1	0	0	1	1	0	1	0	1	0	0	1	1	0	DM1
2	1	0	1	1	1	1	0	0	1	0	0	0	1	1	0	DM1
3	1	1	1	1	1	1	1	0	0	1	0	1	1	1	1	DM1
4	1	1	1	0	1	1	1	1	0	0	0	0	0	0	0	DM1
5	1	1	0	0	1	1	1	1	1	1	1	0	0	1	0	DM1
6	1	0	0	1	1	1	0	1	0	1	0	0	1	1	1	DM1
7	1	0	0	1	1	1	1	1	1	1	1	0	1	1	1	DM1
8	0	1	1	1	1	1	1	0	0	1	1	0	0	0	1	DM1
9	1	0	0	1	1	1	0	1	1	1	0	0	0	0	1	DM1
10	1	1	0	1	1	1	1	1	0	0	0	0	1	1	1	DM1
11	0	0	1	1	1	1	1	0	1	0	1	0	0	1	1	DM1
12	0	1	0	1	1	1	0	1	0	1	1	1	1	1	0	DM1
13	0	1	1	0	1	1	0	1	0	0	1	1	1	1	0	DM1
14	1	0	1	1	1	1	1	0	1	1	1	1	1	1	1	DM1

Keterangan:

G1 : Banyak makan (polifagia)	G9 : Sering mengantuk
G2 : Sering buang air kecil (poliuria)	G10 : Timbulnya rasa kesemutan (mati rasa) atau sakit pada tangan atau kaki
G3 : Banyak minum (polidipsia)	G11 : Timbulnya borok (luka) pada kaki yang tak kunjung sembuh

**Gambar 5.5** Antarmuka List Data Training

G3 : Banyak minum (polidipsia)	G11 : Timbulnya borok (luka) pada kaki yang tak kunjung sembuh
G4 : Turunnya berat badan tanpa alasan yang jelas	G12 : Gatal-gatal
G5 : Umur (15-20)	G13 : Disfungsi ereksi pada pria
G6 : Faktor keturunan	G14 : Obesitas 20% dari berat badan normal
G7 : Mulut kering	G15 : Kadar glukosa darah lebih dari 140 mg/dl (sewaktu)
G8 : Mudah kelelahan	
Tentukan nilai k:	1 ▾
<div style="background-color: white; color: green; padding: 5px 20px; border: 1px solid green; display: inline-block; margin-top: 10px;">HITUNG KNN</div>	

**Gambar 5.6** Antarmuka List Data Inputan Nilai k

#### 5.4.4 Implementasi Antarmuka KNN

Implementasi antarmuka KNN disini terdiri dari data jarak, data setelah sorting, dan data sorting menurut k inputan. Data jarak merupakan hasil perhitungan dari data uji dan data training, sedangkan data setelah sorting adalah data jarak setelah disorting. Data sorting menurut k inputan merupakan hasil akhir dari KNN. Tampilan antarmuka KNN ditunjukkan pada gambar 5.7 dan 5.8.



KNN

DATA JARAK			DATA SETELAH SORTING		
No	Jarak	Tipe DM	No	Jarak	Tipe DM
1	2.8284	DM1	15	2.2361	DM1
2	2.4495	DM1	24	2.2361	DM1
3	2.4495	DM1	38	2.2361	DM2
4	3	DM1	54	2.2361	DM2
5	3.1623	DM1	58	2.2361	DM2
6	2.6458	DM1	62	2.2361	DM2
7	2.8284	DM1	92	2.2361	DM2
8	2.6458	DM1	99	2.2361	DM2
9	2.4495	DM1	2	2.4495	DM1
10	2.4495	DM1	3	2.4495	DM1
11	2.6458	DM1	9	2.4495	DM1
12	2.4495	DM1	10	2.4495	DM1

Gambar 5.7 Antarmuka KNN Data Jarak dan Sorting

DATA SORTING MENURUT K INPUTAN		
No	Jarak	Tipe DM
15	2.2361	DM1
24	2.2361	DM1
38	2.2361	DM2

Hasil Perhitungan KNN data test masuk ke dalam DM1

HITUNG CF

Gambar 5.8 Antarmuka KNN Data Sorting Menurut k Inputan

5.4.5 Implementasi Antarmuka CF

Implementasi antarmuka KNN disini terdiri dari tabel cf dan perhitungan cf. Tabel cf berisi tabel cf user, tabel cf pakar dm1, dan tabel cf pakar dm2. Perhitungan cf berisi tabel cf combine, proses manual cf, dan hasil keseluruhan. Tampilan antarmuka CF ditunjukkan pada gambar 5.9 dan 5.10.



TABEL CF USER		TABEL CF PAKAR DM1		TABEL CF PAKAR DM2	
Gejala	CF_User	Gejala	CF_Pakar	Gejala	CF_Pakar
g2	1	g1	0.8	g1	0.8
g4	1	g2	0.9	g2	0.9
g5	1	g3	0.8	g3	0.8
g6	1	g4	0.1	g4	0.5
g9	1	g5	1	g6	0.5
g12	1	g6	1	g7	0.2
g13	1	g8	0.5	g9	0.4
		g10	0.2	g10	0.2
		g11	0.1	g11	0.5
		g12	0.1	g12	0.2
		g15	0.5	g13	0.2
				g14	0.6
				g15	0.5

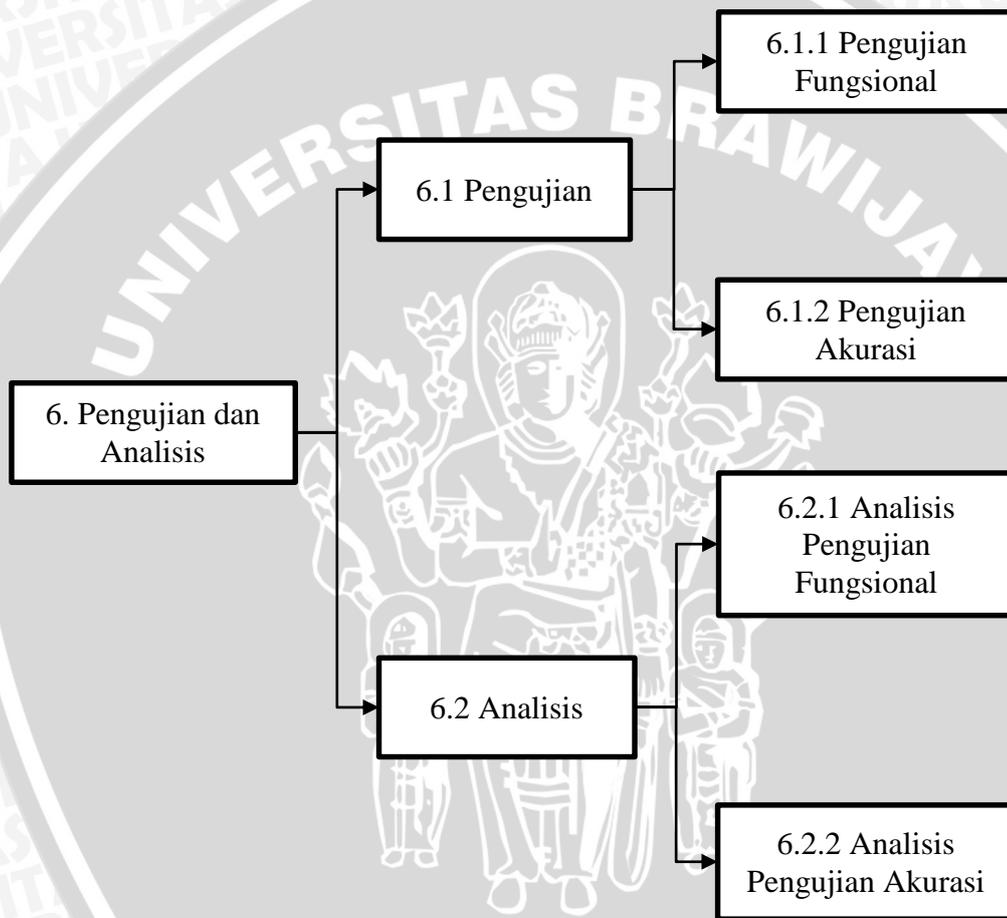
Gambar 5.9 Antarmuka Tabel CF

PERHITUNGAN CF														
TABEL CF COMBINE	PROSES MANUAL CF	HASIL KESELURUHAN												
<table border="1"> <thead> <tr> <th>Gejala</th> <th>CF_Combine</th> </tr> </thead> <tbody> <tr><td>g2</td><td>0.9</td></tr> <tr><td>g4</td><td>0.1</td></tr> <tr><td>g5</td><td>1</td></tr> <tr><td>g6</td><td>1</td></tr> <tr><td>g12</td><td>0.1</td></tr> </tbody> </table>	Gejala	CF_Combine	g2	0.9	g4	0.1	g5	1	g6	1	g12	0.1	$H[0]=0.9+0.1x(1-0.9)=0.91$ $H[1]=1+0.91x(1-1)=1$ $H[2]=1+1x(1-1)=1$ $H[3]=0.1+1x(1-0.1)=1$	<p>Hasil Perhitungan KNN data test masuk ke dalam DM1 &amp; Prosentase Kemungkinan Terkena DM Sebesar = 100%</p>
Gejala	CF_Combine													
g2	0.9													
g4	0.1													
g5	1													
g6	1													
g12	0.1													

Gambar 5.10 Antarmuka Perhitungan CF

## BAB 6 PENGUJIAN DAN ANALISIS

Bab ini membahas tentang pengujian dan analisis dari pemodelan sistem pakar untuk mendeteksi penyakit *diabetes melitus*. Pengujian dalam bab ini meliputi pengujian akurasi nilai k, pengujian akurasi data uji, dan pengujian akurasi data latih. Sedangkan untuk analisis meliputi analisis akurasi nilai k, analisis data uji, dan analisis akurasi data latih. Alur pengujian dari sistem yang akan dibuat ditunjukkan pada gambar 6.1.



Gambar 6.1 Pohon Pengujian dan Analisis

## 6.1 Pengujian

Pada subbab ini membahas tentang pengujian yang dilakukan dalam pemodelan sistem pakar untuk mendeteksi penyakit *diabetes melitus*. Pengujian tersebut meliputi pengujian fungsional, pengujian akurasi nilai k, pengujian akurasi data uji, dan pengujian akurasi data latih.

### 6.1.1 Pengujian Fungsional

Pengujian fungsional merupakan representasi dari pengujian *blackbox* dimana pengujian ini menguji struktur fungsional dari kebutuhan yang telah didefinisikan pada subbab analisis perangkat lunak. Pengujian ini menggunakan kasus uji untuk setiap kebutuhan yang telah didefinisikan di bab analisis kebutuhan sistem. Berikut merupakan kasus uji dari pengujian fungsional.

#### 1. Kasus Uji : Tombol Mulai

Kasus uji mulai bertujuan untuk menguji struktur fungsional dari proses form gejala yang telah didefinisikan pada subbab analisis perangkat lunak. Kasus uji dari tombol mulai ditunjukkan pada tabel 6.1.

**Tabel 6.1** Kasus Tombol Mulai

Nama Kasus Uji	Tombol Mulai
Tujuan Pengujian	Untuk menguji proses tombol mulai pada halaman <i>homepage</i>
Prosedur Uji	<ol style="list-style-type: none"> <li>1. Sistem menampilkan halaman <i>homepage</i></li> <li>2. User menekan tombol mulai</li> </ol>
Hasil yang Diharapkan	- Sistem dapat menampilkan halaman form gejala

#### 2. Kasus Uji : Isi Form Gejala

Kasus uji isi form gejala bertujuan untuk menguji isi form gejala yang didalamnya terdapat beberapa isian gejala-gejala. Kasus uji dari isi form gejala ditunjukkan pada tabel 6.2.

**Tabel 6.2** Kasus Isi Form Gejala

Nama Kasus Uji	Form Gejala
Tujuan Pengujian	Untuk menguji form gejala pada halaman form gejala
Prosedur Uji	<ol style="list-style-type: none"> <li>1. Sistem menampilkan halaman form gejala</li> <li>2. User memasukkan gejala-gejala</li> </ol>
Hasil yang Diharapkan	- Sistem dapat melakukan proses mengisi form gejala

### 3. Kasus Uji : Tombol Proses

Kasus uji tombol proses bertujuan untuk menguji isi tombol proses agar gejala yang dimasukkan dapat diproses. Kasus uji dari isi tombol proses ditunjukkan pada tabel 6.3.

**Tabel 6.3** Kasus Tombol Proses

Nama Kasus Uji	Tombol Proses
Tujuan Pengujian	Untuk menguji tombol proses yang terdapat pada isi gejala
Prosedur Uji	<ol style="list-style-type: none"> <li>1. User mengisi gejala-gejala</li> <li>2. User menekan tombol proses</li> </ol>
Hasil yang Diharapkan	- Sistem dapat melakukan <i>input</i> dari user kedalam database

### 4. Kasus Uji : Tampil List Data

Kasus uji tampil list data bertujuan untuk menguji tampilan list data yang terdiri dari list data uji dan data latih. Kasus uji dari tampil list data ditunjukkan pada tabel 6.4.

**Tabel 6.4** Kasus Tampil List Data

Nama Kasus Uji	Tampil List Data
Tujuan Pengujian	Untuk menguji tampilan list data
Prosedur Uji	<ol style="list-style-type: none"> <li>1. Sistem menampilkan halaman list data</li> </ol>
Hasil yang Diharapkan	- Sistem dapat menampilkan list data uji dan data latih

### 5. Kasus Uji : Tombol Hitung KNN

Kasus uji tombol hitung knn bertujuan untuk menguji tombol hitung KNN agar bisa melakukan perhitungan KNN. Kasus uji dari tombol hitung KNN ditunjukkan pada tabel 6.5.

**Tabel 6.5** Kasus Tombol Hitung KNN

Nama Kasus Uji	Tombol Hitung KNN
Tujuan Pengujian	Untuk menguji proses dari tombol Hitung KNN
Prosedur Uji	<ol style="list-style-type: none"> <li>1. Sistem menampilkan halaman list data</li> <li>2. User menekan tombol hitung KNN</li> </ol>
Hasil yang Diharapkan	- Sistem dapat melakukan proses hitung KNN

### 6. Kasus Uji : Tombol Hitung CF

Kasus uji tombol hitung CF bertujuan untuk menguji tombol hitung CF agar bisa melakukan perhitungan CF. Kasus uji dari tombol hitung CF ditunjukkan pada tabel 6.6.

**Tabel 6.6** Kasus Tombol Hitung CF

Nama Kasus Uji	Tombol Hitung CF
Tujuan Pengujian	Untuk menguji proses dari tombol Hitung CF
Prosedur Uji	1. Sistem menampilkan halaman sorting 2. User menekan tombol hitung CF
Hasil yang Diharapkan	- Sistem dapat melakukan proses hitung CF

**7. Kasus Uji : Tombol Reset**

Kasus uji tombol reset bertujuan untuk menguji tombol hitung reset agar bisa melakukan reset data pada database. Kasus uji dari tombol reset ditunjukkan pada tabel 6.7.

**Tabel 6.7** Kasus Tombol Reset

Nama Kasus Uji	Tombol Reset
Tujuan Pengujian	Untuk menguji proses dari tombol Reset
Prosedur Uji	1. Sistem menampilkan halaman hitung CF 2. User menekan tombol Reset
Hasil yang Diharapkan	- Sistem dapat melakukan reset pada database

**6.1.2 Pengujian Akurasi**

Pengujian akurasi merupakan pengujian yang ditujukan untuk mengetahui tingkat akurasi dari pemodelan sistem pakar untuk mendeteksi penyakit *diabetes melitus* dengan menggunakan metode *k-nearest neighbour-certainty factor*. Pengujian menggunakan 4 variasi data uji masing-masing sebanyak 40 dan 75 untuk data latih. Dalam pengujian ini dilakukan dengan cara membandingkan data hasil identifikasi dari sistem dengan data hasil dari pakar. Penjelasan mengenai pengujian akurasi ditunjukkan pada tabel 6.8.

**Tabel 6.8** Pengujian Akurasi

Uji	Nilai k									
	1	3	5	7	9	11	13	17	23	37
1	100	97	100	100	100	100	92	92	85	75
2	97	95	100	97	97	97	97	90	85	75
3	97	95	100	100	100	100	97	87	82	75
4	100	97	100	97	97	97	92	95	87	75

Pada kolom uji tabel 6.8 terdapat angka 1-4, dimana angka tersebut mewakili satu kali pengujian dengan menggunakan 40 data uji yang diambil secara random dari 75 data uji. Pengujian dilakukan sebanyak 4 kali dengan memberikan nilai k dari 1-37. Dari hasil tersebut didapatkan rata-rata pengujian akurasi yang ditunjukkan pada tabel 6.9.



**Tabel 6.9** Rata-Rata Pengujian Akurasi

Uji	Akurasi (%)
1	94
2	93
3	93
4	94

Dari hasil pengujian akurasi yang telah dilakukan dengan menggunakan 4 macam variasi data, maka diperoleh nilai akurasi terendah sebesar 93% sedangkan nilai akurasi tertinggi sebesar 94%.

## 6.2 Analisis

Pada subbab ini membahas tentang analisis hasil pengujian dalam pemodelan sistem pakar untuk mendeteksi penyakit *diabetes melitus*. Analisis tersebut meliputi analisis pengujian fungsional, analisis akurasi nilai k, analisis akurasi data uji, dan analisis akurasi data latih.

### 6.2.1 Analisis Pengujian Fungsional

Kualitas sistem dilihat dari pengujian validasi pada kasus uji yang telah didefinisikan pada subbab pengujian fungsional. Pengujian validasi ini dilakukan dengan cara menguji prosedur tiap kasus uji kemudian melihat hasilnya. Jika hasil yang diharapkan pada kasus uji dan hasil dari pengujian validasi bernilai valid maka kasus uji tersebut sesuai dengan kebutuhan sistem. Hasil pengujian validasi dapat dilihat pada tabel 6.10.

**Tabel 6.10** Hasil Pengujian Validasi

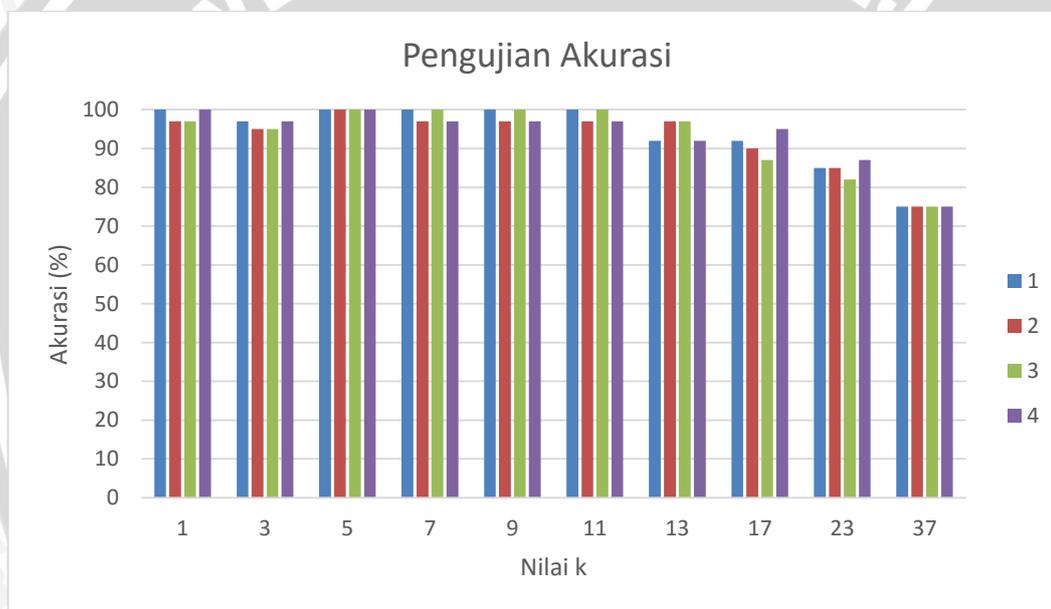
No	Nama Test	Hasil yang diharapkan	Hasil Pengujian	Nilai
1	Tombol Mulai	- Sistem dapat menampilkan halaman form gejala	- Sistem dapat menampilkan halaman form gejala	Valid
2	Isi Form Gejala	- Sistem dapat melakukan proses mengisi form gejala	- Sistem dapat melakukan proses mengisi form gejala	Valid
3	Tombol Proses	- Sistem dapat melakukan input dari user kedalam database	- Sistem dapat melakukan input dari user kedalam database	Valid
4	Tampil List Data	- Sistem dapat menampilkan list data uji dan data latih	- Sistem dapat menampilkan list data uji dan data latih	Valid
5	Tombol Hitung KNN	- Sistem dapat melakukan proses hitung KNN	- Sistem dapat melakukan proses hitung KNN	Valid
6	Tombol Hitung CF	- Sistem dapat melakukan proses hitung CF	- Sistem dapat melakukan proses hitung CF	Valid

**Tabel 6.11** Hasil Pengujian Validasi (lanjutan)

No	Nama Test	Hasil yang diharapkan	Hasil Pengujian	Nilai
7	Tombol Reset	- Sistem dapat melakukan reset pada database	- Sistem dapat melakukan reset pada database	Valid

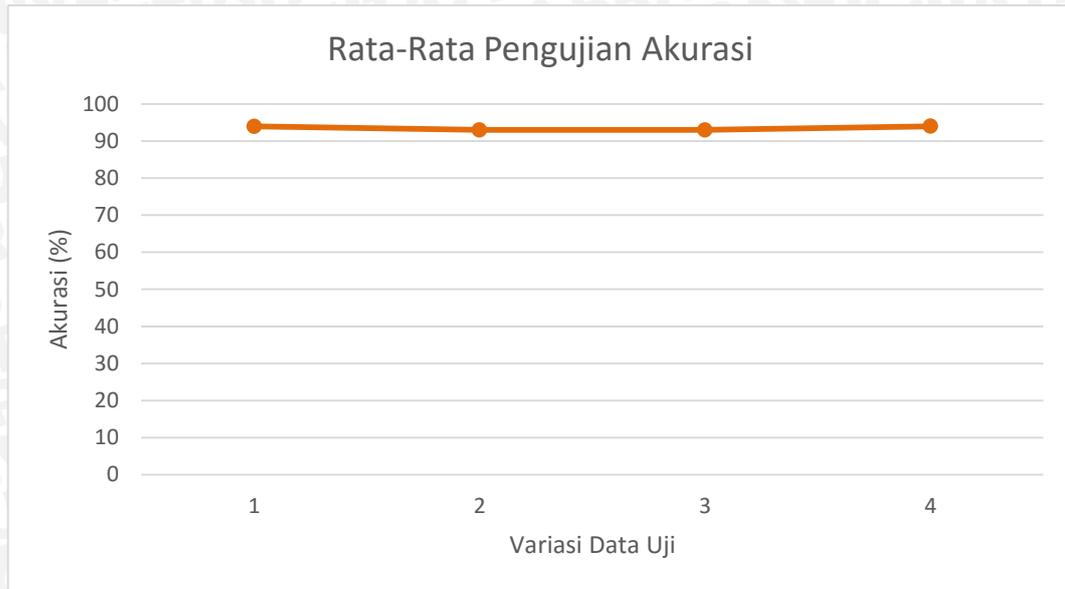
### 6.2.2 Analisis Pengujian Akurasi

Analisis pengujian akurasi merupakan analisis yang ditujukan untuk mengetahui hasil pengujian akurasi dari pemodelan sistem pakar untuk mendeteksi penyakit *diabetes melitus* dengan menggunakan metode *k-nearest neighbour-certainty factor*. Berdasarkan pengujian akurasi pada tabel 6.8, pengujian menggunakan 4 variasi data uji masing-masing sebanyak 40 dan 75 untuk data latih. Grafik analisis pengujian akurasi nilai k ditunjukkan pada gambar 6.2.



**Gambar 6.2** Grafik Pengujian Akurasi

Grafik pengujian akurasi terdiri dari 4 data uji yaitu data uji pertama dengan warna biru, data uji kedua dengan warna merah, data uji ketiga dengan warna hijau, dan data uji keempat dengan warna ungu. Dilihat dari grafik tersebut, apabila nilai k semakin besar maka pada setiap pengujian memiliki tingkat akurasi yang semakin kecil. Analisis hasil rata-rata pengujian akurasi dari yang telah dilakukan dengan nilai k yang berbeda-beda ditunjukkan pada gambar 6.3.



**Gambar 6.3** Grafik Rata-Rata Pengujian

Pada gambar 6.3 dapat dilihat bahwa pada variasi 1 dan 4 memperoleh nilai tertinggi sebesar 94% sedangkan pada variasi 2 dan 3 mengalami penurunan hingga menghasilkan nilai sebesar 93%. Dari keempat hasil analisis tersebut didapatkan nilai rata-rata keseluruhan sebesar 93,5%. Hal ini dapat diartikan bahwa semakin banyak nilai  $k$  yang diberikan maka efek *noise* pada klasifikasi menjadi semakin banyak dan membuat batasan antara setiap klasifikasi menjadi lebih kabur.

## BAB 7 PENUTUP

Bab ini membahas tentang kesimpulan dan saran dari penelitian yang telah dilakukan berdasarkan hasil perancangan, implementasi dan pengujian sistem.

### 7.1 Kesimpulan

Berdasarkan pada hasil perancangan, implementasi, dan pengujian yang telah dilakukan maka kesimpulan yang didapat adalah sebagai berikut :

1. Telah dimodelkan aplikasi sistem pakar untuk mendeteksi penyakit *diabetes melitus* dengan menggunakan metode *k-nearest neighbour-certainty factor* dan dapat digunakan untuk membantu dalam mendiagnosa tipe penyakit *diabetes melitus* tipe 1 atau tipe 2. Masukan dari sistem ini menggunakan 15 gejala dari *diabetes melitus* dan 2 kelas tipe *diabetes melitus* sebagai keluaran. Sistem ini dibangun menggunakan bahasa pemrograman PHP dan MySQL sebagai penyimpanan data. Fitur dari sistem ini yaitu isi gejala, lihat data, hitung knn, dan hitung cf sekaligus hasil akhir berupa diagnosa dan prosentase kemungkinan.
2. Pemodelan sistem pakar untuk mendeteksi penyakit *diabetes melitus* dengan menggunakan metode *k-nearest neighbour – certainty factor* ini telah diuji dengan 2 macam pengujian yaitu:
  - a. Pengujian fungsionalitas yang dilakukan untuk menguji struktur fungsional terhadap kasus uji didapatkan nilai kesuksesan sebesar 100%.
  - b. Pengujian akurasi yang menggunakan 4 macam variasi data uji. Setiap variasi data terdiri dari 75 data latih dan 40 data uji yang berbeda-beda. Berdasarkan pengujian dari variasi data tersebut dihasilkan nilai rata-rata akurasi sebesar 93,5%. Hasil tersebut tidak mencapai nilai maksimal karena dari perhitungan sistem, data uji DM1 ada beberapa data yang hasilnya masuk dalam kelas DM2. Hal tersebut diakibatkan oleh jumlah kelas pada data yang tidak sama yaitu antara kelas DM1 dengan DM2.

### 7.2 Saran

Saran yang diberikan untuk pengembangan sistem dalam penelitian selanjutnya yaitu pada penggunaan data sebaiknya memiliki jumlah kelas yang sama antara kelas satu dengan lainnya, hal ini dapat mempengaruhi hasil dari klasifikasi *k-nearest neighbour* terhadap inputan nilai *k* yang diberikan.

## DAFTAR PUSTAKA

- Arif, Son Wicaksana. 2009. Sistem Pakar Identifikasi Hama Dan Penyakit Tanaman Apel Berbasis Web. Malang, Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang.
- Ayuliana. 2009. Testing dan Implementasi.
- Giarratano, J.C & Riley G. 1994. Expert Systems: Principles and Programming, 2nd edition. PWS Publishing Co, USA.
- Hidayat, Syaeful. 2010. Aplikasi Untuk Mendeteksi Penyakit Pada Tanaman Tebu Dan Cara Penanganannya Berbasis Web. Bandung, Universitas Komputer Indonesia.
- Hidayati, Nur. 2012. Sistem Pakar Berbasis Web Untuk Identifikasi Jenis Hama Dan Penyakit Pada Budidaya Tanaman Jamur Menggunakan Metode Certainty Factor. Malang, Universitas Brawijaya.
- J. Nilsson, Nill. 2009. Introduction To Machine Learning. Stanford University, Stanford, CA 94305.
- Krisnawati, Suci. 2014. Sistem Pakar Pencegahan Dini Kanker Serviks Berdasarkan Faktor Resiko Menggunakan Metode Certainty Factor Berbasis Mobile Web. Jurusan Informatika, Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya. Malang.
- Kusumadewi, Sri. 2003. Artificial Intelligence (Teknik dan Aplikasinya). Graha Ilmu. Yogyakarta.
- Marlissa, Julius. 2013. Pemodelan dan Simulasi Sistem. Pasca Sarjana Universitas Mercu Buana.
- Meristika, Yanita Selly. 2013. Perbandingan K-Nearest Neighbor dan Fuzzy K-Nearest Neighbor pada Diagnosis Penyakit Diabetes Melitus. Program Studi Teknik Informatika, Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya. Malang.
- Moradian, M. dan Baarani, A. 2009. KNNBA: K-Nearest Neighbour Based Asociation Algorithm.
- Muhammad. 2005. Konsep Sistem Pakar. Andi Offset. Yogyakarta.
- Pressman, Roger S. 2010. Software Engineering: Practitioner's Approach, Seventh Edition, McGraw-Hill, New York.
- Puspasari, Maslikha. 2013. Prediksi Tingkat Resiko Penyakit Jantung Koroner (PJK) Menggunakan Metode Fuzzy K-Nearest Neighbour (FK-NN). Skripsi. Malang.
- Smeltzer, Suzanne C. 2001. Buku Ajar Keperawatan Medikal Bedah (Edisi 8). EGC. Jakarta.

Soejoeti, Sunanti Z. 2008. Konsep Sehat, Sakit dan Penyakit dalam Konteks Sosial Budaya. Pusat Penelitian Ekologi Kesehatan, Badan Penelitian dan Pengembangan Kesehatan, Departemen Kesehatan RI. Jakarta.

Sutojo T, Edy Mulyanto, Vincent Suhartono. 2011. Kecerdasan Buatan. Yogyakarta. CV Andi Offset.

Tijar, Ainut. 2011. Diabetes Melitus. Tersedia di: <<http://ainuttijar.blogspot.com/2011/01/diabetes-melitus.html>> [Diakses 24 Maret 2015]



### LAMPIRAN

#### Data Diabetes Melitus

No	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14	G15	DM1	DM2
1	-	V	-	V	V	V	-	-	V	-	-	V	V	-	V	V	
2	-	V	-	V	V	V	-	-	-	V	V	V	-	V	-	V	
3	V	V	-	-	V	V	V	-	V	V	V	V	-	V	V	V	
4	-	V	V	-	V	V	-	V	-	V	-	V	-	V	V	V	
5	-	V	V	V	V	V	-	V	-	V	-	-	V	-	-	V	
6	V	V	-	V	V	V	-	-	V	-	V	V	-	-	V	V	
7	-	-	V	V	V	V	-	-	-	-	-	-	-	V	V	V	
8	V	-	V	-	V	V	V	-	V	-	-	-	-	-	V	V	
9	V	V	V	-	V	V	-	V	V	V	V	-	V	-	V	V	
10	-	V	V	-	V	V	-	-	-	V	V	-	V	V	V	V	
11	V	V	-	-	V	V	-	V	-	V	-	-	V	V	-	V	
12	V	-	V	V	V	V	-	-	V	-	-	-	V	V	-	V	
13	V	V	V	V	V	V	V	-	-	V	-	V	V	V	V	V	
14	V	V	V	-	V	V	V	V	-	-	-	-	-	-	-	V	
15	V	V	-	-	V	V	V	V	V	V	V	-	-	V	-	V	
16	V	-	-	V	V	V	-	V	-	V	-	-	V	V	V	V	
17	V	-	-	V	V	V	V	V	V	V	V	-	V	V	V	V	
18	-	V	V	V	V	V	V	-	-	V	V	-	-	-	V	V	
19	V	-	-	V	V	V	-	V	V	V	-	-	-	-	V	V	

20	V	V	-	V	V	V	V	V	-	-	-	-	V	V	V	V	
21	-	-	V	V	V	V	V	-	V	-	V	-	-	V	V	V	
22	-	V	-	V	V	V	-	V	-	V	V	V	V	V	-	V	
23	-	V	V	-	V	V	-	V	-	-	V	V	V	V	-	V	
24	V	-	V	V	V	V	V	-	V	V	V	V	V	V	V	V	
25	V	V	-	-	V	V	-	-	V	V	-	-	-	-	V	V	
26	-	-	V	V	V	V	-	-	-	-	V	-	-	V	V	V	
27	V	-	V	V	V	V	-	V	-	V	-	-	V	-	V	V	
28	V	V	V	V	V	V	-	-	-	V	-	-	V	-	-	V	
29	V	V	-	-	V	V	-	V	-	-	V	V	-	-	-	V	
30	-	V	V	V	V	V	-	V	-	V	V	-	V	-	-	V	
31	V	V	V	-	V	V	-	V	V	V	-	-	V	-	V	V	
32	V	V	V	-	V	V	V	V	-	-	-	V	-	-	-	V	
33	V	-	V	V	V	V	-	-	V	-	V	-	V	V	-	V	
34	V	V	V	V	V	V	V	-	-	V	-	V	V	-	V	V	
35	-	V	V	V	V	V	V	-	-	V	V	-	V	-	V	V	
36	V	V	-	V	V	V	V	V	-	-	-	-	-	V	V	V	
37	V	-	V	V	V	V	V	-	V	V	-	V	-	V	V	V	
38	V	-	V	V	V	V	-	V	-	-	-	-	V	-	V	V	
39	V	V	V	V	V	V	-	V	-	V	-	-	V	-	-	V	
40	V	V	-	-	V	V	V	V	-	V	V	-	-	V	-	V	
41	V	V	V	-	-	-	V	-	-	V	-	V	-	V	V		V
42	V	V	-	V	-	-	V	V	-	-	V	-	-	-	V		V

43	-	V	-	V	-	-	V	V	V	-	-	V	V	V	V	V	V
44	-	V	V	-	-	-	V	-	V	-	V	V	V	V	V	V	V
45	V	V	V	-	-	-	-	V	-	V	-	-	V	V	V	V	V
46	V	-	V	V	-	-	-	-	-	V	V	-	-	V	V	V	V
47	V	V	V	V	-	-	V	-	V	-	-	-	V	V	V	V	V
48	V	V	V	V	-	-	-	-	V	-	-	-	-	-	V	V	V
49	V	V	V	-	-	-	-	-	-	V	-	-	V	V	V	V	V
50	V	V	-	-	-	-	V	-	-	V	V	-	V	V	V	V	V
51	-	V	V	V	-	-	-	-	V	V	-	-	V	V	V	V	V
52	V	V	V	-	-	-	V	V	V	-	V	V	V	V	-	V	V
53	-	V	V	V	-	-	-	-	-	-	-	V	-	V	V	V	V
54	V	V	V	-	-	-	-	-	V	V	-	V	V	V	V	V	V
55	-	V	V	-	-	-	-	-	V	V	V	-	-	-	V	V	V
56	V	V	V	-	-	-	-	-	V	-	V	V	V	V	V	V	V
57	V	V	V	-	-	-	-	V	-	V	-	-	-	V	V	V	V
58	V	-	V	V	-	-	V	V	V	V	V	V	V	V	V	V	V
59	V	-	V	V	-	-	V	V	V	V	V	-	-	-	V	V	V
60	V	-	V	-	-	-	V	-	-	V	-	V	-	-	V	V	V
61	-	V	V	V	-	-	-	V	V	V	V	-	V	-	V	V	V
62	-	V	V	V	-	-	V	V	-	V	V	V	V	-	V	V	V
63	V	-	V	-	-	-	-	-	V	-	V	V	-	V	V	V	V
64	V	V	-	V	-	-	-	-	-	-	-	-	V	V	V	V	V
65	V	V	-	-	-	-	V	V	V	-	V	V	-	-	V	V	V

66	V	V	-	V	-	-	V	-	V	V	V	-	V	V	V	V	V
67	V	-	V	-	-	-	V	V	V	V	-	V	-	V	V	V	V
68	V	V	-	V	-	-	V	V	V	V	V	-	-	-	V	V	V
69	V	V	V	V	-	-	V	V	V	V	-	V	-	V	V	V	V
70	V	V	V	-	-	-	V	-	-	-	V	-	-	-	V	V	V
71	V	V	-	V	-	-	-	-	-	-	-	-	-	-	V	V	V
72	V	V	-	V	-	-	-	V	-	-	-	-	-	-	V	V	V
73	-	V	V	-	-	-	-	V	-	-	V	-	-	-	V	V	V
74	V	-	V	V	-	-	V	V	V	V	V	-	V	-	V	V	V
75	-	-	V	V	-	-	V	-	-	-	-	-	-	V	V	V	V
76	V	-	V	V	-	-	-	V	-	-	V	-	-	-	V	V	V
77	-	V	V	V	-	-	V	V	V	V	-	-	V	-	V	V	V
78	-	V	-	V	-	-	V	V	V	-	-	V	V	V	V	V	V
79	V	-	V	V	-	-	-	V	-	-	-	-	-	-	V	V	V
80	V	-	V	V	-	-	-	-	-	-	V	-	V	-	V	V	V
81	V	-	V	V	-	-	V	V	V	V	-	-	-	-	V	V	V
82	V	-	V	V	-	-	-	V	V	V	-	-	V	-	V	V	V
83	-	V	V	V	-	-	V	-	-	-	-	V	-	-	-	-	V
84	-	-	V	V	-	-	V	-	-	V	V	V	-	-	-	-	V
85	V	-	V	-	-	-	-	-	-	V	V	V	-	V	-	-	V
86	V	V	-	V	-	-	-	-	-	V	-	V	V	V	V	-	V
87	-	V	-	V	-	-	V	-	V	V	V	-	V	-	-	-	V
88	V	V	V	-	-	-	-	V	-	-	V	-	-	-	-	-	V

89	V	-	V	V	-	-	-	-	-	-	-	-	-	-	V	V
90	V	-	-	V	-	-	-	-	-	V	-	V	V	-	-	V
91	V	V	-	V	-	-	V	-	-	V	V	V	-	V	V	V
92	V	V	V	-	-	-	V	V	-	V	-	V	-	V	V	V
93	V	V	-	V	-	-	V	V	-	-	-	V	V	V	V	V
94	-	V	V	V	-	-	V	-	V	-	-	V	V	V	V	V
95	V	-	V	-	-	-	-	V	-	V	-	-	V	-	V	V
96	V	V	V	-	-	-	-	V	-	V	V	-	-	V	V	V
97	V	V	V	V	-	-	V	-	V	-	-	-	-	V	V	V
98	V	V	V	V	-	-	-	-	V	-	-	V	-	-	V	V
99	V	V	-	V	-	-	V	-	-	V	-	-	V	V	V	V
100	V	V	V	-	-	-	V	-	-	V	V	-	V	V	V	V
101	V	V	V	-	-	-	-	-	V	V	-	V	V	V	V	V
102	-	V	V	V	-	-	V	-	V	-	V	V	V	-	V	V
103	V	V	V	-	-	-	-	V	-	-	-	V	-	V	V	V
104	V	-	V	V	-	-	-	V	V	-	V	V	-	V	V	V
105	-	V	V	-	-	-	-	-	V	V	V	-	V	-	V	V
106	-	V	V	V	-	-	-	-	V	-	V	V	-	V	V	V
107	V	-	V	V	-	-	-	V	-	V	-	-	V	V	V	V
108	-	V	V	V	-	-	V	V	V	V	-	V	V	V	V	V
109	V	V	-	V	-	-	V	V	-	V	V	-	-	-	V	V
110	V	-	V	V	-	-	V	-	-	V	-	V	-	V	V	V
111	V	-	V	V	-	-	-	V	V	V	-	-	V	-	V	V

112	V	V	-	V	-	-	V	V	-	-	V	V	V	-	V		V
113	V	-	V	V	-	-	-	-	V	-	V	V	-	V	V		V
114	V	V	-	V	-	-	-	-	-	-	-	-	V	V	V		V
115	V	V	V	-	-	-	-	V	V	-	V	-	-	-	V		V
116	V	-	V	V	-	-	V	-	V	-	V	-	V	V	V		V
117	V	-	V	-	-	-	V	-	V	V	-	V	-	V	V		V
118	V	V	V	-	-	-	V	V	-	V	V	-	-	-	V		V
119	V	V	V	V	-	-	V	V	-	V	-	V	-	V	V		V
120	V	-	V	V	-	-	V	-	-	-	V	-	V	-	V		V
121	-	V	V	V	-	-	-	V	-	V	-	-	-	-	V		V
122	V	V	V	-	-	-	-	V	-	-	-	-	V	-	V		V
123	V	-	V	-	-	-	-	V	-	-	V	-	-	-	V		V
124	-	V	V	V	-	-	V	V	-	V	V	-	V	-	V		V
125	V	-	V	V	-	-	V	-	-	-	V	-	-	V	V		V
126	V	V	V	-	-	-	-	V	-	-	V	-	V	-	V		V
127	V	-	V	V	-	-	V	V	-	V	-	-	V	-	V		V
128	-	V	V	-	-	-	V	V	V	-	-	V	-	V	V		V
129	V	V	-	V	-	-	-	V	-	-	V	-	-	-	V		V
130	V	V	V	-	-	-	-	-	V	-	V	-	V	-	V		V
131	V	-	V	V	-	-	V	-	V	V	-	-	-	-	V		V
132	V	V	-	V	-	-	-	V	V	-	-	-	V	-	V		V
133	V	V	V	-	-	-	V	-	-	V	-	V	-	-	-		V
134	-	V	V	V	-	-	V	-	-	V	-	V	-	-	-		V

135	V	-	V	V	-	-	-	-	-	-	V	V	-	V	-	V
136	V	V	V	-	-	-	-	-	-	V	-	V	-	V	-	V
137	-	V	V	-	-	-	V	-	V	-	V	-	V	-	-	V
138	V	-	V	V	-	-	-	V	-	-	V	-	V	-	-	V
139	V	V	-	V	-	-	-	-	V	-	-	-	-	-	V	V
140	-	V	V	V	-	-	-	V	-	-	V	-	V	V	-	V
141	V	-	V	V	-	-	V	-	-	V	-	V	-	V	V	V
142	-	V	V	V	-	-	-	V	-	V	-	V	V	-	V	V
143	V	V	-	V	-	-	-	V	-	V	-	-	-	V	V	V
144	V	-	V	V	-	-	V	-	V	-	-	-	-	V	V	V
145	V	V	V	-	-	-	-	-	V	V	-	V	-	-	V	V
146	-	V	V	V	-	-	-	V	V	-	V	V	-	V	V	V
147	V	V	V	-	-	-	-	-	V	V	-	-	V	-	V	V
148	V	-	V	V	-	-	-	-	V	-	-	V	-	V	V	V
149	-	V	V	V	-	-	-	V	-	V	-	-	V	-	V	V
150	V	V	-	V	-	-	V	V	V	V	-	V	-	V	V	V

Keterangan

G1 = banyak makan (polifagia)

G2 = sering buang air kecil (poliuria)

G3 = banyak minum (polodipsia)

G4 = turunnya berat badan tanpa alasan yang jelas

G5 = umur (15-20)

G6 = faktor keturunan

G7 = mulut kering

G8 = mudah kelelahan

G9 = sering mengantuk

G10 = timbulnya rasa kesemutan (mati rasa) atau sakit pada tangan atau kaki

G11 = timbulnya rasa borok (luka) pada kaki yang tak kunjung sembuh

G12 = gatal-gatal

G13 = disfungsi ereksi pada pria

G14 = obesitas 20% dari berat badan normal

G15 = kadar glukosa darah lebih dari 140 mg/dl (sewaktu)



## Nilai Bobot Pakar

### Diabetes Tipe 1

Gejala	MB	MD
Faktor keturunan	1	0
Umur kurang atau sama dengan (15-20) tahun	1	0
Mudah kelelahan	0,5	0,2
Banyak minum (polodipsia)	0,8	0,1
Turunnya berat badan tanpa alasan yang jelas	0,1	0,4
Timbulnya rasa kesemutan (mati rasa) atau sakit pada tangan atau kaki	0,2	0,7
Timbulnya rasa borok (luka) pada kaki yang tak kunjung sembuh	0,1	0,7
Gatal-gatal	0,1	0,5
Kadar glukosa darah lebih dari 140 mg/dl (sewaktu)	0,5	0,5

### Diabetes Tipe 2

Gejala	MB	MD
Banyak makan (polifagia)	0,8	0,4
Sering buang air kecil (poliuria)	0,9	0,4
Banyak minum (polodipsia)	0,8	0,1
Turunnya berat badan tanpa alasan yang jelas	0,5	0,5
Faktor keturunan	0,5	0,5
Mulut kering	0,2	0
Sering mengantuk	0,4	0,2
Timbulnya rasa kesemutan (mati rasa) atau sakit pada tangan atau kaki	0,2	0,5
Timbulnya rasa borok (luka) pada kaki yang tak kunjung sembuh	0,5	0,3
Gatal-gatal	0,2	0,3
Disfungsi ereksi pada pria	0,2	0,7
Obesitas 20% dari berat badan normal	0,6	0,2
Berumur 20 tahun atau lebih	0,2	0,5
Kadar glukosa darah lebih dari 140 mg/dl (sewaktu)	0,5	0,5