

**SISTEM PAKAR DIAGNOSIS PENYAKIT DEMAM: DBD,
MALARIA DAN TIFOID MENGGUNAKAN METODE K-
NEAREST NEIGHBOR – CERTAINTY FACTOR**

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:
Elsa Nur Amilus Shofia
NIM: 125150202111006



**PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2017**

PENGESAHAN

SISTEM PAKAR DIAGNOSIS PENYAKIT DEMAM: DBD, MALARIA DAN TIFOID

MENGGUNAKAN METODE *K-NEAREST NEIGHBOR – CERTAINTY FACTOR*

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
Elsa Nur Amilus Shofia

NIM:
125150202111006

Skripsi ini telah diuji dan dinyatakan lulus pada
30 Maret 2017

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II

Rekyan Regasari Mardi Putri, S.T, M.T
NIK: 2011027704142001

Achmad Arwan, S.Kom, M.Kom
NIK: 198408152008121005

Mengetahui
Ketua Jurusan Teknik Informatika

Tri Astoto Kurniawan, S.T, M.T, Ph.D
NIP: 19710518 200312 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, 30 Maret 2017



Elsa Nur Amilus Shofia

NIM: 125150202111006

KATA PENGANTAR

Puji dan syukur penulis panjatkan kehadirat Allah SWT atas anugerah serta kasih sayang-Nya sehingga penulis dapat menyelesaikan skripsi dengan judul: **“Sistem Pakar Diagnosis Penyakit Demam: DBD, Malaria dan Tifoid Menggunakan Metode K-Nearest Neighbor - Certainty Factor”** ini dengan baik. Skripsi ini disusun sebagai syarat memperoleh gelar sarjana pada Program Studi Informatika / Ilmu Komputer, Fakultas Ilmu Komputer Universitas Brawijaya.

Dalam proses penyelesaian skripsi ini penulis mendapatkan banyak bantuan, baik bantuan moral maupun materiil dari berbagai pihak. Oleh karena itu, penulis mengucapkan banyak terimakasih kepada:

1. Ibu Rekyan Regasari Mardi Putri, S.T, M.T selaku Pembimbing I dan Bapak Achmad Arwan, S.Kom, M.Kom selaku Pembimbing II yang telah memberikan bimbingan, arahan, ilmu, dan masukan dalam penyelesaian skripsi ini.
2. Eko Sakti Pramukantoro, S.Kom, M.Kom selaku Dosen Pembimbing Akademik yang telah memberi dukungan moril maupun masukan selama masa perkuliahan.
3. Agus Wahyu Widodo, S.T, M.Cs selaku Ketua Program Studi Informatika/Illmu Komputer beserta jajarannya.
4. Bapak Tri Astoto Kurniawan, S.T, M.T, Ph.D, selaku Ketua Jurusan Teknik Informatika beserta jajarannya.
5. Seluruh Civitas akademik Informatika/Illmu Komputer Universitas Brawijaya yang telah membantu dan mendukung penulis selama menempuh pendidikan di kampus dan selama penyelesaian skripsi ini.
6. Dr. Anda Citra Utama, SpPD selaku pakar penyakit dalam yang memberikan pengetahuannya dan membantu penulis mengumpulkan data terkait.
7. Alm. Ayahanda tercinta Achmad Irfan Santoso yang selalu mengingatkan penulis untuk tidak lupa beribadah kepada Tuhan YME, selalu memotivasi untuk tidak mudah putus asa dan pantang menyerah karena menjadi sukses perlulah sebuah usaha didalamnya.
8. Ibunda tercinta Elok Suryaningsih yang selama ini berdoa, berjuang dan memberikan semangat serta nasehat yang semata-mata untuk keberhasilan penulis.
9. Adik-adik tersayang Elfananda Istiqlalia, Elmadio Muhammad Dhiaulhaq, Elnafriel Nuris Shoba dan keluarga besar yang telah mendukung penulis dengan segala usahanya, mulai dari doa, materi, dukungan moral, semangat hidup dan tauladan yang semata-mata untuk keberhasilan penulis.
10. Andhica Pratama, Andro Wibowo, Lintang Resita Mayangsari, Nurul Wahidatul Fitriyah, Wulan Sari Simarmata dan Yunika Tria Melati yang telah membantu memberikan semangat persahabatan, dan bantuan-bantuan yang tak akan pernah terlupakan sejak semester 1 hingga sekarang.

11. Andrea Widya Ismunarto Putri, Pamela Agri Cintya dan Wiwied Indiana Loren yang senantiasa mendukung, membantu dan memberi doa serta motivasi pada penulis.
12. Teman-teman Program Studi Informatika/Illmu Komputer yang selalu memberikan semangat, dukungan, dan kebersamaan selama Penulis menempuh pendidikan di Fakultas Ilmu Komputer Universitas Brawijaya.
13. Semua pihak yang tidak dapat disebutkan satu per satu, yang telah membantu dan terlibat baik secara langsung maupun tidak langsung dalam penulisan skripsi ini.

Semoga Allah SWT memberikan balasan yang berlipat ganda kepada semua pihak yang telah membantu dan memberikan dukungan kepada penulis dalam proses penyelesaian skripsi ini. Akhir kata penulis berharap laporan skripsi ini dapat bermanfaat bagi semua pihak yang menggunakan.

Malang, 30 Maret 2017

Elsa Nuramilus Shofia
echanuramilus@gmail.com



ABSTRAK

Elsa Nur Amilus Shofia. 2017. Sistem Pakar Diagnosis Penyakit Demam: DBD, Malaria dan Tifoid Menggunakan Metode *K-Nearest Neighbor - Certainty Factor*. Skripsi Program Studi Informatika/Illmu Komputer, Fakultas Ilmu Komputer, Universitas Brawijaya.

Pembimbing: Rekyan Regasari Mardi Putri, S.T, M.T dan Achmad Arwan, S.Kom, M.Kom

Demam merupakan salah satu jenis gangguan kesehatan yang mengganggu produktivitas setiap orang bahkan dapat menyebabkan kematian dan masih menjadi permasalahan kesehatan Indonesia. Terdapat beberapa jenis demam yang perlu diwaspadai antara lain Demam. Ketiga penyakit ini memiliki gejala yang mirip, sehingga banyak tenaga medis dan dokter internship yang seringkali melakukan kesalahan dalam mendiagnosis. Untuk itu diperlukan sebuah sistem yang mengadopsi pengetahuan pakar untuk mengatasi masalah tersebut yaitu sistem pakar. Metode yang digunakan untuk mendukung sistem pakar tersebut adalah metode *K-Nearest Neighbor - Certainty Factor* yang merupakan penggabungan 2 metode dimana hasil klasifikasi dari metode *K-Nearest Neighbor* akan diberi nilai kepastian oleh metode *Certainty Factor* sehingga menghasilkan suatu diagnosis penyakit. Pada penelitian ini data latih dan data uji yang digunakan berjumlah 143 data. Berdasarkan hasil pengujian variasi nilai *K* didapatkan akurasi sebesar 88.37%. Pada pengujian variasi data latih didapatkan akurasi sebesar 86.04%. Pada pengujian rasio data latih dan data uji didapatkan akurasi sebesar 95%. Pada pengujian variasi jumlah data uji didapatkan akurasi sebesar 90%. Pada pengujian variasi data uji didapatkan rata-rata akurasi sebesar 97.22%. Pada pengujian perbandingan metode, metode *k-nearest neighbor-certainty factor* menghasilkan akurasi sebesar 84.79%.

Kata kunci: Sistem Pakar, DBD, Malaria, Tifoid, *K-Nearest Neighbor* dan *Certainty Factor*

ABSTRACT

Elsa Nur Amilus Shofia. 2017. Expert System for Diagnosis of Fever: DBD, Malaria and Thypoid Using K-Nearest Neighbor - Certainty Factor. Computer Sciency Essay, Faculty of Computer Sciency, University of Brawijaya.

Advisor: Rekyan Regasari Mardi Putri, S.T, M.T and Achmad Arwan, S.Kom, M.Kom

Fever is one of the health problems that disrupt everyone's productivity, even it can cause death and remain a health problem in Indonesia. There are several types of fever that needs to be wary, it includes dengue, malaria and typhoid. These three diseases have similar symptoms, so many medical personnel and doctors internship often make mistakes in diagnosing the disease. Therefore, an expert system is required to resolve the issue. The method used to support the expert system is K-Nearest Neighbor - Certainty Factor which is a merger of two methods in which the classification results of K-Nearest Neighbor to be rated certainty by a Certainty Factor method and resulting a diagnosis of the disease. In this study, the training data and test data used were 143 data. Based on test results obtained *K* value variation accuracy of 88.37%. On testing variations training data obtained an accuracy of 86.04%. In testing the ratio of training data and test data obtained an accuracy of 95%. In testing the variation of the number of test data obtained an accuracy of 90%. In testing the variety of test data obtained an average accuracy of 97.22%. In comparison testing method, the method k-nearest neighbor certainty factor gets an accuracy of 84.79%.

Keywords: *Expert System, Dengue, Malaria, Typhoid, K-Nearest Neighbor and Certainty Factor*



DAFTAR ISI

PENGESAHAN	II
PERNYATAAN ORISINALITAS	III
KATA PENGANTAR	IV
ABSTRAK	VI
ABSTRACT	VII
DAFTAR ISI	VIII
DAFTAR TABEL	XI
DAFTAR GAMBAR	XII
BAB 1 PENDAHULUAN	1
1.1 Latar belakang	1
1.2 Rumusan masalah	3
1.3 Tujuan	3
1.4 Manfaat	3
1.5 Batasan masalah	3
1.6 Sistematika pembahasan	3
BAB 2 LANDASAN KEPUSTAKAAN	5
2.1 Kajian Pustaka	5
2.2 Demam Berdarah Dengue (DBD)	8
2.3 Malaria	9
2.4 Tifoid	9
2.5 Sistem Pakar	9
2.5.1 Struktur Sistem Pakar	10
2.5.2 Konsep Dasar Sistem Pakar	10
2.5.3 Ciri-ciri Sistem Pakar	11
2.5.4 Kelebihan Sistem Pakar	11
2.5.5 Kekurangan Sistem Pakar	12
2.5.6 Tujuan Sistem Pakar	12
2.6 Basis Pengetahuan	12
2.7 Metode Pemecahan Masalah (Metode Inferensi)	13
2.8 K-Nearest Neighbor	14
2.9 Certainty Factor	14
2.10 Pengujian Akurasi	15
2.11 Data Flow Diagram	16
2.12 Entity Relationship Diagram	16
BAB 3 METODOLOGI PENELITIAN	18
3.1 Studi Literatur	19
3.2 Analisis Kebutuhan	19

3.3 Pengumpulan Data.....	19
3.4 Perancangan Sistem.....	19
3.4.1 Struktur Sistem Pakar	20
3.4.2 Diagram Blok Sistem	21
3.5 Implementasi Sistem.....	22
3.6 Pengujian Sistem.....	22
3.7 Penutup.....	22
BAB 4 PERANCANGAN	23
4.1 Analisis Kebutuhan Perangkat Lunak.....	24
4.1.1 Analisis Kebutuhan Data	24
4.1.2 Identifikasi Aktor	26
4.1.3 Analisis Kebutuhan Masukan.....	26
4.1.4 Analisis Kebutuhan Proses	27
4.1.5 Analisis Kebutuhan Keluaran	27
4.2 Perancangan Sistem Pakar.....	27
4.2.1 Akuisisi Pengetahuan	28
4.2.2 Basis Pengetahuan	28
4.2.3 Mesin Inferensi	31
4.2.4 Antarmuka	42
4.2.5 Workplace	46
4.2.6 Fasilitas Penjelas	46
4.3 Perancangan Database	46
4.3.1 Data Flow Diagram.....	46
4.3.2 Entity Relationship Diagram	47
BAB 5 IMPLEMENTASI.....	49
5.1 Spesifikasi Sistem	49
5.1.1 Spesifikasi Perangkat Keras.....	49
5.1.2 Spesifikasi Perangkat Lunak.....	50
5.2 Batasan Implementasi.....	50
5.3 Implementasi Algoritma	50
5.3.1 Implementasi Algoritma Normalisasi Data Latih	51
5.3.2 Implementasi Algoritma Normalisasi Data Input	53
5.3.3 Implementasi Algoritma K-Nearest Neighbor	55
5.3.4 Implementasi Algoritma Certainty Factor	57
5.4 Implementasi User Interface	64
5.4.1 Implementasi Halaman Sign In	64
5.4.2 Implementasi Halaman Beranda (Admin).....	65
5.4.3 Implementasi Halaman Riwayat Pasien (Admin).....	65
5.4.4 Implementasi Halaman Olah Data User (Admin).....	66
5.4.5 Implementasi Halaman Daftar Gejala (Admin).....	67
5.4.6 Implementasi Halaman Beranda (Tenaga Medis)	68
5.4.7 Implementasi Halaman Riwayat Pasien (Tenaga Medis).....	68
5.4.8 Implementasi Halaman Diagnosis (Tenaga Medis).....	68

BAB 6 PENGUJIAN DAN ANALISIS.....	70
6.1 Pengujian.....	70
6.1.1 Pengujian Variasi Nilai K.....	70
6.1.2 Pengujian Variasi Data Latih	71
6.1.3 Pengujian Rasio Data Latih dan Data Uji.....	72
6.1.4 Pengujian Variasi Jumlah Data Uji	72
6.1.5 Pengujian Variasi Data Uji	73
6.1.6 Pengujian Perbandingan Metode	73
6.2 Analisis Hasil.....	74
6.2.1 Analisis Hasil Variasi Nilai K.....	74
6.2.2 Analisis Hasil Variasi Data Latih	75
6.2.3 Analisis Hasil Rasio Data Latih dan Data Uji.....	76
6.2.4 Analisis Hasil Variasi Jumlah Data Uji.....	77
6.2.5 Analisis Hasil Variasi Data Uji	77
6.2.6 Analisis Hasil Perbandingan Metode	78
BAB 7 PENUTUP.....	81
7.1 Kesimpulan.....	81
7.2 Saran	82
DAFTAR PUSTAKA.....	83



DAFTAR TABEL

Tabel 2.1 Kajian pustaka metode Certainty Factor.....	5
Tabel 2.2 Kajian pustaka metode K-Nearest Neighbor.....	6
Tabel 4.1 Contoh data pasien	24
Tabel 4.2 Data gejala.....	25
Tabel 4.3 Contoh data nilai tingkatan gejala	25
Tabel 4.4 Identifikasi aktor.....	26
Tabel 4.5 Kebutuhan fungsional	26
Tabel 4.6 CF pakar pada tiap penyakit dan gejala	29
Tabel 4.7 Nilai CF user pada tiap gejala	29
Tabel 4.8 Contoh input data user	35
Tabel 4.9 Contoh nilai tiap gejala yang telah diinputkan user.....	35
Tabel 4.10 Data penyakit Demam.....	36
Tabel 4.11 Data latih setelah dinormalisasi	37
Tabel 4.12 Data uji setelah dinormalisasi	37
Tabel 4.13 Jarak Euclidean penyakit Tifoid	38
Tabel 4.14 Jarak Euclidean penyakit Malaria	38
Tabel 4.15 Jarak Euclidean penyakit DBD	39
Tabel 4.16 Sorting jarak terdekat-terjauh (penyakit Tifoid)	39
Tabel 4.17 Sorting jarak terdekat-terjauh (penyakit Malaria)	40
Tabel 4.18 Sorting jarak terdekat-terjauh (penyakit DBD)	40
Tabel 4.19 Hasil perkalian CF user dengan CF pakar	41
Tabel 4.20 Hasil perhitungan CF combine penyakit Tifoid, Malaria dan DBD	42
Tabel 5.1 Spesifikasi perangkat keras	49
Tabel 5.2 Spesifikasi perangkat lunak	50
Tabel 6.1 Tabel pengujian variasi nilai K	71
Tabel 6.2 Tabel pengujian variasi data latih	71
Tabel 6.3 Tabel pengujian rasio data latih dan data uji	72
Tabel 6.4 Tabel pengujian variasi jumlah data uji.....	73
Tabel 6.5 Tabel pengujian variasi data uji.....	73
Tabel 6.6 Tabel pengujian perbandingan metode	74

DAFTAR GAMBAR

Gambar 2.1 Struktur sistem pakar	10
Gambar 2.2 Proses Forward Chaining	13
Gambar 2.3 Proses Backward Chaining	14
Gambar 2.4 Diagram alur metode Certainty Factor	15
Gambar 2.5 Entitas/Terminator	16
Gambar 2.6 Proses	16
Gambar 2.7 Aliran data	16
Gambar 2.8 Penyimpan data	16
Gambar 2.9 Entitas ERD	17
Gambar 2.10 Relasi ERD	17
Gambar 2.11 Atribut ERD	17
Gambar 3.1 Alur metodologi penelitian	18
Gambar 3.2 Struktur sistem pakar	20
Gambar 3.3 Diagram blok sistem	21
Gambar 4.1 Pohon perancangan sistem pakar diagnosis penyakit Demam: DBD, Malaria dan Tifoid menggunakan metode K-Nearest Neighbor – Certainty Factor	23
Gambar 4.2 Alur perhitungan menggunakan K-Nearest Neighbor – Certainty Factor	27
Gambar 4.3 Proses K-Nearest Neighbor – Certainty Factor	32
Gambar 4.4 Proses normalisasi	33
Gambar 4.5 Proses K-Nearest Neighbor	33
Gambar 4.6 Proses Euclidean	34
Gambar 4.7 Proses Certainty Factor	34
Gambar 4.8 Halaman Sign In admin dan tenaga medis pada system	43
Gambar 4.9 Halaman utama admin	43
Gambar 4.10 Halaman riwayat pasien	44
Gambar 4.11 Halaman olah data user	44
Gambar 4.12 Halaman daftar gejala	44
Gambar 4.13 Halaman beranda tenaga medis	45
Gambar 4.14 Halaman riwayat pasien	45

Gambar 4.15 Halaman diagnosis tenaga medis	46
Gambar 4.16 DFD Level 0.....	47
Gambar 4.17 DFD Level 1.....	47
Gambar 4.18 ER Diagram	48
Gambar 5.1 Pohon implementasi	49
Gambar 5.2 Halaman Sign In.....	65
Gambar 5.3 Halaman beranda (admin)	65
Gambar 5.4 Halaman riwayat pasien (admin)	66
Gambar 5.5 Halaman olah data user (admin).....	66
Gambar 5.6 Halaman setelah admin menambah user (admin)	66
Gambar 5.7 Halaman setelah admin menambah user (admin)	67
Gambar 5.8 Halaman daftar gejala (admin)	67
Gambar 5.9 Halaman setelah admin memasukkan gejala baru	67
Gambar 5.10 Halaman beranda (tenaga medis).....	68
Gambar 5.11 Halaman riwayat pasien (tenaga medis)	68
Gambar 5.12 Halaman diagnosis (tenaga medis)	69
Gambar 5.13 Halaman diagnosis (tenaga medis)	69
Gambar 5.14 Halaman diagnosis setelah tenaga medis memasukkan data pasien dan keluhannya.....	69
Gambar 6.1 Grafik hasil pengujian variasi nilai k.....	74
Gambar 6.2 Grafik hasil pengujian variasi data latih	75
Gambar 6.3 Grafik hasil pengujian rasio data latih dan data uji	76
Gambar 6.4 Grafik hasil pengujian variasi jumlah data uji	77
Gambar 6.5 Grafik hasil pengujian variasi data uji	78
Gambar 6.6 Grafik hasil pengujian perbandingan metode	79

BAB 1 PENDAHULUAN

1.1 Latar belakang

Kesehatan merupakan hal yang paling didambakan oleh setiap orang, karena dengan hidup sehat segala aktifitas dapat dilakukan dengan baik. Sehingga tidak heran jika seseorang melakukan berbagai cara untuk menjaga kesehatannya. Namun demikian, bukan tidak mungkin seseorang tersebut dapat terserang suatu penyakit. Demam merupakan salah satu jenis gangguan kesehatan yang mengganggu produktivitas setiap orang bahkan dapat menyebabkan kematian dan masih menjadi permasalahan kesehatan Indonesia. Terdapat 8 jenis demam yang perlu diwaspadai yaitu Demam Berdarah Dengue (DBD), Tifoid, Malaria, *Viral*, *Chicken Guinea*, Meningitis, Infeksi Saluran Kemih dan HIV.

Pada tahun 2014 sampai pertengahan bulan Desember di Indonesia terhitung 71.668 orang menderita penyakit DBD dan 641 orang meninggal dunia. Tahun 2013, yaitu sejumlah 12.511 orang dan 871 orang meninggal dunia (Adam Rizal, 2014). Di Provinsi Jawa Timur terjadi peningkatan sebesar 46% pada tahun 2014. Berdasarkan data dari Dinas Kesehatan Provinsi Jawa Timur terdapat 10 data Kabupaten/Kota di Jawa Timur dengan jumlah penderita DBD sebanyak 1112 orang (Kementrian Kesehatan RI, 2014). Berbeda dengan wilayah Papua, menurut Radio Australia disana pembunuh nomor satu adalah Malaria. Tercatat lebih dari 431.000 orang yang terserang malaria pada tahun 2012 (Iffah Nur, 2013). Selain itu terdapat penyakit lain yang gejalanya hampir sama dengan DBD dan Malaria yaitu Tifoid. Penyakit tifoid dapat ditemui sepanjang tahun dengan jumlah mencapai 157 per seratus ribu populasi pada daerah semi rural, dan terus meningkat setiap tahunnya. Di Indonesia penyakit demam tifoid masih sangat tinggi dan menempati urutan ketiga di dunia (Agung, 2015).

Untuk mengetahui seseorang terserang penyakit demam jenis apa harus diketahui dari gejala yang dialami oleh orang tersebut. Padahal gejala pada penyakit Demam sangat mirip dan sulit dibedakan sehingga seringkali terjadi kesalahan dalam mendiagnosis karena tenaga medis dan dokter internship kesulitan dalam membedakan gejala ketiga penyakit tersebut (Dr. Anda Citra Utama, 2016). Dokter internship adalah dokter yang baru menyelesaikan masa pendidikan profesi dan sedang melakukan program magang selama kurang lebih satu tahun (Amir, 2007).

Untuk membantu tenaga medis dan dokter internship dalam mengetahui sejak dini jenis penyakit demam yang diderita, maka diperlukan suatu sistem yang memiliki kemampuan layaknya seorang dokter dalam mendiagnosis penyakit. Sistem tersebut adalah sistem pakar yang diadopsi oleh pengetahuan manusia ke dalam komputer agar dapat membantu menyelesaikan masalah yang biasanya dilakukan oleh pakar, salah satunya dengan menerapkan metode *K-Nearest Neighbor – Certainty Factor*. Metode *K-Nearest Neighbor* merupakan salah satu metode klasifikasi yang dapat mengklasifikasikan suatu penyakit dengan beberapa kriteria yang mirip. Pada metode *K-Nearest Neighbor* tidak bisa mengetahui seberapa besar suatu kondisi masuk ke dalam klasifikasi penyakit tersebut, maka

dibutuhkan metode yang dapat menutupi kelemahan dari permasalahan metode *K-Nearest Neighbor* yaitu metode *Certainty Factor*. Metode *Certainty Factor* merupakan metode yang mendefinisikan ukuran kepastian terhadap suatu fakta atau aturan, untuk menggambarkan tingkat keyakinan pakar terhadap masalah yang sedang dihadapi (Nur Anjas, 2013). Dimana data gejala yang diberikan akan digunakan dalam perhitungan *K-Nearest Neighbor* yang selanjutnya hasil dari perhitungan *K-Nearest Neighbor* tersebut akan dilanjutkan dengan metode *Certainty Factor* yang menghasilkan suatu nilai kepastian.

Ada beberapa penelitian yang membahas tentang metode *Certainty Factor*. Penelitian tersebut adalah Pengembangan Sistem Pakar Diagnosis Penyakit *Tuberculosis* dan Demam Berdarah Berbasis Web Menggunakan Metode *Certainty Factor*. Penelitian yang dilakukan oleh Yulianti Paula Bria dan Engelbertus Agung S. Takung ini bertujuan untuk membantu masyarakat dalam mendiagnosis penyakit TBC dan DBD sehingga dapat ditanggulangi lebih dini dan membantu dokter dalam pengambilan keputusan (Yulianti, 2015). Penelitian lain dilakukan oleh Luther A. Latumakulita dengan judul Sistem Pakar Pendiagnosis Penyakit Anak Menggunakan *Certainty Factor*. Dari hasil penelitian tersebut dengan menggunakan metode *Certainty Factor* dapat digunakan untuk mendiagnosis penyakit anak dengan tingkat kepercayaan yang telah ditentukan oleh pakar terhadap gejala-gejala yang mempengaruhi probabilitas terjadinya suatu penyakit anak (Luther, 2012).

Sedangkan penelitian yang menggunakan metode *K-Nearest Neighbor* adalah Model Diagnosis *Tuberculosis* Menggunakan *K-Nearest Neighbor* Berbasis Seleksi Atribut. Ratih Sari dan Purwanto melakukan perbandingan *K-Nearest Neighbor* dengan dua algoritma lain yaitu *Logistic Regression* dan *Decision Tree*. Input pada penelitian ini adalah karakteristik pasien, *anamnesis*, pemeriksaan tanda vital dan pemeriksaan radiologi sesuai dengan standar TB yang ditetapkan serta lama batuk dan warna dahak. Hasil dari eksperimen menunjukkan bahwa model *K-Nearest Neighbor* berbasis *backward elimination* memberikan kinerja yang terbaik jika dibandingkan dengan model-model lain. Model *K-Nearest Neighbor* berbasis *backward elimination* memberikan nilai akurasi sebesar 78.66% dan nilai *AUC* sebesar 0.806 yang mengindikasikan bahwa model adalah *good classification* (Ratih Sari, 2015).

Berdasarkan paparan masalah yang telah disebutkan diatas maka penulis mengusulkan penelitian dengan judul **“Sistem Pakar Diagnosis Penyakit Demam: DBD, Malaria dan Tifoid Menggunakan Metode *K-Nearest Neighbor* – *Certainty Factor*”**. Pada penelitian sebelumnya kedua metode tersebut dianggap memiliki akurasi yang cukup baik. Oleh karena itu, dengan menggunakan sistem pakar dan metode *K-Nearest Neighbor* – *Certainty Factor* diharapkan dapat membantu tenaga medis dan dokter internship.

1.2 Rumusan masalah

Berdasarkan masalah yang telah dijelaskan pada latar belakang, maka rumusan masalah pada penelitian ini adalah sebagai berikut:

1. Bagaimana mengimplementasikan metode *K-Nearest Neighbor – Certainty Factor* untuk mendiagnosis penyakit Demam: DBD, Malaria dan Tifoid?
2. Bagaimana tingkat akurasi metode *K-Nearest Neighbor – Certainty Factor* dalam mendiagnosis penyakit Demam: DBD, Malaria dan Tifoid?
3. Bagaimana perbandingan nilai akurasi metode *K-Nearest Neighbor – Certainty Factor* dengan metode lain dalam mendiagnosis penyakit Demam: DBD, Malaria dan Tifoid?

1.3 Tujuan

Tujuan dari penelitian ini adalah:

1. Untuk mengetahui bagaimana mengimplementasikan metode *K-Nearest Neighbor – Certainty Factor* untuk diagnosis penyakit Demam: DBD, Malaria dan Tifoid.
2. Untuk mengetahui tingkat akurasi menggunakan metode *K-Nearest Neighbor – Certainty Factor* untuk diagnosis penyakit Demam: DBD, Malaria dan Tifoid.
3. Untuk mengetahui perbandingan nilai akurasi metode *K-Nearest Neighbor – Certainty Factor* dengan metode lain dalam mendiagnosis penyakit Demam: DBD, Malaria dan Tifoid.

1.4 Manfaat

Dari penelitian ini diharapkan dapat membantu tenaga medis dan dokter *internship* untuk mengetahui gejala-gejala dari penyakit Demam (Demam Berdarah Dengue, Malaria dan Tifoid) sehingga tidak terjadi kesalahan dalam mendiagnosis ketiga penyakit tersebut.

1.5 Batasan masalah

Batasan yang diberikan pada penelitian ini adalah :

1. Sistem ini hanya mendiagnosis 3 penyakit Demam yaitu Demam Berdarah Dengue, Malaria dan Tifoid.
2. Data pasien yang digunakan adalah 143 data yang diperoleh dari pakar Dr. Anda Citra Utama., SpPD.
3. Keluaran sistem berupa data pasien dan hasil diagnosis penyakit.
4. Pembobotan tingkatan gejala diberikan oleh pakar berdasarkan pengetahuan dan ilmu pakar.

1.6 Sistematika pembahasan

1. BAB I Pendahuluan

Bab pendahuluan menjelaskan tentang latar belakang, rumusan masalah, tujuan penelitian, manfaat penelitian, batasan penelitian, sistematika

pembahasan dari Sistem Pakar Diagnosis Penyakit Demam: DBD, Malaria dan Tifoid Menggunakan Metode *K-Nearest Neighbor - Certainty Factor*.

2. **BAB II Landasan Kepustakaan**

Bab Landasan Kepustakaan menjelaskan tentang teori yang mendukung analisis dan perancangan Sistem Pakar Diagnosis Penyakit Demam: DBD, Malaria dan Tifoid Menggunakan Metode *K-Nearest Neighbor - Certainty Factor*. Teori-teori yang dibahas mencakup sistem pakar, metode *K-Nearest Neighbor*, metode *Certainty Factor*, Penyakit Demam (DBD, Malaria dan Tifoid).

3. **BAB III Metodologi Penelitian**

Bab Metodologi Penelitian membahas mengenai langkah-langkah yang akan dilakukan dalam penelitian ini seperti studi literature, analisis kebutuhan, pengumpulan data, perancangan sistem, implementasi, dan penutup.

4. **BAB IV Perancangan**

Bab ini berisi tentang perancangan aplikasi Sistem Pakar Diagnosis Penyakit Demam: DBD, Malaria dan Tifoid Menggunakan Metode *K-Nearest Neighbor - Certainty Factor* yang disertai juga dengan manualisasi.

5. **BAB V Implementasi**

Pada bab ini akan dilakukan implementasi dari aplikasi Sistem Pakar Diagnosis Penyakit Demam: DBD, Malaria dan Tifoid Menggunakan Metode *K-Nearest Neighbor - Certainty Factor*.

6. **BAB VI Pengujian dan Analisis**

Bab yang membahas tentang pengujian dan analisis pembahasan Sistem Pakar Diagnosis Penyakit Demam: DBD, Malaria dan Tifoid Menggunakan Metode *K-Nearest Neighbor - Certainty Factor*.

7. **BAB VII Penutup**

Bab ini memuat kesimpulan terhadap akurasi dari pengujian yang diperoleh dari pembuatan dan pengujian perangkat lunak yang telah dibuat dalam tugas akhir ini serta memuat saran-saran untuk pengembangan selanjutnya.

BAB 2 LANDASAN KEPUSTAKAAN

Pada bab ini berisi landasan kepastakaan dan dasar teori yang digunakan untuk menunjang topik pembahsan. Dasar teori yang dibutuhkan adalah mengenai Sistem Pakar, Penyakit Demam (DBD, Malaria, Tifoid) dan Metode *K-Nearest Neighbor – Certainty Factor*.

2.1 Kajian Pustaka

Kajian pustaka yang ada dalam penelitian ini terdiri dari beberapa referensi terkait. Berdasarkan judul skripsi yang dibahas, terdapat beberapa jurnal atau penelitian yang berkaitan untuk mendukung pengerjaan penelitian ini. Pada Tabel 2.1 akan dijabarkan beberapa jurnal terkait.

Tabel 2.1 Kajian pustaka metode *Certainty Factor*

No	Judul	Input (Obyek)	Proses	Output (Hasil)
1	Sistem Pakar Mendiagnosis Penyakit Demam Berdarah Menggunakan Metode <i>Certainty Factor</i>	Obyek: Penyakit Demam Berdarah Input: Gejala yang berhubungan dengan penyakit Demam Berdarah Dengue	Metode <i>Certainty Factor</i> 1. Pakar menentukan nilai CF untuk masing-masing gejala 2. Setelah user memilih jawaban/gejala, kemudian ditentukan nilai bobot dari jawaban user 3. Menghitung nilai CF dengan mengkalikan CF dari pakar dengan CF dari user 4. Menghitung nilai <i>CF Combine</i> dengan mengkalikan tiap hasil CF sebelumnya	- Pilihan gejala user - Penyakit yang diderita user - Nilai presentas e penyakit
2	Sistem Pakar Pendiagnosis Penyakit Anak	Obyek: Penyakit Anak	Metode <i>Certainty Factor</i>	- 3 jenis penyakit yang

No	Judul	Input (Obyek)	Proses	Output (Hasil)
	Menggunakan Metode <i>Certainty Factor</i>	Input: Data user, data gejala	1. User memilih gejala dengan mencentang <i>button check</i> yang tersedia. Berdasarkan gejala tersebut sistem akan menjalankan mesin referensi. 2. Gunakan CF untuk menghitung nilai setiap penyakit. 3. Ditampilkan 3 jenis penyakit yang memiliki nilai CF paling tinggi.	memiliki nilai CF paling tinggi dan nilai CF tersebut diurutkan secara <i>descending</i> .

Tabel 2.2 Kajian pustaka metode *K-Nearest Neighbor*

No	Judul	Input (Obyek)	Proses	Output (Hasil)
1	Komparasi Algoritma <i>Backpropagation</i> , <i>Neighbor</i> dan <i>Decision Tree</i> Untuk Mendeteksi Demam Berdarah Pada Pasien Opname	Obyek: Penyakit Demam Berdarah Input: Jenis kelamin, suhu badan, trombosit, <i>hematocrit</i> dan hasil diagnosis.	Perbandingan algoritma <i>backpropagation</i>, <i>Nearest Neighbor</i> dan <i>Decision Tree</i> 1. Terdapat data pasien dengan penyakit DBD rawat inap. 2. Data pasien yang sudah diperoleh kemudian diproses dengan perhitungan menggunakan ketiga algoritma diatas untuk mengetahui perbandingannya.	- Akurasi hasil perbandingan ketiga algoritma

No	Judul	Input (Obyek)	Proses	Output (Hasil)
2	Model Diagnosis Tuberculosis Menggunakan K-Nearest Neighbor Berbasis Seleksi Atribut	Obyek: Penyakit Tuberculosis Input: Anamnesis, Lama Batuk, Karakteristik pasien, warna dahak, pemeriksaan radiologi dan pemeriksaan tanda vital.	Perbandingan algoritma <i>logistic regression</i>, <i>Nearest Neighbor</i> dan <i>Decision Tree</i>. Dalam melakukan analisis terdapat 2 tahap yaitu: 1. Melakukan analisis dengan individual model 2. Menggabungkan ketiga model tersebut dengan metode seleksi variabel <i>Backward elimination</i>. 3. Melakukan perbandingan hasil kinerja semua model individu dan kombinasi metode seleksi variabel dengan ketiga model, dan membandingkan dengan metode seleksi lain yaitu <i>forward selection</i>.	- Akurasi hasil perbandingan ketiga model

Penelitian pertama dilakukan oleh Nur Anjas Sari (Nur Anjas, 2013) dengan judul Sistem Pakar Mendiagnosis Penyakit Demam Berdarah Menggunakan Metode *Certainty Factor*. Gejala yang berhubungan dengan penyakit Demam Berdarah Dengue seperti pusing, mual, nyeri, berkurangnya nafsu makan, tubuh terasa dingin, kesadaran menurun, dan mengalami pendarahan merupakan inputan pada penelitian ini. Proses yang dilakukan dengan menggunakan Metode *Certainty Factor* yang pertama adalah Pakar menentukan nilai CF untuk tiap gejala, setelah *user* memilih jawaban/gejala, kemudian ditentukan nilai bobot dari jawaban *user*, menghitung nilai CF dengan mengkalikan CF dari pakar dengan CF dari *user*, lalu mengkombinasikan nilai CF dari masing-masing kaidah. Dari hasil

penelitian didapatkan hasil perhitungan *certainty factor* pada penyakit Demam Berdarah Dengue memiliki tingkat keyakinan 92%.

Penelitian kedua dilakukan oleh Luther A. Latumakulita dengan judul Sistem Pakar Pendiagnosis Penyakit Anak Menggunakan *Certainty Factor*. Dari hasil penelitian tersebut dengan menggunakan metode *Certainty Factor* dapat digunakan untuk mendiagnosis penyakit anak dengan tingkat kepercayaan yang telah ditentukan oleh pakar terhadap gejala-gejala yang mempengaruhi probabilitas terjadinya suatu penyakit anak (Luther, 2012).

Penelitian terkait lainnya yang membahas tentang sistem pakar dengan metode *K-Nearest Neighbor* adalah penelitian berjudul Komparasi Algoritma *Backpropagation*, *Nearest Neighbor*, dan *Decision Tree* Untuk Mendeteksi Penyakit Demam Berdarah Pada Pasien Opname. Input pada sistem ini yaitu jenis kelamin pasien, suhu badan, trombosit, *hematocrit*, dan hasil diagnosis. Pada penelitian ini peneliti membandingkan keefektifan dari algoritma *Backpropagation*, *Nearest Neighbor*, dan *Decision Tree*. Hasil penelitian komparasi ketiga algoritma pada penyakit Demam Berdarah Dengue pasien rawat inap menunjukkan bahwa algoritma *K-Nearest Neighbor* dapat mendeteksi penyakit demam lebih baik dibandingkan dua metode algoritma lainnya. Berdasarkan hasil penelitian, menyimpulkan bahwa komparasi algoritma lebih efektif dalam mendeteksi klasifikasi penyakit demam berdarah dengue menggunakan *K-Nearest Neighbor* (Untung Suwardi, 2012).

Penelitian yang menggunakan *K-Nearest Neighbor* lainnya adalah Model Diagnosis *Tuberculosis* Menggunakan *K-Nearest Neighbor* Berbasis Seleksi Atribut. Peneliti melakukan perbandingan *K-Nearest Neighbor* dengan dua algoritma lain yaitu *Logistic Regression* dan *Decision Tree*. Input pada penelitian ini adalah karakteristik pasien, anamnesis, pemeriksaan tanda vital dan pemeriksaan radiologi sesuai dengan standar TB yang ditetapkan serta lama batuk dan warna dahak. Hasil dari eksperimen menunjukkan bahwa model *K-Nearest Neighbor* berbasis *backward elimination* memberikan kinerja yang terbaik jika dibandingkan dengan model-model lain. Model *K-Nearest Neighbor* berbasis *backward elimination* memberikan nilai akurasi sebesar 78.66% dan nilai AUC sebesar 0.806 yang mengindikasikan bahwa model adalah *good classification* (Ratih Sari, 2015).

Penelitian terakhir adalah penelitian yang diusulkan oleh penulis. Penelitian ini berjudul Sistem Pakar Diagnosis Penyakit Demam: DBD, Malaria dan Tifoid Menggunakan Metode *K-Nearest Neighbor – Certainty Factor*. Proses perhitungan menggunakan metode *K-Nearest Neighbor* yang selanjutnya akan dihitung dengan *Certainty Factor* untuk mendapatkan nilai kepastian. Dari hasil kedua perhitungan tersebut maka akan diketahui apakah seseorang dengan gejala-gejala yang telah diinputkan masuk kedalam penyakit Demam jenis Malaria, Tifoid atau Demam Berdarah Dengue.

2.2 Demam Berdarah Dengue (DBD)

Demam Berdarah Dengue atau DBD merupakan penyakit yang dapat membuat suhu tubuh penderitanya sangat tinggi. Penyakit demam berdarah

dengue disebabkan oleh *virus dengue* yang disebarkan melalui gigitan nyamuk Nyamuk *Aedes Aegypti*. Demam berdarah dengue tidak bisa menular dari orang ke orang secara langsung seperti penyakit batuk dan flu. Setelah digigit oleh Nyamuk *Aedes Aegypti* biasanya penderita DBD mengalami demam tinggi yang berlangsung selama sehari-hari, nyeri pada kepala dan punggung kadang terdapat tanda-tanda pendarahan, nyeri pada ulu hati dan pendarahan pada saluran cerna (Kemenkes, 2014).

2.3 Malaria

Malaria adalah penyakit akibat infeksi *parasite plasmodium*. Penularan *parasite plasmodium* tidak ditularkan melalui kontak langsung dari satu orang ke orang lainnya melainkan melalui gigitan nyamuk *Anopheles* betina (Depkes, 2013). Gigitan ini lebih sering terjadi pada malam hari. Setelah terjadinya gigitan parasit akan masuk ke dalam aliran darah manusia dan menyerang sel darah merah. Parasit akan memanfaatkan sel darah merah untuk tempat berkembang biak. Jika sel darah merah sudah penuh terisi dengan parasit, sel akan meletus dan parasit akan menyebar ke aliran darah. Bila hal ini terjadi penderita akan mengalami berkeriang, mual, demam tinggi, diare, menggigil, batuk, nyeri pada kepala dan otot persendian, dan muntah.

2.4 Tifoid

Tifoid atau tipus merupakan penyakit yang disebabkan oleh bakteri *Salmonella thypi* dan umumnya menyebar melalui makanan dan minuman yang tidak higienis dan telah terkontaminasi. Infeksi demam terjadi ketika seseorang mengonsumsi makanan atau minuman yang mengandung bakteri, lalu bakteri masuk ke usus melalui makanan atau minuman kemudian berkembang biak di dalam saluran cerna dan menimbulkan sakit perut dan diare. Kondisi penderita tifoid dapat memburuk dalam beberapa minggu jika tidak segera ditangani dengan tepat. Gejala-gejala pada tifoid yang biasa terjadi adalah mual, demam tinggi, batuk, lidah berubah warna menjadi agak putih di bagian tengah dan merah pada bagian pinggir, muntah, nyeri pada kepala dan otot persendian dan sesak napas.

2.5 Sistem Pakar

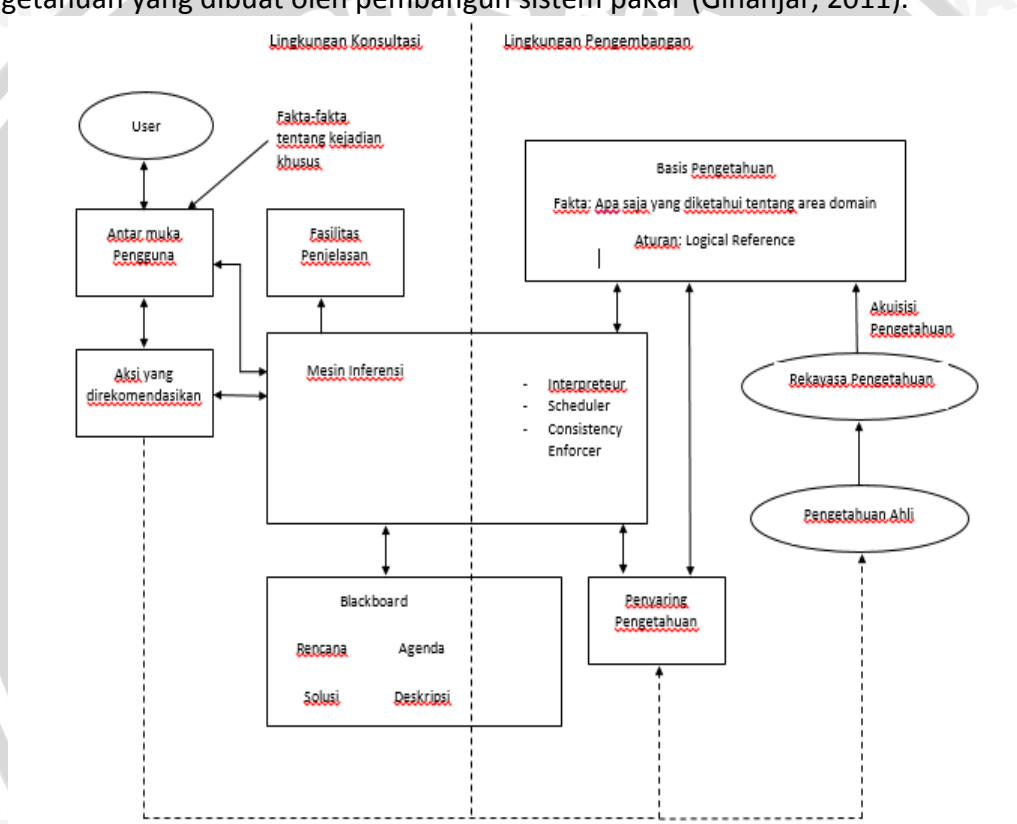
Sistem Pakar adalah sistem yang berusaha mengadopsi pengetahuan para pakar dalam menyelesaikan permasalahan berbasis sistem komputer. Berdasarkan pendapat para ahli (Sri Hartati, 2008), Martin dan Oxman mengatakan bahwa "sistem pakar merupakan sistem yang berbasis komputer dengan menggunakan fakta, pengetahuan, dan teknik penalaran dalam memecahkan permasalahan yang biasanya hanya dapat diselesaikan oleh pakar dalam bidang tertentu". Sedangkan menurut Turban dan Aronson "sistem pakar merupakan sistem yang menggunakan pengetahuan manusia yang dimasukkan kedalam komputer untuk memecahkan permasalahan yang biasanya diselesaikan oleh pakar" (Turban, 2005). Dibawah ini merupakan komponen yang dibutuhkan dalam membangun sistem pakar, antara lain:

1. Memori Kerja (*Working Memory*)

2. Antar Muka Pengguna (*User Interface*)
3. Mekanisme Inferensi (*Inference Machine*)
4. Basis Pengetahuan (*Knowledge Base*)

2.5.1 Struktur Sistem Pakar

Berdasarkan struktur sistem pakar, terdapat bagian penting yaitu lingkungan konsultasi dan lingkungan pengembangan. Dalam melakukan konsultasi dengan sistem, pengguna menggunakan lingkungan konsultasi sehingga mendapatkan pengetahuan dan nasihat dari sistem seperti berkonsultasi dengan seorang pakar. Sedangkan, lingkungan pengembangan berguna untuk membangun komponen dan memperkenalkan pengetahuan ke dalam basis pengetahuan yang dibuat oleh pembangun sistem pakar (Ginanjari, 2011).



Gambar 2.1 Struktur sistem pakar

Sumber: (Ginanjari, 2011)

2.5.2 Konsep Dasar Sistem Pakar

Konsep dasar sistem pakar yaitu kemampuan menjelaskan, ahli/pakar, aturan, keahlian, pengalihan keahlian, aturan dan pengambilan keputusan. Kepakaran memiliki sifat yang luas dan pengetahuan yang didapatkan merupakan pengetahuan dari pengalaman, pelatihan dan membaca. Bentuk pengetahuan yang termasuk dalam keahlian yaitu teori tentang suatu permasalahan tertentu, fakta tentang suatu permasalahan tertentu, aturan dan prosedur dalam suatu permasalahan tertentu.

Seorang ahli/pakar adalah seseorang yang mempunyai kemampuan dalam menjelaskan suatu tanggapan, mempelajari hal-hal baru tentang sebuah topik permasalahan, memecahkan masalah dengan cepat dan tepat. Seorang pakar harus mempunyai pengetahuan, pengalaman, dan metode khusus dan mampu menerapkannya dalam memecahkan suatu masalah atau dalam memberikan saran maupun nasihat.

Pengalihan Keahlian merupakan proses memindahkan keahlian dari seorang pakar ke dalam computer kemudian keahlian tersebut diinformasikan kepada orang lain yang bukan pakar. Pada proses ini terdapat 4 langkah, yaitu pengetahuan yang diperoleh dari pakar atau sumber lainnya, representasi pengetahuan ke komputer, inferensi pengetahuan dan pengalihan pengetahuan ke pengguna.

Pengambilan keputusan dalam sistem pakar adalah tugas sebuah komponen yang disebut dengan mesin inferensi. Mesin inferensi akan menampilkan suatu komponen yang mencakup suatu prosedur dalam menyelesaikan suatu permasalahan. Semua pengetahuan yang dimiliki oleh pakar akan disimpan pada basis pengetahuan yang ada pada mesin inferensi. Sistem pakar merupakan sistem yang dibuat berdasarkan pada aturan-aturan yang akan menyimpan program dalam bentuk rule sebagai prosedur-prosedur pemecahan masalah. Aturan yang ada pada sistem pakar berbentuk *IF-THEN*.

2.5.3 Ciri-ciri Sistem Pakar

Dibawah ini merupakan ciri-ciri pada Sistem pakar (Sutojo, 2011):

1. Bekerja sesuai dengan aturan tertentu.
2. Memiliki fasilitas informasi yang terjamin.
3. Dapat beradaptasi dengan mudah.
4. Jika data tidak lengkap dan tidak pasti sistem mampu memberikan penalaran untuk data-data tersebut.
5. Dapat digunakan dalam berbagai jenis komputer.
6. Keluaran sistem bersifat masukan atau berupa saran.
7. Sistem dapat mengaktifkan aturan secara searah yang sesuai.

2.5.4 Kelebihan Sistem Pakar

Selain kelemahan banyak juga manfaat yang dapat dipelajari dengan adanya sistem pakar, antara lain (Sutojo, 2011):

1. Dapat melakukan proses berulang secara otomatis.
2. Membantu masyarakat awam agar dapat mengerjakan pekerjaan tanpa bantuan pakar.
3. Mampu mengadopsi keahlian para pakar.
4. Meningkatkan output dan produktifitas.
5. Mampu memperoleh informasi meskipun data tidak lengkap dan tidak pasti.
6. Menyimpan pengetahuan dan kemampuan ahli pakar.
7. Memerlukan biaya yang sedikit saat digunakan.

8. Tidak memakan banyak waktu dalam mengambil keputusan.
9. Lebih cepat dalam menyelesaikan permasalahan dibandingkan dengan kemampuan manusia pada data yang sama.
10. Menghabiskan dana yang sedikit meskipun telah dilipatgandakan sesuai dengan kebutuhan.

2.5.5 Kekurangan Sistem Pakar

Selain memiliki kelebihan, sistem pakar juga memiliki kekurangan yaitu:

1. Ketersediaan pakar yang terbatas sesuai dengan bidangnya membuat sistem pakar mengalami kesulitan dalam proses pengembangan.
2. Untuk membuat, memelihara, mengembangkan sistem pakar memerlukan biaya yang tidak sedikit.
3. Sistem pakar perlu diuji sebelum digunakan, sistem pakar tidak 100% benar karena seseorang yang membuat sistem pakar belum tentu benar.

2.5.6 Tujuan Sistem Pakar

Tujuan dari sistem pakar adalah membantu masyarakat agar dapat menyelesaikan permasalahan yang hanya dapat diselesaikan oleh ahli/pakar dengan menggunakan bantuan sistem komputer. Sistem pakar mempunyai 4 proses aktifitas antara lain (Hidayat, 2010):

1. Akuisisi pengetahuan (*knowledge acquisition*) adalah proses menghimpun data dari para ahli/pakar yang berkaitan dengan masalah yang akan diselesaikan.
2. Inferensi pengetahuan (*knowledge inferencing*) adalah proses inferensi berdasarkan dengan pengetahuan yang diperoleh dan metode yang digunakan.
3. Pemindahan pengetahuan (*knowledge transfer*) adalah kegiatan pemindahan pengetahuan dari komputer ke *user* yang tidak ahli.
4. Representasi pengetahuan (*knowledge representation*) adalah proses penyimpanan pengetahuan ke dalam sistem komputer. Data yang disimpan akan menjadi basis pengetahuan

2.6 Basis Pengetahuan

Basis pengetahuan merupakan suatu pengetahuan yang didapat dari pakar untuk menyelesaikan masalah tertentu. Bentuk basis pengetahuan yang umum digunakan yaitu (Arif, 2009):

1. Penalaran berbasis kasus

Pada penalaran berbasis kasus atau *Case-Based Reasoning*, basis pengetahuan diperoleh dari kejadian-kejadian sebelumnya yang telah terjadi dan digunakan untuk memperkirakan kejadian selanjutnya. Selain itu *Case-Based Reasoning* juga dapat digunakan apabila telah memiliki sejumlah kasus tertentu dalam basis pengetahuan.

2. Penalaran Berbasis Aturan

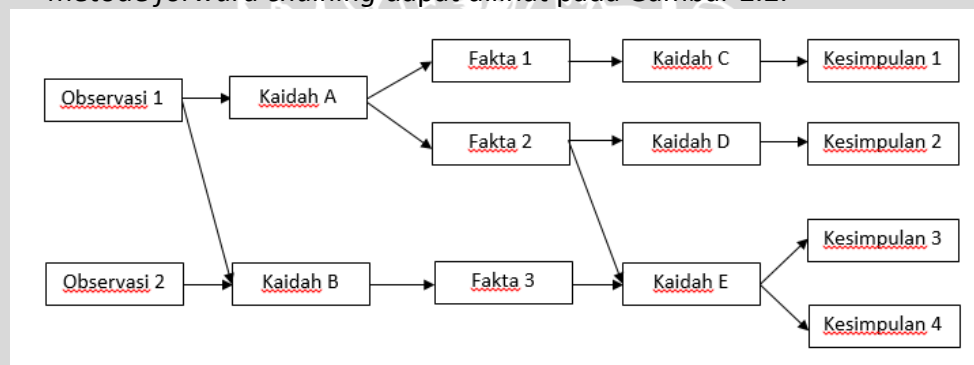
Penalaran berbasis aturan atau *Rule-Based Reasoning* yang digunakan adalah berupa *IF-THEN*. *Rule-Based Reasoning* digunakan ketika terdapat beberapa pengetahuan pakar pada suatu permasalahan tertentu, dan pakar dapat menyelesaikan masalah tersebut secara berurutan. Selain itu, dapat digunakan juga apabila dibutuhkan penjabaran tentang langkah-langkah dalam mencapai solusi.

2.7 Metode Pemecahan Masalah (Metode Inferensi)

Inferensi adalah proses untuk menghasilkan sebuah informasi dari fakta yang diketahui. Dalam sistem pakar proses inferensi disebut mesin inferensi. Pada metode inferensi terdapat dua cara pemecahan masalah yaitu (Hidayati, 2012):

1. *Forward Chaining*

Strategi pada *forward chaining* adalah dimulai dengan mengetahui fakta yang ada, kemudian fakta-fakta tersebut dicocokkan dengan bagian IF dari aturan *IF-THEN*. Bila ada aturan yang cocok dengan bagian IF, aturan tersebut akan dieksekusi. Bila aturan dieksekusi maka ditambahkan sebuah fakta baru (bagian *THEN*) ke dalam basis data. *Forward chaining* disebut juga dengan penelusuran deduktif. Alur metode *forward chaining* dapat dilihat pada Gambar 2.2.

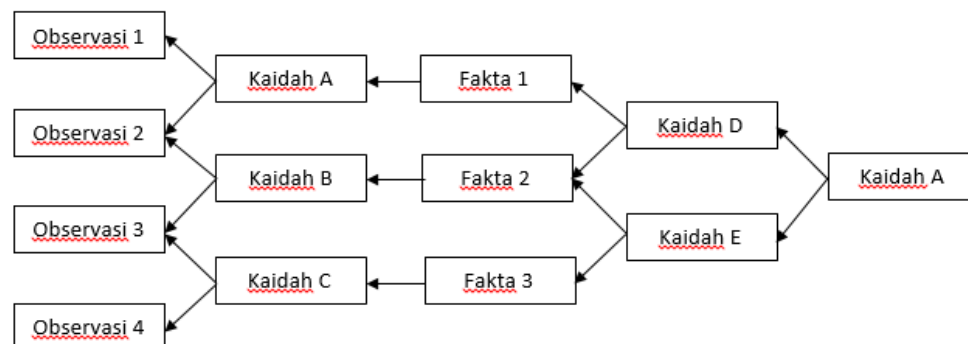


Gambar 2.2 Proses *Forward Chaining*

Sumber: (Hidayati, 2012)

2. *Backward Chaining*

Strategi pada *backward chaining* merupakan strategi penarikan keputusan berdasarkan dari dugaan yang telah diperoleh berdasarkan informasi yang ada. Proses dimulai dari *goal* (yang berada pada bagian *THEN* dari aturan *IF-THEN*), kemudian mulai melakukan pencarian fakta-fakta yang ada untuk dicocokkan dengan premis-premis pada bagian *IF*. Jika cocok, maka aturan akan dieksekusi, kemudian hipotesis di bagian *THEN* ditempatkan sebagai fakta baru pada basis data. Jika tidak cocok premis akan disimpan pada bagian *IF* ke dalam subgoal. Proses berakhir ketika ditemukan *goal* atau tidak ada aturan yang dapat membuktikan kebenaran subgoal atau *goal*. Alur metode *backward chaining* dapat dilihat pada Gambar 2.3.



Gambar 2.3 Proses Backward Chaining

Sumber: (Hidayati, 2012)

2.8 K-Nearest Neighbor

Algoritma *K-Nearest Neighbor* (k-NN atau KNN) merupakan metode untuk melakukan *clustering* terhadap objek berdasarkan data pembelajaran yang jaraknya paling dekat dengan objek tersebut dan nantinya objek tersebut akan dimasukkan kedalam kelas-kelas yang telah disediakan sesuai dengan kelas data pembelajaran. Dimana data pembelajaran disimpan sehingga klasifikasi untuk *record* baru yang belum diklasifikasikan dapat ditemukan dengan membandingkan kemiripan yang paling banyak dalam data pembelajaran. Dalam algoritma *k-nearest neighbor* ini, kemiripan data biasanya dihitung berdasarkan *euclidean distance* yang dipresentasikan seperti Persamaan 2.1 (Ernawati, dkk, 2014) :

$$d = \sqrt{(X_1 - X_2)^2 + (Y_1 - Y_2)^2 + \dots + (N_1 - N_2)^2} \quad (2.1)$$

Keterangan :

d = Jarak Skalar antara kedua atribut (Euclidean)

X_1 = Atribut pertama dari record pertama

X_2 = Atribut pertama dari record kedua

Y_1 = Atribut kedua dari record pertama

Y_2 = Atribut kedua dari record kedua

N_1 = Atribut ke-n dari record pertama

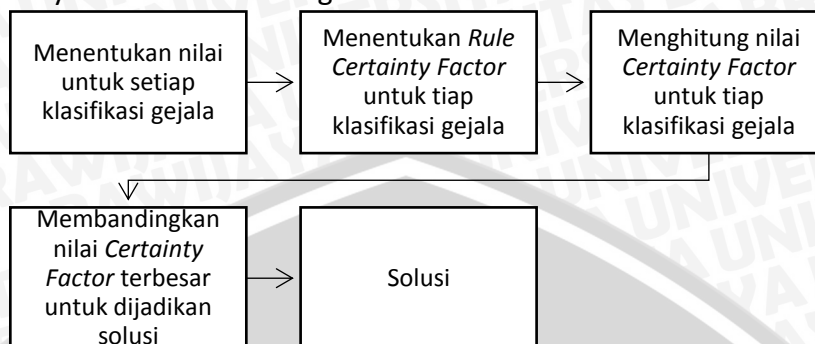
N_2 = Atribut ke-n dari record kedua

Dalam *Euclidean* perhitungan yang dipilih merupakan jarak terpendek atau terdekat antara dua titik.

2.9 Certainty Factor

Certainty Factor adalah teknik yang digunakan untuk mengatasi permasalahan ketidakpastian dalam pengambilan keputusan. *Certainty Factor* dapat terjadi dalam berbagai kondisi, misalnya terdapat beberapa *rule* yang berbeda dengan satu kondisi yang sama. Pada contoh kondisi ini harus dilakukan agregasi pada nilai *Certainty Factor* keseluruhan dari setiap kondisi yang ada. Konsep pada *Certainty Factor* juga sering disebut dengan adanya *believe* dan *disbelieve*.

Pada Certainty Factor terdapat alur dari metode tersebut. Adapun alur pada metode Certainty Factor adalah sebagai berikut:



Gambar 2.4 Diagram alur metode Certainty Factor

Sumber: (Luther, 2012)

Alur pada metode *Certainty Factor* adalah sebagai berikut:

1. Menentukan nilai dari setiap gejala yang sudah ditentukan pakar dan yang diinputkan oleh *user*.
2. Menentukan *Rule Certainty Factor* untuk setiap gejala dari setiap penyakit. *Rule* ini berisi gabungan nilai dari pakar dan *user*.
3. Nilai dari *rule* diatas akan dihitung pada tiap penyakit.
4. Membandingkan nilai terbesar tiap penyakit untuk mendapatkan solusi.
5. Solusi yang telah didapatkan merupakan hasil akhir.

Secara umum rumusan dasar dari metode *Certainty Factor* adalah seperti pada persamaan 2.3 berikut:

$$CF[h, e] = MB[h, e] - MD[h, e] \dots \dots \dots (2.2)$$

Keterangan:

CF [h, e]: *Certainty Factor* dalam hipotesis *h* yang dipengaruhi oleh fakta *e*

MB [h, e]: *Measure of Believe*, adalah nilai kenaikan dari kepercayaan hipotesis *h* yang dipengaruhi oleh fakta *e*

MD [h, e]: *Measure of Disbelieve* adalah nilai kenaikan dari kepercayaan hipotesis *h* yang dipengaruhi oleh fakta *e*

h: hipotesis

e: *evidence*

2.10 Pengujian Akurasi

Akurasi merupakan hasil perhitungan seberapa dekat suatu angka hasil pengukuran terhadap angka sebenarnya. Pengujian tingkat akurasi sistem dengan cara menghitung nilai kebenaran setiap melakukan pengujian pada setiap data. Tingkat akurasi diperoleh dengan perhitungan sesuai dengan persamaan 2.4.

$$\text{Nilai Akurasi} = \frac{\text{jumlah n data akurat}}{\text{jumlah n seluruh data}} \times 100\% \dots \dots \dots (2.3)$$

2.11 Data Flow Diagram

Data Flow Diagram atau DFD adalah diagram untuk menggambarkan sistem jaringan kerja antar fungsi-fungsi yang berhubungan satu sama lain dengan aliran dan penyimpanan data menggunakan notasi-notasi atau simbol-simbol (Nugroho, 2011). DFD memiliki 4 komponen utama, yaitu:

1. Entitas/Terminator

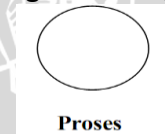
Entitas merupakan suatu unit kerja/jabatan, atau sejenisnya yang berada diluar sistem tetapi memberi andil atas pengiriman dan penerimaan data dari sistem secara langsung. Entitas di representasikan menggunakan gambar persegi panjang.



Gambar 2.5 Entitas/Terminator

2. Proses

Proses merupakan suatu tindakan yang diambil terhadap data yang masuk, karena proses adalah tindakan maka berisi kata kerja. Proses di representasikan menggunakan gambar lingkaran.



Gambar 2.6 Proses

3. Aliran Data

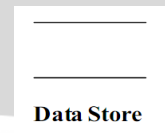
Aliran data berfungsi untuk menyalurkan data dari entitas/terminator menuju ke proses atau dari proses ke proses lainnya. Aliran data di representasikan dengan garis panah.



Gambar 2.7 Aliran data

4. Penyimpan Data

Penyimpan data atau *Data Store* digunakan untuk proses penyimpanan data. Penyimpan data direpresentasikan dengan persegi panjang tanpa kedua sisi kanan dan kirinya.



Gambar 2.8 Penyimpan data

2.12 Entity Relationship Diagram

Entity Relationship Diagram (ERD) merupakan suatu model untuk menjelaskan hubungan antar data dalam basis data berdasarkan objek-objek

dasar data yang mempunyai hubungan antar relasi. Notasi yang digunakan pada ERD adalah sebagai berikut:

1. Entitas

Entitas merupakan suatu obyek yang dapat dibedakan dari lainnya yang dapat diwujudkan dalam basis data. Entitas direpresentasikan dengan gambar persegi panjang.



Gambar 2.9 Entitas ERD

2. Relasi

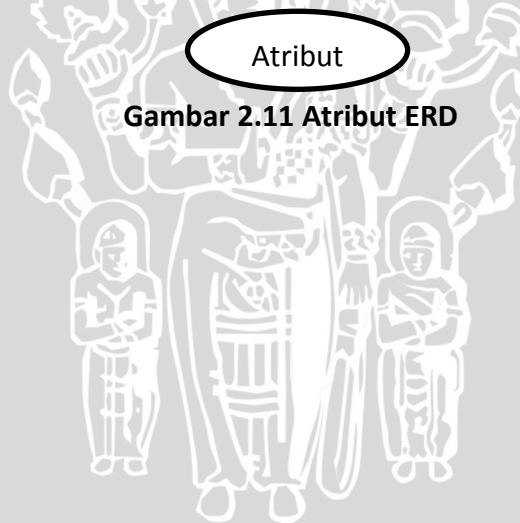
Hubungan antara beberapa entitas dan direpresentasikan dengan gambar belah ketupat.



Gambar 2.10 Relasi ERD

3. Atribut

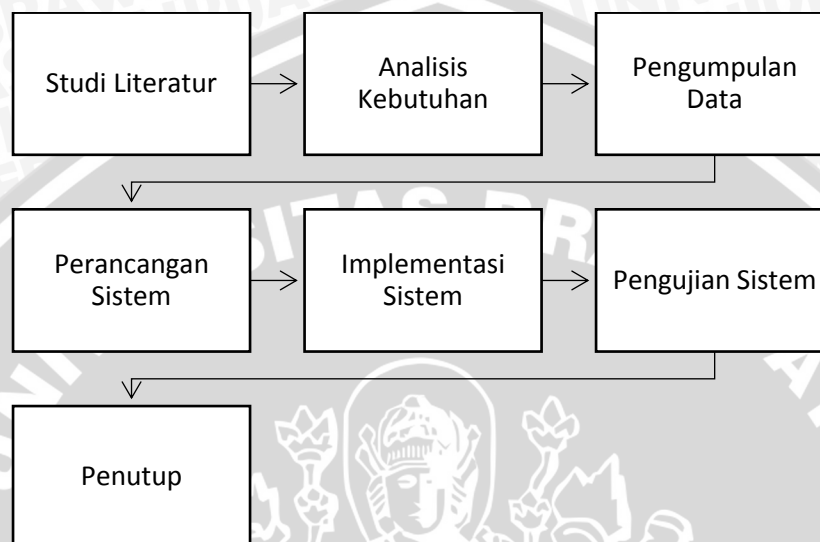
Atribut berfungsi untuk mendeskripsikan karakteristik dari sebuah entitas. Atribut direpresentasikan dengan gambar oval.



Gambar 2.11 Atribut ERD

BAB 3 METODOLOGI PENELITIAN

Pada bab metodologi penelitian ini membahas tentang metode-metode yang dilakukan dalam pembuatan tugas akhir. Pada gambar 3.1 berikut ini akan ditunjukkan alur metodologi penelitian.



Gambar 3.1 Alur metodologi penelitian

Dari Gambar 3.1 dapat diketahui diagram alir dari penelitian ini. Setiap langkah mempunyai peran masing-masing dan saling berkaitan. Penjelasan mengenai langkah-langkah dari alur penelitian adalah sebagai berikut:

1. Melakukan Studi literatur mengenai diagnosis penyakit Demam (DBD, Malaria dan Tifoid), *K-Nearest Neighbor – Certainty Factor*, dan literature lain yang berhubungan dengan metode yang digunakan.
2. Melakukan analisis kebutuhan sistem untuk mengetahui kebutuhan yang diperlukan dalam membangun implementasi sistem meliputi kebutuhan *software*, dan *hardware*.
3. Melakukan pengumpulan data. Data yang digunakan dalam penelitian ini merupakan data yang diperoleh dari pakar.
4. Melakukan perancangan sistem untuk menjelaskan secara umum jalannya sistem yang akan dibuat.
5. Melakukan implementasi sistem dalam bentuk perangkat lunak yang dapat mengklasifikasi data diagnosis penyakit Demam: DBD, Malaria dan Tifoid menggunakan *K-Nearest Neighbor – Certainty Factor*.
6. Melakukan pengujian sistem sehingga dapat diketahui akurasi yang dihasilkan.
7. Pada bagian penutup, dapat dilakukan penarikan kesimpulan dan saran yang digunakan sebagai referensi dan untuk pengembangan pada penelitian selanjutnya.

3.1 Studi Literatur

Studi Pustaka akan difokuskan pada pencarian referensi relevan yang berkaitan dengan permasalahan yang sedang diteliti oleh penulis. Hal ini dilakukan agar peneliti mampu meningkatkan pemahaman dan pengetahuan teori tentang permasalahan yang sedang diteliti. Teori-teori tersebut didapatkan dari berbagai sumber seperti buku, jurnal, *e-book*, laporan penelitian, wawancara serta sumber lain yang teorinya dapat mendukung penelitian tugas akhir ini. Adapun literature yang berhubungan dengan penelitian ini adalah:

- a. Penyakit Demam (DBD, Malaria dan Tifoid)
- b. Sistem Pakar
- c. Metode *K-Nearest Neighbor*
- d. Metode *Certainty Factor*

3.2 Analisis Kebutuhan

Analisis kebutuhan bertujuan untuk mengetahui kebutuhan-kebutuhan apa saja yang diperlukan Sistem Pakar Diagnosis Penyakit Demam: DBD, Malaria dan Tifoid Menggunakan Metode *K-Nearest Neighbor – Certainty Factor*. Analisis kebutuhan terbagi atas analisis kebutuhan data, analisis kebutuhan proses dan analisis kebutuhan database. Analisis kebutuhan data yaitu dengan mengumpulkan data-data mengenai pasien penyakit Demam serta data gejala-gejala pada penyakit dengan melakukan wawancara kepada pakar. Analisis kebutuhan proses bertujuan untuk mengetahui tahapan-tahapan proses kerja pada sistem meliputi analisis kebutuhan masukan, proses dan keluaran serta identifikasi aktor untuk mengetahui peran pengguna dalam sistem pakar ini. Analisis kebutuhan proses dilakukan dengan menganalisis data masukan dengan menggunakan metode *K-Nearest Neighbor – Certainty Factor*, menganalisis kebutuhan fungsional sistem, menganalisis alur proses perhitungan menggunakan metode *K-Nearest Neighbor – Certainty Factor*, menganalisis kebutuhan keluaran sistem dan membuat sebuah *Data Flow Diagram* dan *Entity Relationship Diagram*.

3.3 Pengumpulan Data

Pengumpulan data dalam diagnosis penyakit Demam dilakukan dengan wawancara kepada narasumber yaitu Dr. Anda Citra Utama SpPD melalui telepon, dikarenakan jarak pakar yang berada di Riau (Rumah Sakit Umum Selasih Pangkalan Kerinci Kabupaten Pelalawan). Data yang didapatkan berjumlah 143 data dan berupa data pengetahuan tentang penyakit Demam beserta gejala-gejalanya, data nilai gejala serta data pasien yang nantinya digunakan sebagai data latih dalam perhitungan menggunakan metode *K-Nearest Neighbor – Certainty Factor*. Data pasien yang digunakan adalah data pasien yang mengidap penyakit DBD, Malaria dan Tifoid.

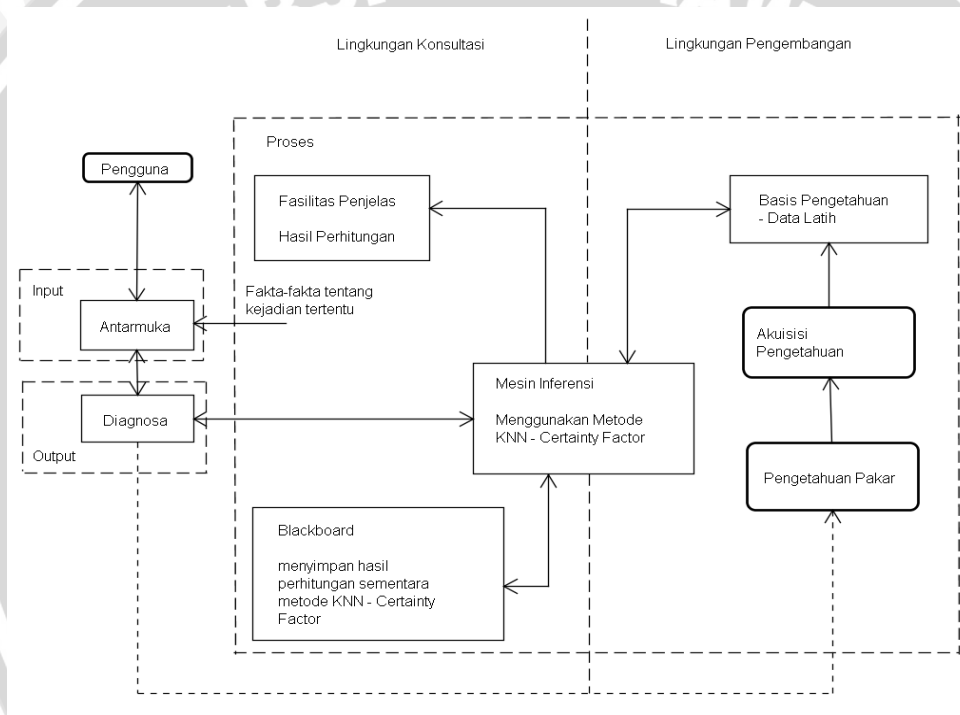
3.4 Perancangan Sistem

Tahap perancangan sistem ini akan menjelaskan secara umum jalannya sistem yang akan dibuat. Perancangan sistem dibuat sebagai acuan dasar dari

tahap implementasi. Tipe yang digunakan pada penelitian ini adalah tipe implementatif yaitu dapat menghasilkan sebuah sistem yang terdiri dari subsistem yang saling berkaitan untuk membentuk proses yang dapat mengolah input menjadi output yang diharapkan. Untuk mengimplementasikannya diperlukan perancangan sistem agar dapat menggambarkan secara umum jalannya sistem yang akan dibuat. Sistem yang akan dibuat adalah Sistem Pakar Diagnosis Penyakit Demam: DBD, Malaria dan Tifoid Menggunakan Metode *K-Nearest Neighbor* – *Certainty Factor*.

3.4.1 Struktur Sistem Pakar

Struktur sistem pakar diagnosis penyakit Demam: DBD, Malaria dan Tifoid menggunakan metode *K-Nearest Neighbor* – *Certainty Factor* akan ditunjukkan pada Gambar 3.2.



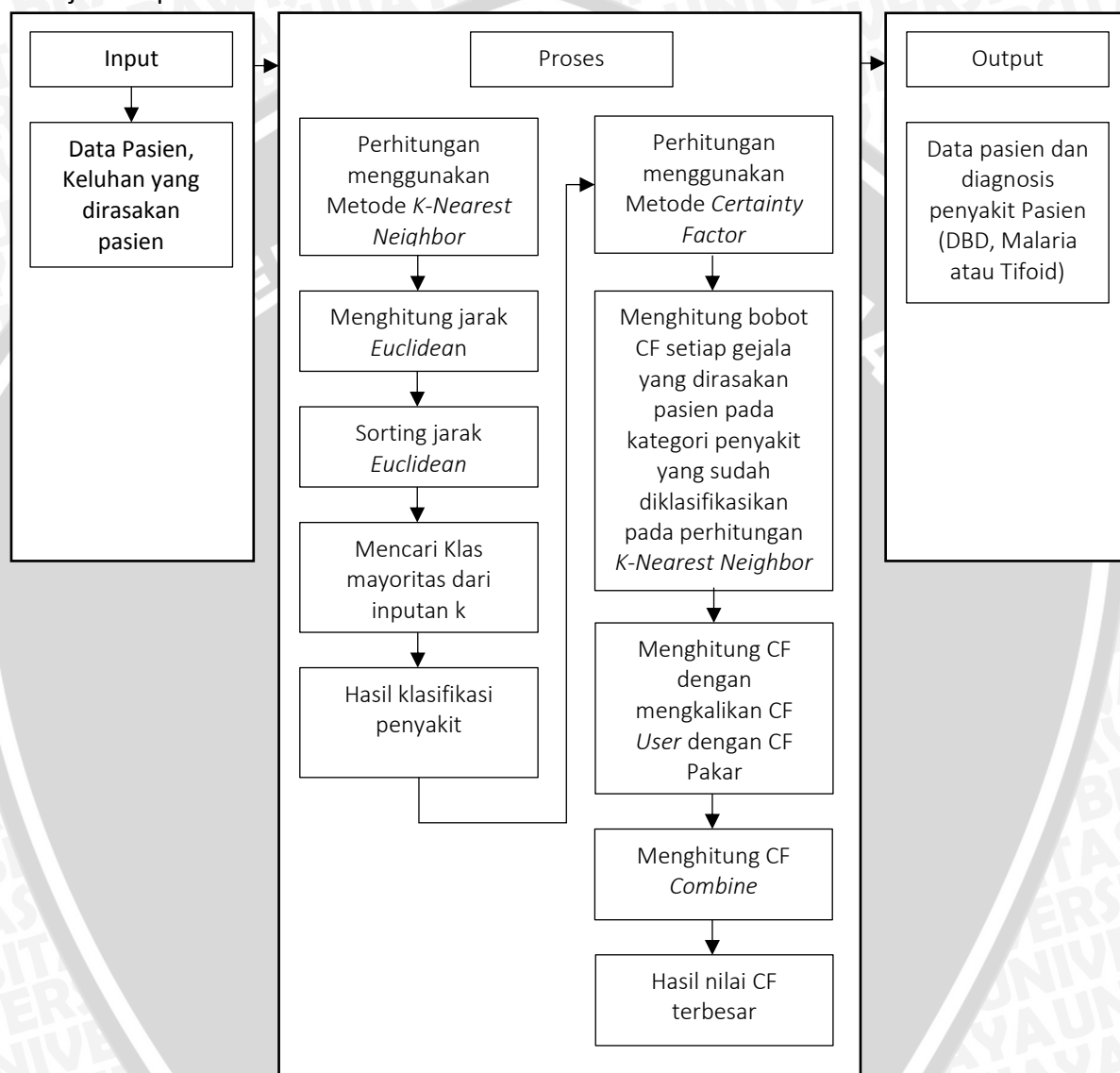
Gambar 3.2 Struktur sistem pakar

Pada Gambar 3.2 dijelaskan struktur sistem pakar berdasarkan pada komponen-komponen yang akan dibangun. Di dalam sistem ini terdapat pengguna yaitu dokter internship dan tenaga medis yang akan memasukkan data pasien dan gejala yang dirasakan pasien dari pemeriksaan fisik. Antarmuka digunakan sebagai jembatan antara pengguna dengan sistem. Pada antarmuka, sistem akan menyediakan halaman diagnosis yang akan diisi oleh pengguna.

Blackboard digunakan untuk menyimpan hasil perhitungan sementara metode *K-Nearest Neighbor* – *Certainty Factor*. Data latih disimpan pada basis pengetahuan yang nantinya digunakan untuk melakukan penalaran pada mesin inferensi.

3.4.2 Diagram Blok Sistem

Diagram blok sistem merupakan tahapan-tahapan perhitungan dari input awal kemudian proses pengolahan data hingga mendapatkan hasil *output* yang berupa hasil diagnosis penyakit. Diagram blok sistem pada diagnosis penyakit Demam menggunakan metode *K-Nearest Neighbor – Certainty Factor* akan ditunjukkan pada Gambar 3.3.



Gambar 3.3 Diagram blok sistem

Gambar 3.3 dijelaskan bagaimana sistem bekerja. Tenaga medis dapat memasukkan data pasien dan keluhan-keluhan yang dirasakan pada form yang telah disediakan oleh sistem. Setelah tenaga medis menekan tombol proses, maka dari data yang telah didapat kemudian dihitung menggunakan metode *K-Nearest Neighbor* untuk mengklasifikasikan pasien tersebut masuk ke jenis penyakit DBD, Malaria atau Tifoid, setelah mengetahui pasien masuk ke jenis penyakit apa lalu

menghitung nilai kepastian penyakit yang diderita pasien dengan *Certainty Factor*, dan hasil keluaran adalah berupa data pasien dan hasil diagnosis penyakit.

3.5 Implementasi Sistem

Pada tahap implementasi sistem dilakukan dengan mengacu pada perancangan sistem. Implementasi perangkat meliputi:

- a. Pembuatan antarmuka sistem (*user interface*)
- b. Implementasi basis data menggunakan *database MySQL* yang nantinya digunakan untuk memasukkan data.
- c. Implementasi algoritma dan melakukan perhitungan menggunakan metode *K-Nearest Neighbor – Certainty Factor* kedalam bahasa pemrograman *PHP*.

Dalam implementasi sistem, perangkat yang digunakan antara lain :

1. Perangkat Keras
 - a. Laptop dengan *Processor Intel® Core™ i5-4200U CPU @ 1.60GHz* 2.30GHz
 - b. *RAM 4.00 GB*
2. Perangkat Lunak
 - a. Sistem Operasi *Windows 8.0*
 - b. *XAMPP v.3.2.1*
 - c. *Macromedia Dreamweaver 8*

3.6 Pengujian Sistem

Pada tahap ini akan dilakukan pengujian dengan data penyakit Demam (DBD, Malaria dan Tifoid). Tujuan dilakukannya pengujian sistem ini adalah untuk mengetahui apakah sistem yang telah dibuat dapat bekerja sesuai dengan yang diharapkan. Pengujian yang akan dilakukan antara lain:

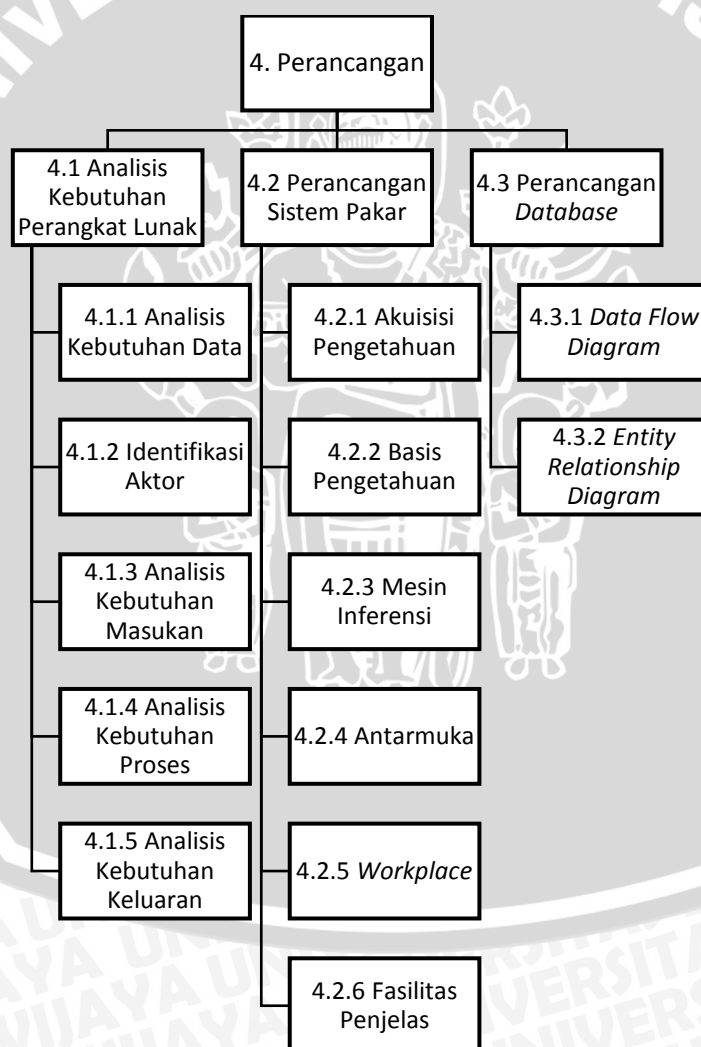
1. Pengujian Variasi Nilai K
2. Pengujian Variasi Data Latih
3. Pengujian Rasio Data Latih dan Data Uji
4. Pengujian Variasi Jumlah Data Uji
5. Pengujian Variasi Data Uji
6. Pengujian Perbandingan Metode

3.7 Penutup

Pada bagian penutup akan dilakukan penarikan kesimpulan dan saran yang berhubungan dengan hasil pengujian yang telah dilakukan. Kesimpulan diambil dari tahap perancangan hingga pengujian sistem. Saran bermanfaat untuk memperbaiki kesalahan pada penelitian dan dapat dilakukan pengembangan terhadap sistem lebih lanjut pada masa yang akan datang.

BAB 4 PERANCANGAN

Bab ini akan menjelaskan mengenai perancangan pada sistem pakar diagnosis penyakit Demam (DBD, Malaria dan Tifoid). Perancangan sistem memiliki tahap-tahap yang terdiri dari Analisis Kebutuhan Perangkat Lunak dan Perancangan Sistem Pakar. Tahap analisis kebutuhan perangkat lunak yaitu analisis kebutuhan data, identifikasi aktor, analisis kebutuhan masukan, analisis kebutuhan proses dan analisis kebutuhan keluaran. Tahap perancangan sistem pakar yaitu akuisisi pengetahuan, basis pengetahuan, mesin inferensi, antarmuka, *workplace* dan fasilitas penjelas. Tahap perancangan *database* yaitu *Data Flow Diagram* dan *Entity Relationship Diagram*. Berikut ilustrasi pohon perancangan Sistem Pakar Diagnosis Penyakit Demam: DBD, Malaria dan Tifoid Menggunakan Metode *K-Nearest Neighbour – Certainty Factor*.



Gambar 4.1 Pohon perancangan sistem pakar diagnosis penyakit Demam: DBD, Malaria dan Tifoid menggunakan metode *K-Nearest Neighbor – Certainty Factor*

4.1 Analisis Kebutuhan Perangkat Lunak

Tahapan analisis kebutuhan perangkat lunak memiliki tujuan untuk memodelkan informasi yang akan digunakan dalam tahapan perancangan. Analisis kebutuhan sistem yang diperlukan meliputi analisis kebutuhan data, identifikasi aktor, analisis kebutuhan masukan, analisis kebutuhan proses dan analisis kebutuhan keluaran. Sub bab selanjutnya merupakan penjelasan dari masing-masing tahapan.

4.1.1 Analisis Kebutuhan Data

Data yang dibutuhkan pada penelitian ini adalah data pasien penyakit Demam yang akan dijadikan sebagai data latih dan data uji, data gejala dan data nilai tingkatan gejala pada ketiga penyakit tersebut.

4.1.1.1 Data Latih dan Data Uji

Data pasien dengan penyakit Demam (DBD, Malaria dan Tifoid) yang didapat berjumlah 143 data, yang terbagi menjadi 100 data digunakan untuk data latih dan 43 data digunakan untuk data uji. Data latih digunakan sebagai pengetahuan untuk menyelesaikan masalah menggunakan penalaran berbasis kasus. Atribut pada data latih dan data uji berupa nilai gejala sesuai dengan jenis gejala dan diagnosis penyakit. Tabel 4.1 merupakan contoh data pasien penyakit Demam (DBD, Malaria dan Tifoid).

Tabel 4.1 Contoh data pasien

PASIEN	GEJALA															PENYAKIT
	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14	G15	
1	0	0.5	0.75	0.75	0.9	0.75	0.85	0.6	0.8	0	0.85	0.75	0.5	0.65	0	TYPHOID
2	0.8	0.85	0	0.5	0.9	0	0	0.2	0	0	0	0	0.5	0.65	0.85	MALARIA
3	0	0	0.75	0.5	0	0.75	0.6	0.6	0	0	0	0.75	0.5	0.65	0	TYPHOID
4	0.8	0.85	0	0.5	0.7	0	0	0.2	0.4	0	0	0	0.5	0.65	0.85	MALARIA
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
132	0	0	0	0.2	0.9	0.9	0.85	0.2	0	1	0	0	0	0	0	DBD
133	0	0	0	0.2	0.9	0.9	0.85	0.2	0	1	0	0	0	0	0	DBD

4.1.1.2 Data Gejala

Data gejala pada penyakit Demam (DBD, Malaria dan Tifoid) berjumlah 15 gejala yaitu demam intermitten, demam menggigil, demam terutama di malam hari, lama demam, mual, sakit kepala, sakit tulang dan sendi, dan muntah, mencret atau susah BAB, nyeri perut, bintik merah pada kulit, lidah kotor, bradikardi relative, pembesaran hati, pembesaran limpa dan kulit lembab. Data gejala dapat dilihat pada Tabel 4.2.

Tabel 4.2 Data gejala

NO	GEJALA	Penyakit		
G1	Demam Intermittent (putus putus)		Malaria	
G2	Demam menggigil	Tifoid	Malaria	DBD
G3	Demam terutama malam hari	Tifoid	Malaria	DBD
G4	Demam lebih dari 1 minggu	Tifoid	Malaria	DBD
G5	Sakit Kepala	Tifoid	Malaria	DBD
G6	Sakit sakit tulang dan sendi	Tifoid	Malaria	DBD
G7	Mual dan muntah	Tifoid		DBD
G8	Mencret atau susah BAB (konstipasi)	Tifoid		
G9	Nyeri Perut	Tifoid		DBD
G10	Bintik merah (ptekie) pada kulit			DBD
G11	Lidah kotor (coated tongue)	Tifoid		
G12	Bradikardi relatif	Tifoid		
G13	Pembesaran hati	Tifoid	Malaria	DBD
G14	Pembesaran limpa	Tifoid	Malaria	DBD
G15	Kulit lembab/ Keringat		Malaria	

4.1.1.3 Data Nilai Tingkatan Gejala

Data nilai tingkatan gejala adalah nilai tingkatan tiap gejala yang digunakan untuk menentukan nilai CF User pada perhitungan *Certainty Factor*. Contoh Data nilai tingkatan gejala ditunjukkan pada tabel 4.3

Tabel 4.3 Contoh data nilai tingkatan gejala

NO	GEJALA	SKOR
G1	Demam Intermittent (putus putus)	
	Ya	0.8
	Tidak	0
.	.	.
.	.	.
G14	Pembesaran limpa	
	Ya	0.65
	Tidak ada	0
G15	Kulit lembab/ Keringat	
	Berat	0.85
	Sedang	0.4
	Tidak ada	0

4.1.2 Identifikasi Aktor

Pada bagian ini bertujuan untuk mengidentifikasi aktor-aktor yang akan berinteraksi dengan aplikasi sistem pakar. Pada kolom aktor menjelaskan siapa saja aktor yang ada pada sistem dan pada kolom deskripsi aktor menjelaskan apa saja yang dapat dilakukan aktor pada sistem. Penjelasan lebih lanjut mengenai identifikasi aktor dapat dilihat pada Tabel 4.4.

Tabel 4.4 Identifikasi aktor

No	Aktor	Definisi Aktor
1	Admin	Admin merupakan aktor yang dapat mengelola siapa saja yang dapat mengakses sistem. Admin dapat melakukan <i>sign in</i> dan <i>sign out</i> , mengelola data <i>user</i> , menambah daftar gejala berdasarkan pengetahuan pakar dan melihat data riwayat pasien.
2	Tenaga Medis	Tenaga Medis merupakan dokter internship dan tenaga medis. Tenaga Medis dapat melakukan <i>sign in</i> dan <i>sign out</i> , dapat melakukan diagnosis dan melihat data riwayat pasien.

4.1.3 Analisis Kebutuhan Masukan

Pada Tahapan ini ada beberapa data masukan yang dibutuhkan oleh sistem yaitu data pasien penyakit demam (dbd, malaria, tifoid) sebagai data latih dan data uji, data gejala penyakit dbd, malaria, tifoid dan data nilai tingkatan tiap gejala. Tabel 4.5 merupakan tabel daftar kebutuhan fungsional yang bertujuan untuk menjelaskan kebutuhan sistem yang harus dipenuhi saat aktor melakukan sebuah aksi.

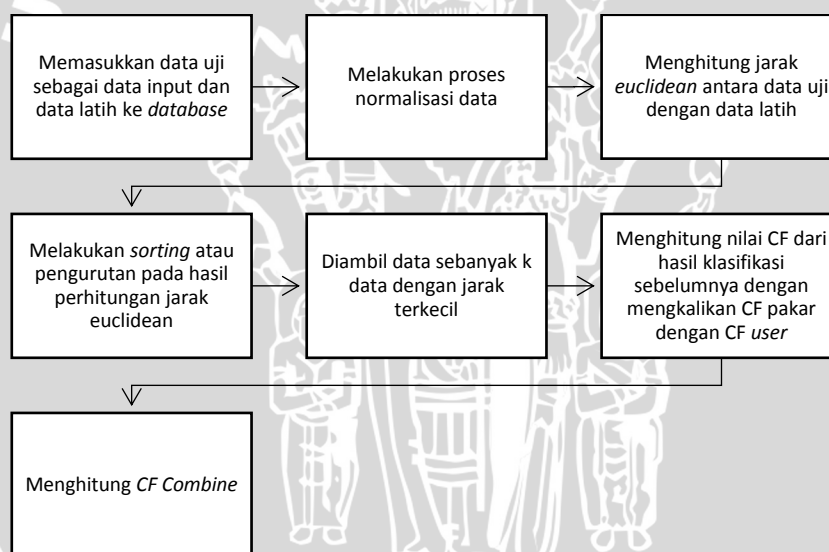
Tabel 4.5 Kebutuhan fungsional

ID	Requirement	Entitas	Nama Aliran Data
KF_01	Sistem dapat menyediakan menu <i>sign in</i>	Admin, Tenaga Medis	<i>Sign In</i>
KF_02	Sistem dapat menyediakan <i>sign out</i>	Admin, Tenaga Medis	<i>Sign Out</i>
KF_03	Sistem dapat menyediakan menu diagnosis	Tenaga Medis	Diagnosis
KF_04	Sistem dapat menampilkan hasil diagnosis	Tenaga Medis	Diagnosis

ID	Requirement	Entitas	Nama Aliran Data
KF_05	Sistem dapat menampilkan data riwayat pasien	Admin, Tenaga Medis	Tampil riwayat pasien
KF_06	Sistem dapat menambah gejala	Admin	Tambah gejala
KF_07	Sistem dapat menghapus gejala	Admin	Hapus gejala
KF_08	Sistem dapat mengelola data <i>user</i>	Admin	Olah data <i>user</i>

4.1.4 Analisis Kebutuhan Proses

Pada tahap ini, sistem melakukan proses penalaran untuk menentukan jenis penyakit yang diderita pasien sesuai dengan gejala-gejala yang diinputkan oleh tenaga medis. Proses pada sistem yaitu perhitungan menggunakan *K-Nearest Neighbor – Certainty Factor*. Pada Gambar 4.2 merupakan alur perhitungan menggunakan metode *K-Nearest Neighbor – Certainty Factor*.



Gambar 4.2 Alur perhitungan menggunakan *K-Nearest Neighbor – Certainty Factor*

4.1.5 Analisis Kebutuhan Keluaran

Analisis kebutuhan keluaran pada sistem ini berupa data pasien dan hasil diagnosis antara lain nama pasien, alamat, no hp dan jenis penyakit (DBD, Malaria atau Tifoid). Hasil dari diagnosis tersebut berdasarkan dari data gejala-gejala yang telah diinputkan tenaga medis.

4.2 Perancangan Sistem Pakar

Pada tahap perancangan sistem pakar terdiri dari beberapa proses antara lain akuisisi pengetahuan, basis pengetahuan, mesin inferensi, antarmuka,

workplace dan fasilitas penjas. Sistem yang akan dibuat adalah Sistem Pakar Diagnosis Penyakit Demam: DBD, Malaria dan Tifoid Menggunakan Metode *K-Nearest Neighbor – Certainty Factor*. Dalam penelitian ini terdapat 15 gejala penyakit Demam (DBD, Malaria dan Tifoid). Metode yang diterapkan dalam penelitian ini yaitu *K-Nearest Neighbor – Certainty Factor* yang digunakan untuk mendiagnosis penyakit Demam: DBD, Malaria dan Tifoid berdasarkan pada gejala-gejala yang ada pada data sesuai dengan pengetahuan dari pakar.

4.2.1 Akuisisi Pengetahuan

Akuisisi pengetahuan merupakan transformasi keahlian, akumulasi dan transfer untuk menyelesaikan masalah dari sumber pengetahuan ke dalam program komputer. Akuisisi pengetahuan dilakukan dengan mencari berbagai informasi yang dapat diperoleh dari buku, jurnal, laporan penelitian, skripsi yang telah ada sebelumnya, dan internet. Ada tiga metode dalam akuisisi pengetahuan, antara lain: wawancara, observasi pada pakar dan analisis. Metode akuisisi pengetahuan yang digunakan dalam penelitian ini adalah wawancara.

Wawancara dilakukan melalui tanya jawab dengan pakar yaitu dr. Anda Citra Utama, SpPD. Dari hasil wawancara dengan pakar, diperoleh beberapa informasi yaitu:

- a. Pengetahuan seputar Penyakit Demam (DBD, Malaria dan Tifoid)
- b. Gejala-gejala pada Penyakit Demam (DBD, Malaria dan Tifoid)
- c. Nilai bobot tiap gejala dari Penyakit Demam (DBD, Malaria dan Tifoid)

4.2.2 Basis Pengetahuan

Basis pengetahuan berisi pengetahuan-pengetahuan pakar berupa prosedur, konsep, fakta-fakta, aturan yang telah digambarkan dalam bentuk yang mudah dimengerti sistem. Pada penelitian ini digunakan pendekatan penalaran *Rule Base Reasoning (RBS)*. *Rule Base Reasoning* menggunakan aturan *IF-THEN* yang akan berguna ketika terdapat beberapa pengetahuan pakar pada permasalahan tertentu dan permasalahan tersebut dapat diselesaikan oleh pakar secara berurutan. Berdasarkan data dari pakar terdapat 15 gejala yaitu demam putus-putus, demam menggigil, demam terutama malam hari, demam lebih dari satu minggu, sakit kepala, sakit tulang dan sendi, mual dan muntah, mencret atau susah BAB, nyeri perut, bintik merah pada kulit, lidah kotor, bradikardi *relative*, pembesaran hati, pembesaran limpa, dan kulit lembab/keringat. Setiap gejala memiliki tingkatan yang telah diberikan skor oleh pakar. Dari inputan yang dirasakan pasien, kemudian akan diproses sehingga menghasilkan output berupa jenis penyakit DBD, Malaria atau Tifoid. Tabel 4.6 merupakan gejala yang ada pada penyakit Demam (DBD, Malaria dan Tifoid).

Tabel 4.6 CF pakar pada tiap penyakit dan gejala

NO	VARIABEL	Tifoid	Malaria	DBD
		P1	P2	P3
I	KELUHAN			
G1	Demam Intermittent (putus putus)	0.3	0.9	0.2
G2	Demam menggigil	0.4	0.9	0.4
G3	Demam terutama malam hari	0.8	0.2	0.6
G4	Demam lebih dari 1 minggu	0.8	0.7	0.1
G5	Sakit Kepala	0.9	0.7	0.8
G6	Sakit sakit tulang dan sendi	0.5	0.7	1
G7	Mual dan muntah	0.9	0.3	0.7
G8	Mencoret atau susah BAB (konstipasi)	0.7	0.2	0.2
G9	Nyeri Perut	0.5	0	0.9
II	PEMERIKSAAN FISIK			
G10	Bintik merah (ptekie) pada kulit	0	0	1
G11	Lidah kotor (coated tongue)	0.9	0.4	0.2
G12	Bradikardi relative	0.8	0	0
G13	Pembesaran hati	0.5	0.5	0.3
G14	Pembesaran limpa	0.6	0.8	0.3
G15	Kulit lembab/ Keringat	0	0.9	0.2

Nilai dari P1, P2, dan P3 merupakan *range* nilai kepercayaan yang diberikan antara 0 sampai 1 terhadap gejala-gejala yang ada pada penyakit Demam berdasarkan hasil wawancara dengan pakar. Tiap gejala memiliki tingkatan yang telah diberikan oleh pakar. Nilai tersebut digunakan sebagai acuan untuk data latih dan data uji. Nilai tingkatan gejala dapat dilihat pada tabel 4.7.

Tabel 4.7 Nilai CF user pada tiap gejala

NO	VARIABEL	SKOR
I	KELUHAN	
G1	Demam Intermittent (putus putus)	
	Ya	0.8
	Tidak	0
G2	Demam menggigil	
	Berat	0.85
	Sedang	0.5
	Tidak ada	0
G3	Demam terutama malam hari	
	Ya	0.75
	Tidak	0
G4	Lama demam	
	> 7 hari	0.75

NO	VARIABEL	SKOR
I	KELUHAN	
	4 - 7 hari	0.5
	1 - 3 hari	0.2
G5	Sakit Kepala	
	Berat	0.9
	Sedang	0.7
	Tidak ada	0
G6	Sakit sakit tulang dan sendi	
	Berat	0.9
	Sedang	0.75
	Tidak ada	0
G7	Mual dan muntah	
	Berat	0.85
	Sedang	0.6
	Tidak ada	0
G8	Mencoret atau susah BAB (konstipasi)	
	Ada	0.6
	Tidak ada	0.2
G9	Nyeri Perut	
	Berat	0.8
	Sedang	0.4
	Tidak ada	0
II	PEMERIKSAAN FISIK	
G10	Bintik merah (ptekie) pada kulit	
	Berat	1
	Sedang	0.7
	Tidak ada	0
G11	Lidah kotor (coated tongue)	
	Berat	0.85
	Sedang	0.4
	Tidak ada	0
G12	Bradikardi relatif	
	Ya	0.75
	Tidak ada	0
G13	Pembesaran hati	
	Ya	0.5
	Tidak ada	0
G14	Pembesaran limpa	
	Ya	0.65
	Tidak ada	0

NO	VARIABEL	SKOR
I	KELUHAN	
G15	Kulit lembab/ Keringat	
	Berat	0.85
	Sedang	0.4
	Tidak ada	0

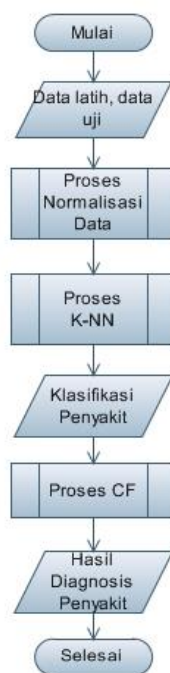
4.2.3 Mesin Inferensi

Mesin inferensi merupakan suatu program komputer untuk memberikan penalaran tentang informasi pada basis pengetahuan. Mesin inferensi ini akan dimodelkan bagaimana sistem memproses data dimulai dari gejala-gejala yang telah dimasukkan oleh tenaga medis sebagai data masukan ke sistem. Kemudian data yang di dapat dari inputan tenaga medis tersebut dihitung sesuai dengan perumusan *K-Nearest Neighbor – Certainty Factor* untuk mengetahui hasil diagnosis penyakit pasien. Langkah pertama pada proses perhitungan adalah memasukkan data latih ke dalam *database* untuk dijadikan sebagai *knowledge* pada proses perhitungan. Kemudian tenaga medis akan memasukkan data gejala yang dirasakan pasien. Data gejala tersebut telah diberi nilai skor oleh pakar sebelumnya seperti pada tabel 4.7. Selanjutnya adalah melakukan normalisasi pada data latih, setelah dilakukan normalisasi kemudian menghitung jarak terhadap data latih, hasil dari perhitungan jarak tersebut akan disorting dan dipilih sebanyak *K* data jarak. Setelah didapatkan hasil klasifikasi penyakit, lalu dicari nilai CF yaitu mengkalikan nilai gejala sesuai seperti yang dirasakan pasien (CF User) dengan nilai tiap gejala berdasarkan penyakitnya (CF Pakar). Selanjutnya cari nilai *CF Combine* untuk mengetahui nilai kepastian diagnosis penyakit dari pasien tersebut.

4.2.3.1 Perancangan Algoritma

1. Flowchart Proses *K-Nearest Neighbor – Certainty Factor*

Pada proses ini dijelaskan alur sistem dalam pemrosesan data latih dan data uji menggunakan *K-Nearest Neighbor – Certainty Factor* secara umum. Gambar 4.3 merupakan gambaran secara umum pada sistem.



Gambar 4.3 Proses *K-Nearest Neighbor – Certainty Factor*

Dari Gambar 4.5 terdapat beberapa proses pada metode *K-Nearest Neighbor – Certainty Factor*, yaitu:

a. Proses Normalisasi Data

Proses normalisasi data merupakan proses yang bertujuan untuk menyeragamkan nilai pada rentang tertentu. Proses normalisasi ini dilakukan pada setiap nilai hasil perhitungan gejala.

b. Proses *K-Nearest Neighbor*

Proses *K-Nearest Neighbor* adalah suatu proses klasifikasi terhadap data uji untuk dimasukkan kedalam kelas jenis penyakit. Proses *K-Nearest Neighbor* dilakukan dengan menghitung jarak *Euclidean* (tetangga terdekat) antara data uji dengan data latih. Kemudian dilakukan sorting dari jarak terkecil ke terbesar. Selanjutnya diambil sejumlah data sebanyak *K* data yang nantinya akan digunakan dalam perhitungan *certainty factor*.

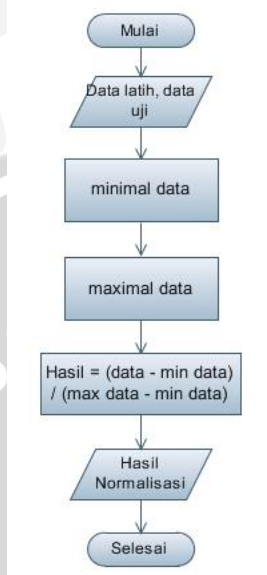
c. Proses *Certainty Factor*

Proses *Certainty Factor* adalah proses mencari nilai kepastian untuk mengambil keputusan. Setelah mendapatkan hasil klasifikasi penyakit dari proses *K-Nearest Neighbor* sebelumnya, lalu hitung nilai CF dan *CF Combine* agar dapat diketahui nilai keyakinan dari jenis penyakit tersebut.

2. *Flowchart* Proses Normalisasi Data

Pada Proses normalisasi data, langkah pertama adalah menghitung nilai minimal dan maksimal antara data uji dan data latih. Kemudian

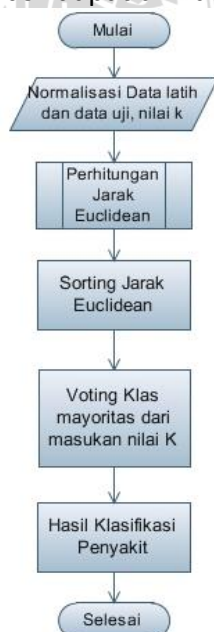
setelah proses normalisasi selesai, data uji dan data latih yang telah di normalisasi akan digunakan pada perhitungan *K-Nearest Neighbor*. Alur proses normalisasi ditunjukkan oleh Gambar 4.4.



Gambar 4.4 Proses normalisasi

3. Flowchart Proses *K-Nearest Neighbor*

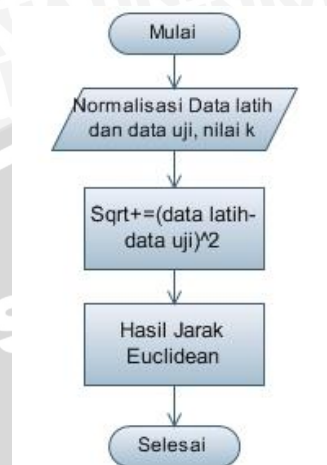
Pada Proses *K-Nearest Neighbor*, langkah pertama adalah mencari jarak *Euclidean* antara data uji dengan data latih kemudian dilakukan pengurutan data dari jarak terkecil ke terbesar. Selanjutnya diambil nilai sebanyak *K* data untuk dilakukan perhitungan *certainty factor*. Alur proses *K-Nearest Neighbor* dapat dilihat pada gambar berikut.



Gambar 4.5 Proses *K-Nearest Neighbor*

4. Flowchart Proses Euclidean

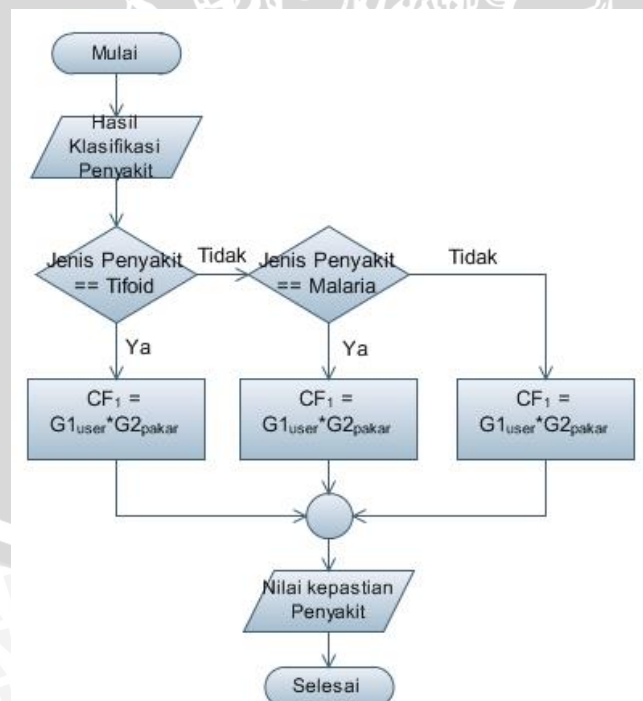
Pada Proses *euclidean*, langkah pertama adalah mencari jarak *Euclidean* antara data uji dengan data latih. Alur proses *euclidean* dapat dilihat pada gambar berikut.



Gambar 4.6 Proses Euclidean

5. Flowchart Proses Certainty Factor

Pada proses cf, langkah pertama adalah menghitung nilai CF, kemudian mencari nilai *CF Combine* untuk memperoleh nilai kepastian jenis penyakit tersebut. Alur proses *Certainty Factor* dijelaskan pada gambar berikut.



Gambar 4.7 Proses Certainty Factor

4.2.3.2 Perhitungan Manual

Pada sistem ini perhitungan manual menggunakan *K-Nearest Neighbor – Certainty Factor*. Berikut ini adalah langkah-langkah pada perhitungan *K-Nearest Neighbor – Certainty Factor*. Sebagai inputan awal, misalkan saja *user* memasukkan data gejala sebagai berikut.

Tabel 4.8 Contoh input data *user*

Gejala	Nama Gejala	Tingkatan	Nilai Bobot Gejala
G1	Demam Intermittent (putus putus)	Tidak	0
G2	Demam menggigil	Tidak	0
G3	Demam terutama malam hari	Ya	0.75
G4	Demam lebih dari 1 minggu	4-7 Hari	0.5
G5	Sakit Kepala	Berat	0.9
G6	Sakit sakit tulang dan sendi	Tidak	0
G7	Mual dan muntah	Berat	0.85
G8	Mencret atau susah BAB (konstipasi)	Ada	0.6
G9	Nyeri Perut	Sedang	0.4
G10	Bintik merah (ptekie) pada kulit	Tidak ada	0
G11	Lidah kotor (coated tongue)	Berat	0.85
G12	Bradikardi relatif	Ya	0.75
G13	Pembesaran hati	Ya	0.5
G14	Pembesaran limpa	Ya	0.65
G15	Kulit lembab/ Keringat	Tidak	0

Dari inputan *user* diatas, akan dijadikan data uji untuk mengetahui jenis penyakit yang diderita pasien tersebut. Pada tabel 4.9 ditunjukkan contoh data uji beserta nilai bobot tiap gejala yang didapat dari masukkan *user* pada tabel 4.8.

Tabel 4.9 Contoh nilai tiap gejala yang telah diinputkan *user*

No Urut Pasien	Gejala															Penyakit
	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14	G15	
25	0	0	0.75	0.5	0.9	0	0.85	0.6	0.4	0	0.85	0.75	0.5	0.65	0	TYPHOID

1. Data Penyakit Demam (DBD, Malaria dan Tifoid)

Berikut ini adalah data-data yang digunakan untuk perhitungan manual menggunakan metode *K-Nearest Neighbor – Certainty Factor*. Terdiri dari 15 data yaitu 12 data latih dan 3 data uji pada setiap kelas. Data latih berada pada no. urut pasien 10 - 109 dan data uji berada pada no. urut pasien 25 – 56, seperti yang ditunjukkan pada tabel 4.10.

Tabel 4.10 Data penyakit Demam

No Urut Pasien	Gejala															Penyakit
	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14	G15	
10	0.8	0.85	0	0.5	0.7	0	0	0.2	0	0	0	0	0	0	0.4	MALARIA
14	0	0.85	0	0.2	0.9	0	0	0.2	0	0	0	0	0	0	0.4	MALARIA
18	0	0.85	0	0.5	0.9	0	0	0.2	0	0	0	0	0	0	0.4	MALARIA
45	0.8	0.85	0	0.5	0.7	0	0	0.2	0	0	0	0	0.5	0.65	0.85	MALARIA
17	0	0	0	0.75	0.7	0.75	0.85	0.6	0	0	0.4	0.75	0.5	0.65	0	TYPHOID
110	0	0	0.75	0.75	0.9	0	0	0.6	0.4	0.7	0.4	0.75	0	0.65	0	TYPHOID
84	0	0	0	0.75	0.9	0	0.85	0.6	0	0	0	0.75	0	0	0	TYPHOID
120	0	0	0.75	0.5	0.9	0	0.85	0.2	0.8	0	0.85	0	0.5	0.65	0.45	TYPHOID
20	0	0.5	0.75	0.5	0.7	0.75	0.6	0.2	0.8	0.7	0.4	0	0.5	0.65	0	DBD
38	0	0	0	0.2	0.9	0.75	0.6	0.2	0.8	0.7	0	0	0.5	0	0	DBD
66	0	0	0	0.2	0.9	0.9	0.85	0.2	0	0.7	0.4	0	0	0	0	DBD
109	0	0	0	0.2	0.7	0.9	0.85	0.2	0.4	0.7	0	0	0	0	0	DBD
25	0	0	0.75	0.5	0.9	0	0.85	0.6	0.4	0	0.85	0.75	0.5	0.65	0	TYPHOID
57	0.8	0.5	0	0.5	0.7	0	0	0.2	0	0	0	0	0	0.65	0.4	MALARIA
56	0	0	0	0.2	0.9	0.75	0.85	0.2	0.8	1	0	0	0	0	0	DBD

2. Normalisasi

Proses normalisasi merupakan proses yang pertama dilakukan. Normalisasi dilakukan untuk menyamakan nilai pada rentang tertentu. Contoh perhitungan normalisasi menggunakan persamaan dapat dilihat pada perhitungan dibawah ini.

$$V_1 = \frac{data - \min A}{\max A - \min A}$$

$$V_{1,1} = \frac{0.8 - 0}{0.8 - 0} = 1$$

$$V_{1,2} = \frac{0.85 - 0}{0.85 - 0} = 1$$

....

$$V_{15,15} = \frac{0 - 0}{0.85 - 0} = 0$$

Untuk contoh Data Uji misalnya *user* memasukkan nilai seperti pada Tabel 4.9. Hasil pada Tabel 4.11 merupakan hasil normalisasi dari data sebelumnya yang telah dipisah menjadi data uji dan data latih.

Tabel 4.11 Data latih setelah dinormalisasi

No Urut Pasien	Gejala															Penyakit
	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14	G15	
10	1	1	0	0.54	0	0	0	0	0	0	0	0	0	0	0.47	MALARIA
14	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0.47	MALARIA
18	0	1	0	0.54	1	0	0	0	0	0	0	0	0	0	0.47	MALARIA
45	1	1	0	0.54	0	0	0	0	0	0	0	0	1	1	1	MALARIA
17	0	0	0	1	0	0.83	1	1	0	0	0.47	1	1	1	0	TYPHOID
110	0	0	1	1	1	0	0	1	0.5	0.7	0.47	1	0	1	0	TYPHOID
84	0	0	0	1	1	0	1	1	0	0	0	1	0	0	0	TYPHOID
120	0	0	1	0.54	1	0	1	0	1	0	1	0	1	1	0.52	TYPHOID
20	0	0.58	1	0.54	0	0.83	0.70	0	1	0.7	0.47	0	1	1	0	DBD
38	0	0	0	0	1	0.83	0.70	0	1	0.7	0	0	1	0	0	DBD
66	0	0	0	0	1	1	1	0	0	0.7	0.47	0	0	0	0	DBD
109	0	0	0	0	0	1	1	0	0.5	0.7	0	0	0	0	0	DBD

Tabel 4.12 Data uji setelah dinormalisasi

No Urut Pasien	Gejala															Penyakit
	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14	G15	
25	0	0	1	0.54	1	0	1	1	0.5	0	1	1	1	1	0	TYPHOID
57	1	0.58	0	0.54	0	0	0	0	0	0	0	0	0	1	0.47	MALARIA
56	0	0	0	0	1	0.83	1	0	1	1	0	0	0	0	0	DBD

3. Jarak Euclidean

Setelah data latih dan data uji di normalisasi, kemudian menghitung jarak *Euclidean* data latih terhadap data uji dengan menggunakan persamaan 2.2. Tujuannya adalah agar diketahui perbedaan antara nilai tiap atribut pada *record* data pertama dan selanjutnya. Contoh perhitungan jarak Euclidean adalah sebagai berikut:

$$d_{(1,1)} = \sqrt{(1-0)^2 + (1-0)^2 + (0-1)^2 + (0.54-0.54)^2 + (0-1)^2 + (0-0)^2 + (0-1)^2 + (0-1)^2 + (0-0.5)^2 + (0-0)^2 + (0-1)^2 + (0-1)^2 + (0-1)^2 + (0-1)^2 + (0-1)^2 + (0-0)^2} = 3.23596$$

...,dst

Pada tabel 4.13 merupakan hasil perhitungan jarak *Euclidean* untuk data uji penyakit Tifoid.

Tabel 4.13 Jarak *Euclidean* penyakit Tifoid

No Urut Pasien	Jarak	Penyakit
10	3.2359625	MALARIA
14	2.9612453	MALARIA
18	2.9105761	MALARIA
45	3.0413813	MALARIA
17	1.8523857	TYPHOID
110	1.7253662	TYPHOID
84	2.1110688	TYPHOID
120	1.5906844	TYPHOID
20	2.2687546	DBD
38	2.6112201	DBD
66	2.7051428	DBD
109	2.9643753	DBD

Pada tabel 4.14 merupakan hasil perhitungan jarak *Euclidean* untuk data uji penyakit Malaria.

Tabel 4.14 Jarak *Euclidean* penyakit Malaria

No Urut Pasien	Jarak	Penyakit
10	1.433654	MALARIA
14	2.086357	MALARIA
18	2.013793	MALARIA
45	1.825426	MALARIA
17	2.5865	TYPHOID
110	2.595292	TYPHOID
84	2.236068	TYPHOID
120	2.888948	TYPHOID
20	2.474999	DBD
38	2.747309	DBD
66	2.564459	DBD
109	2.367487	DBD

Pada tabel 4.15 merupakan hasil perhitungan jarak *Euclidean* untuk data uji penyakit DBD.

Tabel 4.15 Jarak *Euclidean* penyakit DBD

No Urut Pasien	Jarak	Penyakit
10	2.6857808	MALARIA
14	2.2171824	MALARIA
18	2.2832911	MALARIA
45	3.161007	MALARIA
17	2.8673077	TYPHOID
110	2.6936774	TYPHOID
84	2.3863035	TYPHOID
120	2.5044444	TYPHOID
20	2.2453285	DBD
38	1.0846682	DBD
66	1.1572515	DBD
109	1.1695203	DBD

4. *Sorting* hasil Jarak *Euclidean*

Setelah diketahui jarak *Euclidean* setiap *record* data latih terhadap data uji, selanjutnya dilakukan *sorting* dari jarak terkecil ke jarak terbesar dan mengambil data k, misalnya saja sebanyak 5. Pada Tabel 4.16 akan ditunjukkan hasil *sorting* pada Penyakit Tifoid, dimana pada kolom no. urut pasien 120-20 merupakan hasil *sorting* 5 jarak terdekat.

Tabel 4.16 *Sorting* jarak terdekat-terjauh (penyakit Tifoid)

No Urut Pasien	Jarak	Penyakit
120	1.5906844	TYPHOID
110	1.7253662	TYPHOID
17	1.8523857	TYPHOID
84	2.1110688	TYPHOID
20	2.2687546	DBD
38	2.6112201	DBD
66	2.7051428	DBD
18	2.9105761	MALARIA
14	2.9612453	MALARIA
109	2.9643753	DBD
45	3.0413813	MALARIA
10	3.2359625	MALARIA

Pada Tabel 4.17 akan ditunjukkan hasil *sorting* pada Penyakit Malaria, dimana pada kolom no. urut pasien 10-84 merupakan hasil *sorting* 5 jarak terdekat.

Tabel 4.17 *Sorting* jarak terdekat-terjauh (penyakit Malaria)

No Urut Pasien	Jarak	Penyakit
10	1.433654	MALARIA
45	1.825426	MALARIA
18	2.013793	MALARIA
14	2.086357	MALARIA
84	2.236068	TYPHOID
109	2.367487	DBD
20	2.474999	DBD
66	2.564459	DBD
17	2.5865	TYPHOID
110	2.595292	TYPHOID
38	2.747309	DBD
120	2.888948	TYPHOID

Pada Tabel 4.18 akan ditunjukkan hasil *sorting* pada Penyakit DBD, dimana pada kolom no. urut pasien 38-20 merupakan hasil *sorting* 5 jarak terdekat.

Tabel 4.18 *Sorting* jarak terdekat-terjauh (penyakit DBD)

No Urut Pasien	Jarak	Penyakit
38	1.0846682	DBD
66	1.1572515	DBD
109	1.1695203	DBD
14	2.2171824	MALARIA
20	2.2453285	DBD
18	2.2832911	MALARIA
84	2.3863035	TYPHOID
120	2.5044444	TYPHOID
10	2.6857808	MALARIA
110	2.6936774	TYPHOID
17	2.8673077	TYPHOID
45	3.161007	MALARIA

Dari tabel 4.16 dapat disimpulkan bahwa hasil klasifikasi adalah penyakit Tifoid, tabel 4.17 adalah penyakit Malaria dan tabel 4.18

adalah penyakit DBD. Hal ini terlihat dari jumlah terbanyak dengan permisalan $K = 5$.

5. Perhitungan *Certainty Factor*

Kemudian menentukan nilai *Certainty Factor* berdasarkan klasifikasi sebelumnya. Tabel 4.12 merupakan contoh kasus dari CF *User* yang diambil dari data uji yang telah dinormalisasi. Selain menggunakan CF *User* diperlukan nilai dari pakar yaitu berupa CF Pakar yang terdiri dari CF Pakar DBD, CF Pakar Malaria dan CF Pakar Tifoid yang akan dapat dilihat pada tabel 4.6 dimana nilai tersebut telah didapatkan dari pakar.

6. Perhitungan Nilai CF

Langkah selanjutnya adalah menghitung nilai perkalian antara CF *User* dengan CF Pakar. Pada CF *User* jika salah satu gejala bernilai 0 maka hasilnya akan selalu 0 dan tidak perlu dihitung. Contoh perkalian CF *User* dengan CF Pakar pada kelas Tifoid dapat dilihat pada perhitungan dibawah ini.

$$CF_3 = G3_{User} \times G3_{Pakar} = 1 \times 0.8 = 0.8$$

$$CF_4 = G4_{User} \times G4_{Pakar} = 0.54545 \times 0.8 = 0.43636$$

...,dst

Tabel 4.19 merupakan hasil perkalian CF *User* dengan CF Pakar (dapat dilihat pada Tabel 4.6) pada Kelas Tifoid, Malaria dan DBD.

Tabel 4.19 Hasil perkalian CF *user* dengan CF pakar

CF	Tifoid	Malaria	DBD
CF 1	0	0.9	0
CF 2	0	0.5294118	0
CF 3	0.8	0	0
CF 4	0.436364	0.3818182	0
CF 5	0.9	0	0.8
CF 6	0	0	0.833333
CF 7	0.9	0	0.7
CF 8	0.7	0	0
CF 9	0.25	0	0.9
CF 10	0	0	1
CF 11	0.9	0	0
CF 12	0.8	0	0
CF 13	0.5	0	0
CF 14	0.6	0.8	0
CF 15	0	0.4235294	0

7. Perhitungan *CF Combine*

Untuk menghitung *CF Combine* dapat digunakan persamaan 2.3. Proses perhitungan dapat dilihat seperti dibawah ini:

$$CFCombine_3 = CF_3 + (CF_4(1 - CF_3)) = 0.8 + (0.436364(1 - 0.8)) = 0.88727$$

$$CFCombine_4 = CF_{Combine_3} + (CF_4(1 - CF_{Combine_3})) = 0.88727 + (0.43636(1 - 0.88727)) = 0.93646$$

...,dst

Pada tabel 4.20 menunjukkan hasil perhitungan *CF Combine* pada penyakit Tifoid, Malaria dan DBD.

Tabel 4.20 Hasil perhitungan CF combine penyakit Tifoid, Malaria dan DBD

CF Combine	Tifoid	Malaria	DBD
Combine 1	0	0.9529412	0
Combine 2	0	0.9778547	0
Combine 3	0.8872727	0	0
Combine 4	0.9364628	0.9863102	0
Combine 5	0.9936463	0	0.966666667
Combine 6	0	0	0.994444444
Combine 7	0.9993646	0	0.998333333
Combine 8	0.9998094	0	0
Combine 9	0.999857	0	0.999833333
Combine 10	0	0	1
Combine 11	0.9999857	0	0
Combine 12	0.9999971	0	0
Combine 13	0.9999986	0	0
Combine 14	0.9999994	0.997262	0
Combine 15	0	0.9984216	0
Max Combine	0.9999994	0.9984216	1

Setelah mendapatkan hasil perhitungan dari seluruh *CF Combine*, cari nilai terbesar dari hasil *CF Combine* tersebut. Nilai terbesar dari *CF Combine* Tifoid adalah 0.9999994, *CF Combine* Malaria adalah 0.9984216 dan *CF Combine* DBD adalah 1.

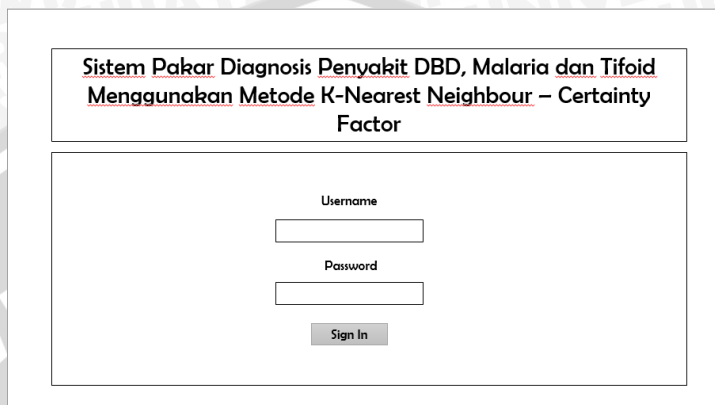
4.2.4 Antarmuka

Untuk mempermudah dalam proses implementasi program, diperlukan adanya rancangan yang menggambarkan program aplikasi yang akan dibuat ini. Secara garis besar, aplikasi Sistem Pakar Diagnosis Penyakit Demam: DBD, Malaria dan Tifoid ini terdiri dari halaman Admin dan halaman untuk tenaga medis. Untuk lebih jelasnya, dapat dilihat pada gambar-gambar berikut.

4.2.4.1 Admin

1. Halaman *Sign In*

Halaman *Sign In* merupakan halaman utama yang digunakan admin dan tenaga medis untuk masuk ke dalam aplikasi sistem pakar diagnosis penyakit Demam. Tampilan halaman *sign in* ditunjukkan oleh Gambar 4.8.



Sistem Pakar Diagnosis Penyakit DBD, Malaria dan Tifoid
Menggunakan Metode K-Nearest Neighbour – Certainty
Factor

Username

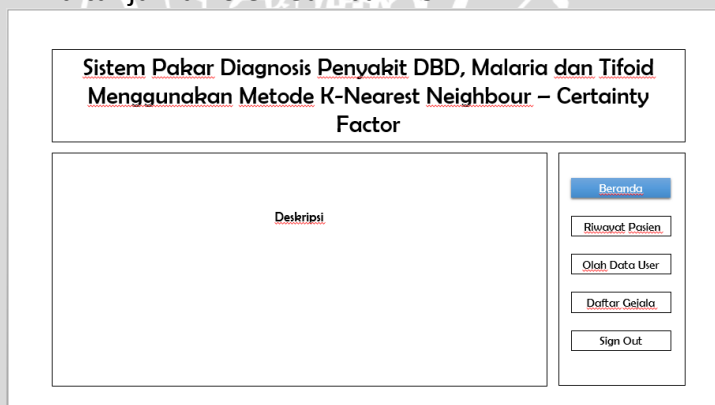
Password

Sign In

Gambar 4.8 Halaman *Sign In* admin dan tenaga medis pada system

2. Halaman Utama Admin

Halaman utama admin berisi deskripsi singkat tentang sistem, dan menu-menu yang dapat diakses oleh admin yaitu olah data *user*, daftar gejala, bobot gejala, riwayat pasien dan *sign out*. Tampilan halaman utama admin ditunjukkan oleh Gambar 4.9.



Sistem Pakar Diagnosis Penyakit DBD, Malaria dan Tifoid
Menggunakan Metode K-Nearest Neighbour – Certainty
Factor

Deskripsi

Beranda
Riwayat Pasien
Olah Data User
Daftar Gejala
Sign Out

Gambar 4.9 Halaman utama admin

3. Halaman Riwayat Pasien

Pada Halaman Riwayat Pasien, admin dapat melihat daftar riwayat penyakit yang diderita oleh pasien. Tampilan halaman riwayat pasien ditunjukkan pada Gambar 4.10.

**Sistem Pakar Diagnosis Penyakit DBD, Malaria dan Tifoid
Menggunakan Metode K-Nearest Neighbour – Certainty
Factor**

RIWAYAT PASIEN

No	ID Pasien	Nama Pasien	Alamat	No HP	Tamagol Periksa	Hasil Diagnosis

[Beranda](#)

[Riwayat Pasien](#)

[Olah Data User](#)

[Daftar Gejala](#)

[Sign Out](#)

Gambar 4.10 Halaman riwayat pasien

4. Halaman Olah Data User

Halaman Olah Data User merupakan halaman yang digunakan admin untuk mengedit, menambah maupun menghapus user. Tampilan halaman olah data user ditunjukkan pada Gambar 4.11.

**Sistem Pakar Diagnosis Penyakit DBD, Malaria dan Tifoid
Menggunakan Metode K-Nearest Neighbour – Certainty
Factor**

OLAH DATA USER

[Tambah User](#)



Pakar 1

[Edit](#)
[Hapus](#)



Pakar 2

[Edit](#)
[Hapus](#)



User 1

[Edit](#)
[Hapus](#)



User 2

[Edit](#)
[Hapus](#)

[Beranda](#)

[Riwayat Pasien](#)

[Olah Data User](#)

[Daftar Gejala](#)

[Sign Out](#)

Gambar 4.11 Halaman olah data user

5. Halaman Daftar Gejala

Halaman Daftar Gejala merupakan halaman yang digunakan oleh admin untuk mengedit, menambah maupun menghapus daftar gejala. Tampilan halaman daftar gejala ditunjukkan oleh Gambar 4.12.

**Sistem Pakar Diagnosis Penyakit DBD, Malaria dan Tifoid
Menggunakan Metode K-Nearest Neighbour – Certainty
Factor**

DAFTAR GEJALA

Nama Gejala

Tingkat Gejala

Nilai

[Simpan](#)

[Beranda](#)

[Riwayat Pasien](#)

[Olah Data User](#)

[Daftar Gejala](#)

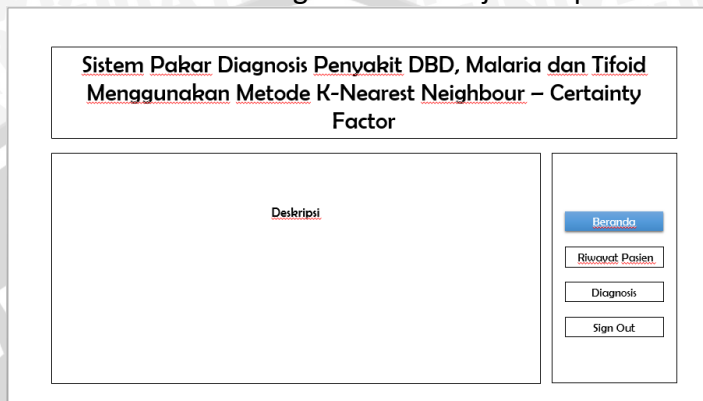
[Sign Out](#)

Gambar 4.12 Halaman daftar gejala

4.2.4.2 Tenaga Medis

1. Halaman Beranda

Halaman Beranda merupakan halaman utama yang digunakan oleh tenaga medis setelah melakukan *sign in*. Tenaga Medis hanya dapat mengakses menu riwayat pasien, diagnosis dan melakukan *sign out*. Tampilan halaman beranda tenaga medis ditunjukkan pada Gambar 4.13.



Sistem Pakar Diagnosis Penyakit DBD, Malaria dan Tifoid
Menggunakan Metode K-Nearest Neighbour - Certainty
Factor

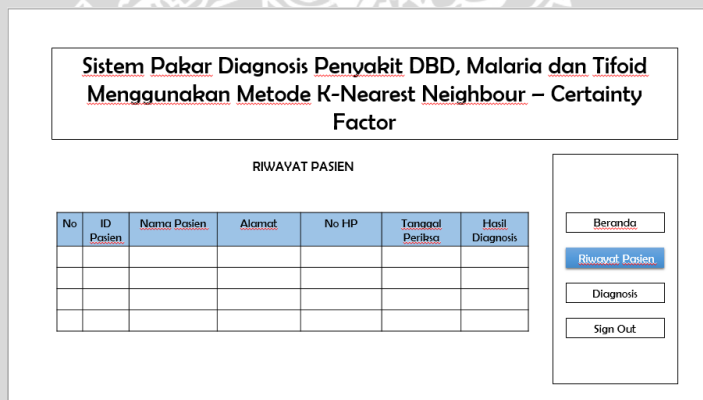
Deskripsi

Beranda
Riwayat Pasien
Diagnosis
Sign Out

Gambar 4.13 Halaman beranda tenaga medis

2. Halaman Riwayat Pasien

Tampilan halaman riwayat pasien ditunjukkan oleh Gambar 4.14.



Sistem Pakar Diagnosis Penyakit DBD, Malaria dan Tifoid
Menggunakan Metode K-Nearest Neighbour - Certainty
Factor

RIWAYAT PASIEN

No	ID Pasien	Nama Pasien	Alamat	No HP	Tanggal Periksa	Hasil Diagnosis

Beranda
Riwayat Pasien
Diagnosis
Sign Out

Gambar 4.14 Halaman riwayat pasien

3. Halaman Diagnosis

Pada Halaman Diagnosis, tenaga medis dapat menginputkan data pasien dan gejala pasien. Tampilan halaman diagnosis ditunjukkan oleh Gambar 4.15.

**Sistem Pakar Diagnosis Penyakit DBD, Malaria dan Tifoid
Menggunakan Metode K-Nearest Neighbor – Certainty
Factor**

DIAGNOSIS

Tanggal Periksa

Nama Pasien

Alamat

No HP

A. Keluhan

1. Demam Intermiten ☐ Ya ☒ Tidak

B. Pemeriksaan Fisik

2. Bintik Merah ☒ Berat ☐ Sedang ☐ Tidak

Beranda

Riwayat Pasien

Gambar 4.15 Halaman diagnosis tenaga medis

4.2.5 Workplace

Workplace adalah kumpulan memori kerja untuk menyimpan fakta-fakta yang diketahui. *Workplace* dalam penelitian ini adalah *database* yang digunakan sebagai tempat penyimpanan sementara data pasien, dan gejala-gejala penyakit Demam (DBD, Malaria dan Tifoid) yang telah diinputkan sebelumnya, kemudian data tersebut akan diolah oleh mesin inferensi menggunakan perhitungan *K-Nearest Neighbor – Certainty Factor* hingga akhirnya mendapatkan hasil diagnosis penyakit pasien.

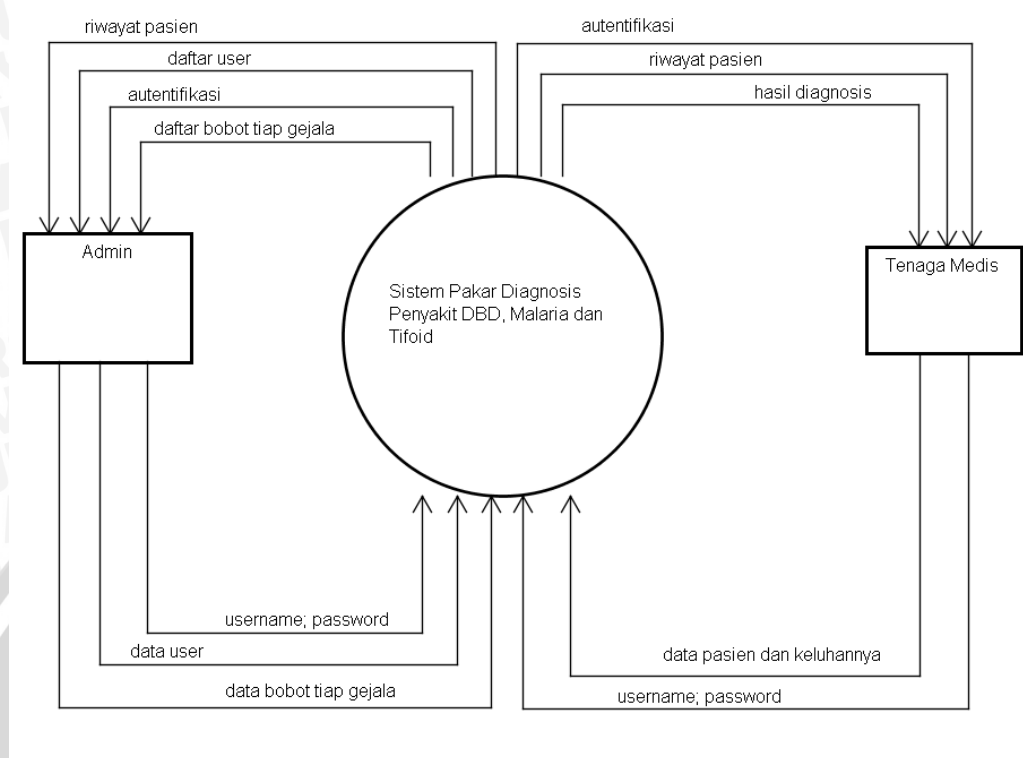
4.2.6 Fasilitas Penjelas

Fasilitas penjelas dalam sistem ini akan menjawab pertanyaan bagaimana kesimpulan hasil diagnosis yang telah dicapai dengan menampilkan penjelasan perhitungan sampai di dapatkan kesimpulan hasil diagnosis dari gejala-gejala yang sebelumnya dimasukkan.

4.3 Perancangan *Database*

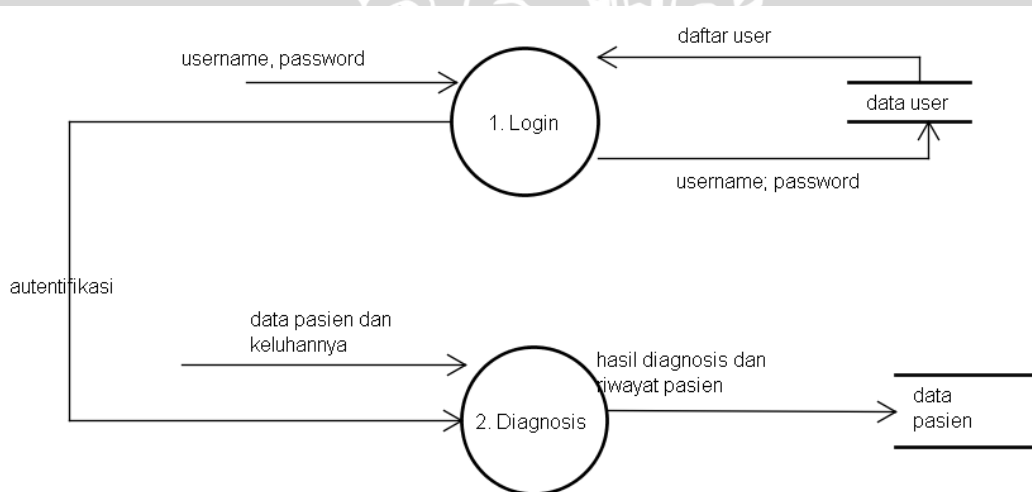
4.3.1 Data Flow Diagram

Langkah awal membuat DFD adalah mengidentifikasi semua entitas luar yang terlibat di sistem. Entitas luar pada sistem ini adalah pengguna sistem. Diagram konteks adalah DFD yang menggambarkan hubungan antar sistem dengan lingkungan luarnya (entitas luar). Diagram konteks atau diagram level 0 menunjukkan sebuah proses utama yang terjadi pada sistem pakar. Gambar 4.16 menunjukkan DFD Level 0 untuk menampilkan semua proses utama dari sistem dan bagaimana setiap proses dapat saling terkait.



Gambar 4.16 DFD Level 0

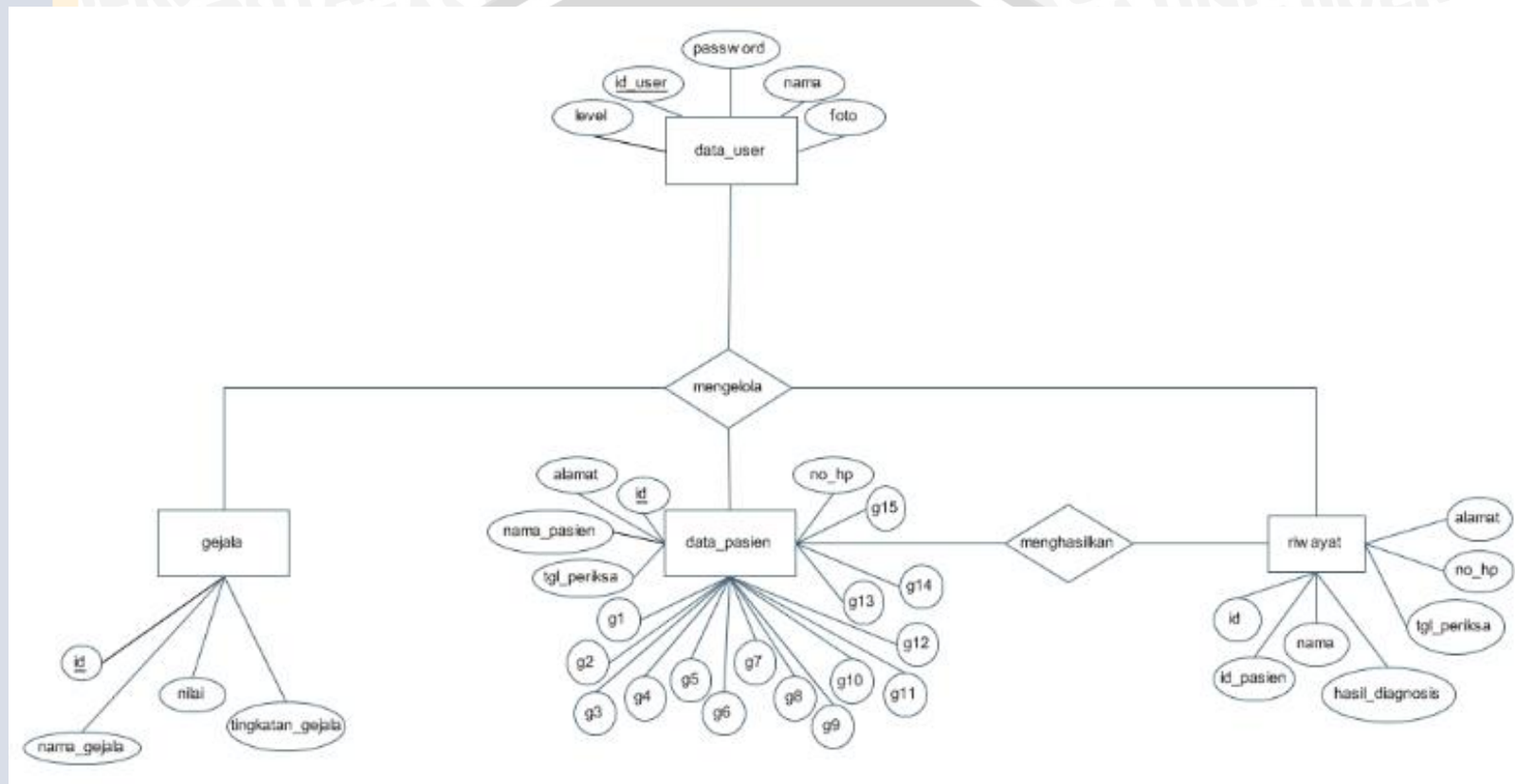
Setiap proses pada DFD Level 0 dapat diuraikan ke dalam DFD yang lebih eksplisit, yang disebut dengan diagram tingkat 1 atau DFD level 1, yang menunjukkan bagaimana sebuah sistem beroperasi secara lebih rinci. DFD Level 1 dapat dilihat pada Gambar 4.17 berikut.



Gambar 4.17 DFD Level 1

4.3.2 Entity Relationship Diagram

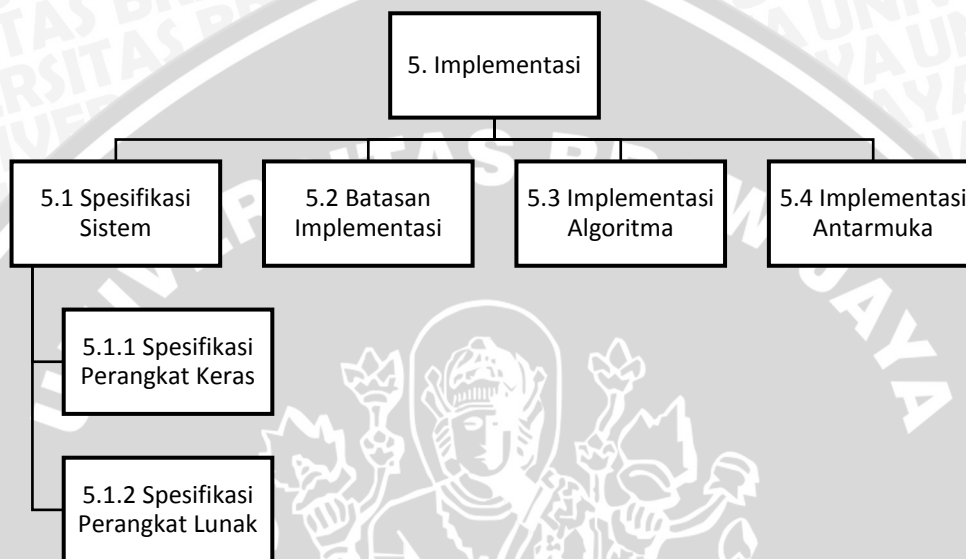
Pada bagian ini akan digambarkan relasi antar entitas dan juga atribut yang diperlukan dalam perancangan sistem pakar ini. Relasi antar entitas akan digambarkan menggunakan ERD sebagai berikut.



Gambar 4.18 ER Diagram

BAB 5 IMPLEMENTASI

Bab ini menjelaskan tentang implementasi perangkat lunak berdasarkan hasil yang telah diperoleh dari proses perancangan *user interface* yang dibuat. Pembahasan meliputi penjelasan mengenai spesifikasi sistem, batasan implementasi, implementasi algoritma, dan implementasi antarmuka. Pohon implementasi dapat dilihat pada Gambar 5.1.



Gambar 5.1 Pohon implementasi

5.1 Spesifikasi Sistem

Hasil perancangan pada bab 4 menjadi acuan untuk melakukan implementasi menjadi sistem. Spesifikasi sistem diimplementasikan pada spesifikasi perangkat keras dan perangkat lunak.

5.1.1 Spesifikasi Perangkat Keras

Spesifikasi perangkat keras pada Sistem Pakar Diagnosis Penyakit Demam: DBD, Malaria dan Tifoid Menggunakan Metode *K-Nearest Neighbor – Certainty Factor* menggunakan komputer yang dijelaskan pada Tabel 5.1.

Tabel 5.1 Spesifikasi perangkat keras

Nama Komponen	Spesifikasi
Processor	Intel® Core™ i5-4200U CPU @ 1.60GHz 2.30GHz
Memori (RAM)	4GB RAM
Kartu Grafis	Intel®HD Graphic Family
Harddisk	500 GB HDD

5.1.2 Spesifikasi Perangkat Lunak

Pada penelitian ini akan dibuat program penerapan algoritma *K-Nearest Neighbor – Certainty Factor* menggunakan sebuah komputer dengan spesifikasi perangkat lunak yang dijelaskan pada Tabel 5.2.

Tabel 5.2 Spesifikasi perangkat lunak

Sistem Operasi	<i>Microsoft Windows 8</i>
Bahasa Pemrograman	<i>PHP</i>
Tools Pemrograman	<i>XAMPP v.3.2.1</i>
Browser	<i>Mozilla Firefox</i>
DBMS	<i>MySQL</i>

5.2 Batasan Implementasi

Batasan implementasi pada Sistem Pakar Diagnosis Penyakit Demam: DBD, Malaria dan Tifoid Menggunakan Metode *K-Nearest Neighbor – Certainty Factor* adalah sebagai berikut:

1. Sistem yang dibuat menggunakan bahasa pemrograman *PHP* dan berbasis web.
2. Data-data yang digunakan pada sistem disimpan dalam *database MySQL*.
3. Admin dapat menambah data gejala dan data bobot sesuai dengan pengetahuan pakar, namun tidak dapat menambah data penyakit karena dalam penelitian ini hanya ada 3 jenis penyakit yaitu DBD, Malaria dan Tifoid.
4. Data pasien dan gejala-gejala yang dirasakan oleh pasien merupakan input yang diterima oleh sistem dan data tersebut akan dimasukkan oleh tenaga medis.
5. Proses perhitungan menggunakan metode *K-Nearest Neighbor – Certainty Factor* akan ditampilkan apabila tenaga medis menekan tombol Lihat Perhitungan, karena metode *K-Nearest Neighbor – Certainty Factor* hanya sebagai fasilitas penjelas.
6. Hasil diagnosis dapat dilihat pada menu riwayat apabila tenaga medis menekan tombol Simpan ke Riwayat.
7. *Output* yang diterima adalah nama pasien, alamat pasien, no hp pasien, dan hasil diagnosis penyakit.

5.3 Implementasi Algoritma

Pada pembuatan program penerapan algoritma *K-Nearest Neighbor – Certainty Factor* ini memiliki beberapa proses pada implementasinya, yaitu normalisasi data latih, normalisasi data input, *K-Nearest Neighbor (euclidean dan sorting)* dan *Certainty Factor*.

5.3.1 Implementasi Algoritma Normalisasi Data Latih

Proses normalisasi data digunakan untuk proses normalisasi data input atau data uji yang nantinya akan digunakan pada tahapan proses berikutnya. Tahapan proses membaca file ditunjukkan pada *source code* 5.1.

```

1  <?php
2      $truncate = mysql_query("Truncate table
3  normal_data_latih");
4      $query = ('Select * from data_latih');
5      $hasil = mysql_query($query);
6      $id = 1;
7
8      while ($row = mysql_fetch_array($hasil)) {
9          $g1[$id] = $row['g1'];
10         $g2[$id] = $row['g2'];
11         $g3[$id] = $row['g3'];
12         $g4[$id] = $row['g4'];
13         $g5[$id] = $row['g5'];
14         $g6[$id] = $row['g6'];
15         $g7[$id] = $row['g7'];
16         $g8[$id] = $row['g8'];
17         $g9[$id] = $row['g9'];
18         $g10[$id] = $row['g10'];
19         $g11[$id] = $row['g11'];
20         $g12[$id] = $row['g12'];
21         $g13[$id] = $row['g13'];
22         $g14[$id] = $row['g14'];
23         $g15[$id] = $row['g15'];
24         $diagnosis[$id] = $row['diagnosis'];
25         $id++;
26     }
27
28     // mencari nilai minimum dan nilai maksimum dari
29     tiap gejala
30     $ming1 = min($g1);
31     $ming2 = min($g2);
32     $ming3 = min($g3);
33     $ming4 = min($g4);
34     $ming5 = min($g5);
35     $ming6 = min($g6);
36     $ming7 = min($g7);
37     $ming8 = min($g8);
38     $ming9 = min($g9);
39     $ming10 = min($g10);
40     $ming11 = min($g11);
41     $ming12 = min($g12);
42     $ming13 = min($g13);
43     $ming14 = min($g14);
44     $ming15 = min($g15);
45     $maxg1 = max($g1);
46     $maxg2 = max($g2);
47     $maxg3 = max($g3);
48     $maxg4 = max($g4);
49
50     $maxg5 = max($g5);
51     $maxg6 = max($g6);
52     $maxg7 = max($g7);
53     $maxg8 = max($g8);
54     $maxg9 = max($g9);
55     $maxg10 = max($g10);
56     $maxg11 = max($g11);
57     $maxg12 = max($g12);
58     $maxg13 = max($g13);

```

```

59         $maxg14 = max($g14);
60         $maxg15 = max($g15);
61
62         for ($i = 1; $i < $id; $i++) { // rumus
63 normalisasi data latih
64             $g1normal[$i] = ($g1[$i] - $ming1) /
65 ($maxg1 - $ming1);
66             $g2normal[$i] = ($g2[$i] - $ming2) /
67 ($maxg2 - $ming2);
68             $g3normal[$i] = ($g3[$i] - $ming3) /
69 ($maxg3 - $ming3);
70             $g4normal[$i] = ($g4[$i] - $ming4) /
71 ($maxg4 - $ming4);
72             $g5normal[$i] = ($g5[$i] - $ming5) /
73 ($maxg5 - $ming5);
74             $g6normal[$i] = ($g6[$i] - $ming6) /
75 ($maxg6 - $ming6);
76             $g7normal[$i] = ($g7[$i] - $ming7) /
77 ($maxg7 - $ming7);
78             $g8normal[$i] = ($g8[$i] - $ming8) /
79 ($maxg8 - $ming8);
80             $g9normal[$i] = ($g9[$i] - $ming9) /
81 ($maxg9 - $ming9);
82             $g10normal[$i] = ($g10[$i] - $ming10) /
83 ($maxg10 - $ming10);
84             $g11normal[$i] = ($g11[$i] - $ming11) /
85 ($maxg11 - $ming11);
86             $g12normal[$i] = ($g12[$i] - $ming12) /
87 ($maxg12 - $ming12);
88             $g13normal[$i] = ($g13[$i] - $ming13) /
89 ($maxg13 - $ming13);
90             $g14normal[$i] = ($g14[$i] - $ming14) /
91 ($maxg14 - $ming14);
92             $g15normal[$i] = ($g15[$i] - $ming15) /
93 ($maxg15 - $ming15);
94
95             // masukkan data latih yang telah di
96 normalisasi ke dalam tabel normal_data latih
97             $masuk = mysql_query("INSERT INTO
98 normal_data_latih VALUES
99 ('','.$g1normal[$i].','.$g2normal[$i].',
100
101 '.$g3normal[$i].','.$g4normal[$i].','.$g5normal[$i].','.$
102 g6normal[$i].','.$g7normal[$i].',
103
104 '.$g8normal[$i].','.$g9normal[$i].','.$g10normal[$i].','.$
105 $g11normal[$i].','.$g12normal[$i].',
106 '.$g13normal[$i].','.$g14normal[$i].','.$g15normal[$i].','.'
107 '.$diagnosis[$i].'"));
108             mysql_query($masuk);
109         }
110     }
111 }

```

Source code 5.1 Normalisasi Data Latih

Source code 5.1 adalah potongan algoritma untuk melakukan normalisasi dari data latih yang dimasukkan.

5.3.2 Implementasi Algoritma Normalisasi Data Input

Proses normalisasi data digunakan untuk proses normalisasi data input atau data uji yang nantinya akan digunakan pada tahapan proses berikutnya. *Source code* proses normalisasi data ditunjukkan pada *source code* 5.2.

```

1  <?php
2      $truncate = mysql_query("Truncate table normal_data_input");
3      $query = ('Select * from data_input');
4      $hasil = mysql_query($query);
5      $id = 1;
6
7      while ($row = mysql_fetch_array($hasil)) {
8          $g1[$id] = $row['g1'];
9          $g2[$id] = $row['g2'];
10         $g3[$id] = $row['g3'];
11         $g4[$id] = $row['g4'];
12         $g5[$id] = $row['g5'];
13         $g6[$id] = $row['g6'];
14         $g7[$id] = $row['g7'];
15         $g8[$id] = $row['g8'];
16         $g9[$id] = $row['g9'];
17         $g10[$id] = $row['g10'];
18         $g11[$id] = $row['g11'];
19         $g12[$id] = $row['g12'];
20         $g13[$id] = $row['g13'];
21         $g14[$id] = $row['g14'];
22         $g15[$id] = $row['g15'];
23         $id++;
24     }
25
26     // mencari nilai minimum dan nilai maksimum dari
27     tiap gejala
28     $ming1 = min($g1);
29     $ming2 = min($g2);
30     $ming3 = min($g3);
31     $ming4 = min($g4);
32     $ming5 = min($g5);
33     $ming6 = min($g6);
34     $ming7 = min($g7);
35     $ming8 = min($g8);
36     $ming9 = min($g9);
37     $ming10 = min($g10);
38     $ming11 = min($g11);
39     $ming12 = min($g12);
40     $ming13 = min($g13);
41     $ming14 = min($g14);
42     $ming15 = min($g15);
43     $maxg1 = max($g1);
44     $maxg2 = max($g2);
45     $maxg3 = max($g3);
46     $maxg4 = max($g4);
47     $maxg5 = max($g5);
48     $maxg6 = max($g6);
49     $maxg7 = max($g7);
50     $maxg8 = max($g8);
51     $maxg9 = max($g9);
52     $maxg10 = max($g10);
53     $maxg11 = max($g11);
54     $maxg12 = max($g12);
55     $maxg13 = max($g13);
56     $maxg14 = max($g14);
57     $maxg15 = max($g15);
58

```



```

59         for ($i = 1; $i < $id; $i++) { // rumus
60     normalisasi data input
61         $g1normal[$i] = ($g1[$i] - $ming1) /
62     ($maxg1 - $ming1);
63         $g2normal[$i] = ($g2[$i] - $ming2) /
64     ($maxg2 - $ming2);
65         $g3normal[$i] = ($g3[$i] - $ming3) /
66     ($maxg3 - $ming3);
67         $g4normal[$i] = ($g4[$i] - $ming4) /
68     ($maxg4 - $ming4);
69         $g5normal[$i] = ($g5[$i] - $ming5) /
70     ($maxg5 - $ming5);
71         $g6normal[$i] = ($g6[$i] - $ming6) /
72     ($maxg6 - $ming6);
73         $g7normal[$i] = ($g7[$i] - $ming7) /
74     ($maxg7 - $ming7);
75         $g8normal[$i] = ($g8[$i] - $ming8) /
76     ($maxg8 - $ming8);
77         $g9normal[$i] = ($g9[$i] - $ming9) /
78     ($maxg9 - $ming9);
79         $g10normal[$i] = ($g10[$i] - $ming10) /
80     ($maxg10 - $ming10);
81         $g11normal[$i] = ($g11[$i] - $ming11) /
82     ($maxg11 - $ming11);
83         $g12normal[$i] = ($g12[$i] - $ming12) /
84     ($maxg12 - $ming12);
85         $g13normal[$i] = ($g13[$i] - $ming13) /
86     ($maxg13 - $ming13);
87         $g14normal[$i] = ($g14[$i] - $ming14) /
88     ($maxg14 - $ming14);
89         $g15normal[$i] = ($g15[$i] - $ming15) /
90     ($maxg15 - $ming15);
91
92         // masukkan data input yang telah di
93     normalisasi ke dalam tabel normal_data_input
94         $masuk = mysql_query ("INSERT INTO
95     normal_data_input VALUES
96     ('$g1normal[$i].','.$g2normal[$i].','
97
98     '$g3normal[$i].','.$g4normal[$i].','.$g5normal[$i].','.$
99     g6normal[$i].','.$g7normal[$i].','
100
101     '$g8normal[$i].','.$g9normal[$i].','.$g10normal[$i].','.$
102     $g11normal[$i].','.$g12normal[$i].','
103
104     '$g13normal[$i].','.$g14normal[$i].','.$g15normal[$i].')")
105     );
106         mysql_query($masuk);
107         if (!$masuk) {
108             echo 'tidak berhasil menyimpan';
109         }
110     }
111 }
112
113 ?>

```

Source code 5.2 Normalisasi Data Input

Source code 5.2 adalah potongan algoritma untuk melakukan normalisasi dari data input (data uji) yang dimasukkan.

5.3.3 Implementasi Algoritma K-Nearest Neighbor

Proses *K-Nearest Neighbor* ini digunakan untuk melakukan perhitungan jarak kedekatan data uji terhadap data latih yang dalam melakukan perhitungan jarak ini menggunakan *euclidean distance*. *Source code K-Nearest Neighbor* meliputi perhitungan jarak *euclidean* dan *sorting* jarak terkecil ke terbesar yang ditunjukkan pada *source code 5.3*.

```

1  <!-- PROSES K-NEAREST NEIGHBOR -->
2  <?php
3      $truncate = mysql_query("TRUNCATE table jarak");
4      $query1 = 'Select * from normal_data_latih';
5      $query2 = 'Select * from normal_data_input';
6
7      $hasil1 = mysql_query($query1);
8      $hasil2 = mysql_query($query2);
9
10     $id = 1;
11     while ($row = mysql_fetch_array($hasil1)) {
12         $g1latih[$id] = $row['g1'];
13         $g2latih[$id] = $row['g2'];
14         $g3latih[$id] = $row['g3'];
15         $g4latih[$id] = $row['g4'];
16         $g5latih[$id] = $row['g5'];
17         $g6latih[$id] = $row['g6'];
18         $g7latih[$id] = $row['g7'];
19         $g8latih[$id] = $row['g8'];
20         $g9latih[$id] = $row['g9'];
21         $g10latih[$id] = $row['g10'];
22         $g11latih[$id] = $row['g11'];
23         $g12latih[$id] = $row['g12'];
24         $g13latih[$id] = $row['g13'];
25         $g14latih[$id] = $row['g14'];
26         $g15latih[$id] = $row['g15'];
27         $diagnosislatih[$id] = $row['diagnosis'];
28         $id++;
29     }
30     $id = 1;
31     while ($row = mysql_fetch_array($hasil2)) {
32         $g1input[$id] = $row['g1'];
33         $g2input[$id] = $row['g2'];
34         $g3input[$id] = $row['g3'];
35         $g4input[$id] = $row['g4'];
36         $g5input[$id] = $row['g5'];
37         $g6input[$id] = $row['g6'];
38         $g7input[$id] = $row['g7'];
39         $g8input[$id] = $row['g8'];
40         $g9input[$id] = $row['g9'];
41         $g10input[$id] = $row['g10'];
42         $g11input[$id] = $row['g11'];
43         $g12input[$id] = $row['g12'];
44         $g13input[$id] = $row['g13'];
45         $g14input[$id] = $row['g14'];
46         $g15input[$id] = $row['g15'];
47         $id++;
48     }
49
50     for ($i = 1; $i < 101; $i++) {
51         $jarak[$i] = sqrt(pow($g1input[1] - $g1latih[$i],
52 2)
53         + pow($g2input[1] - $g2latih[$i], 2) +
54         pow($g3input[1] - $g3latih[$i], 2)
55

```



```

56         + pow($g4input[1] - $g4latih[$i], 2) +
57     pow($g5input[1] - $g5latih[$i], 2)
58         + pow($g6input[1] - $g6latih[$i], 2) +
59     pow($g7input[1] - $g7latih[$i], 2)
60         + pow($g8input[1] - $g8latih[$i], 2) +
61     pow($g9input[1] - $g9latih[$i], 2)
62         + pow($g10input[1] - $g10latih[$i], 2) +
63     pow($g11input[1] - $g11latih[$i], 2)
64         + pow($g12input[1] - $g12latih[$i], 2) +
65     pow($g13input[1] - $g13latih[$i], 2)
66         + pow($g14input[1] - $g14latih[$i], 2) +
67     pow($g15input[1] - $g15latih[$i], 2));
68
69     $query = "INSERT INTO jarak (hasil_jarak, diagnosis)
70     VALUES ('".$jarak[$i]."', '".$diagnosislatih[$i]."'");
71
72     mysql_query($query);
73
74     }
75
76     ?>
77
78     <!-- FUNGSI PENGURUTAN -->
79     <!-- PENGURUTAN DARI NILAI TERKECIL -->
80     <?php
81         function insertionSort(array $entering) {
82             $length = count($entering);
83             for ($i = 0; $i < ($length + 1); $i++) {
84                 $index_e[$i] = $i;
85             }
86             for ($i = 0; $i < $length; $i++) {
87                 $temp_e = $entering[$i];
88                 $temp_index_e = $index_e[$i];
89                 $j = $i;
90                 while ($j > 0 && $entering[$j - 1] > $temp_e)
91                 {
92                     //move value to right and key to previous
93                     smaller index
94                     $entering[$j] = $entering[$j - 1];
95                     $index_e[$j] = $index_e[$j - 1];
96                     $j = $j - 1;
97                 }
98                 //put the element at index $j
99                 $entering[$j] = $temp_e;
100                 $index_e[$j] = $temp_index_e;
101             }
102             return $index_e;
103         }
104     ?>
105
106     <!-- SORTING NILAI JARAK
107     <?php
108         $truncate = mysql_query("TRUNCATE table sorting_jarak");
109         $query = ('Select * from jarak');
110         $hasil = mysql_query($query);
111
112         $sid = 0;
113         while ($row = mysql_fetch_array($hasil)) {
114             $sid_jarak[$sid] = $row[0]; // memanggil id_jarak
115             $hasil_jarak[$sid] = $row[1]; // memanggil hasil jarak
116             $target[$sid] = $row[2]; // memanggil target kelas
117         }
118         penyakit
119         $sid++;

```



```

119     }
120     $sid_j = insertionSort($hasil_jarak);
121
122     for ($i = 0; $i < 100; $i++) {
123         $sid1 [$i] = $sid_jarak[$sid_j[$i]];
124         $jarak [$i] = $hasil_jarak[$sid_j[$i]];
125         $diagnosis [$i] = $target[$sid_j[$i]];
126         $query = "INSERT INTO sorting_jarak
127 (id_jarak, hasil_jarak, diagnosis) VALUES
128 ('".$sid1[$i]."', '".$jarak[$i]."', '".$diagnosis[$i]."'");
129         mysql_query($query);
130     }
131
132     ?>
133
134 <!-- AMBIL SEBANYAK K TETANGGA DENGAN JARAK TERDEKAT -->
135 <?php
136     $truncate = mysql_query("TRUNCATE table k_jarak");
137     $query = 'Select * from sorting_jarak';
138     $hasil = mysql_query($query);
139
140     $sid = 0;
141     while ($row = mysql_fetch_array($hasil)) {
142         $datajarak0[$sid] = $row[0];
143         $datajarak1[$sid] = $row[1];
144         $datajarak2[$sid] = $row[2];
145         $datajarak3[$sid] = $row[3];
146         $sid++;
147     }
148     for ($i = 0; $i < $k_klasifikasi; $i++) {
149         $sid_jarak = $datajarak1[$i];
150         $hasil_jarak = $datajarak2[$i];
151         $diagnosis = $datajarak3[$i];
152
153         $query = "INSERT INTO k_jarak (id_jarak, hasil_jarak,
154 diagnosis) VALUES
155 ('".$sid_jarak."', '".$hasil_jarak."', '".$diagnosis."')";
156         mysql_query($query);
157     }
158
159     ?>

```

Source code 5.3 K-Nearest Neighbor

Source code 5.3 merupakan fungsi perhitungan algoritma *K-Nearest Neighbor* yang didalamnya terdapat proses pencarian jarak tetangga terdekat dan proses pengurutan nilai jarak.

5.3.4 Implementasi Algoritma *Certainty Factor*

Proses *Certainty Factor* ini digunakan untuk melakukan perhitungan nilai kepastian setelah didapatkan hasil klasifikasi dari perhitungan *K-Nearest Neighbor*. Source code *Certainty Factor* dapat dilihat pada source code 5.4.

```

1  <?php
2  $truncate = mysql_query("Truncate table cf");
3  $sql=mysql_query("select * from normal_data_input");
4  $data=mysql_fetch_array($sql);
5
6  //MENGAMBIL DATA DARI TABLE NORMAL_DATA_INPUT
7  $g1=$data['g1'];
8  $g2=$data['g2'];
9  $g3=$data['g3'];
10 $g4=$data['g4'];
11 $g5=$data['g5'];
12 $g6=$data['g6'];
13 $g7=$data['g7'];
14 $g8=$data['g8'];
15 $g9=$data['g9'];
16 $g10=$data['g10'];
17 $g11=$data['g11'];
18 $g12=$data['g12'];
19 $g13=$data['g13'];
20 $g14=$data['g14'];
21 $g15=$data['g15'];
22
23 $sql1=mysql_query("select count(diagnosis) as jumlah from
24 k_jarak where diagnosis='1'");
25 //menghitung jumlah diagnosis yang bernilai 1 atau tifoid
26 $sql2=mysql_query("select count(diagnosis) as jumlah from
27 k_jarak where diagnosis='2'");
28 //menghitung jumlah diagnosis yang bernilai 2 atau malaria
29 $sql3=mysql_query("select count(diagnosis) as jumlah from
30 k_jarak where diagnosis='3'");
31 //menghitung jumlah diagnosis yang bernilai 3 atau dbd
32 $hasil1=mysql_fetch_array($sql1);
33 $hasil2=mysql_fetch_array($sql2);
34 $hasil3=mysql_fetch_array($sql3);
35
36 if ($hasil1 > $hasil2 && $hasil1 > $hasil3) {
37
38     $sql=mysql_query("select * from bobot_tifoid");
39     $data=mysql_fetch_array($sql);
40
41     //MENGAMBIL DATA DARI TABLE NORMAL_DATA_INPUT(CF USER) KEMUDIAN
42     DIKALIKAN
43     //DENGAN DATA PADA TABLE BOBOT_TIFOID(CF PAKAR)
44     $cf[1]=$data['g1']*$g1;
45     $cf[2]=$data['g2']*$g2;
46     $cf[3]=$data['g3']*$g3;
47     $cf[4]=$data['g4']*$g4;
48     $cf[5]=$data['g5']*$g5;
49     $cf[6]=$data['g6']*$g6;
50     $cf[7]=$data['g7']*$g7;
51     $cf[8]=$data['g8']*$g8;
52     $cf[9]=$data['g9']*$g9;
53     $cf[10]=$data['g10']*$g10;
54     $cf[11]=$data['g11']*$g11;
55     $cf[12]=$data['g12']*$g12;
56     $cf[13]=$data['g13']*$g13;
57     $cf[14]=$data['g14']*$g14;
58     $cf[15]=$data['g15']*$g15;
59
60     //proses CF combine
61     //kondisi untuk CF Combine 1
62     if($cf[1] == 0){
63         $cf_combine[1]=0;
64     2 }elseif($cf[1] > 0 AND $cf[2] > 0){

```



```

65         $cf_combine[1]=$cf[1]+$cf[2]*(1-$cf[1]);
66     }elseif($cf[1] > 0 AND $cf[2] == 0 AND $cf[3] > 0){
67         $cf_combine[1]=$cf[1]+$cf[3]*(1-$cf[1]);
68     }elseif($cf[1] > 0 AND $cf[2] == 0 AND $cf[3] == 0
69     AND $cf[4] > 0){
70         $cf_combine[1]=$cf[1]+$cf[4]*(1-$cf[1]);
71     }
72
73     //kondisi untuk CF Combine 2
74     if($cf[2] == 0){
75         $cf_combine[2]=0;
76     }elseif($cf_combine[1] > 0 AND $cf[2] > 0){
77         $cf_combine[2]=$cf_combine[1]+$cf[2]*(1-$cf_combine[1]);
78     }elseif($cf_combine[1] == 0 AND $cf[2] > 0
79     AND $cf[3] > 0){
80         $cf_combine[2]=$cf[2]+$cf[3]*(1-$cf[2]);
81     }elseif($cf_combine[1] == 0 AND $cf[2] > 0
82     AND $cf[3] == 0 AND $cf[4] > 0){
83
84         //kondisi untuk CF Combine 3
85         if($cf[3] == 0){
86             $cf_combine[3] = 0;
87         }elseif($cf_combine[1] > 0 AND $cf[2] == 0
88         AND $cf[3] > 0){
89             $cf_combine[3]=$cf_combine[1]+$cf[3]*(1-$cf_combine[1]);
90         }elseif($cf_combine[2] > 0
91         AND $cf[3] > 0){
92             $cf_combine[3] = $cf_combine[2]+$cf[3]*(1-
93             $cf_combine[2]);
94         }elseif($cf_combine[2] == 0 AND $cf[3] > 0
95         AND $cf[4] > 0){
96
97         //kondisi untuk CF Combine 4
98         if($cf[4] == 0){
99             $cf_combine[4] = 0;
100        }elseif($cf_combine[1] > 0
101        AND $cf_combine[2] == 0 AND $cf_combine[3] == 0
102        AND $cf[4] > 0){
103            $cf_combine[4]=$cf_combine[1]+$cf[4]*(1-$cf_combine[1]);
104        }elseif($cf_combine[2] > 0
105        AND $cf_combine[3] == 0 AND $cf[4] > 0){
106            $cf_combine[4]=$cf_combine[2]+$cf[4]*(1-$cf_combine[2]);
107        }elseif($cf_combine[3] > 0
108        AND $cf[4] > 0){
109            $cf_combine[4]=$cf_combine[3]+$cf[4]*(1-$cf_combine[3]);
110
111        //kondisi untuk CF Combine 5
112        if($cf[5] == 0){
113            $cf_combine[5] = 0;
114        }elseif($cf_combine[4] > 0
115        AND $cf[5] > 0){
116            $cf_combine[5] = $cf_combine[4] +
117            $cf[5] * (1 - $cf_combine[4]);
118        }elseif($cf_combine[3] > 0
119        AND $cf_combine[4] == 0 AND $cf[5] > 0){
120            $cf_combine[5]=$cf_combine[3]+$cf[5]*
121            (1-$cf_combine[3]);
122        }elseif($cf_combine[2] > 0 AND $cf_combine[3] == 0
123        AND $cf_combine[4] == 0 AND $cf[5] > 0){
124            $cf_combine[5]=$cf_combine[2]+$cf[5]*
125            (1-$cf_combine[2]);
126
127        //kondisi untuk CF Combine 6
128        if($cf[6] == 0){

```



```

129         $cf_combine[6] = 0;
130     }elseif($cf_combine[1] > 0 AND $cf_combine[2] == 0
131     AND $cf_combine[3] == 0 AND $cf_combine[4] == 0
132     AND $cf_combine[5] == 0 AND $cf[6] > 0){
133         $cf_combine[6]=$cf_combine[1]+$cf[6]*
134         (1-$cf_combine[1]);
135     }elseif($cf_combine[2] > 0
136     AND $cf_combine[3] == 0 AND $cf_combine[4] == 0
137     AND $cf_combine[5] == 0 AND $cf[6] > 0){
138         $cf_combine[6]=$cf_combine[2]+$cf[6]*
139         (1-$cf_combine[2]);
140     }elseif($cf_combine[3] > 0
141     AND $cf_combine[4] == 0 AND $cf_combine[5] == 0
142     AND $cf[6] > 0){
143         $cf_combine[6]=$cf_combine[3]+$cf[6]*
144         (1-$cf_combine[3]);
145     }elseif($cf_combine[4] > 0
146     AND $cf_combine[5] == 0 AND $cf[6] > 0){
147         $cf_combine[6]=$cf_combine[4]+$cf[6]*
148         (1-$cf_combine[4]);
149     }elseif($cf_combine[5] > 0
150     AND $cf[6] > 0){
151         $cf_combine[6]=$cf_combine[5]+$cf[6]*
152         (1-$cf_combine[5]);
153
154     //kondisi untuk CF Combine 7
155     if($cf[7] == 0){
156         $cf_combine[7] = 0;
157     }elseif($cf_combine[1] > 0
158     AND $cf_combine[2] == 0
159     AND $cf_combine[3] == 0 AND $cf_combine[4] == 0
160     AND $cf_combine[5] == 0 AND $cf_combine[6] == 0
161     AND $cf[7] > 0){
162         $cf_combine[7]=$cf_combine[1]+$cf[7]*
163         (1-$cf_combine[1]);
164     }elseif($cf_combine[2] > 0
165     AND $cf_combine[3] == 0
166     AND $cf_combine[4] == 0 AND $cf_combine[5] == 0
167     AND $cf_combine[6] == 0 AND $cf[7] > 0){
168         $cf_combine[7]=$cf_combine[2]+$cf[7]*
169         (1-$cf_combine[2]);
170     }elseif($cf_combine[3] > 0
171     AND $cf_combine[4] == 0 AND $cf_combine[5] == 0
172     AND $cf_combine[6] == 0 AND $cf[7] > 0){
173         $cf_combine[7]=$cf_combine[3]+$cf[7]*
174         (1-$cf_combine[3]);
175
176     //kondisi untuk CF Combine 8
177     if($cf[8] == 0){
178         $cf_combine[8] = 0;
179     }elseif($cf_combine[1] > 0 AND $cf_combine[2] == 0
180     AND $cf_combine[3] == 0 AND $cf_combine[4] == 0
181     AND $cf_combine[5] == 0 AND $cf_combine[6] == 0
182     AND $cf_combine[7] == 0 AND $cf[8] > 0){
183         $cf_combine[8]=$cf_combine[1]+$cf[8]*
184         (1-$cf_combine[1]);
185     }elseif($cf_combine[2] > 0
186     AND $cf_combine[3] == 0 AND $cf_combine[4] == 0
187     AND $cf_combine[5] == 0 AND $cf_combine[6] == 0
188     AND $cf_combine[7] == 0 AND $cf[8] > 0){
189         $cf_combine[8]=$cf_combine[2]+$cf[8]*
190         (1-$cf_combine[2]);
191     }elseif($cf_combine[3] > 0
192     AND $cf_combine[4] == 0

```

```

193 AND $scf_combine[5] == 0 AND $scf_combine[6] == 0
194 AND $scf_combine[7] == 0 AND $scf[8] > 0){
195     $scf_combine[8]=$scf_combine[3]+$scf[8]*
196     (1-$scf_combine[3]);
197 }elseif($scf_combine[4] > 0
198 AND $scf_combine[5] == 0 AND $scf_combine[6] == 0
199 AND $scf_combine[7] == 0 AND $scf[8] > 0){
200     $scf_combine[8]=$scf_combine[4]+$scf[8]*
201     (1-$scf_combine[4]);
202
203 //kondisi untuk CF Combine 10
204 if($scf[10] == 0){
205     $scf_combine[10] = 0;
206 }elseif($scf_combine[1] > 0 AND $scf_combine[2] == 0
207 AND $scf_combine[3] == 0 AND $scf_combine[4] == 0
208 AND $scf_combine[5] == 0 AND $scf_combine[6] == 0
209 AND $scf_combine[7] == 0 AND $scf_combine[8] == 0
210 AND $scf_combine[9] == 0 AND $scf[10] > 0){
211     $scf_combine[10]=$scf_combine[1]+$scf[10]*
212     (1-$scf_combine[1]);
213 }elseif($scf_combine[2] > 0 AND $scf_combine[3] == 0
214 AND $scf_combine[4] == 0 AND $scf_combine[5] == 0
215 AND $scf_combine[6] == 0 AND $scf_combine[7] == 0
216 AND $scf_combine[8] == 0 AND $scf_combine[9] == 0
217 AND $scf[10] > 0){
218     $scf_combine[10]=$scf_combine[2]+$scf[10]*
219     (1-$scf_combine[2]);
220 }elseif($scf_combine[3] > 0 AND $scf_combine[4] == 0
221 AND $scf_combine[5] == 0 AND $scf_combine[6] == 0
222 AND $scf_combine[7] == 0 AND $scf_combine[8] == 0
223 AND $scf_combine[9] == 0 AND $scf[10] > 0){
224     $scf_combine[10]=$scf_combine[3]+$scf[10]*
225     (1-$scf_combine[3]);
226 }elseif($scf_combine[4] > 0 AND $scf_combine[5] == 0
227 AND $scf_combine[6] == 0 AND $scf_combine[7] == 0
228 AND $scf_combine[8] == 0 AND $scf_combine[9] == 0
229 AND $scf[10] > 0){
230     $scf_combine[10]=$scf_combine[4]+$scf[10]*
231     (1-$scf_combine[4]);
232
233 //kondisi untuk CF Combine 11
234 if($scf[11] == 0){
235     $scf_combine[11] = 0;
236 }elseif($scf_combine[1] > 0 AND $scf_combine[2] == 0
237 AND $scf_combine[3] == 0 AND $scf_combine[4] == 0
238 AND $scf_combine[5] == 0 AND $scf_combine[6] == 0
239 AND $scf_combine[7] == 0 AND $scf_combine[8] == 0
240 AND $scf_combine[9] == 0 AND $scf_combine[10] == 0
241 AND $scf[11] > 0){
242     $scf_combine[11]=$scf_combine[1]+$scf[11]*
243     (1-$scf_combine[1]);
244 }elseif($scf_combine[2] > 0 AND $scf_combine[3] == 0
245 AND $scf_combine[4] == 0 AND $scf_combine[5] == 0
246 AND $scf_combine[6] == 0 AND $scf_combine[7] == 0
247 AND $scf_combine[8] == 0 AND $scf_combine[9] == 0
248 AND $scf_combine[10] == 0 AND $scf[11] > 0){
249     $scf_combine[11]=$scf_combine[2]+$scf[11]*
250     (1-$scf_combine[2]);
251 }elseif($scf_combine[3] > 0 AND $scf_combine[4] == 0
252 AND $scf_combine[5] == 0 AND $scf_combine[6] == 0
253 AND $scf_combine[7] == 0 AND $scf_combine[8] == 0
254 AND $scf_combine[9] == 0 AND $scf_combine[10] == 0
255 AND $scf[11] > 0){
256     $scf_combine[11]=$scf_combine[3]+$scf[11]*
257

```



```

258 (1-$cf_combine[3]);
259 }elseif($cf_combine[4] > 0 AND $cf_combine[5] == 0
260 AND $cf_combine[6] == 0 AND $cf_combine[7] == 0
261 AND $cf_combine[8] == 0 AND $cf_combine[9] == 0
262 AND $cf_combine[10] == 0 AND $cf[11] > 0){
263     $cf_combine[11]=$cf_combine[4]+$cf[11]*
264 (1-$cf_combine[4]);
265
266 //kondisi untuk CF Combine 12
267 if($cf[12] == 0){
268     $cf_combine[12] = 0;
269 }elseif($cf_combine[1] > 0 AND $cf_combine[2] == 0
270 AND $cf_combine[3] == 0 AND $cf_combine[4] == 0
271 AND $cf_combine[5] == 0 AND $cf_combine[6] == 0
272 AND $cf_combine[7] == 0 AND $cf_combine[8] == 0
273 AND $cf_combine[9] == 0 AND $cf_combine[10] == 0
274 AND $cf_combine[11] == 0 AND $cf[12] > 0){
275     $cf_combine[12]=$cf_combine[1]+$cf[12]*
276 (1-$cf_combine[1]);
277 }elseif($cf_combine[2] > 0 AND $cf_combine[3] == 0
278 AND $cf_combine[4] == 0 AND $cf_combine[5] == 0
279 AND $cf_combine[6] == 0 AND $cf_combine[7] == 0
280 AND $cf_combine[8] == 0 AND $cf_combine[9] == 0
281 AND $cf_combine[10] == 0 AND $cf_combine[11] == 0
282 AND $cf[12] > 0){
283     $cf_combine[12]=$cf_combine[2]+$cf[12]*
284 (1-$cf_combine[2]);
285 }elseif($cf_combine[3] > 0 AND $cf_combine[4] == 0
286 AND $cf_combine[5] == 0 AND $cf_combine[6] == 0
287 AND $cf_combine[7] == 0 AND $cf_combine[8] == 0
288 AND $cf_combine[9] == 0 AND $cf_combine[10] == 0
289 AND $cf_combine[11] == 0 AND $cf[12] > 0){
290     $cf_combine[12]=$cf_combine[3]+$cf[12]*
291 (1-$cf_combine[3]);
292
293 //kondisi untuk CF Combine 13
294 if($cf[13] == 0){
295     $cf_combine[13] = 0;
296 }elseif($cf_combine[1] > 0 AND $cf_combine[2] == 0
297 AND $cf_combine[3] == 0 AND $cf_combine[4] == 0
298 AND $cf_combine[5] == 0 AND $cf_combine[6] == 0
299 AND $cf_combine[7] == 0 AND $cf_combine[8] == 0
300 AND $cf_combine[9] == 0 AND $cf_combine[10] == 0
301 AND $cf_combine[11] == 0 AND $cf_combine[12] == 0
302 AND $cf[13] > 0){
303     $cf_combine[13]=$cf_combine[1]+$cf[13]*
304 (1-$cf_combine[1]);
305 }elseif($cf_combine[2] > 0 AND $cf_combine[3] == 0
306 AND $cf_combine[4] == 0 AND $cf_combine[5] == 0
307 AND $cf_combine[6] == 0 AND $cf_combine[7] == 0
308 AND $cf_combine[8] == 0 AND $cf_combine[9] == 0
309 AND $cf_combine[10] == 0 AND $cf_combine[11] == 0
310 AND $cf_combine[12] == 0 AND $cf[13] > 0){
311     $cf_combine[13]=$cf_combine[2]+$cf[13]*
312 (1-$cf_combine[2]);
313 }elseif($cf_combine[3] > 0 AND $cf_combine[4] == 0
314 AND $cf_combine[5] == 0 AND $cf_combine[6] == 0
315 AND $cf_combine[7] == 0 AND $cf_combine[8] == 0
316 AND $cf_combine[9] == 0 AND $cf_combine[10] == 0
317 AND $cf_combine[11] == 0 AND $cf_combine[12] == 0
318 AND $cf[13] > 0){
319     $cf_combine[13]=$cf_combine[3]+$cf[13]*
320 (1-$cf_combine[3]);
321
322

```



```

323 //kondisi untuk cf combine 14
324 if($cf[14] == 0){
325     $cf_combine[14] = 0;
326 }elseif($cf_combine[1] > 0 AND $cf_combine[2] == 0
327 AND $cf_combine[3] == 0 AND $cf_combine[4] == 0
328 AND $cf_combine[5] == 0 AND $cf_combine[6] == 0
329 AND $cf_combine[7] == 0 AND $cf_combine[8] == 0
330 AND $cf_combine[9] == 0 AND $cf_combine[10] == 0
331 AND $cf_combine[11] == 0 AND $cf_combine[12] == 0
332 AND $cf_combine[13] == 0 AND $cf[14] > 0){
333     $cf_combine[14]=$cf_combine[1]+$cf[14]*
334 (1-$cf_combine[1]);
335 }elseif($cf_combine[2] > 0 AND $cf_combine[3] == 0
336 AND $cf_combine[4] == 0 AND $cf_combine[5] == 0
337 AND $cf_combine[6] == 0 AND $cf_combine[7] == 0
338 AND $cf_combine[8] == 0 AND $cf_combine[9] == 0
339 AND $cf_combine[10] == 0 AND $cf_combine[11] == 0
340 AND $cf_combine[12] == 0 AND $cf_combine[13] == 0
341 AND $cf[14] > 0){
342     $cf_combine[14]=$cf_combine[2]+$cf[14]*
343 (1-$cf_combine[2]);
344 }elseif($cf_combine[3] > 0 AND $cf_combine[4] == 0
345 AND $cf_combine[5] == 0 AND $cf_combine[6] == 0
346 AND $cf_combine[7] == 0 AND $cf_combine[8] == 0
347 AND $cf_combine[9] == 0 AND $cf_combine[10] == 0
348 AND $cf_combine[11] == 0 AND $cf_combine[12] == 0
349 AND $cf_combine[13] == 0 AND $cf[14] > 0){
350     $cf_combine[14]=$cf_combine[3]+$cf[14]*
351 (1-$cf_combine[3]);
352
353 //kondisi untuk cf combine 15
354 if($cf[15] == 0){
355     $cf_combine[15] = 0;
356 }elseif($cf_combine[1] > 0 AND $cf_combine[2] == 0
357 AND $cf_combine[3] == 0 AND $cf_combine[4] == 0
358 AND $cf_combine[5] == 0 AND $cf_combine[6] == 0
359 AND $cf_combine[7] == 0 AND $cf_combine[8] == 0
360 AND $cf_combine[9] == 0 AND $cf_combine[10] == 0
361 AND $cf_combine[11] == 0 AND $cf_combine[12] == 0
362 AND $cf_combine[13] == 0 AND $cf_combine[14] == 0
363 AND $cf[15] > 0){
364     $cf_combine[15]=$cf_combine[1]+$cf[15]*
365 (1-$cf_combine[1]);
366 }elseif($cf_combine[2] > 0 AND $cf_combine[3] == 0
367 AND $cf_combine[4] == 0 AND $cf_combine[5] == 0
368 AND $cf_combine[6] == 0 AND $cf_combine[7] == 0
369 AND $cf_combine[8] == 0 AND $cf_combine[9] == 0
370 AND $cf_combine[10] == 0 AND $cf_combine[11] == 0
371 AND $cf_combine[12] == 0 AND $cf_combine[13] == 0
372 AND $cf_combine[14] == 0 AND $cf[15] > 0){
373     $cf_combine[15]=$cf_combine[2]+$cf[15]*
374 (1-$cf_combine[2]);
375 }elseif($cf_combine[3] > 0 AND $cf_combine[4] == 0
376 AND $cf_combine[5] == 0 AND $cf_combine[6] == 0
377 AND $cf_combine[7] == 0 AND $cf_combine[8] == 0
378 AND $cf_combine[9] == 0 AND $cf_combine[10] == 0
379 AND $cf_combine[11] == 0 AND $cf_combine[12] == 0
380 AND $cf_combine[13] == 0 AND $cf_combine[14] == 0
381 AND $cf[15] > 0){
382     $cf_combine[15]=$cf_combine[3]+$cf[15]*
383 (1-$cf_combine[3]);
384 }elseif($cf_combine[4] > 0 AND $cf_combine[5] == 0
385 AND $cf_combine[6] == 0 AND $cf_combine[7] == 0
386 AND $cf_combine[8] == 0 AND $cf_combine[9] == 0

```

```

387 AND $cf_combine[10] == 0 AND $cf_combine[11] == 0
388 AND $cf_combine[12] == 0 AND $cf_combine[13] == 0
389 AND $cf_combine[14] == 0 AND $cf[15] > 0){
390     $cf_combine[15]=$cf_combine[4]+$cf[15]*
391     (1-$cf_combine[4]);
392
393     $presentase_keyakinan = max($cf_combine[1],$cf_combine[2],
394     $cf_combine[3],$cf_combine[4],$cf_combine[5],
395     $cf_combine[6],$cf_combine[7],$cf_combine[8],
396     $cf_combine[9],$cf_combine[10],$cf_combine[11],
397     $cf_combine[12],$cf_combine[13],$cf_combine[14],$cf_combine[15]);
398     $presentase_keyakinan = round($presentase_keyakinan, 5);
399
400     $query = "INSERT INTO cf
401 (hasil_diagnosis, presentase) VALUES
402 ('Tifoid','".$presentase_keyakinan."')";
403     mysql_query($query)or die(mysql_error());
404
405     if($query){
406         echo '<meta http-equiv = "refresh" content
407 = "0;URL=hasildiagnosis.php" />';
408     }else{
409         echo 'gagal vroh';
410     }
411 }

```

Source code 5.4 Certainty Factor

Source code 5.4 merupakan fungsi perhitungan *certainty factor* yang didalamnya terdapat proses menghitung nilai *cf* dan *cf combine* tiap jenis penyakit.

5.4 Implementasi User Interface

Pada implementasi *user interface* ini akan dibuat beberapa halaman yang terdiri dari halaman *sign in*, beranda admin dan tenaga medis, halaman riwayat pasien admin dan tenaga medis, halaman diagnosis, halaman olah data *user*, dan halaman gejala.

5.4.1 Implementasi Halaman Sign In

Gambar 5.2 merupakan implementasi halaman *sign in* dari sistem pakar diagnosis penyakit Demam: DBD, Malaria dan Tifoid. Pada halaman *sign in* tersebut admin atau tenaga medis dapat masuk ke sistem dengan menggunakan *username* dan *password* yang telah terdaftar.



Gambar 5.2 Halaman Sign In

5.4.2 Implementasi Halaman Beranda (Admin)

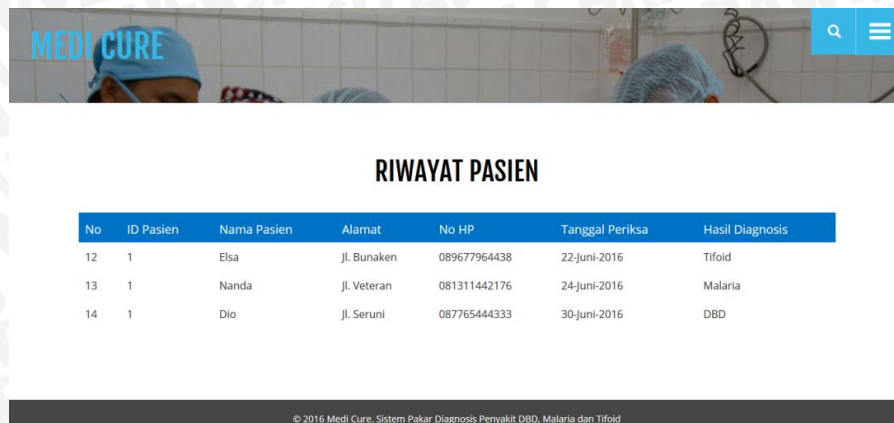
Gambar 5.3 merupakan implementasi halaman beranda milik admin dari sistem pakar diagnosis penyakit Demam: DBD, Malaria dan Tifoid. Pada halaman beranda milik admin, admin dapat mengakses menu riwayat pasien, olah data user, daftar gejala dan melakukan sign out.



Gambar 5.3 Halaman beranda (admin)

5.4.3 Implementasi Halaman Riwayat Pasien (Admin)

Gambar 5.4 merupakan implementasi halaman riwayat pasien milik admin dari sistem pakar diagnosis penyakit Demam: DBD, Malaria dan Tifoid. Pada halaman ini admin dapat melihat daftar riwayat pasien, mulai dari nama, alamat, no telp, tanggal pemeriksaan dan hasil diagnosis.



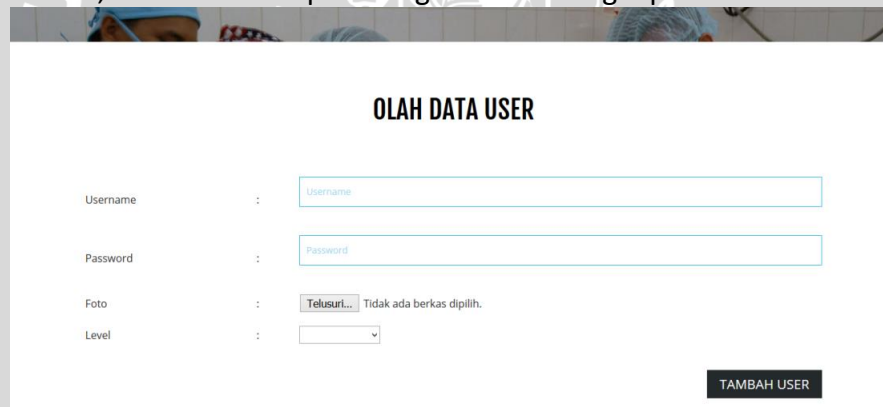
No	ID Pasien	Nama Pasien	Alamat	No HP	Tanggal Periksa	Hasil Diagnosis
12	1	Elsa	Jl. Bunaken	089677964438	22-Juni-2016	Tifoid
13	1	Nanda	Jl. Veteran	081311442176	24-Juni-2016	Malaria
14	1	Dio	Jl. Seruni	087765444333	30-Juni-2016	DBD

© 2016 Medi Cure. Sistem Pakar Diagnosis Penyakit DBD, Malaria dan Tifoid

Gambar 5.4 Halaman riwayat pasien (admin)

5.4.4 Implementasi Halaman Olah Data *User* (Admin)

Gambar 5.5 merupakan implementasi halaman olah data *user* milik admin dari sistem pakar diagnosis penyakit Demam: DBD, Malaria dan Tifoid. Pada halaman ini admin dapat menambah data *user* berupa *username*, *password*, *level*, dan foto *user*, serta admin dapat mengedit dan menghapus data *user*.



OLAH DATA USER

Username :

Password :

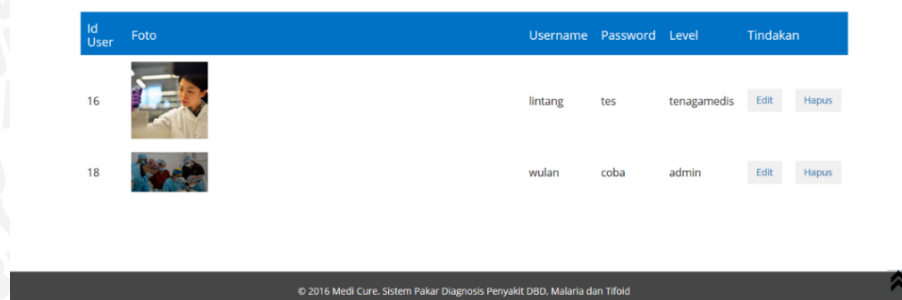
Foto : Telusuri... Tidak ada berkas dipilih.

Level :

TAMBAH USER

Gambar 5.5 Halaman olah data *user* (admin)

Gambar 5.6 pada halaman selanjutnya merupakan implementasi halaman olah data *user* milik admin setelah admin menambahkan *user* baru.



Id User	Foto	Username	Password	Level	Tindakan
16		lintang	tes	tenagamedis	Edit Hapus
18		wulan	coba	admin	Edit Hapus

© 2016 Medi Cure. Sistem Pakar Diagnosis Penyakit DBD, Malaria dan Tifoid

Gambar 5.6 Halaman setelah admin menambah *user* (admin)

Gambar 5.7 merupakan implementasi halaman olah data *user* milik admin ketika admin mengedit data *user*.

Gambar 5.7 Halaman setelah admin menambah *user* (admin)

5.4.5 Implementasi Halaman Daftar Gejala (Admin)

Gambar 5.8 merupakan implementasi halaman daftar gejala milik admin dari sistem pakar diagnosis penyakit Demam: DBD, Malaria dan Tifoid. Pada halaman ini admin dapat menambah data gejala berupa nama gejala, tingkatan gejala, dan nilai.

Gambar 5.8 Halaman daftar gejala (admin)

Gambar 5.9 merupakan implementasi halaman daftar gejala milik admin setelah admin memasukkan gejala baru.

Nomor	ID Gejala	Nama Gejala	Tingkatan Gejala	Nilai	Tindakan
1	4	Demam Intermittent	Ya	0.8	<button>Hapus</button>
2	5	Demam Intermittent	Tidak	0	<button>Hapus</button>
3	6	Demam Terutama Malam Hari	Ya	0.75	<button>Hapus</button>
4	7	Demam Terutama Malam Hari	Tidak	0	<button>Hapus</button>

Gambar 5.9 Halaman setelah admin memasukkan gejala baru

5.4.6 Implementasi Halaman Beranda (Tenaga Medis)

Gambar 5.10 merupakan implementasi halaman beranda milik tenaga medis dari sistem pakar diagnosis penyakit Demam: DBD, Malaria dan Tifoid. Pada halaman ini tenaga medis dapat melihat riwayat pasien, melakukan diagnosis, dan sign out.



Gambar 5.10 Halaman beranda (tenaga medis)

5.4.7 Implementasi Halaman Riwayat Pasien (Tenaga Medis)

Gambar 5.11 merupakan implementasi halaman riwayat pasien milik tenaga medis. Pada halaman ini tenaga medis dapat melihat daftar riwayat pasien, mulai dari nama, alamat, no telp, tanggal periksa dan hasil diagnosis.

No	ID Pasien	Nama Pasien	Alamat	No HP	Tanggal Periksa	Hasil Diagnosis
12	1	Elsa	Jl. Bunaken	089677964438	22-juni-2016	Tifoid
13	1	Nanda	Jl. Veteran	081311442176	24-juni-2016	Malaria
14	1	Dio	Jl. Seruni	087765444333	30-juni-2016	DBD

Gambar 5.11 Halaman riwayat pasien (tenaga medis)

5.4.8 Implementasi Halaman Diagnosis (Tenaga Medis)

Gambar 5.12 dan Gambar 5.13 merupakan implementasi halaman diagnosis milik tenaga medis. Pada halaman ini tenaga medis dapat memasukkan tanggal periksa, data pasien dan keluhan-keluhan yang dirasakan oleh pasien.

DIAGNOSIS

Tanggal Periksa :

Nama Pasien :

Alamat :

No Telp :

A. Keluhan

1. Demam Intermitten ☐ Ya ☒ Tidak

2. Demam Menggigil ☐ Berat ☐ Sedang ☒ Tidak Ada

Gambar 5.12 Halaman diagnosis (tenaga medis)

8. Menceret atau Susah BAB ☐ Ada ☒ Tidak Ada

9. Nyeri Perut ☐ Berat ☐ Sedang ☒ Tidak Ada

10. Pemeriksaan Fisik :

10. Bintik Merah (ptekie) pada kulit ☐ Berat ☐ Sedang ☒ Tidak Ada

11. Lidah Kotor (Coated Tongue) ☐ Berat ☐ Sedang ☒ Tidak Ada

12. Bradikardi relatif ☐ Ya ☒ Tidak Ada

13. Pembesaran Hati ☐ Ya ☒ Tidak Ada

14. Pembesaran Limpa ☐ Ya ☒ Tidak Ada

15. Kulit Lembab/Keringat ☐ Berat ☐ Sedang ☒ Tidak Ada

PROSES

© 2016 Medi Cure. Sistem Pakar Diagnosis Penyakit DBD, Malaria dan Tifoid

Gambar 5.13 Halaman diagnosis (tenaga medis)

Gambar 5.14 merupakan implementasi halaman diagnosis setelah tenaga medis memasukkan data pasien dan keluhannya. Pada halaman terdapat hasil diagnosis pasien dan terdapat pilihan untuk menyimpan data tersebut ke menu riwayat pasien.

HASIL DIAGNOSIS

Id Pasien	Nama Pasien	Alamat Pasien	No HP	Tanggal Periksa	Hasil	Tindakan
1	Ghina	Jl. Melati	087654321888	14-Juni-2016	DBD	Simpan ke Riwayat Pasien

© 2016 Medi Cure. Sistem Pakar Diagnosis Penyakit DBD, Malaria dan Tifoid

Gambar 5.14 Halaman diagnosis setelah tenaga medis memasukkan data pasien dan keluhannya

BAB 6 PENGUJIAN DAN ANALISIS

Pada bab ini akan membahas mengenai tahapan pengujian dan analisis dari Sistem Pakar Diagnosis Penyakit Demam: DBD, Malaria dan Tifoid Menggunakan Metode *K-Nearest Neighbor – Certainty Factor*.

6.1 Pengujian

Tahap pengujian ini dilakukan untuk mengetahui hasil akurasi dari metode *K-Nearest Neighbor – Certainty Factor* dalam mendiagnosis penyakit Demam: DBD, Malaria dan Tifoid. Pada pengujian ini terdapat 6 macam mekanisme pengujian yaitu:

1. Pengujian Variasi Nilai *K*

Pada pengujian ini digunakan nilai *K* yang bervariasi mulai dari 1 sampai dengan 50 dan menggunakan 100 data latih dan 43 data uji.

2. Pengujian Variasi Data Latih

Pada pengujian ini digunakan variasi data latih yang berbeda-beda dengan data uji yang sama dan nilai *K* yang tetap. Nilai *K* latih yang digunakan memiliki tingkat akurasi yang tinggi yang dihasilkan oleh pengujian pertama dan data latih yang digunakan berjumlah 100 data.

3. Pengujian Rasio Data Latih dan Data Uji

Pada pengujian ini digunakan rasio data latih dan data uji yang berbeda-beda. Dimana masing-masing kelas penyakit Demam memiliki jumlah data yang seimbang.

4. Pengujian Variasi Jumlah Data Uji

Pada pengujian ini digunakan jumlah data uji yang berbeda-beda dengan data latih yang sama dan nilai *K* yang sama. Nilai *K* latih yang digunakan memiliki tingkat akurasi yang tinggi yang dihasilkan oleh pengujian pertama dan data latih yang digunakan berjumlah 100 data.

5. Pengujian Variasi Data Uji

Pada pengujian ini digunakan variasi data uji yang berbeda-beda dengan jumlah yang sama, data latih yang sama dan nilai *K* yang tetap. Nilai *K* latih yang digunakan memiliki tingkat akurasi yang tinggi yang dihasilkan oleh pengujian pertama dan data latih yang digunakan berjumlah 100 data.

6. Pengujian Perbandingan Metode

Pada pengujian ini adalah pengujian perbandingan nilai akurasi antara metode *K-Nearest Neighbor – Certainty Factor* dengan metode *K-Nearest Neighbor*, *Naïve Bayes* dan *J48*. Pengujian yang akan dilakukan pada metode *K-Nearest Neighbor* sama dengan pengujian pertama (variasi nilai *k*).

6.1.1 Pengujian Variasi Nilai *K*

Pada pengujian variasi nilai *K* bertujuan untuk mengetahui apakah nilai *K* berpengaruh terhadap tingkat akurasi hasil diagnosis sistem dengan hasil diagnosis pakar dan untuk mengetahui nilai *K* yang memiliki nilai akurasi tertinggi.

Pada pengujian ini digunakan 100 data latih dan 43 data uji, dengan beberapa variasi nilai K mulai dari $k=1$ hingga $k=50$. Pengujian variasi nilai K ditunjukkan pada tabel 6.1.

Tabel 6.1 Tabel pengujian variasi nilai K

Nilai K	Akurasi	Nilai K	Akurasi	Nilai K	Akurasi	Nilai K	Akurasi	Nilai K	Akurasi
1	88.37%	11	86.05%	21	83.72%	31	83.72%	41	83.72%
2	88.37%	12	83.72%	22	83.72%	32	86.05%	42	83.72%
3	88.37%	13	83.72%	23	83.72%	33	86.05%	43	83.72%
4	88.37%	14	83.72%	24	83.72%	34	86.05%	44	83.72%
5	88.37%	15	83.72%	25	83.72%	35	86.05%	45	83.72%
6	86.05%	16	83.72%	26	83.72%	36	86.05%	46	83.72%
7	86.05%	17	83.72%	27	83.72%	37	86.05%	47	83.72%
8	88.37%	18	83.72%	28	83.72%	38	83.72%	48	83.72%
9	86.05%	19	83.72%	29	83.72%	39	83.72%	49	83.72%
10	86.05%	20	83.72%	30	83.72%	40	83.72%	50	83.72%
Rata-rata		84.79%							

Dari Tabel 6.1 didapatkan hasil akurasi yang bervariasi dari nilai $k=1$ hingga $k=50$ yaitu akurasi tertinggi sebesar 88.37% didapat dari nilai $k=1$, $k=2$, $k=3$, $k=4$, $k=5$ dan $k=8$. Sedangkan akurasi terendah yaitu sebesar 83.72%.

6.1.2 Pengujian Variasi Data Latih

Pengujian variasi data latih bertujuan untuk mengetahui apakah data latih yang berbeda-beda akan berpengaruh terhadap tingkat akurasi sistem. Pada pengujian ini jumlah data latih yang digunakan adalah sama namun dengan data yang berbeda. Nilai K yang digunakan adalah $k=3$, $k=5$ dan $k=8$. Pengujian variasi data latih ditunjukkan pada Tabel 6.2.

Tabel 6.2 Tabel pengujian variasi data latih

Variasi Data Latih	Akurasi		
	$K = 3$	$K = 5$	$K = 8$
50 Data (1)	83.72%	83.72%	81.39%
50 Data (2)	83.72%	83.72%	83.72%

Variasi Data Latih	Akurasi		
	$K = 3$	$K = 5$	$K = 8$
50 Data (3)	83.72%	86.04%	83.72%
50 Data (4)	83.72%	83.72%	83.72%

Dari hasil pengujian diatas didapatkan hasil akurasi berbeda-beda pada tiap percobaan dimana akurasi tertinggi terdapat pada percobaan ke-3 dengan nilai $k=5$ yaitu sebesar 86.04%, sedangkan akurasi terendah terdapat pada percobaan 1 dengan nilai $k=8$ yaitu 81.39%.

6.1.3 Pengujian Rasio Data Latih dan Data Uji

Pengujian rasio data latih dan data uji bertujuan untuk mengetahui apakah rasio data latih dan data uji yang berbeda-beda akan berpengaruh terhadap tingkat akurasi sistem.

Pada pengujian ini dilakukan percobaan sebanyak 5 kali dengan rasio data latih dan data uji yang berbeda. Data yang dipakai pada pengujian ini berjumlah 120 data. Untuk nilai K yang digunakan adalah $k=3$, $k=5$, dan $k=8$. Pengujian rasio data latih dan data uji ditunjukkan pada Tabel 6.3.

Tabel 6.3 Tabel pengujian rasio data latih dan data uji

Percobaan ke-	Rasio Data Latih dan Data Uji	Jumlah Data Latih	Jumlah Data Uji	Akurasi		
				$K = 3$	$K = 5$	$K = 8$
1	90 : 10	108	12	91.67%	91.67%	91.67%
2	80 : 20	96	24	91.67%	91.67%	95.83%
3	50 : 50	60	60	95%	95%	95%
4	20 : 80	24	96	95.83%	94.79%	92.70%
5	10 : 90	12	108	89.81%	87.96%	71.29%

Dari Tabel 6.3 didapatkan hasil akurasi yang berbeda-beda pada tiap percobaan dimana akurasi tertinggi pada percobaan ke-2 pada nilai $k=8$ dan percobaan ke-4 pada nilai $k=3$ sebesar 95.83%, sedangkan akurasi terendah terdapat pada percobaan ke-5 dengan nilai $k=8$ yaitu 71.29%.

6.1.4 Pengujian Variasi Jumlah Data Uji

Pada pengujian variasi jumlah data uji ini digunakan nilai $k=3$, $k=5$ dan $k=8$. Pengujian ini menggunakan variasi jumlah data uji 1 hingga 40 data dan data latih yang digunakan adalah data latih yang sama yaitu 100 data latih. Pengujian variasi jumlah data uji ini bertujuan untuk mengetahui apakah jumlah data uji yang berbeda-beda dapat mempengaruhi tingkat keakurasian. Pengujian variasi jumlah data uji ini ditunjukkan pada Tabel 6.4.

Tabel 6.4 Tabel pengujian variasi jumlah data uji

Percobaan ke-	Jumlah Data Uji	Akurasi		
		$K = 3$	$K = 5$	$K = 8$
1	10 Data Uji	80%	80%	80%
2	20 Data Uji	90%	90%	90%
3	30 Data Uji	86.67%	86.67%	83.33%
4	40 Data Uji	85%	85%	82.5%

Dari tabel 6.4 didapatkan hasil akurasi yang berbeda-beda pada tiap percobaan dimana akurasi tertinggi pada percobaan ke-2 dengan nilai $k=3$, $k=5$ dan $k=8$ yaitu sebesar 90%, sedangkan akurasi terendah pada percobaan ke-1 dengan nilai $k=3$, $k=5$ dan $k=8$ yaitu 80%.

6.1.5 Pengujian Variasi Data Uji

Pada pengujian variasi data uji ini digunakan nilai K dan data latih dengan tingkat akurasi tertinggi yang didapatkan pada percobaan pertama yaitu nilai $k=3$, $k=5$ dan $k=8$ dan 100 data latih. Pengujian ini menggunakan variasi data uji dengan jumlah yang sama yaitu 36 data. Pengujian variasi data uji ini bertujuan untuk mengetahui apakah variasi data uji yang berbeda-beda dapat mempengaruhi tingkat keakurasian. Pengujian variasi data uji ini ditunjukkan pada Tabel 6.5.

Tabel 6.5 Tabel pengujian variasi data uji

Percobaan ke-	Jumlah Data Uji	Akurasi		
		$K = 3$	$K = 5$	$K = 8$
1	36 Data Uji	94.44%	94.44%	94.44%
2	36 Data Uji	94.44%	94.44%	94.44%
3	36 Data Uji	97.22%	97.22%	94.44%

Dari Tabel 6.5 didapatkan hasil akurasi tertinggi pada percobaan ke-3 dengan nilai $k=3$ dan $k=5$ yaitu sebesar 97.22%, sedangkan akurasi terendah pada percobaan ke-1 dan ke-2 yaitu 94.44%.

6.1.6 Pengujian Perbandingan Metode

Pengujian ini dilakukan untuk mengetahui perbandingan nilai akurasi antara metode *K-Nearest Neighbor – Certainty Factor* dan metode *K-Nearest Neighbor* dengan acuan perbandingan rata-rata akurasi yang dihasilkan untuk setiap K pada setiap data latih. Pengujian ini menggunakan $k=1$ hingga $k=50$, jumlah data latih yang tetap yaitu 100 data dan jumlah 43 data uji yang tetap. Pengujian ini bertujuan untuk mengetahui nilai akurasi terbaik dan ketepatan diagnosis yang

terbaik antara metode *K-Nearest Neighbor* – *Certainty Factor* dengan metode *K-Nearest Neighbor*. Pengujian perbandingan metode ditunjukkan pada Tabel 6.6.

Tabel 6.6 Tabel pengujian perbandingan metode

Nilai k	Akurasi <i>K-Nearest Neighbor</i>	Akurasi <i>K-Nearest Neighbor</i> – <i>Certainty Factor</i>
1	84.79%	84.79%
2	84.79%	84.79%
3	84.79%	84.79%
...	...	...
50	84.79%	84.79%

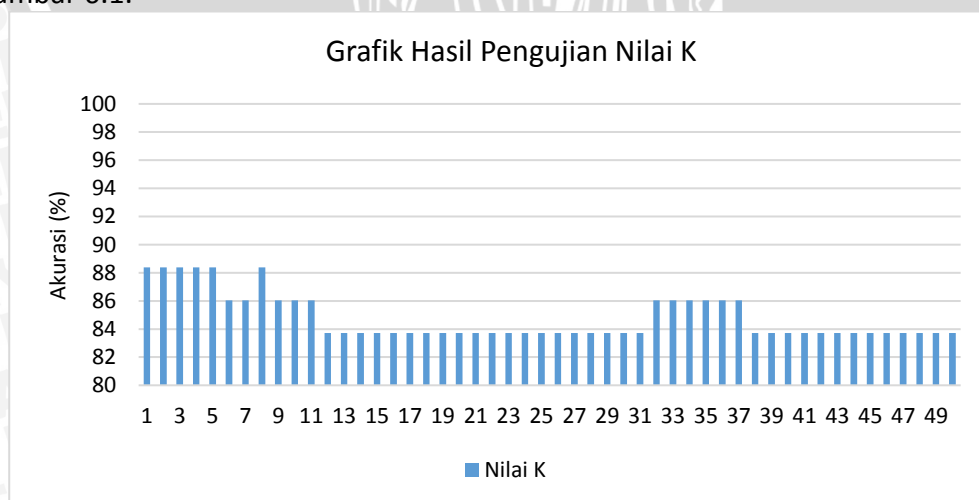
Dari Tabel 6.6 dapat disimpulkan bahwa metode *K-Nearest Neighbor* – *Certainty Factor* dan metode *K-Nearest Neighbor* memiliki akurasi yang sama sebesar 84.79%.

6.2 Analisis Hasil

Pada bagian ini akan dilakukan analisis dari pengujian-pengujian yang telah dilakukan sebelumnya. Pengujian yang dilakukan sebelumnya meliputi pengujian variasi nilai k, pengujian variasi data latih, pengujian rasio data latih dan data uji, pengujian variasi jumlah data uji, pengujian variasi data uji dan pengujian perbandingan metode *K-Nearest Neighbor* dan metode *K-Nearest Neighbor* – *Certainty Factor*.

6.2.1 Analisis Hasil Variasi Nilai K

Pada analisis variasi nilai K digunakan 100 data latih, 43 data uji dan nilai K mulai dari $k=1$ sampai $k=50$. Hasil pengujian pengaruh nilai K ditunjukkan pada Gambar 6.1.



Gambar 6.1 Grafik hasil pengujian variasi nilai k

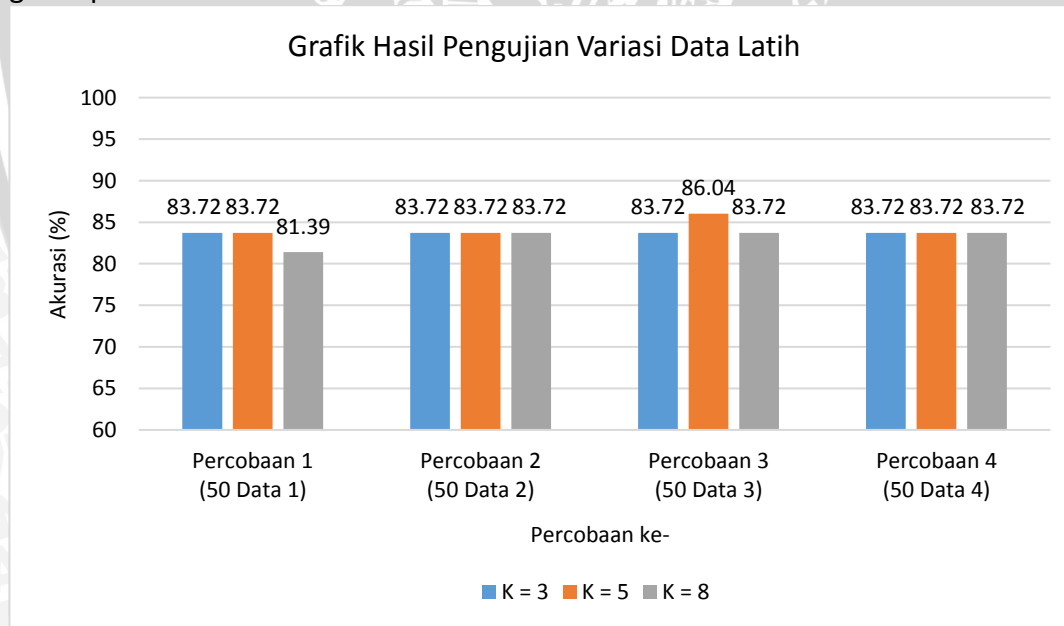
Pada grafik Gambar 6.1 diatas dapat diketahui bahwa dari nilai $k=1$ sampai $k=5$ menghasilkan akurasi sebesar 88.37%. Setelah itu akurasi mengalami naik turun. Pada nilai $k=12$ hingga $k=31$ menghasilkan akurasi 83.72%, pada nilai $k=32$ hingga $k=37$ menghasilkan akurasi 86.05%, pada nilai $k=38$ hingga $k=50$ menghasilkan akurasi 83.72%.

Pada percobaan tersebut nilai K yang semakin bertambah menghasilkan akurasi yang semakin rendah hingga mengalami naik turun akurasi, namun naik turun akurasi tidak begitu besar dan cenderung stabil. Nilai K yang bertambah akan membuat semakin banyak data tidak relevan terhadap data uji dan hal tersebut mengakibatkan tingkat kesalahan prediksi yang semakin besar dan menyebabkan tingkat akurasi turun.

Dari percobaan diatas dapat disimpulkan bahwa variasi nilai K berpengaruh pada akurasi. Untuk nilai K yang akan digunakan pada pengujian-pengujian adalah nilai K yang memiliki akurasi tertinggi yaitu nilai $k=3$, $k=5$ dan $k=8$ dengan akurasi sebesar 88.37%.

6.2.2 Analisis Hasil Variasi Data Latih

Pada analisis variasi data latih digunakan nilai K dengan tingkat akurasi yang tertinggi yang didapatkan pada pengujian variasi nilai K yaitu $k=3$, $k=5$ dan $k=8$ dimana digunakan variasi data latih dengan jumlah 50 data dan data uji berjumlah 43 data. Hasil pengujian pengaruh jumlah data latih ditampilkan dalam sebuah grafik pada Gambar 6.2.



Gambar 6.2 Grafik hasil pengujian variasi data latih

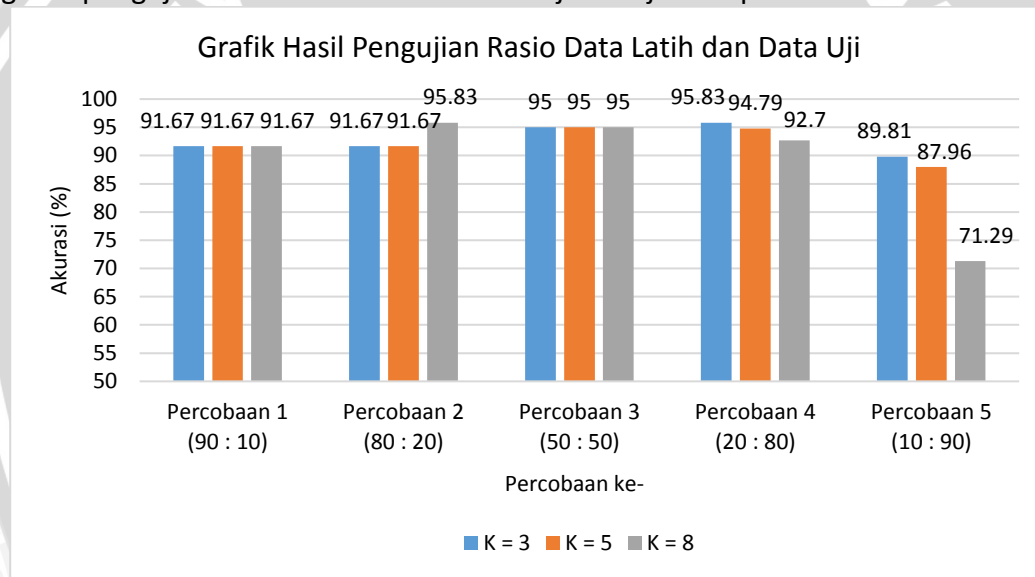
Pada grafik Gambar 6.2 diatas dapat diketahui bahwa percobaan ke-3 dengan nilai $k=5$ memiliki hasil akurasi yang tertinggi di banding nilai K yang lain yaitu sebesar 86.04% sedangkan akurasi terendah terdapat pada percobaan ke-1 dengan nilai $k=8$ yaitu 81.39%. Hal ini disebabkan karena meskipun data latih yang

digunakan berjumlah sama, akan tetapi variasi data latihnya berbeda. Dimana pada percobaan ke-3 memiliki variasi data dengan jarak yang lebih berdekatan dengan data uji dibandingkan dengan percobaan ke-1, ke-2 dan ke-4 sehingga menghasilkan tingkat akurasi yang lebih tinggi.

Dari hal tersebut dapat disimpulkan bahwa percobaan yang dilakukan dengan nilai $k=3$, $k=5$ dan $k=8$ memiliki akurasi yang naik turun walaupun sebagian besar sama. Perbedaan variasi pada setiap percobaan membuat tingkat akurasi berbeda-beda dan mempengaruhi nilai jarak antara data latih dan data uji. Jika dalam suatu variasi data latih memiliki nilai jarak yang berdekatan dengan data uji akan membuat akurasi semakin tinggi dan begitu sebaliknya.

6.2.3 Analisis Hasil Rasio Data Latih dan Data Uji

Pengujian rasio data latih dan data uji ini diambil dari persentase perbandingan data latih dan data uji, yaitu 90:10, 80:20, 50:50, 20:80 dan 10:90. Berikut hasil grafik pengujian rasio data latih dan data uji ditunjukkan pada Gambar 6.3.

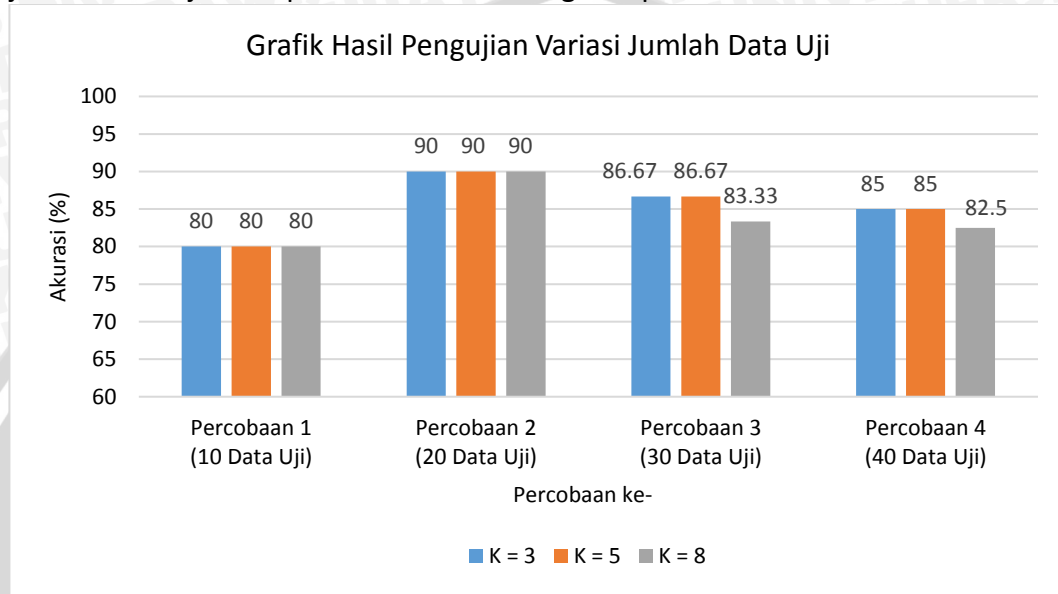


Gambar 6.3 Grafik hasil pengujian rasio data latih dan data uji

Pada grafik Gambar 6.3 diatas dapat diketahui bahwa pada nilai $k=8$ percobaan ke-2 dan nilai $k=3$ pada percobaan ke-4 memiliki hasil akurasi tertinggi yaitu 95.83%, ini dikarenakan jumlah kesalahan pada setiap percobaan dan banyaknya jumlah data uji pada setiap percobaan berpengaruh pada tingkat akurasi sistem. Seperti pada percobaan ke-2 jumlah kesalahan yang terjadi adalah 2 dan ke-3 jumlah kesalahan yang terjadi adalah 4, tetapi data uji yang digunakan pada percobaan ke-2 adalah 24 data dan pada percobaan ke-3 adalah 60 data, karena jumlah data uji pada percobaan ke-3 lebih banyak maka tingkat akurasi pada percobaan ke-3 lebih besar meskipun memiliki jumlah kesalahan lebih besar daripada percobaan ke-2. Jadi jumlah kesalahan pada setiap percobaan dan banyaknya jumlah data uji berpengaruh besar pada tingkat akurasi.

6.2.4 Analisis Hasil Variasi Jumlah Data Uji

Pada analisis variasi jumlah data uji ini digunakan nilai K dengan tingkat akurasi tertinggi yang didapatkan pada percobaan pertama yaitu $k=3$, $k=5$ dan $k=8$ dimana digunakan data uji dengan jumlah 10, 20, 30 dan 40 data dan data latih yang digunakan adalah data latih yang sama yaitu 100 data. Hasil pengujian variasi jumlah data uji ditampilkan dalam sebuah grafik pada Gambar 6.4.



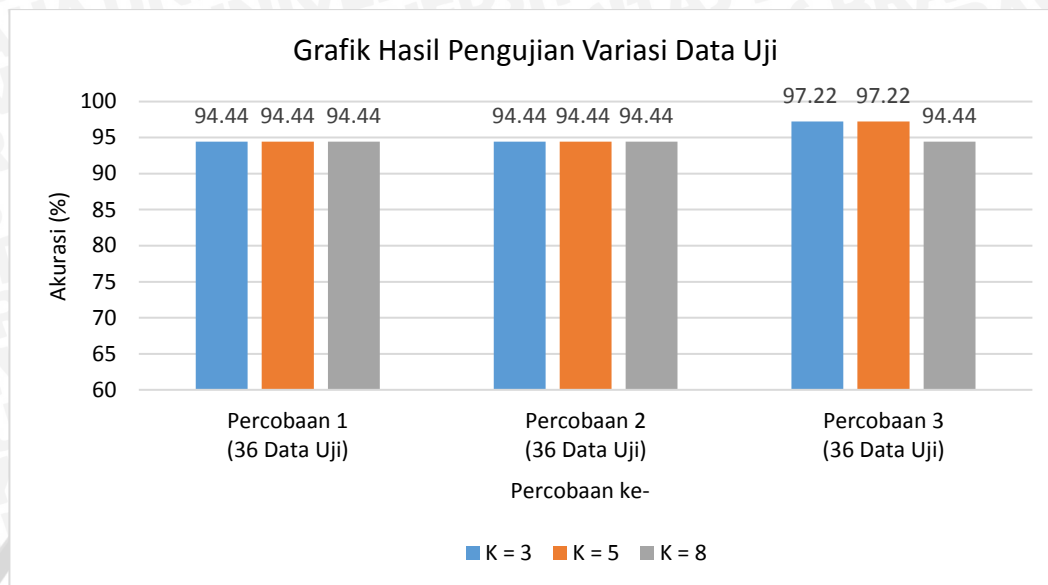
Gambar 6.4 Grafik hasil pengujian variasi jumlah data uji

Pada grafik Gambar 6.4 diatas dapat diketahui bahwa pada percobaan ke-1 dengan nilai $k=3$, $k=5$ dan $k=8$ memiliki nilai akurasi 80%. Sedangkan pada percobaan ke-2 dengan nilai $k=3$, $k=5$ dan $k=8$ memiliki nilai akurasi tertinggi sebesar 90%. Pada percobaan ke-3 dan ke-4 tingkat akurasi mengalami naik turun tetapi tidak terlalu besar dan cenderung stabil.

Dari grafik diatas dapat disimpulkan bahwa yang dapat mempengaruhi tingkat akurasi adalah jumlah kesalahan pada setiap percobaan dan banyaknya jumlah data uji pada setiap percobaan. Seperti pada percobaan ke-1 dan ke-2 jumlah kesalahan yang terjadi adalah 2, tetapi data uji yang digunakan pada percobaan ke-1 adalah 10 data dan pada percobaan ke-2 adalah 20 data, karena jumlah data uji pada percobaan ke-2 lebih banyak maka tingkat akurasi pada percobaan ke-2 lebih besar. Jadi dapat disimpulkan bahwa pada pengujian variasi jumlah data uji dipengaruhi oleh 2 hal yaitu jumlah kesalahan pada setiap percobaan dan banyaknya jumlah data uji.

6.2.5 Analisis Hasil Variasi Data Uji

Pada analisis variasi data uji ini digunakan nilai K dan data latih dengan tingkat akurasi tertinggi yang didapatkan pada percobaan pertama yaitu $k=3$, $k=5$ dan $k=8$ dimana digunakan variasi data uji berjumlah sama yaitu 36 data pada 3 kali percobaan dan data latih sebanyak 100 data latih. Hasil pengujian variasi data uji ditampilkan dalam sebuah grafik pada Gambar 6.5.

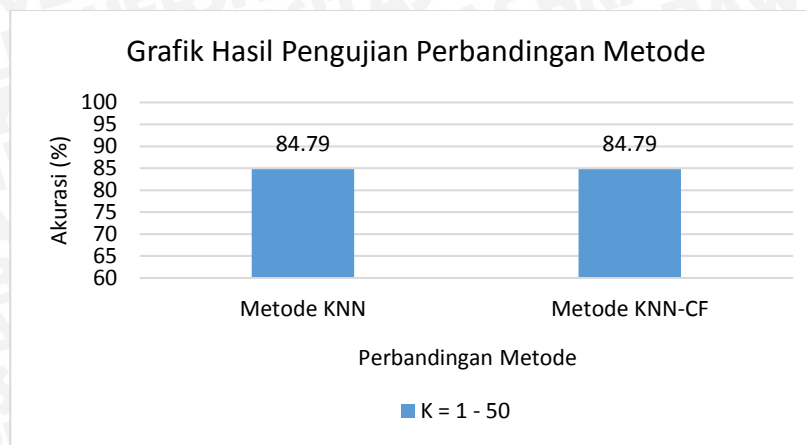


Gambar 6.5 Grafik hasil pengujian variasi data uji

Pada grafik Gambar 6.5 diatas dapat diketahui bahwa percobaan yang dilakukan dengan nilai $k=3$, $k=5$ dan $k=8$ percobaan ke-1 dan ke-2 menghasilkan akurasi 94.44%. Akurasi tertinggi dihasilkan oleh percobaan ke-3 dengan nilai $k=3$ dan $k=5$ sebesar 97.22, hal ini disebabkan karena meskipun variasi data uji yang digunakan berjumlah sama, akan tetapi variasi data ujinya berbeda. Dimana pada percobaan ke-3 memiliki variasi data dengan jarak yang lebih berdekatan dengan data latih dibandingkan dengan percobaan lainnya sehingga menghasilkan tingkat akurasi yang lebih tinggi.

6.2.6 Analisis Hasil Perbandingan Metode

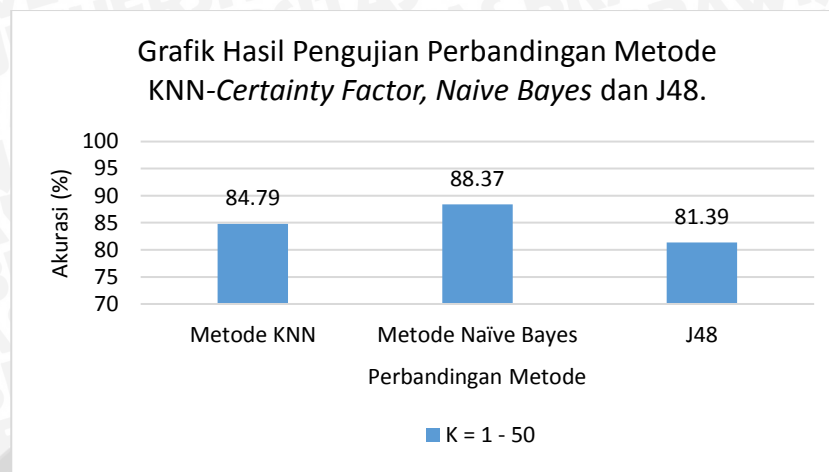
Pada analisis perbandingan metode ini digunakan nilai akurasi dari dua metode yaitu metode *K-Nearest Neighbor – Certainty Factor* dan metode *K-Nearest Neighbor* dengan acuan perbandingan rata-rata akurasi yang dihasilkan untuk setiap k . Pengujian ini menggunakan $k=1$ hingga $k=5$, 100 data latih dan jumlah data uji yang tetap yaitu sebanyak 43 data. Hasil pengujian perbandingan metode ditampilkan dalam sebuah grafik pada Gambar 6.6.



Gambar 6.6 Grafik hasil pengujian perbandingan metode

Dari grafik diatas dapat disimpulkan metode *K-Nearest Neighbor – Certainty Factor* memiliki tingkat akurasi yang sama dengan metode *K-Nearest Neighbor* yaitu sebesar 84.79%. Akurasi adalah menghitung hasil ketepatan diagnosis system dengan pakar dan hasil akurasi metode *K-Nearest Neighbor – Certainty Factor* mengikuti hasil klasifikasi *K-Nearest Neighbor*. Jadi dapat disimpulkan antara metode *K-Nearest Neighbor – Certainty Factor* dan *K-Nearest Neighbor* yang berperan dalam penentuan hasil akurasi adalah metode *K-Nearest Neighbor*, sedangkan metode *Certainty Factor* hanya menghitung nilai kepastian suatu penyakit berdasarkan hasil diagnosis yang dilakukan oleh metode *K-Nearest Neighbor*, apabila metode *K-Nearest Neighbor* tidak menghasilkan output maka metode *Certainty Factor* tidak dapat melakukan proses perhitungan nilai kepastian.

Untuk menguji kinerja metode *K-Nearest Neighbor – Certainty Factor* pada kasus Sistem Pakar Diagnosis Penyakit Demam: DBD, Malaria dan Tifoid maka dilakukan pengujian perbandingan dengan 2 metode klasifikasi lainnya yaitu *naïve bayes* dan J48 menggunakan Aplikasi Weka. *Naïve Bayes* adalah metode Bayesian learning yang paling cepat dan sederhana. Metode *Naive Bayes* merepresentasikan data gejala kedalam bentuk probability model. Probability model tersebut yang digunakan sebagai model untuk mengelompokkan suatu penyakit. Sedangkan metode *K-Nearest Neighbor* menyimpan sekelompok data training set sebagai acuan untuk mengelompokkan suatu penyakit. Metode J48 merupakan salah satu kelas di paket classifier pada sistem weka yang mengimplementasikan algoritma C45. Algoritma C4.5 adalah salah satu metode untuk membuat decision tree berdasarkan training data yang telah disediakan. Dalam pengujian ini data latih yang digunakan berjumlah 100 data dan 43 data uji. Hasil pengujian perbandingan metode *KNN-Certainty Factor*, *naïve bayes* dan J48 ditampilkan dalam sebuah grafik pada Gambar 6.7.



Gambar 6.7 Grafik hasil pengujian perbandingan metode KNN-CF, Naive Bayes dan J48

Dari grafik diatas dapat disimpulkan bahwa metode *K-Nearest Neighbor – Certainty Factor* lebih baik daripada metode J48 yang menghasilkan akurasi 81.39%. Hal ini dikarenakan metode J48 dalam membuat cabang node tree hanya terbatas sehingga informasi yang didapatkan kurang optimal dan membuat akurasi menjadi turun. Tetapi metode *K-Nearest Neighbor – Certainty Factor* memiliki akurasi kurang dari *Naive Bayes* yang menghasilkan akurasi sebesar 88.37%. Hal ini disebabkan karena metode *naive bayes* sudah mampu menentukan parameter yang diperlukan dalam proses pengklasifikasian hanya dengan jumlah data latih yang kecil atau sedikit sedangkan pada metode *K-Nearest Neighbor – Certainty Factor* diperlukan data latih yang besar sehingga hasil lebih efektif dan tangguh terhadap training data yang noisy.

BAB 7 PENUTUP

Pada bab ini akan membahas tentang kesimpulan dan saran dari peneliti berdasarkan hasil perancangan, implementasi dan pengujian sistem yang telah dilakukan.

7.1 Kesimpulan

Berdasarkan hasil penelitian Sistem Pakar Diagnosis Penyakit Demam: DBD, Malaria dan Tifoid Menggunakan Metode *K-Nearest Neighbor – Certainty Factor*, dapat diambil kesimpulan sebagai berikut:

1. Sistem Pakar Diagnosis Penyakit Demam: DBD, Malaria dan Tifoid Menggunakan Metode *K-Nearest Neighbor – Certainty Factor* ini dibuat menggunakan Bahasa pemrograman PHP dan basis data MySQL sebagai tempat penyimpanan data masukan dan data *knowledge*. Sistem pakar ini mampu mengidentifikasi penyakit Demam dengan menginputkan 15 gejala dan kemudian dilakukan proses perhitungan menggunakan metode *K-Nearest Neighbor – Certainty Factor* untuk mendapatkan hasil diagnosis berupa jenis penyakit sesuai dengan nilai kepastian terbesar.
2. Metode *K-Nearest Neighbor – Certainty Factor* menghasilkan tingkat akurasi yang tergolong baik dalam mengklasifikasikan dan memberi nilai kepastian jenis penyakit DBD, Malaria atau Tifoid dimana pada beberapa mekanisme pengujian menghasilkan tingkat akurasi yaitu:
 - a. Didapatkan nilai akurasi maksimum sebesar 88.37% pada pengujian variasi nilai k , dengan $k=3$, $k=5$ dan $k=8$. Dimana nilai akurasi maksimum diperoleh dari nilai K yang kecil. Semakin banyak nilai K maka akan membuat semakin banyak data tidak relevan terhadap data uji dan hal tersebut mengakibatkan tingkat kesalahan prediksi yang semakin besar dan menyebabkan tingkat akurasi turun.
 - b. Didapatkan nilai akurasi maksimum sebesar 86.04% pada pengujian variasi data latih, dimana perbedaan variasi data latih memberikan perbedaan tingkat akurasi. Hal ini disebabkan karena perbedaan tersebut akan mempengaruhi nilai jarak antara data latih dan data uji. Jika dalam suatu variasi memiliki nilai jarak yang berdekatan dengan data uji akan membuat akurasi semakin tinggi, begitupun sebaliknya.
 - c. Didapatkan hasil akurasi maksimum sebesar 95.83% pada pengujian rasio data latih dan data uji pada percobaan 2 dengan nilai $k=8$ dan pada percobaan ke-4 dengan nilai $k=2$. Hal ini dikarenakan jumlah kesalahan pada setiap percobaan dan banyaknya jumlah data uji pada setiap percobaan berpengaruh pada tingkat akurasi sistem. Jumlah data uji pada percobaan ke-3 lebih banyak maka tingkat akurasi pada percobaan ke-3 lebih besar meskipun memiliki jumlah kesalahan lebih besar daripada percobaan ke-2.

- d. Didapatkan hasil akurasi maksimum sebesar 90% pada pengujian variasi jumlah data uji dengan jumlah 20 data. Hal ini disebabkan karena jumlah kesalahan pada setiap percobaan dan banyaknya jumlah data uji pada setiap percobaan mempengaruhi tingkat akurasi yang dihasilkan.
 - e. Didapatkan hasil akurasi maksimum sebesar 97.22% pada pengujian variasi data uji dimana perbedaan variasi data uji memberikan perbedaan tingkat akurasi. Hal ini disebabkan karena perbedaan variasi data uji akan mempengaruhi nilai jarak antara data latih dan data uji. Jika dalam suatu variasi memiliki nilai jarak yang berdekatan dengan data latih akan menghasilkan nilai akurasi semakin tinggi, dan sebaliknya.
 - f. Didapatkan hasil akurasi maksimum sebesar 84.79% pengujian perbandingan metode. Perbandingan nilai akurasi antara metode *K-Nearest Neighbor – Certainty Factor* dengan metode *K-Nearest Neighbor* menghasilkan akurasi yang sama, karena metode *Certainty Factor* hanya menghitung nilai kepastian suatu penyakit berdasarkan hasil diagnosis dari metode *K-Nearest Neighbor*.
3. Perbandingan nilai akurasi metode antara *K-Nearest Neighbor – Certainty Factor* dengan *K-Nearest Neighbor* menghasilkan akurasi yang sama yaitu 84.79%. Untuk perbandingan dengan metode yang lainnya, metode *K-Nearest Neighbor – Certainty Factor* lebih baik daripada metode J48 yang menghasilkan akurasi 81.39%. Tetapi metode *K-Nearest Neighbor – Certainty Factor* memiliki akurasi kurang dari *Naïve Bayes* yang menghasilkan akurasi sebesar 88.37%.

7.2 Saran

Beberapa saran untuk pengembangan lebih lanjut yang dapat diberikan oleh penulis antara lain:

1. Sistem ini hanya untuk mendiagnosis 3 jenis penyakit demam (DBD, Malaria dan Tifoid) sedangkan masih terdapat banyak jenis demam sehingga dapat ditambahkan jenis-jenis demam yang lain.
2. Pada penelitian selanjutnya perlu dilakukan penambahan data latih dan data uji, solusi dan cara pencegahan penyakit Demam sehingga informasi dan data pembelajaran sistem semakin banyak.
3. Sebaiknya pada penelitian selanjutnya metode sistem pakar yang dapat digunakan adalah dengan menggunakan metode *Fuzzy – K-Nearest Neighbor* atau *Modified – K-Nearest Neighbor* dan penggabungan metode lainnya dengan *K-Nearest Neighbor*.

DAFTAR PUSTAKA

- Abdurrahman, Ginanjar. 2011. *Thesis Project: Penerapan Metode Tsukamoto (Logika Fuzzy) Dalam Sistem Pendukung Keputusan Untuk Menentukan Jumlah Produksi Barang Berdasarkan Data Persediaan dan Jumlah Permintaan*. Yogyakarta: Universitas Negeri Yogyakarta.
- Adam Rizal. 2014. Demam Berdarah Perlu Diwaspadai. Tersedia di: <http://www.sinarharapan.co/sehat/read/150113061/awas-dbd-pada-januari-2015> [Diakses 20 Agustus 2016]
- Agung. 2015. Demam Tifoid. Tersedia di: <http://ugm.ac.id/id/berita/10156teliti.demam.tifoid.nataniel.tan-dirogang.raih.doktor> [Diakses 20 Agustus 2016]
- Amir, Syarif. 2007. *Farmakologi dan Terapi*, Edisi V. Jakarta: Badan Penerbit FKUI.
- Anjas Sari, Nur. 2013. *Sistem Pakar Mendiagnosis Penyakit Demam Berdarah Menggunakan Metode Certainty Factor*. Medan: STMIK Budidarma Medan.
- Arhami, Muhammad. 2005. *Konsep Dasar Sistem Pakar*. Yogyakarta: Andi Offset.
- Arif, Son Wicaksana. 2009. *Sistem Pakar Identifikasi Hama dan Penyakit Tanaman Apel Berbasis Web*. Malang: Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang.
- Departemen Kesehatan. 2013. DBD Musim Pancaroba. Tersedia di: <http://www.depkes.go.id/article/print/15010200002/waspada-dbd-di-musim-pancaroba.html> [Diakses 28 Agustus 2016]
- Djunaedi, D. 2006. *Demam Berdarah Dengue (Dengue DBD) Epidemiologi, Imunopatologi, Patogenesis, Diagnosis dan Penatalaksanaannya*. Malang: UMM Press.
- Dr. Anda Citra Utama. 2016. Wawancara Sistem Pakar Diagnosis Penyakit Demam. Diwawancarai oleh Elsa Nur Amilus Shofia. 22 Maret 2016.
- Ernawati, dkk. 2014. *Komparasi Algoritma Fuzzy C-Means dan Algoritma K-Nearest Neighbor dalam Pengelompokan Rombongan Belajar Siswa Baru (Studi Kasus : Siswa Baru Madrasah Aliyah Negeri 01 Kota Bengkulu)*. Universitas Bengkulu. Jurnal Teknik Informatika. Vol. 7. No. 2.
- Hidayat, Syaeful. 2010. *Aplikasi Untuk Mendeteksi Penyakit Pada Tanaman Tebu dan Cara Penanganannya Berbasis Web*. Bandung: Universitas Komputer Indonesia.
- Hidayati, N. 2012. *Thesis Project: Analisis Regresi Poisson Terhadap Faktor-faktor yang Mempengaruhi Penyakit Demam Typhoid di Provinsi Jawa Timur*. Surabaya: FMIPA ITS.
- Hartati, Sri dan Sari Iswanti. 2008. *Sistem Pakar dan Pengembangannya*. Yogyakarta: Graha Ilmu.
- Iffah Nur Arifah. 2013. Pembunuh No. 1 di Papua. Tersedia di: <http://www.radioaustralia.net.au/indonesian/radio/onairhighlig>

<https://malaria-pembunuh-nomor-satu-di-papua/1121524>> [Diakses 20 Agustus 2016]

Kementrian Kesehatan RI. 2014. Laporan Peningkatan Kasus DBD di Jawa Timur. Tersedia di:

<http://www.depkes.go.id/article/view/15013000002/kemenkes-terima-laporan-peningkatan-kasus-dbd-di-jawa-timur.html>> [Diakses 28 Agustus 2016]

Kusrini. 2006. *Sistem Pakar "Teori dan Aplikasinya"*. Yogyakarta: Andi Offset.

Latumakulita, Luther. 2012. *Sistem Pakar Pendiagnosis Penyakit Anak Menggunakan Certainty Factor (CF)*. Manado: Jurnal Ilmiah Sains Vol. 12 No. 2.

Nugroho, Adi. 2011. *Perancangan dan Implementasi Sistem Basis Data*. Andi Offset. Yogyakarta

Paula, Yulianti. 2015. *Pengembangan Sistem Pakar Diagnosis Penyakit Tuberculosis dan Demam Berdarah Berbasis Web Menggunakan Metode Certainty Factor*. Yogyakarta: Seminar Nasional Teknologi Informasi dan Komunikasi.

Prasetyo, Eko. 2012. *Data Mining Konsep dan Aplikasi Menggunakan Matlab*. Yogyakarta: Andi Offset.

Sutojo, T., 2011. *Kecerdasan Buatan*. Yogyakarta: Andi Offset.

Turban, e., Aronson, J. E., & Liang, T. 2005. *Decision Support Systems and Intelligent Systems*. 7th Ed. Jilid 1. Yogyakarta: Andi Offset.

Ulya, Dzurratul. 2014. *Thesis Project: Sistem Pakar Diagnosis Penyakit Kulit Pada Anak Menggunakan Metode Certainty Factor*. Malang: Universitas Brawijaya.

Umar Fahmi, A. 2005. *Manajemen Penyakit Berbasis Wilayah*. Jakarta: Kompas.

Wardhani, Indira Kusuma. 2012. *Seleksi Supplier Bahan Baku dengan Metode TOPSIS Fuzzy MADM (Studi Kasus PT. Giri Sekar Kedaton, Gresik)*. Surabaya: Institut Teknologi Sepuluh Nopember.

LAMPIRAN 1

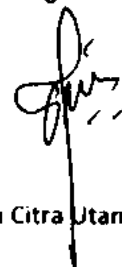
Hasil Wawancara

Wawancara dengan pakar Dr. Anda Citra Utama, SpPD dilakukan melalui media telepon (dikarenakan jarak pakar yang berada di Riau), pada tanggal 22 Maret 2016.

No.	Pertanyaan	Jawaban
1.	Apakah ada kendala yang dialami oleh dokter khususnya dokter internship di rumah sakit ini dok?	Ada. Karena keterbatasan pengalaman membuat kecepatan mendiagnosis penyakit sangat terbatas dan terkadang kurang akurat saat menegakkan diagnosis.
2.	Apakah sering terjadi kesalahan dalam penegakan diagnosis oleh dokter internship tersebut dok?	Kadang-kadang. Khususnya sebelum dilakukan pemeriksaan penunjang seperti tes laboratorium, usg, rontgen dan lain lain.
3.	Apa yang menyebabkan terjadinya kesalahan dalam penegakan diagnosis dok?	1. Karena beberapa penyakit memiliki beberapa gejala yang mirip 2. Tidak rinci dalam menggali gejala-gejala pasien
4.	Penyakit apa saja yang memiliki gejala yang sama dok?	Ada beberapa penyakit dengan keluhan demam yang memiliki gejala yang mirip seperti malaria, demam berdarah dengue dan tifoid. Ini menjadi kendala dalam menegakkan diagnosis dengan cepat dan dini.
5.	Saya ingin membuat sebuah sistem pakar untuk membantu penegakkan diagnosis penyakit, apakah sekiranya sistem ini membantu dan dibutuhkan tenaga medis dok?	Bisa membantu, terutama pada tempat-tempat yang belum tersedia pemeriksaan penunjang dan laboratorium seperti di puskesmas, puskesmas pembantu atau praktek pribadi.
6.	Apa saja tahapan dokter dalam menegakkan diagnosis?	1. Menganamesa (bertanya kepada pasien tentang keluhan dan riwayat penyakit). Pada tahap ini biasanya sudah bisa memperkirakan 60% diagnosis.

No.	Pertanyaan	Jawaban
		2. Melakukan pemeriksaan fisik dengan terlebih dahulu memeriksa tanda-tanda vital 3. Pemeriksaan seluruh tubuh 4. Melakukan pemeriksaan penunjang seperti pemeriksaan laboratorium, <i>rontgen</i> , <i>USG</i> dll.
7.	Apa saja parameter yang digunakan dalam menegakkan diagnosis?	Keluhan utama dan gejala yang dirasakan pasien, riwayat penyakit dan kelainan pada fisik.
8.	Pernahkah terjadi kesalahan dalam penegakkan diagnosis penyakit Demam?	Sering juga. Contohnya seorang pasien yang dianggap terkena penyakit tifoid karena gejala yang mirip tetapi setelah beberapa hari perawatan bahwa pasien tersebut terbukti malaria.
9.	Apa yang dimaksud dengan demam intermitten dok?	Demam intermitten merupakan demam yang terjadi tidak sepanjang hari dan tidak setiap hari. Ada fase bebas demam sehingga seakan-akan seperti tidak sakit.
10.	Apa yang dimaksud dengan bradikardi relative dok?	Bradikardi relative merupakan peningkatan frekuensi denyut nadi tidak sebanding dengan peningkatan suhu. Biasanya tiap kenaikan suhu 1 ^o , nadi meningkat 15 kali/menit.

Mengetahui,



Dr. Anda Citra Utama, SpPD

LAMPIRAN 2

Data Gejala Penyakit Demam

Wawancara dengan pakar Dr. Anda Citra Utama, SpPD dilakukan melalui media telepon (dikarenakan jarak pakar yang berada di Riau), pada tanggal 30 Maret 2016.

Kode Gejala	Gejala	Penyakit		
G1	Demam Intermittent (putus putus)		Malaria	
G2	Demam Menggigil	Tifoid	Malaria	DBD
G3	Demam terutama malam hari	Tifoid	Malaria	DBD
G4	Demam lebih dari 1 minggu	Tifoid	Malaria	DBD
G5	Sakit kepala	Tifoid	Malaria	DBD
G6	Sakit sakit tulang dan sendi	Tifoid	Malaria	DBD
G7	Mual dan Muntah	Tifoid		DBD
G8	Mencret atau susah BAB (konstipasi)	Tifoid		
G9	Nyeri perut	Tifoid		DBD
G10	Bintik merah (ptekie) pada kulit			DBD
G11	Lidah kotor (coated tongue)	Tifoid		
G12	Bradikardi relative	Tifoid		
G13	Pembesaran hati	Tifoid	Malaria	DBD
G14	Pembesaran limpa	Tifoid	Malaria	DBD
G15	Kulit lembab/keringat		Malaria	

Mengetahui,



Dr. Anda Citra Utama, SpPD

LAMPIRAN 3

Data Nilai Bobot Gejala Penyakit Demam

Wawancara dengan pakar Dr. Anda Citra Utama, SpPD dilakukan melalui media telepon (dikarenakan jarak pakar yang berada di Riau), pada tanggal 30 Maret 2016.

NO	VARIABEL	SKOR
I	KELUHAN	
G1	Demam Intermittent (putus putus)	
	Ya	0.8
	Tidak	0
G2	Demam menggigil	
	Berat	0.85
	Sedang	0.5
	Tidak ada	0
G3	Demam terutama malam hari	
	Ya	0.75
	Tidak	0
G4	Lama demam	
	> 7 hari	0.75
	4 - 7 hari	0.5
	1 - 3 hari	0.2
G5	Sakit Kepala	
	Berat	0.9
	Sedang	0.7
	Tidak ada	0
G6	Sakit sakit tulang dan sendi	
	Berat	0.9
	Sedang	0.75
	Tidak ada	0
G7	Mual dan muntah	
	Berat	0.85
	Sedang	0.6
	Tidak ada	0
G8	Mencret atau susah BAB (konstipasi)	
	Ada	0.6
	Tidak ada	0.2
G9	Nyeri Perut	

NO	VARIABEL	SKOR
I	KELUHAN	
	Berat	0.8
	Sedang	0.4
	Tidak ada	0
II	PEMERIKSAAN FISIK	
G10	Bintik merah (ptekie) pada kulit	
	Berat	1
	Sedang	0.7
	Tidak ada	0
G11	Lidah kotor (coated tongue)	
	Berat	0.85
	Sedang	0.4
	Tidak ada	0
G12	Bradikardi relatif	
	Ya	0.75
	Tidak ada	0
G13	Pembesaran hati	
	Ya	0.5
	Tidak ada	0
G14	Pembesaran limpa	
	Ya	0.65
	Tidak ada	0
G15	Kulit lembab/ Keringat	
	Berat	0.85
	Sedang	0.4
	Tidak ada	0

Mengetahui,



Dr. Anda Citra Utama, SpPD

LAMPIRAN 4 Data Latih

Wawancara dengan pakar Dr. Anda Citra Utama, SpPD dilakukan melalui media telepon (dikarenakan jarak pakar yang berada di Riau), pada tanggal 30 Maret 2016.

PASIE	GEJALA															PENYAKIT
	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14	G15	
1	0	0.5	0.75	0.75	0.9	0.75	0.85	0.6	0.8	0	0.85	0.75	0.5	0.65	0	1
2	0	0	0.75	0.5	0	0.75	0.6	0.6	0	0	0	0.75	0.5	0.65	0	1
3	0	0.5	0.75	0.5	0.9	0	0.6	0.6	0.8	0	0.85	0.75	0.5	0.65	0	1
4	0	0.5	0.75	0.5	0.7	0.75	0.85	0.6	0.4	0	0.85	0	0	0.65	0	1
5	0	0.5	0	0.5	0.7	0	0	0.2	0.8	0	0.85	0.75	0.5	0.65	0	1
6	0	0	0.75	0.75	0.9	0	0.6	0.6	0.8	0	0	0.75	0.5	0	0	1
7	0	0.5	0.75	0.75	0.7	0	0.6	0.2	0.8	0.7	0.85	0.75	0.5	0.65	0	1
8	0	0	0.75	0.75	0.9	0.75	0	0.6	0.8	0	0.85	0.75	0.5	0	0	1
9	0	0	0	0.75	0.7	0.75	0.85	0.6	0	0	0.4	0.75	0.5	0.65	0	1
10	0	0	0	0.75	0.9	0	0.6	0.6	0.4	0	0.85	0.75	0.5	0	0	1
11	0	0.5	0.75	0.75	0.7	0	0.6	0.2	0.8	0	0.4	0.75	0.5	0.65	0	1
12	0	0	0	0.5	0.9	0	0.6	0.6	0.8	0	0.85	0.75	0.5	0	0	1
13	0	0	0.75	0.5	0.9	0	0.85	0.6	0.4	0	0.85	0.75	0.5	0.65	0	1
14	0	0.5	0	0.5	0.9	0.75	0.6	0.2	0.8	0	0.85	0	0.5	0.65	0.45	1
15	0	0	0.75	0.75	0.9	0	0.6	0.6	0.8	0	0	0.75	0.5	0	0	1
16	0	0.5	0.75	0.75	0.7	0	0.6	0.2	0.8	0.7	0.85	0.75	0.5	0.65	0	1
17	0	0	0.75	0.75	0.9	0.75	0	0.6	0.8	0	0.85	0.75	0.5	0	0	1

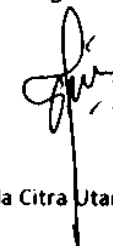
PASIEAN	GEJALA															PENYAKIT
	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14	G15	
18	0	0	0	0.75	0.7	0.75	0.85	0.6	0	0	0.4	0.75	0.5	0.65	0	1
19	0	0	0	0.5	0.7	0	0.6	0.6	0.4	0	0.4	0	0	0	0	1
20	0	0	0.75	0.75	0.9	0.75	0	0.2	0.8	0	0.85	0.75	0.5	0.65	0	1
21	0	0	0.75	0.75	0.9	0	0.85	0.6	0.8	0	0.85	0.75	0	0	0	1
22	0	0	0.75	0.75	0.9	0	0.85	0.2	0.8	0	0.4	0.75	0.5	0.65	0.45	1
23	0	0	0.75	0.75	0.7	0	0.85	0.6	0.4	0	0.85	0.75	0.5	0.65	0	1
24	0	0	0.75	0.75	0.7	0	0.85	0.6	0.8	0	0.85	0	0.5	0.65	0	1
25	0	0	0	0.75	0.9	0	0.85	0.6	0	0	0	0.75	0	0	0	1
26	0	0	0	0.75	0.7	0.75	0	0.6	0.8	0	0.85	0	0	0.65	0	1
27	0	0	0.75	0.75	0.7	0.75	0.85	0.6	0.8	0	0	0.75	0	0	0	1
28	0	0	0.75	0.75	0.7	0	0.85	0.6	0.4	0	0.85	0.75	0.5	0.65	0	1
29	0	0	0	0.5	0.9	0	0.85	0.6	0.4	0	0.4	0.75	0.5	0	0	1
30	0	0	0	0.75	0.9	0	0.85	0.6	0	0	0	0.75	0	0	0	1
31	0.8	0.85	0	0.5	0.9	0	0	0.2	0	0	0	0	0.5	0.65	0.85	2
32	0.8	0.85	0	0.5	0.7	0	0	0.2	0.4	0	0	0	0.5	0.65	0.85	2
33	0.8	0.5	0	0.5	0.9	0.75	0	0.2	0	0	0	0	0.5	0.65	0.85	2
34	0.8	0.85	0	0.5	0.7	0	0	0.2	0	0	0	0	0.5	0.65	0.85	2
35	0.8	0.85	0	0.5	0.7	0	0	0.2	0	0	0	0	0	0	0.4	2
36	0.8	0.85	0	0.5	0.9	0	0	0.2	0	0	0	0	0	0	0.4	2
37	0	0.85	0	0.2	0.9	0	0	0.2	0	0	0	0	0	0	0.4	2
38	0.8	0.85	0	0.5	0.9	0	0	0.2	0	0	0	0	0	0	0.4	2
39	0	0.85	0	0.5	0.9	0	0	0.2	0	0	0	0	0	0	0.4	2
40	0.8	0.85	0	0.5	0.7	0	0	0.2	0	0	0	0	0.5	0.65	0.85	2

PASIE	GEJALA															PENYAKIT
	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14	G15	
41	0.8	0.85	0	0.5	0.9	0	0	0.2	0	0	0	0	0.5	0.65	0.85	2
42	0.8	0.85	0	0.5	0.7	0	0	0.2	0.4	0	0	0	0.5	0.65	0.85	2
43	0.8	0.5	0	0.5	0.9	0.75	0	0.2	0	0	0	0	0.5	0.65	0.85	2
44	0.8	0.85	0	0.5	0.7	0	0	0.2	0	0	0	0	0.5	0.65	0.85	2
45	0.8	0.85	0	0.2	0.9	0	0	0.2	0	0	0	0	0.5	0.65	0.85	2
46	0.8	0.85	0	0.5	0.7	0.75	0	0.2	0	0	0	0	0.5	0.65	0.85	2
47	0.8	0.85	0	0.5	0.7	0	0	0.2	0	0	0	0	0.5	0.65	0.85	2
48	0.8	0.5	0	0.5	0.7	0	0	0.2	0	0	0.4	0	0.5	0.65	0.85	2
49	0.8	0.85	0	0.5	0.7	0	0	0.2	0.4	0	0	0	0.5	0.65	0.85	2
50	0.8	0.5	0	0.5	0.7	0	0	0.2	0	0	0	0	0	0.65	0.4	2
51	0.8	0.85	0	0.5	0.7	0	0	0.2	0	0	0	0	0	0.65	0.4	2
52	0	0.85	0	0.5	0.7	0	0	0.2	0	0	0	0	0	0	0.4	2
53	0.8	0.85	0	0.5	0.9	0	0	0.2	0	0	0	0	0	0	0.4	2
54	0.8	0.85	0.75	0.2	0.7	0	0.6	0.2	0	0	0	0	0	0	0.4	2
55	0	0.85	0.75	0.5	0.7	0	0	0.2	0	0	0	0	0.5	0.65	0.85	2
56	0.8	0.85	0	0.5	0.9	0.75	0	0.2	0.4	0	0	0	0	0	0.4	2
57	0.8	0.5	0	0.5	0.7	0	0	0.6	0	0	0	0	0.5	0.65	0.85	2
58	0.8	0.85	0	0.5	0.9	0	0	0.2	0	0	0	0	0.5	0.65	0.85	2
59	0.8	0.85	0	0.5	0.9	0	0.6	0.2	0	0	0	0	0.5	0.65	0.85	2
60	0	0.85	0.75	0.5	0.7	0	0	0.2	0	0	0	0	0.5	0.65	0.85	2
61	0.8	0.85	0	0.2	0.9	0	0	0.2	0	0	0	0	0.5	0.65	0.85	2
62	0	0.85	0	0.5	0.9	0	0	0.2	0	0	0	0	0.5	0.65	0.85	2
63	0.8	0.85	0	0.5	0.9	0	0	0.6	0	0	0	0	0.5	0.65	0.4	2

PASIE	GEJALA															PENYAKIT
	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14	G15	
64	0.8	0.85	0	0.5	0.9	0	0.6	0.2	0	0	0	0	0	0.65	0.4	2
65	0.8	0.85	0	0.2	0.9	0.75	0	0.2	0	0	0	0	0	0.65	0.4	2
66	0.8	0.85	0	0.5	0.7	0	0	0.2	0	0	0	0	0	0	0.4	2
67	0.8	0.85	0	0.5	0.7	0	0	0.2	0	0	0	0	0	0	0.4	2
68	0.8	0.85	0	0.5	0.9	0	0	0.2	0	0	0	0	0	0	0.4	2
69	0.8	0.85	0	0.2	0.7	0	0	0.2	0	0	0	0	0.5	0.65	0.85	2
70	0	0.5	0.75	0.5	0.7	0.75	0.6	0.2	0.8	0.7	0.4	0	0.5	0.65	0	3
71	0	0	0	0.2	0.9	0.9	0.85	0.2	0.4	0.7	0	0	0	0.65	0	3
72	0	0	0	0.2	0.7	0.9	0	0.2	0.4	0.7	0	0	0	0.65	0	3
73	0	0.5	0	0.2	0.9	0.9	0.85	0.2	0.4	0.7	0.4	0	0.5	0	0	3
74	0	0	0	0.2	0.7	0.9	0.85	0.2	0.8	1	0	0	0.5	0	0	3
75	0	0	0	0.2	0.9	0.9	0.85	0.2	0.8	1	0	0	0.5	0	0	3
76	0	0	0	0.2	0	0.9	0.6	0.2	0.4	0.7	0	0.75	0	0	0	3
77	0	0	0	0.2	0.9	0.9	0.6	0.2	0.4	1	0	0	0	0.65	0.4	3
78	0	0	0	0.5	0.9	0.9	0.6	0.2	0.4	1	0.4	0	0	0	0	3
79	0	0	0	0.2	0.9	0.75	0.6	0.2	0.8	0.7	0	0	0.5	0	0	3
80	0	0.5	0	0.5	0.9	0.9	0.85	0.2	0.8	0.7	0	0	0.5	0.65	0	3
81	0	0	0	0.2	0.9	0.9	0.85	0.2	0.8	0	0	0	0.5	0	0	3
82	0	0	0	0.2	0.9	0.75	0.85	0.2	0.4	1	0	0	0	0	0	3
83	0	0.5	0	0.5	0.9	0.9	0.85	0.2	0.4	0.7	0	0	0	0	0	3
84	0	0	0	0.2	0.9	0.9	0.85	0.2	0.4	1	0.4	0	0	0	0	3
85	0	0	0	0.2	0.9	0.9	0.85	0.2	0.4	0.7	0	0	0	0	0.4	3
86	0	0	0	0.5	0.9	0.75	0.85	0.2	0.4	0.7	0	0	0.5	0	0	3

PASIEN	GEJALA															PENYAKIT
	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14	G15	
87	0	0	0	0.2	0.9	0.75	0.85	0.2	0.8	1	0	0	0.5	0.65	0	3
88	0	0	0	0.2	0.9	0.75	0.85	0.2	0.8	1	0	0	0	0	0	3
89	0	0	0	0.5	0.9	0.9	0	0.2	0.8	0.7	0	0	0.5	0	0	3
90	0	0	0	0.2	0.9	0.9	0.6	0.2	0.4	0.7	0	0	0.5	0	0	3
91	0	0.5	0	0.2	0.9	0.75	0	0.2	0.4	1	0	0.75	0.5	0	0	3
92	0	0	0	0.5	0.9	0.9	0	0.2	0.4	0.7	0	0	0	0	0	3
93	0	0	0	0.2	0.9	0.9	0.85	0.2	0	0.7	0.4	0	0	0	0	3
94	0	0	0.75	0.2	0.9	0.9	0.6	0.2	0.8	0.7	0	0	0	0.65	0	3
95	0	0.5	0	0.5	0.9	0.75	0.6	0.2	0.4	0.7	0	0	0	0	0	3
96	0	0	0	0.2	0.9	0.9	0.85	0.2	0.4	1	0	0	0	0	0	3
97	0	0	0	0.2	0.9	0.9	0	0.2	0.4	1	0	0	0	0	0	3
98	0	0	0	0.2	0.9	0.9	0.85	0.2	0.4	0.7	0	0	0	0	0.4	3
99	0	0	0	0.2	0.9	0.9	0.6	0.2	0.4	1	0	0	0	0	0	3
100	0	0	0	0.2	0.9	0.75	0.85	0.2	0.8	1	0	0	0.5	0.65	0	3

Mengetahui,



Dr. Anda Citra Utama, SpPD

LAMPIRAN 5

Data Uji

Wawancara dengan pakar Dr. Anda Citra Utama, SpPD dilakukan melalui media telepon (dikarenakan jarak pakar yang berada di Riau), pada tanggal 30 Maret 2016.

PASIE	GEJALA															PENYAKIT
	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14	G15	
101	0	0	0	0.5	0.9	0	0.85	0.6	0.4	0	0.4	0.75	0.5	0	0	1
102	0	0	0.75	0.5	0.9	0	0.85	0.2	0.8	0	0.85	0	0.5	0.65	0.45	1
103	0	0	0.75	0.75	0.9	0	0	0.6	0.4	0.7	0.4	0.75	0	0.65	0	1
104	0	0.5	0	0.75	0.9	0	0.85	0.2	0.8	0	0.85	0	0	0.65	0	1
105	0	0	0.75	0.75	0.7	0	0.85	0.6	0.8	0	0.85	0	0.5	0.65	0	1
106	0	0	0.75	0.5	0.9	0	0.85	0.2	0.8	0	0.85	0	0.5	0.65	0.45	1
107	0.8	0.85	0	0.5	0.9	0	0.6	0.2	0	0	0	0	0.5	0.65	0.85	2
108	0.8	0.5	0	0.5	0.7	0	0	0.2	0	0	0.4	0	0	0	0.4	2
109	0.8	0.5	0	0.5	0.9	0	0.6	0.2	0	0	0	0	0	0	0.4	2
110	0.8	0.85	0	0.5	0.9	0	0	0.2	0	0	0	0	0	0	0.4	2
111	0.8	0.5	0	0.2	0.7	0.75	0.6	0.6	0	0	0.4	0	0	0.65	0.4	2
112	0.8	0.85	0	0.2	0.7	0	0	0.2	0	0	0	0	0.5	0.65	0.85	2
113	0.8	0.85	0	0.5	0.9	0	0.6	0.2	0	0	0	0	0.5	0.65	0.85	2
114	0	0	0	0.2	0.9	0.9	0.6	0.2	0.4	0.7	0	0	0.5	0.65	0	3
115	0	0	0	0.2	0.7	0.9	0.6	0.2	0.8	0	0	0	0.5	0	0	3
116	0	0	0	0.2	0.9	0.9	0.85	0.2	0.8	1	0	0	0.5	0	0	3
117	0	0	0.75	0.5	0.9	0.9	0	0.2	0.4	1	0	0	0	0	0	3

PASIE	GEJALA															PENYAKIT
	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14	G15	
118	0	0	0	0.2	0.7	0.9	0.85	0.2	0.4	0.7	0	0	0	0	0	3
119	0	0	0	0.2	0.9	0.9	0	0.2	0.4	1	0	0	0	0.65	0	3
120	0	0	0	0.2	0.9	0.9	0.85	0.2	0.4	0.7	0	0	0	0.65	0	3
121	0	0	0	0.2	0.7	0.9	0	0.2	0.4	0.7	0	0	0	0.65	0	3
122	0	0.5	0	0.2	0.9	0.9	0.85	0.2	0.4	0.7	0.4	0	0.5	0	0	3
123	0	0	0	0.2	0.7	0.9	0.85	0.2	0.8	1	0	0	0.5	0	0	3
124	0	0	0	0.2	0.7	0.9	0.85	0.2	0.8	1	0	0	0.5	0	0	3
125	0	0	0	0.2	0.9	0.75	0.6	0.2	0.8	0.7	0	0	0.5	0	0.4	3
126	0	0	0	0.2	0.9	0.9	0.6	0.2	0.4	1	0	0	0.5	0	0	3
127	0	0	0	0.5	0.9	0.75	0.85	0.2	0.4	0.7	0	0	0.5	0	0	3
128	0	0	0	0.2	0.9	0.9	0	0.2	0.4	1	0	0	0	0.65	0	3
129	0	0	0	0.2	0.7	0.9	0	0.2	0.4	0.7	0	0	0	0	0	3
130	0	0	0	0.5	0.9	0.75	0.85	0.2	0.8	1	0	0	0.5	0.65	0	3
131	0	0	0	0.5	0.9	0.9	0.85	0.2	0	1	0	0	0	0.65	0	3
132	0	0	0	0.2	0.9	0.9	0.85	0.2	0	1	0	0	0	0	0	3
133	0	0	0	0.2	0.9	0.9	0.85	0.2	0	1	0	0	0	0	0	3
134	0.8	0.5	0.75	0.75	0	0	0.85	0.2	0	0	0.85	0.75	0.5	0.65	0	1
135	0.8	0.5	0.75	0.75	0	0	0.6	0.6	0	0.7	0	0	0	0	0.45	3
136	0.8	0.85	0	0.5	0	0	0	0.2	0	0	0	0	0.5	0.65	0.85	1
137	0	0.85	0.75	0.2	0	0.75	0.6	0.2	0	0	0	0	0	0	0.4	3
138	0.8	0.5	0	0.2	0	0.9	0.85	0.2	0.4	0.7	0.4	0	0.5	0	0	2
139	0.8	0	0.75	0.2	0.7	0.9	0	0.2	0.4	1	0	0	0	0.65	0.85	3
140	0	0.85	0	0.75	0.9	0	0.6	0.6	0.8	0	0	0.75	0.5	0	0	1

PASIEN	GEJALA															PENYAKIT
	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14	G15	
141	0	0	0	0.5	0.7	0	0.6	0.6	0.4	0.7	0.4	0	0	0	0	1
142	0	0	0	0.5	0.9	0.9	0.85	0.2	0.8	1	0.4	0	0	0	0	2
143	0	0.85	0.75	0.2	0.7	0.75	0	0.2	0	0	0	0	0.5	0.65	0.85	1

Mengetahui,



Dr. Anda Citra Utama, SpPD

LAMPIRAN 6

Hasil Pengujian Dengan Aplikasi Weka

1. Hasil Pengujian dengan metode Naïve Bayes menghasilkan akurasi 88.37%.

```
Classifier output

=== Evaluation on test set ===

Time taken to test model on supplied test set: 0.01 seconds

=== Summary ===

Correctly Classified Instances      38      88.3721 %
Incorrectly Classified Instances    5      11.6279 %
Kappa statistic                    0.8056
Mean absolute error                0.0788
Root mean squared error            0.276
Relative absolute error             17.4877 %
Root relative squared error        57.5948 %
Coverage of cases (0.95 level)     88.3721 %
Mean rel. region size (0.95 level) 34.1085 %
Total Number of Instances          43

=== Detailed Accuracy By Class ===

          TP Rate  FP Rate  Precision  Recall  F-Measure  MCC      ROC Area  PRC Area  Class
          0.818   0.031   0.900     0.818   0.857     0.813   0.932   0.907   Tifoid
          0.778   0.059   0.778     0.778   0.778     0.719   0.820   0.756   Malaria
          0.957   0.100   0.917     0.957   0.936     0.860   0.943   0.948   DBD
Weighted Avg.   0.884   0.074   0.883     0.884   0.883     0.819   0.915   0.897

=== Confusion Matrix ===

 a  b  c  <-- classified as
 9  2  0  |  a = Tifoid
 0  7  2  |  b = Malaria
 1  0  22 |  c = DBD
```

2. Hasil pengujian dengan metode J48 menghasilkan akurasi 81.39%.

```
Classifier output

=== Evaluation on test set ===

Time taken to test model on supplied test set: 0.01 seconds

=== Summary ===

Correctly Classified Instances      35      81.3953 %
Incorrectly Classified Instances    8      18.6047 %
Kappa statistic                    0.6929
Mean absolute error                0.1263
Root mean squared error            0.3507
Relative absolute error             28.0501 %
Root relative squared error        73.1841 %
Coverage of cases (0.95 level)     81.3953 %
Mean rel. region size (0.95 level) 33.3333 %
Total Number of Instances          43

=== Detailed Accuracy By Class ===

          TP Rate  FP Rate  Precision  Recall  F-Measure  MCC      ROC Area  PRC Area  Class
          0.727   0.063   0.800     0.727   0.762     0.687   0.830   0.638   Tifoid
          0.778   0.088   0.700     0.778   0.737     0.664   0.845   0.591   Malaria
          0.870   0.150   0.870     0.870   0.870     0.720   0.862   0.834   DBD
Weighted Avg.   0.814   0.115   0.816     0.814   0.814     0.699   0.850   0.733

=== Confusion Matrix ===

 a  b  c  <-- classified as
 8  2  1  |  a = Tifoid
 0  7  2  |  b = Malaria
 2  1  20 |  c = DBD
```