

# IMPLEMENTASI METODE F-KNN (*FUZZY K-NEAREST NEIGHBOR*) UNTUK DIAGNOSIS PENYAKIT ANJING

## SKRIPSI

Untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Komputer

Disusun oleh:

Dizka Maryam Febri Shanti

NIM: 115060807113028



PROGRAM STUDI TEKNIK INFORMATIKA  
JURUSAN TEKNIK INFORMATIKA  
FAKULTAS ILMU KOMPUTER  
UNIVERSITAS BRAWIJAYA  
MALANG  
2018

## PENGESAHAN

IMPLEMENTASI METODE F-KNN (*FUZZY K-NEAREST NEIGHBOR*) UNTUK  
DIAGNOSIS PENYAKIT ANJING

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Komputer

Disusun Oleh :  
Dizka Maryam Febri Shanti  
NIM: 115060807113028

Skripsi ini telah diuji dan dinyatakan lulus pada  
03 Agustus 2018

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II

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

Randy Cahya Wihandika, S.ST., M.Kom  
NIK: 201405 880206 1 001

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, 03 Agustus 2018



Dizka Maryam Febri Shanti

NIM: 115060807113028



## KATA PENGANTAR

Segala puji syukur atas Allah SWT sehingga penulis dapat menyelesaikan skripsi “Implementasi Metode *Fuzzy K-Nearest Neighbor* (FK-NN) Untuk Diagnosis Penyakit Anjing” dengan baik. Penulis ingin mengucapkan terima kasih kepada semua pihak yang telah membantu penulis dalam menyelesaikan skripsi ini. Ucapan terima kasih penulis berikan kepada:

1. Nurul Hidayat, S.Pd, M.Sc selaku dosen pembimbing pertama yang telah membimbing, memberikan saran dan motivasi kepada penulis.
2. Randy Cahya Wihandika, S.ST., M.Kom selaku dosen pembimbing kedua yang telah membimbing, memberikan saran dan motivasi kepada penulis.
3. Seluruh dosen dan civitas akademika Fakultas Ilmu Komputer Universitas Brawijaya yang telah memberikan ilmu, wawasan, dukungan dan bantuan kepada penulis.
4. Kedua orang tua penulis, kedua adik, dan keluarga penulis yang telah memberikan doa, semangat, dukungan dan pengertian kepada penulis.
5. Teman-teman penulis yang telah banyak memerikan bantuan dan inspirasi kepada penulis.
6. Klinik Hewan Yudisthira yang telah bersedia memberikan data yang diperlukan dalam penyusunan skripsi.
7. Semua pihak yang tidak dapat disebutkan satu persatu yang telah membantu penulis menyelesaikan skripsi ini.

Penulis menyadari masih banyak terdapat kesalahan dalam penyusunan skripsi ini. Kritik dan saran penulis harapkan untuk pengembangan penelitian selanjutnya. Adapun harapan penulis skripsi ini dapat memberikan manfaat bagi pembaca.

Malang, 03 Agustus 2018

Penulis

dizkamfs@gmail.com

## ABSTRAK

Anjing adalah salah satu hewan yang banyak dijadikan peliharaan. Ketika membelai (*petting*) dan bermain dengan hewan peliharaan, oksitosin, hormon yang berkaitan dengan stres dan rasa lega dilepaskan, membantu menurunkan tekanan darah juga level kortisol. Meskipun memelihara anjing memiliki banyak manfaat, para pemilik harus telaten dalam merawat anjing mereka. Tidak sedikit anjing yang terserang berbagai macam penyakit yang disebabkan oleh virus, protozoa, bakteri dan parasit. Anjing yang sakit apabila tidak segera mendapatkan perawatan dan pengobatan memiliki resiko dapat menularkan kepada anjing dan hewan lainnya atau bahkan kepada manusia. Metode yang digunakan *Fuzzy K-Nearest Neighbor*. (FK-NN) ialah varian metode K-Nearest Neighbor (K-NN) dengan teknik fuzzy. Metode FK-NN memberikan nilai keanggotaan kelas pada vektor sampel dan bukan menempatkan vektor pada kelas tertentu. FK-NN dapat diimplementasikan untuk diagnosis penyakit pada anjing dengan beberapa tahap yaitu menghitung jarak antara data latih dengan data uji, mengambil jarak terkecil antara data latih dengan data uji sebanyak K, Fuzzifikasi dan Defuzzifikasi, Kelas dengan nilai defuzzifikasi tertinggi adalah yang dijadikan sebagai kelas untuk hasil klasifikasi. Nilai K mempengaruhi tingkat akurasi sistem dimana semakin tinggi nilai k maka terjadi kecenderungan akurasi akan semakin turun. Akurasi tertinggi yang didapatkan dari hasil pengujian adalah ketika  $K = 5$  yaitu dengan nilai 98,67%.

**Kata kunci:** *Anjing, Penyakit Anjing, Fuzzy, Fuzzy K-Nearest Neighbor.*

## ABSTRACT

*Dogs are one of the favorite animals used as pets. When petting and playing with dogs, oxytocin, stress-related and relieved hormones are released, helping to lower blood pressure as well as cortisol levels. Although dog maintenance has many benefits, the owners should be careful in caring for their dogs. Not a few dogs are attacked by various diseases caused by viruses, protozoa, bacteria and parasites. Dogs who are sick if not immediately get treatment and treatment have the risk of transmitting to dogs and other animals or even to humans. The method used Fuzzy K-Nearest Neighbor. (FK-NN) is a variant of K-Nearest Neighbor (K-NN) method with fuzzy technique. The FK-NN method assigns a class membership value to the sample vector instead of placing the vector in a particular class. FK-NN can be implemented for the diagnosis of diseases in dogs by several stages: calculating the distance between the train data and the test data, taking the smallest distance between the train data and the test data as much as K, Fuzzification and Defuzzification, Class with the highest defuzzification value used as the class for the result classification. The value of K affects the accuracy of the system where the higher the value of k then the tendency of accuracy will decrease. The highest accuracy obtained from the test results is when K = 5 ie with a value of 98.67%.*

**Keywords:** Dog, Dog's Disease, Fuzzy, Fuzzy K-Nearest Neighbor.

## DAFTAR ISI

IMPLEMENTASI METODE F-KNN ( <i>FUZZY K-NEAREST NEIGHBOR</i> ) UNTUK DIAGNOSIS PENYAKIT ANJING .....	i
PENGESAHAN .....	ii
PERNYATAAN ORISINALITAS .....	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	v
ABSTRACT.....	vi
DAFTAR ISI.....	vii
DAFTAR TABEL.....	x
DAFTAR GAMBAR.....	xi
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.....	4
BAB 2 LANDASAN KEPUSTAKAAN .....	5
2.1 Anjing .....	5
2.2 Penyakit-Penyakit Anjing.....	5
2.3 Kajian Pustaka .....	6
2.3.1 <i>Demodicosis</i> .....	7
2.3.2 <i>Scabiosis</i> .....	8
2.3.3 <i>Otitis</i> .....	8
2.3.4 <i>Ringworm</i> .....	9
2.3.5 <i>Helminthiasis</i> .....	9
2.3.6 <i>Canine Parvovirus</i> .....	9
2.3.7 <i>Canine Distemper</i> .....	10
2.3.8 <i>Ehrlichiosis</i> .....	10
2.3.9 <i>Coccidiosis</i> .....	11
2.3.10 <i>Urolithiasis</i> .....	11



2.4 Data Mining .....	12
2.5 Klasifikasi.....	13
2.6 K-Nearest Neighbor (K-NN).....	14
2.7 Proses K-Nearest Neighbor.....	14
2.8 Logika Fuzzy .....	15
2.9 Himpunan Fuzzy.....	15
2.10 Fuzzy K-Nearest Neighbor (FK-NN).....	16
2.11 Pengujian Akurasi .....	17
<b>BAB 3 METODOLOGI .....</b>	<b>18</b>
3.1 Studi Literatur .....	18
3.2 Analisis Kebutuhan .....	19
3.3 Pengumpulan Data .....	19
3.4 Perancangan Sistem .....	19
3.5 Implementasi Sistem .....	20
3.6 Pengujian dan Analisis .....	20
3.7 Kesimpulan dan Saran .....	21
<b>BAB 4 PERANCANGAN.....</b>	<b>22</b>
4.1 Struktur Fuzzy K-Nearest Neighbor (FK-NN).....	22
4.1.1 Proses K-Nearest Neighbor (K-NN) .....	23
4.1.2 Proses Fuzzy K-Nearest Neighbor (FKNN).....	25
4.2 Contoh Perhitungan Manual .....	27
4.2.1 Hitung Euclidian Distance Data Latih Terhadap Data Uji.....	27
4.2.2 Sorting Berdasarkan Euclidian Distance .....	29
4.2.3 Pemilihan data sebanyak nilai K .....	31
4.2.4 Fuzzyfikasi .....	32
4.2.5 Defuzzyfikasi .....	33
4.3 Perancangan Antarmuka .....	34
4.3.1 Perancangan Antarmuka Data Latih .....	34
4.3.2 Perancangan Antarmuka Data Uji.....	35
4.3.3 Perancangan Antarmuka Proses FKNN .....	36
<b>BAB 5 IMPLEMENTASI .....</b>	<b>38</b>
5.1 Implementasi sistem.....	38



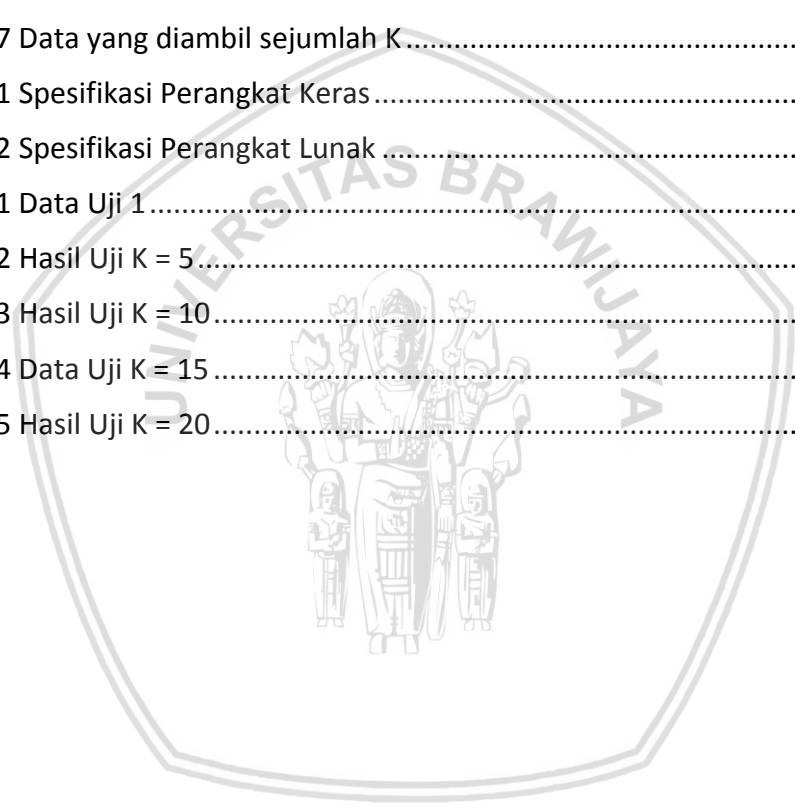


5.1.1 Spesifikasi Perangkat Keras.....	38
5.1.2 Spesifikasi Perangkat Lunak .....	38
5.2 Batasan Implementasi .....	38
5.3 Implementasi Algoritma .....	39
5.3.1 Implementasi Perhitungan Jarak Euclidean Data Latih dan Data Uji .....	39
5.3.2 Implementasi Pengurutan Jarak Euclidean.....	40
5.3.3 Implementasi Pengambilan Hasil Penghitungan Jarak Euclidean Sebanyak K .....	41
5.3.4 Implementasi Fuzzifikasi .....	42
5.3.5 Implementasi Defuzzifikasi .....	44
5.3.6 Implementasi Penentuan Kelas .....	51
5.4 Implementasi Antarmuka .....	53
5.4.1 Implementasi Antarmuka Awal .....	53
5.4.2 Implementasi Antarmuka Proses FKNN.....	55
5.4.3 Implementasi Hasil Klasifikasi dan Akurasi .....	57
<b>BAB 6 PENGUJIAN DAN ANALISIS .....</b>	<b>58</b>
6.1 Pengujian Akurasi Sistem.....	58
6.1.1 Pengujian dengan nilai $K = 5$ .....	62
6.1.2 Pengujian dengan nilai $K = 10$ .....	66
6.1.3 Pengujian dengan nilai $K = 15$ .....	70
6.1.4 Pengujian dengan nilai $K = 20$ .....	74
6.1.5 Akurasi Rata-Rata.....	78
6.2 Analisis .....	78
<b>BAB 7 KESIMPULAN DAN SARAN .....</b>	<b>80</b>
7.1 Kesimpulan .....	80
7.2 Saran .....	80
<b>DAFTAR PUSTAKA.....</b>	<b>81</b>



## DAFTAR TABEL

Tabel 3.1 Pengumpulan Data .....	20
Tabel 4.1 Contoh data latihan manualisasi.....	27
Tabel 4.2 Contoh data uji manualisasi .....	27
Tabel 4.3 Hasil perhitungan eucledian distance .....	29
Tabel 4.4 Data sebelum disorting .....	30
Tabel 4.5 Data yang telah disorting .....	30
Tabel 4.6 Data sebelum diambil sejumlah K.....	31
Tabel 4.7 Data yang diambil sejumlah K.....	31
Tabel 5.1 Spesifikasi Perangkat Keras.....	38
Tabel 5.2 Spesifikasi Perangkat Lunak .....	38
Tabel 6.1 Data Uji 1 .....	58
Tabel 6.2 Hasil Uji K = 5.....	62
Tabel 6.3 Hasil Uji K = 10 .....	66
Tabel 6.4 Data Uji K = 15 .....	70
Tabel 6.5 Hasil Uji K = 20 .....	74



## DAFTAR GAMBAR

Gambar 2.1 Kulit Anjing yang Terserang <i>Demodicosis</i> .....	7
Gambar 2.2 Kulit Anjing yang Terserang <i>Scabiosis</i> .....	8
Gambar 2.3 Telinga Anjing yang Terserang <i>Otitis</i> .....	9
Gambar 2.4 Batu Struvit.....	11
Gambar 2.5 Batu Urat Kandung Kemih dari Anjing Dalmatian.....	12
Gambar 2.6 Proses Klasifikasi .....	13
Gambar 3.1 Diagram Alir Metodologi Penelitian.....	18
Gambar 4.1 Diagram Alir Tahapan Fuzzy K-NN .....	23
Gambar 4.2 Diagram Alir Proses K-Nearest Neighbor .....	24
Gambar 4.3 Diagram Alir Eucledian Distance .....	25
Gambar 4.4 Diagram Alir Fuzzyfikasi FKNN .....	26
Gambar 4.5 Perancangan Antarmuka Data Latih .....	34
Gambar 4.6 Perancangan Antarmuka Data Uji.....	35
Gambar 4.7 Perancangan Antarmuka Proses FKNN .....	36
Gambar 5.1 Antarmuka Halaman Pilih Data Uji.....	53
Gambar 5.2 Antarmuka Halaman Pilih Data Latih .....	54
Gambar 5.3 Implementasi Antarmuka Jarak Eucledian Sebelum Diurutkan.....	55
Gambar 5.4 Implementasi Antarmuka Jarak Eucledian Setelah Diurutkan.....	55
Gambar 5.5 Implementasi Antarmuka Defuzzifikasi .....	57
Gambar 5.6 Implementasi Antarmuka Hasil Klasifikasi dan Implentasi .....	57

## BAB 1 PENDAHULUAN

### 1.1 Latar belakang

Hewan peliharaan mempunyai manfaat yang baik bagi manusia. Salah satunya adalah dapat meningkatkan kesehatan mental manusia. Anjing adalah salah satu hewan yang banyak dijadikan peliharaan selain kucing. Ketika membelai (*petting*) dan bermain dengan hewan peliharaan, oksitosin, hormon yang berkaitan dengan stres dan rasa lega dilepaskan, membantu menurunkan tekanan darah juga level kortisol. Hewan peliharaan juga mampu membuatnya pemiliknya memiliki gaya hidup yang lebih aktif, mengurangi rasa kesepian, mengurangi alergi dan meningkatkan imunitas (Hark, 2013).

Sebuah studi tahun 2002 berjudul tentang pengaruh hewan peliharaan terhadap aktivitas kardiovaskular manusia menunjukkan bahwa mereka yang memiliki anjing atau kucing berkemungkinan kecil mengalami lonjakan denyut jantung dan tekanan darah ketika berada pada kondisi stres seperti saat mengerjakan tugas matematika dengan batasan waktu. Denyut jantung dan tekanan darah mereka kembali normal lebih cepat daripada mereka yang tidak memiliki anjing atau kucing (Allen et al, 2002). Studi lainnya adalah tentang pengaruh kehadiran hewan peliharaan pada anak-anak saat dilakukan pemeriksaan fisik. Studi ini mengukur tekanan darah, denyut jantung dan perilaku tertekan (*behavioral distress*) anak-anak sehat berumur 3 sampai 6 tahun dengan mengunjungi dua dokter berbeda untuk pemeriksaan fisik rutin. Ditempatkan seekor anjing pada kunjungan pertama, sementara pada kunjungan kedua tidak. Hasil menunjukkan bahwa ketika ada anjing, tekanan darah anak-anak tersebut menjadi lebih rendah dengan denyut jantung yang juga lebih rendah serta sedikit perilaku tertekan (Nagengast et al, 1997).

Meskipun memelihara anjing memiliki banyak manfaat bagi manusia, para pemilik harus telaten dalam merawat anjing mereka, mulai dari lingkungan tempat tinggal yang bersih, ukuran kandang yang memadai, pemberian gizi yang cukup serta pemberian vaksin rutin. Tidak sedikit anjing yang terserang berbagai macam penyakit yang disebabkan oleh virus, protozoa, bakteri dan parasit. Anjing yang sakit apabila tidak segera mendapatkan perawatan dan pengobatan memiliki resiko dapat menularkannya kepada anjing dan hewan peliharaan lainnya atau bahkan kepada manusia. Kondisi menjadi sedikit sulit apabila pemilik baru menyadari bahwa anjingnya sedang dalam keadaan tidak sehat pada jam-jam di luar jam kerja. Terbatasnya jumlah klinik hewan yang dapat menangani pasien di luar jam kerja membuat pemilik anjing harus dapat memberikan penanganan awal dengan segera. Oleh karena itu, diperlukan suatu sistem yang dapat melakukan klasifikasi penyakit anjing untuk membantu pemilik anjing melakukan deteksi dini penyakit anjing agar dapat dilakukan penanganan awal dengan segera (Radostits, 2016).

Klasifikasi merupakan sebuah metode untuk mengelompokkan data atau objek ke dalam kelas yang sebelumnya sudah didefinisikan. Data yang diinputkan dalam

klasifikasi berupa sekumpulan atribut-atribut atau fitur-fitur yang merupakan ciri-ciri dari suatu data atau objek. Atribut atau fitur tersebut bisa berupa data diskrit maupun kontinyu. Ada bermacam-macam algoritme klasifikasi yang dapat digunakan saat ini, salah satu contohnya adalah *Fuzzy K-Nearest Neighbor* (FK-NN). Fuzzy k-Nearest Neighbor (FK-NN) ialah varian metode K-Nearest Neighbor (K-NN) dengan teknik fuzzy. Metode FK-NN memberikan nilai keanggotaan kelas pada vektor sampel dan bukan menempatkan vektor pada kelas tertentu. FK-NN adalah metode klasifikasi yang digunakan untuk memprediksi data uji menggunakan nilai derajat keanggotaan data uji pada setiap kelas. Kemudian diambil kelas dengan nilai derajat keanggotaan terbesar dari data uji sebagai kelas hasil prediksi. (Aroquiaraj, 2014).

Adapun salah satu penelitian dengan objek anjing mengenai diagnosis penyakit anjing yang menggunakan *Voting Feature Intervals* (VF15). Penelitian ini mengklasifikasi pasien ke dalam dua kelas penyakit yaitu parvo dan distemper, yang bertujuan untuk melakukan perbandingan diagnosis tahap awal yang dilakukan oleh dokter dengan hasil klasifikasi menggunakan algoritme VF15 dan juga untuk mengetahui tingkat akurasinya. Dari penelitian ini didapatkan akurasi yang cukup tinggi untuk setiap iterasinya. Pada iterasi pertama dan kedua didapatkan akurasi sebesar 100%, sementara pada iterasi ketiga didapatkan akurasi sebesar 70%. Rata-rata akurasi adalah sebesar 90% dengan standar deviasi sebesar 17,32% (Iqbal et al, 2014).

Algoritma *Fuzzy K-Nearest Neighbor* (FK-NN) juga telah banyak digunakan untuk melakukan pengklasifikasian penyakit. Contoh penelitian pertama yang menggunakan Algoritma *Fuzzy K-Nearest Neighbor* (FK-NN) adalah penelitian yang dilakukan oleh Deny Andrianto untuk melakukan diagnosis penderita gagal ginjal, data yang dibutuhkan pada penelitian ini adalah usia pasien, tekanan darah, kadar gula darah dan lain – lain, pada penelitian ini dihasilkan sebuah perangkat lunak yang dapat menentukan seseorang terserang penyakit gagal ginjal atau tidak, pada pengujiannya didapatkan hasil akurasi terbaik ketika Nilai  $k = 3$  yaitu dengan akurasi sebesar 98%.

Penelitian selanjutnya adalah penelitian yang dilakukan oleh Andika Satria yang digunakan untuk menentukan ketepatan waktu kelulusan, data yang dibutuhkan pada penelitian ini adalah nilai indeks prestasi mulai dari semester 1 hingga semester 14, pada penelitian ini dihasilkan sebuah perangkat lunak yang dapat menentukan seseorang lulus tepat waktu atau tidak, pada pengujiannya didapatkan rata – rata akurasi sebesar 100%.

Penelitian selanjutnya adalah penelitian yang dilakukan oleh Satria Dwi Nugraha yang digunakan untuk menentukan status gizi balita, data yang dibutuhkan pada penelitian ini adalah jenis kelamin, umur berat badan dan tinggi badan, pada penelitian ini dihasilkan sebuah perangkat lunak yang dapat menentukan gizi balita sudah terpenuhi atau tidak, pada pengujiannya didapatkan hasil akurasi terbaik ketika Nilai  $k = 4$  yaitu dengan akurasi sebesar 84,37%.

Berdasarkan uraian permasalahan yang telah disebutkan dan pemaparan penelitian sebelumnya, maka judul yang diusulkan dalam penelitian ini adalah “Implementasi Metode FKNN (*Fuzzy K-Nearest Neighbor*) untuk Diagnosis Penyakit Anjing”.

## 1.2 Rumusan masalah

Rumusan masalah penelitian yang didapatkan berdasarkan uraian latar belakang adalah sebagai berikut:

1. Bagaimana implementasi metode *Fuzzy K-Nearest Neighbor* (FK-NN) untuk mendiagnosis penyakit anjing.
2. Bagaimana tingkat akurasi sistem diagnosis penyakit anjing yang diperoleh dengan menggunakan metode *Fuzzy K-Nearest Neighbor* (FK-NN).

## 1.3 Tujuan

Tujuan yang ingin dicapai dalam penelitian ini adalah sebagai berikut:

1. Mengimplementasikan metode *Fuzzy K-Nearest Neighbor* (FK-NN) untuk melakukan diagnosis penyakit anjing.
2. Menguji tingkat akurasi sistem diagnosis penyakit anjing yang diperoleh dengan menggunakan metode *Fuzzy K-Nearest Neighbor* (FK-NN).

## 1.4 Manfaat

Penelitian ini diharapkan dapat membantu pemilik anjing melakukan deteksi dini penyakit pada anjing sehingga dapat dilakukan tindakan penanganan awal terhadap penyakit secepatnya.

## 1.5 Batasan masalah

Untuk menjaga agar penelitian ini berjalan sesuai dengan masalah yang telah dirumuskan, maka diberikan batasan-batasan sebagai berikut:

1. Data penelitian bersumber dari data rekam medis hewan di klinik hewan Yudhistira, Denpasar yang berjumlah 250 data.
2. Terdapat 10 penyakit yang akan diklasifikasikan yaitu *Demodicosis*, *Scabiosis*, *Otitis*, *Ringworm*, *Helminthiasis*, *Canine Parvovirus*, *Canine Distemper*, *Ehrlichiosis*, *Coccidiosis* dan *Urolithiasis* dengan total 46 gejala.
3. Keluaran dari sistem klasifikasi ini adalah diagnosis penyakit anjing berdasarkan masukan berupa gejala-gejala penyakit.
4. Sistem diagnosis yang dibangun menggunakan bahasa pemrograman Java.



## 1.6 Sistematika pembahasan

Sistematika penyusunan penelitian ini adalah sebagai berikut:

### **BAB I PENDAHULUAN**

Berisi uraian tentang latar belakang, rumusan masalah, tujuan, manfaat, batasan masalah dan sistematika pembahasan penelitian.

### **BAB II LANDASAN KEPUSTAKAAN**

Berisi uraian kajian pustaka dari penelitian-penelitian sebelumnya terkait implementasi algoritme *Fuzzy K-Nearest Neighbor* (FK-NN). Bagian ini juga memuat dasar teori yang berisi teori-teori lainnya yang berhubungan penelitian.

### **BAB III METODOLOGI**

Berisi bahasan metodologi yang digunakan dalam penelitian yang terdiri dari studi literatur, pengumpulan data, analisis kebutuhan, perancangan sistem, implementasi sistem, pengujian sistem, serta pengambilan kesimpulan dan saran.

### **BAB IV PERANCANGAN SISTEM**

Berisi bahasan mengenai rancangan implementasi sistem klasifikasi penyakit anjing menggunakan algoritme *Fuzzy K-Nearest Neighbor* (FK-NN).

### **BAB V IMPLEMENTASI PERANGKAT LUNAK**

Berisi implementasi dari hasil rancangan yang sudah dibuat sebelumnya juga berisi penjelasan langkah-langkah melakukan implementasi.

### **BAB VI PEMBAHASAN**

Berisi penjelasan langkah-langkah pengujian sistem serta analisis hasil pengujian sistem.

### **BAB VII PENUTUP**

Berisi kesimpulan yang didapat berdasarkan hasil penelitian serta pemberian saran untuk pengembangan penelitian selanjutnya.



## BAB 2 LANDASAN KEPUSTAKAAN

Bab ini berisi penjelasan singkat penelitian-penelitian sebelumnya juga pemaparan dasar-dasar teori untuk menunjang penyusunan penelitian ini. Beberapa teori yang dibutuhkan meliputi kajian pustaka, *data mining*, klasifikasi, metode *Fuzzy K-Nearest Neighbor* (FK-NN), dan penyakit-penyakit pada anjing.

### 2.1 Anjing

Tidak diketahui dengan pasti darimana anjing berasal. Anjing (*Canis familiaris*) dipercaya sebagai keturunan serigala abu-abu (*grey wolf*). Domestikasi serigala awalnya diyakini muncul sekitar 15.000 tahun yang lalu di Timur Tengah. Namun sebuah studi yang dipublikasi oleh ahli genetika Swedia, Pontus Skoglund, mendeskripsikan tentang penemuan tulang serigala Siberia berumur 35.000 tahun. Dari penemuan tersebut disimpulkan bahwa domestikasi *canine* atau anjing kemungkinan terjadi pertama kali sekitar 27.000 sampai 40.000 tahun yang lalu. Studi genetika menyebutkan bahwa anjing-anjing peliharaan modern berasal dari Cina, Timur Tengah, dan Eropa Timur. Arkeolog dan ahli genetika, Greger Larson, menyebutkan bahwa domestikasi serigala abu-abu dilakukan oleh manusia di Eurasia bagian barat (Jacobson, 2016).

Namun ada studi lainnya dalam PLOS Genetics yang menunjukkan bahwa anjing tidak berevolusi dari serigala abu-abu, atau setidaknya bukan serigala abu-abu yang ada saat ini. Dilakukan sebuah penelitian dengan mengambil urutan genom tiga serigala abu-abu yang berasal dari Cina, Israel, dan Kroasia yang dipercaya sebagai lokasi asal anjing. Selain itu diambil juga urutan genom anjing Basenji dan Dingo yang berasal dari Afrika Tengah dan Australia, di mana menurut siaran pers dari The University of Chicago Medical Center, kedua daerah tersebut dalam sejarahnya terisolasi dari populasi serigala. Hasil penelitian urutan genom anjing dan serigala tersebut menunjukkan bahwa anjing-anjing tersebut memiliki kekerabatan lebih dekat satu sama lain dibanding dengan serigala. Seriga-serigala tersebut juga memiliki kekerabatan lebih dekat satu sama lain dibandingkan dengan anjing. Para ilmuwan tidak menemukan adanya bukti yang menghubungkan anjing dengan berbagai jenis serigala yang dijadikan sampel. Salah satu kemungkinan yang ada yaitu anjing berasal dari jenis serigala yang sudah punah (Chowdury, 2014).

### 2.2 Penyakit-Penyakit Anjing

Memelihara anjing memiliki banyak manfaat yang baik bagi manusia. Namun, manusia harus waspada terhadap penyakit-penyakit yang dapat menyerang anjing. Beberapa penyakit tersebut dapat menular ke manusia. Oleh karena itu, memelihara kesehatan anjing sangatlah penting. Berikut beberapa penyakit anjing yang digunakan dalam penelitian ini.

## 2.3 Kajian Pustaka

Kajian pustaka membahas tentang penelitian-penelitian sebelumnya yang menggunakan metode *Fuzzy K-Nearest Neighbor* (FK-NN) dengan berbagai macam objek. Contoh penelitian pertama yang menggunakan Algoritma *Fuzzy K-Nearest Neighbor* (FK-NN) adalah penelitian yang dilakukan oleh Deny Andrianto untuk melakukan diagnosis penderita gagal ginjal, data yang dibutuhkan pada penelitian ini adalah usia pasien, tekanan darah, kadar gula darah dan lain – lain, pada penelitian ini dihasilkan sebuah perangkat lunak yang dapat menentukan seseorang terserang penyakit gagal ginjal atau tidak, pada pengujiannya didapatkan hasil akurasi terbaik ketika Nilai  $k = 3$  yaitu dengan akurasi sebesar 100%.

Penelitian selanjutnya adalah penelitian yang dilakukan oleh Andika Satria yang digunakan untuk menentukan ketepatan waktu kelulusan, data yang dibutuhkan pada penelitian ini adalah nilai indeks prestasi mulai dari semester 1 hingga semester 14, pada penelitian ini dihasilkan sebuah perangkat lunak yang dapat menentukan seseorang lulus tepat waktu atau tidak, pada pengujiannya didapatkan rata – rata akurasi sebesar 98%.

Penelitian selanjutnya adalah penelitian yang dilakukan oleh Satria Dwi Nugraha yang digunakan untuk menentukan status gizi balita, data yang dibutuhkan pada penelitian ini adalah jenis kelamin, umur berat badan dan tinggi badan, pada penelitian ini dihasilkan sebuah perangkat lunak yang dapat menentukan gizi balita sudah terpenuhi atau tidak, pada pengujiannya didapatkan hasil akurasi terbaik ketika Nilai  $k = 4$  yaitu dengan akurasi sebesar 84,37%. Kajian pustaka ditunjukkan oleh Tabel 2.1.

**Tabel 2.1**

No	Judul	Object	Metode (Proses)	Hasil
1.	Penerapan Fuzzy K-Nearest Neighbor Untuk Diagnosis Penderita Gagal Ginjal Kronis Berdasarkan Dataset Indians Chronic Kidney Disease (CKD)	Pasien terduga terserang penyakit gagal ginjal	Fuzzy K-Nearest Neighbor (F-KNN)	Menentukan pasien terserang penyakit gagal ginjal atau tidak, tingkat akurasi hingga 100%.
2.	Implementasi Algoritme Fuzzy K-Nearest Neighbor untuk	Alumni Fakultas Ilmu Komputer Universitas	Fuzzy K-Nearest Neighbor (F-KNN)	Menentukan alumni dinyatakan lulus tepat waktu atau tidak, rata-

	Penentuan Lulus Tepat Waktu (Studi Kasus : Fakultas Ilmu Komputer Universitas Brawijaya)	Brawijaya Malang		rata akurasi sebesar 98%.
3.	Penerapan Fuzzy K-Nearest Neighbor (FK-NN) Dalam Menentukan Status Gizi Balita	Balita	Fuzzy K-Nearest Neighbor (F-KNN)	Menentukan apakah gizi balita sudah terpenuhi atau tidak, tingkat akurasi hingga 84,37%.

### 2.3.1 Demodicosis

*Demodicosis* merupakan penyakit kulit yang disebabkan oleh tungau *Demodex canis* yang berada di folikel rambut atau kelenjar minyak pada anjing. *Demodicosis* bisa berupa lokal atau general dan tidak menular. *Demodicosis* lokal biasanya terjadi pada anak-anak anjing berumur kurang dari 1 tahun dan biasanya bisa sembuh dengan sendirinya tanpa pengobatan. Lesi *demodicosis* lokal biasanya terjadi di bagian tubuh tertentu saja seperti sekitar mulut atau kaki. *Demodicosis* lokal bisa berkembang menjadi *demodicosis* general, di mana lesi menyebar lebih luas di tubuh anjing atau bahkan ke seluruh tubuh anjing (Dryden, 2016). Gejala-gejala *demodicosis* adalah rambut rontok, kulit berketombe (*squama*), kulit berkerak atau berkeropeng (*krustae*), gatal ringan (*pruritus*), kulit kemerahan (eritema) bahkan kehitaman (hiperpigmentasi). Penyakit *demodicosis* pada anjing ditunjukkan pada Gambar 2.1



Gambar 2.1 Kulit Anjing yang Terserang *Demodicosis*

Sumber: Dryden, 2016

### 2.3.2 Scabiosis

*Scabiosis*, atau biasa disebut scabies, disebabkan oleh tungau *Sarcoptes scabiei* yang dapat menular pada hewan lain maupun manusia. Tungau ini masuk ke dalam kulit dan menyebabkan gatal-gatal yang parah. Penyakit ini dapat menyerang semua jenis anjing dari berbagai usia. Gejala awal yang muncul berupa adanya bentol-bentol (papula) pada kulit, gatal, lepuhan pada kulit (vesikula), dan rambut patah-patah. Lesi biasanya muncul di perut, dada, telinga, dan siku (Dryden, 2016). Jika tidak diobati, maka akan menyebar ke seluruh tubuh. Apabila sudah parah, anjing yang terserang *scabiosis* bahkan mengalami penebalan kulit (*hyperkeratosis*). Penyakit *scabiosis* pada anjing ditunjukkan pada Gambar 2.2.



Gambar 2.2 Kulit Anjing yang Terserang *Scabiosis*

Sumber: Dryden, 2016

### 2.3.3 Otitis

*Otitis externa* atau biasa disebut *otitis*, merupakan salah satu kelainan saluran telinga pada anjing yang disebabkan oleh berbagai faktor. Beberapa faktor seperti parasit, benda asing, dan alergi, menyebabkan peradangan secara langsung, sementara beberapa faktor lainnya seperti bakteri, membuat kondisinya semakin buruk. Bentuk saluran telinga anjing juga sangat mempengaruhi anjing terjangkit otitis. *Otitis* terjadi ketika lapisan sel yang melapisi telinga bagian luar mengalami peradangan (Neer, 2016). Penyakit *otitis* pada anjing ditunjukkan pada Gambar 2.3





**Gambar 2.3 Telinga Anjing yang Terserang Otitis**

Sumber: Neer, 2016

Anjing yang terserang *otitis* sering mengguncangkan kepala karena telinga yang gatal dan juga sering menggaruk telinga. Gejala lainnya yaitu keluarnya leleran dari telinga serta adanya serumen (kotoran telinga) baru yang berwarna gelap.

#### **2.3.4 Ringworm**

*Ringworm* merupakan infeksi kulit, rambut, dan cakar, yang disebabkan oleh jamur. Pada anjing, ada 3 jamur yang menjadi penyebab *ringworm* di mana jamur-jamur dapat menyebar dengan sangat mudah. 70% disebabkan oleh jamur *Microsporum canis*, 20% disebabkan oleh jamur *Microsporum canis*, dan 10% disebabkan oleh jamur *Trichophyton mentagrophytes*. *Ringworm* umumnya terjadi karena adanya kontak dengan hewan lain yang terjangkit penyakit ini, atau bisa juga karena terkontaminasi alat-alat seperti alat pencukur rambut anjing. Bagian tubuh anjing yang sering terkena *ringworm* adalah wajah, ujung telinga, ekor, dan kaki. Anjing yang terjangkit *ringworm* mengalami pitak dan lesi berbentuk lingkaran menyerupai cincin dengan pinggiran berketombe (Moriello et al, 2016).

#### **2.3.5 Helminthiasis**

*Helminthiasis* merupakan penyakit yang disebabkan oleh infeksi cacing parasit. Penyakit ini selain dapat terjadi pada hewan, dapat terjadi juga pada manusia. Ada banyak spesies cacing yang dapat menyebabkan *helminthiasis*. Salah satunya adalah *ascaridoid nematodes* atau biasa dikenal sebagai *roundworms*. Jenis *roundworms* yang dianggap paling berbahaya adalah *Toxocara canis* karena mudah menyebabkan infeksi terutama pada anak anjing. Selain itu, cacing ini juga dapat berpindah ke manusia. Gejala-gejala *helminthiasis* pada anjing adalah rambut anjing berubah menjadi kusam, perut buncit, tidak nafsu makan (anoreksia), diare, kotoran mata berlebih, dan muntah. Pada penyakit ini, cacing bisa ditemukan pada muntahan atau pada feses (Bowman et al, 2014).

#### **2.3.6 Canine Parvovirus**

*Canine Parvovirus* (CPV) atau biasa disebut parvo merupakan penyakit menular yang menyebabkan infeksi gastrointestinal akut yang umumnya

menyerang anjing muda. Penyakit ini banyak menyerang anjing berusia antara 6 minggu sampai 6 bulan yang belum divaksin atau yang belum mendapatkan vaksin lengkap. Virus ini dilepaskan melalui feses anjing yang terinfeksi 4 sampai 5 hari setelah terpapar, selama masa penyakit, dan selama 10 hari setelah pemulihan. Infeksi terjadi melalui kontak langsung oral atau nasal dengan feses yang mengandung virus atau secara tidak langsung melalui objek yang sudah terkontaminasi virus, misalnya lingkungan atau peralatan. Gejala klinis parvo berkembang 5 sampai 7 hari infeksi, tetapi bisa berkisar antara 2 sampai 14 hari. Gejala klinis awal biasanya berupa demam, tidak nafsu makan (anoreksia), dan lemas (*lethargy*), yang kemudian berkembang menjadi dehidrasi, muntah, dan diare berdarah dalam kurun waktu 24 sampai 48 jam (Mitchell, 2016).

### 2.3.7 Canine Distemper

*Canine Distemper* atau biasa disebut distemper merupakan penyakit yang sangat menular yang disebabkan oleh virus. Sebagian besar kasus distemper pada anjing terjadi setelah anjing menghirup uap air yang mengandung virus yang berasal dari anjing atau hewan lainnya yang terinfeksi virus distemper. Anjing yang terjangkit distemper mengalami demam pertama selama 3 sampai 6 hari disertai dengan hilangnya nafsu makan (anoreksia) dan berubah lemas (*lethargy*). Demam pertama akan hilang selama beberapa hari sebelum terjadi demam kedua. Demam kedua terjadi selama satu minggu disertai dengan keluarnya leleran dari hidung dan mulut (salivasi) serta anoreksia. Distemper juga menyerang sistem saraf yang menyebabkan kejang-kejang (*seizure*), kejang otot (*spasmus*), dan pergerakan bola mata yang tidak terkendali (*nystagmus*). Gejala lainnya yang dapat dilihat pada tubuh anjing yaitu munculnya bintik-bintik bernanah (pustul) pada kulit abdomen. Anjing akan mengalami kejang-kejang lebih sering seiring dengan berkembangnya penyakit (Radostits et al, 2016).

### 2.3.8 Ehrlichiosis

*Canine monocytic ehrlichiosis* atau yang biasa hanya disebut *ehrlichiosis* umumnya disebabkan oleh bakteri *Ehrlichia canis* dan *Anaplasma platys*. Bakteri ini, dibawa oleh kutu, menginfeksi sel darah putih tertentu dan menyebabkan demam serta gejala-gejala lainnya (Radostits et al, 2016).

Gejala yang disebabkan oleh bakteri *Ehrlichia canis* berkembang dari jangka pendek ke jangka panjang, tergantung pada sistem imun anjing. Gejala jangka pendeknya antara lain demam, pembesaran limpa, hilangnya nafsu makan (anoreksia), kaku, lemas (*lethargy*), dan sulit bernafas (*dyspnea*). Ini biasanya terjadi saat musim hangat, saat banyaknya kutu yang aktif. Pada jangka pendek, kondisi anjing yang terjangkit bisa pulih dengan cepat dan jarang terjadi kasus kematian. Anjing yang sudah pulih bisa terbebas dari gejala-gejala tersebut, atau penyakitnya bisa berkembang menjadi jangka panjang. *Ehrlichiosis* jangka panjang bisa terjadi kapan saja di luar musim hangat. Sedangkan *ehrlichiosis* yang disebabkan oleh *Anaplasma platys* umumnya hanya menunjukkan sedikit gejala, bahkan tidak ada tanda-tanda infeksi. Gejala yang umum adalah menurun trombosit (*thrombocytopenia*) dan adanya edema pada kaki. Gejala-gejala lainnya

pada *ehrlichiosis* adalah mimisan, kencing berdarah (*hematuria*), dan feses yang berwarna hitam (*melena*).

### 2.3.9 *Coccidiosis*

*Coccidiosis* merupakan penyakit infeksi saluran pencernaan yang disebabkan oleh protozoa *coccidia*. Ada 3 spesies protozoa *coccidia* yang menyerang anjing yaitu *Isospora*, *Hammondia*, dan *Sarcocytic*, dengan *Isospora* sebagai yang paling umum menyerang anjing. Gejala-gejala penyakit ini yaitu dehidrasi, diare atau diare berdarah, hilangnya nafsu makan (*anoreksia*), dan menurunnya berat badan. Anjing yang sebelumnya sudah terjangkit *coccidiosis*, menyebarkan ookista (sel *coccidia* yang belum matang), melalui feses. Pada kondisi suhu dan kelembapan yang tepat, ookista ini menjadi infeksius. Apabila seekor anjing menelan ookista, maka ookista akan mengeluarkan sporozoit yang menyerang sel-sel pada usus (Constable, 2016).

### 2.3.10 *Urolithiasis*

*Urolithiasis* merupakan penyakit yang disebabkan karena adanya batu (*uroliths*) atau kristal di saluran kencing. Batu-batu yang umumnya ada pada anjing yaitu magnesium amonium fosfat, kalsium oksalat, atau amonium urat. Magnesium amonium fosfat pada anjing contohnya adalah struvit hingga disebut batu struvit, yang terbentuk bersamaan dengan infeksi saluran kencing. Batu struvit ditunjukkan pada Gambar 2.4.



**Gambar 2.4 Batu Struvit**

Sumber: Brown, 2016

Selain struvit, ada amonium urat yang umumnya ditemukan pada anjing Dalmatian. Batu kencing lainnya yang banyak ditemukan pada anjing mengandung kalsium oksalat. Batu ini terbentuk karena adanya peningkatan kalsium di ginjal. Adapun karakteristiknya adalah (Brown, 2016):

- Peningkatan ekskresi kalsium urin, normalnya konsentrasi serum kalsium, dan normal atau rendahnya serum *parathormone*. Keadaan ini disebut *absorptive hypercalciuria*.



- Normalnya konsentrasi serum kalsium, meningkatnya ekskresi kalsium urin, dan meningkatnya konsentrasi serum *parathormone*. Keadaan ini disebut *renal leak hypercalciuria*.
- Adanya filtrasi dan ekskresi kalsium berlebihan pada urin. Keadaan ini disebut *resoptive hypercalciuria*.

Adapun gejala-gejala yang terjadi pada anjing yang terserang *urolithiasis* adalah sakit pada abdomen, muntah, lemas (*lethargy*), kandung kemih membesar dan keras, serta mengalami masalah urinasi seperti tidak bisa kencing (*dysuria*), kencing berdarah (*hematuria*), atau kencing sedikit-sedikit (*stranguria*). Gambar 2.5 menunjukkan batu urat kandung kemih dari anjing Dalmatian.



**Gambar 2.5 Batu Urat Kandung Kemih dari Anjing Dalmatian**

Sumber: Brown, 2016

## 2.4 Data Mining

*Data mining*, atau biasa disebut sebagai *Knowledge Discovery in Databases* (KDD), mengacu pada ekstraksi, penggalian, atau penemuan suatu pengetahuan (*knowledge*) yang berasal dari data yang berjumlah besar. Adapun beberapa yang menganggap bahwa data mining itu sendiri hanya merupakan suatu tahap penting dalam proses penemuan pengetahuan (Han, 2012).

Proses penemuan pengetahuan terdiri dari urutan iterasi langkah-langkah berikut.

- *Data cleaning*, untuk menghilangkan noise atau data yang tidak relevan dan tidak konsisten.
- *Data integration*, menggabungkan berbagai sumber data.
- *Data selection*, mengambil data dari basis data yang relevan dengan analisis.
- *Data transformation*, mengubah data ke dalam bentuk lain yang sesuai untuk ekstraksi.
- *Data mining*, proses penting yang menggunakan sebuah metode untuk ekstraksi pola data.
- *Pattern evaluation*, untuk mengidentifikasi pola menarik yang mewakili pengetahuan berdasarkan beberapa pengukuran.
- *Knowledge presentation*, menyajikan pengetahuan yang sudah diekstraksi kepada pengguna.

Berdasarkan langkah-langkah tersebut, *data mining* merupakan suatu langkah dalam proses penemuan pengetahuan yang sangat penting karena dapat menemukan suatu pola menarik. Oleh karena itu, dapat disimpulkan bahwa *data mining* adalah proses menemukan pengetahuan dan pola menarik dari sejumlah data yang berukuran besar. Data-data tersebut bisa bersumber dari basis data, *data warehouse*, web, dan penyimpanan informasi lainnya. Bentuk data yang umum digunakan untuk aplikasi *data mining* adalah basis data, data warehouse, dan data transaksional (Han, 2012).

*Data mining* memiliki beberapa fungsionalitas yang digunakan untuk menentukan jenis pola yang ditemukan dalam tugas *data mining*. Tugas tersebut secara umum dikategorikan menjadi 2 macam yaitu deskriptif dan prediktif. Tugas deskriptif adalah untuk mengkarakterisasi properti-properti data dalam data set yang menjadi target, sedangkan prediktif adalah membuat prediksi dengan melakukan induksi pada data yang ada (Han, 2012).

## 2.5 Klasifikasi

Salah satu fungsionalitas *data mining* adalah klasifikasi. Klasifikasi merupakan proses yang terdiri dari dua langkah yaitu pembelajaran atau pelatihan (*learning*) dan klasifikasi (*classification*). Pembelajaran merupakan langkah membuat model klasifikasi sedangkan klasifikasi merupakan langkah menentukan kelas dari data yang diberikan menggunakan model yang sudah dibuat (Han, 2012). Proses klasifikasi ditunjukkan oleh Gambar 2.7.



Gambar 2.6 Proses Klasifikasi

Sumber: Tan et al, 2006

Pada proses pelatihan, suatu set data latih dengan kelas yang telah diketahui dianalisis dan dibangun sebuah model dari setiap kelas dengan bantuan algoritme klasifikasi. Proses pelatihan pada klasifikasi disebut juga *supervised learning* karena setiap data latih sudah diketahui kelasnya masing-masing. Model yang didapatkan merupakan aturan-aturan klasifikasi (*classification rules*). Aturan-aturan ini kemudian diuji dengan data uji untuk memperkirakan akurasi. Data uji bersifat independen dan tidak digunakan dalam proses pelatihan. Bila akurasi baik, maka aturan-aturan tersebut dapat digunakan untuk klasifikasi data baru yang kelasnya tidak diketahui (Han, 2012).

## 2.6 K-Nearest Neighbor (K-NN)

Salah satu metode klasifikasi yang sering digunakan adalah metode *K-Nearest Neighbor*. Penggunaan *K-Nearest Neighbor* bertujuan untuk mengklasifikasikan objek baru berdasarkan data pembelajaran yang jaraknya paling dekat dengan objek baru tersebut. Teknik algoritma *K-Nearest Neighbor* mudah untuk diimplementasikan. Dalam hal ini jumlah data atau biasa disebut dengan tetangga terdekat ditentukan oleh user yang dinyatakan oleh  $k$ . Misalkan ditentukan  $k = 5$ , maka setiap data uji dihitung jarak kedekatannya terhadap data latih. Untuk  $k = 5$  berarti data latih yang dipilih adalah sebanyak lima yang jaraknya paling dekat terhadap data uji. Selanjutnya dipilih kelompok kelas terbanyak dari sebanyak  $k = 5$  yang telah ditentukan. Dari sebanyak  $k = 5$  tersebut kelompok kelas yang memiliki frekuensi terbanyak adalah hasil kelas prediksi yang diharapkan dari metode K-NN (Shouman, 2012).

## 2.7 Proses K-Nearest Neighbor

Langkah pertama untuk melakukan proses perhitungan jarak antara *record* data uji dan data latih perlu dilakukan *praprocessing* data pada setiap atribut terlebih dahulu. *Praprocessing* data tersebut berupa normalisasi data. Hal ini bertujuan untuk standarisasi skala nilai pada semua atribut / parameter data yang digunakan dalam perhitungan. Proses normalisasi yang umum digunakan adalah *Min-Max normalization*. Perhitungan *Min-Max normalization* ditunjukkan pada persamaan 2-1.

$$V' = \frac{V(x) - \min(x)}{\text{Range}(x)} \quad (2-1)$$

$$\text{Range}(x) = \text{Nilai dari } (\max(x) - \min(x)) \quad (2-2)$$

Keterangan :

- $V'$  = Hasil normalisasi yang nilainya berkisar antara 0 dan 1
- $V(x)$  = Nilai atribut yang akan dinormalisasikan
- $\text{Max}(x)$  = Nilai maksimum dari suatu atribut / parameter
- $\text{Min}(x)$  = Nilai minimum dari suatu atribut / parameter

Selanjutnya dihitung jarak kedekatan antara data uji dan data latih terlebih dahulu. Untuk perhitungan jarak terdekat digunakan fungsi *euclidean distance* yang ditunjukkan pada persamaan 2-3.

$$d_i = \sqrt{\sum_{i=1}^p (x_{2i} - x_{1i})^2} \quad (2-3)$$

Keterangan :

- $d_i$  = Jarak kedekatan

- $p$  = Jumlah atribut data  
 $x_1$  = Data latih  
 $x_2$  = Data uji

Variabel  $x_1$  dan  $x_2$  merupakan dua *record* dengan  $n$  atribut. Perhitungan dengan menggunakan persamaan (2-3) bertujuan untuk mengetahui jarak antara data  $x_1$  dan  $x_2$  pada masing – masing *record*.

## 2.8 Logika Fuzzy

Logika *fuzzy* pertama kali diperkenalkan oleh Prof. Lotfi A. Zadeh, seorang profesor dari University of California pada tahun 1965. Logika *fuzzy* digunakan sebagai suatu cara untuk memetakan permasalahan dari *input* ke *output* yang diharapkan. Logika *fuzzy* bekerja menggunakan derajat keanggotaan dari sebuah nilai yang kemudian digunakan untuk menentukan hasil yang ingin dihasilkan berdasarkan atas spesifikasi yang telah ditentukan [KUS-10].

Dalam konsepnya, logika *fuzzy* meniru cara berpikir manusia dengan menggunakan konsep sifat kesamaan suatu nilai. Dengan menggunakan logika *fuzzy*, nilai yang dihasilkan bukan hanya 1 (iya, *high*, *on*) atau 0 (tidak, *low*, *off*) tetapi seluruh kemungkinan antara 0 dan 1.

Menurut Cox (1994), alasan mengapa orang menggunakan logika *fuzzy*, di antaranya :

1. Konsep logika *fuzzy* mudah dimengerti karena menggunakan dasar teori himpunan dan konsep matematis yang mendasari penalarannya.
2. Logika *fuzzy* sangat fleksibel.
3. Logika *fuzzy* memiliki toleransi terhadap data yang tidak tepat.
4. Logika *fuzzy* mampu memodelkan fungsi nonlinier yang sangat kompleks.
5. Logika *fuzzy* dapat membangun dan mengaplikasikan pengalaman para pakar secara langsung tanpa melalui proses pelatihan.
6. Logika *fuzzy* dapat bekerja sama dengan teknik kendali secara konvensional seperti teknik mesin dan teknik elektro.
7. Logika *fuzzy* didasarkan pada bahasa alami (bahasa sehari – hari).

## 2.9 Himpunan Fuzzy

Himpunan tegas (*crisp*), nilai keanggotaan  $x$  dalam himpunan  $A$ , sering ditulis  $\mu_A(x) = 1$ , memiliki dua kemungkinan (Kusnawi,2007) yaitu :

1. Satu (1) artinya suatu item menjadi anggota dalam suatu himpunan.
2. Nol (0) artinya suatu item tidak menjadi anggota dalam suatu himpunan.

Jika di himpunan tegas, nilai keanggotaan hanya ada 0 dan 1, pada himpunan *fuzzy* nilai keanggotaan terletak pada rentang 0 sampai 1. Apabila  $x$  memiliki nilai keanggotaan *fuzzy*  $\mu_A(x) = 1$ , artinya  $x$  menjadi anggota penuh himpunan  $A$ . Himpunan *fuzzy* memiliki 2 atribut, yaitu (Kusnawi,2007):

1. Linguistik, yaitu penamaan suatu grup yang mewakili suatu keadaan atau kondisi tertentu dengan menggunakan bahasa alami, seperti : muda, parobaya, tua.
2. Numeris, yaitu suatu angka yang menunjukkan ukuran dari suatu variabel seperti : 40, 25, 50 dan sebagainya.

## 2.10 Fuzzy K-Nearest Neighbor (FK-NN)

Fuzzy K-NN merupakan metode klasifikasi yang menggabungkan teknik fuzzy dengan *K-Nearest Neighbor Classifier*. Algoritma Fuzzy *K-Nearest Neighbor* memberikan nilai keanggotaan kelas pada data uji bukan menempatkan data uji pada kelas tertentu. FK-NN merupakan metode klasifikasi yang digunakan untuk memprediksi data uji menggunakan nilai derajat keanggotaan data uji pada setiap kelas. Kemudian diambil kelas dengan nilai derajat keanggotaan terbesar dari data uji sebagai kelas hasil prediksi. Keuntungannya adalah nilai – nilai keanggotaan pada kelas data uji memberikan tingkat jaminan pada hasil klasifikasi. Sebagai contoh jika vektor diberikan nilai keanggotaan 0,9 pada kelas pertama dan 0,1 pada dua kelas lainnya peneliti dapat cukup yakin bahwa kelas dengan nilai keanggotaan 0,9 adalah kelas milik vektor tersebut. Di sisi lain, jika sebuah vektor diberi nilai keanggotaan 0,55 pada kelas pertama, 0,44 pada kelas kedua dan 0,01 pada kelas ketiga, maka peneliti harus ragu – ragu untuk menetapkan vektor berdasarkan hasil ini. Namun, dapat diyakinkan bahwa vektor tersebut bukan milik kelas ketiga. Dalam kasus seperti ini, perlu diperiksa lebih lanjut untuk menentukan klasifikasinya, karena memiliki derajat keanggotaan yang tinggi pada dua kelas yaitu satu dan dua. Pemberian nilai keanggotaan oleh algoritma ini jelas berguna dalam proses klasifikasi.

Sebuah data memiliki nilai keanggotaan pada setiap kelas yang berbeda dengan nilai derajat keanggotaan dalam interval [0,1]. Teori himpunan fuzzy menggeneralisasi teori K-NN klasik dengan mendefinisikan nilai keanggotaan sebuah data pada masing – masing kelas. Untuk mencari nilai Fuzzy *K-Nearest Neighbor* pada suatu data digunakan persamaan 2-5. Tetapi sebelum mencari nilai keanggotaan tiap kelas  $u_i(x)$ , terlebih dahulu  $u_{ij}$  diproses dengan menggunakan persamaan 2-4.

$$u_{ij} = \begin{cases} 0,51 + (n_j/n) * 0,49, & \text{jika } j = i \\ (n_j/n) * 0,49, & \text{jika } j \neq i \end{cases} \quad (2-4)$$

Keterangan :

- $u_{ij}$  = Nilai keanggotaan kelas i pada vektor j
- $n_j$  = Jumlah anggota kelas j pada suatu data latih n
- n = Jumlah data latih yang digunakan
- j = Kelas data
- i = Kelas data

Selanjutnya barulah nilai keanggotaan tiap kelas dapat dicari dengan menggunakan persamaan 2-5

$$u_i(x) = \frac{\sum_{j=1}^k u_{ij}(|x-x_j|^{-2/(m-1)})}{\sum_{j=1}^k (|x-x_j|^{-2/(m-1)})} \quad (2-5)$$



Keterangan :

$u_i(x)$  = Nilai keanggotaan suatu data  $x$  ke dalam kelas  $i$

$k$  = Jumlah tetangga terdekat.

$u_{ij}$  = Nilai keanggotaan kelas  $i$  pada vektor  $j$

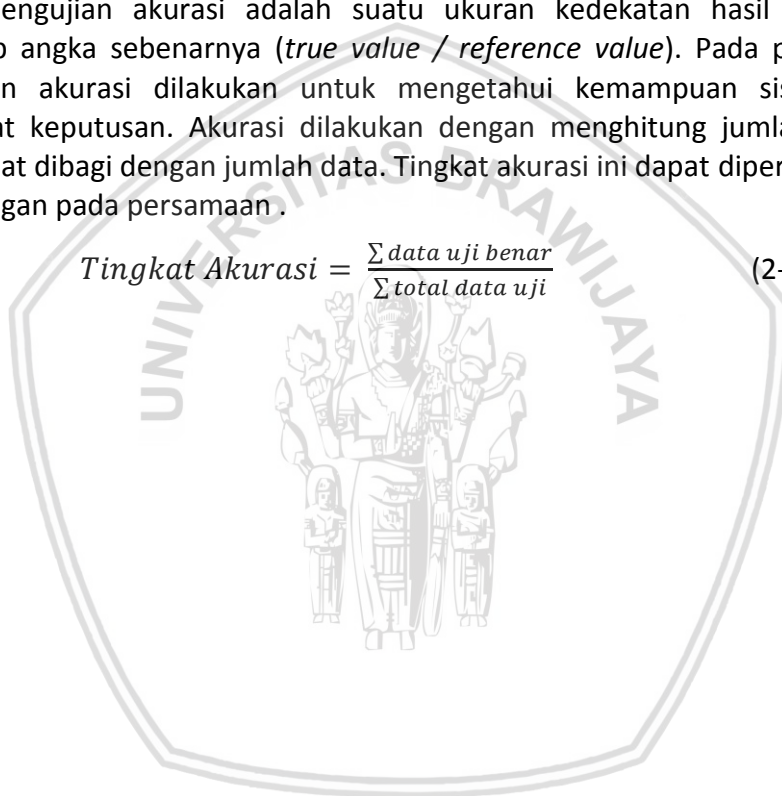
$x - x_j$  = Selisih jarak data uji  $x$  ke data latih  $x_j$  dalam  $k$  tetangga terdekat

$m$  = Bobot pangkat (*weight exponent*) yang besarnya 2.

## 2.11 Pengujian Akurasi

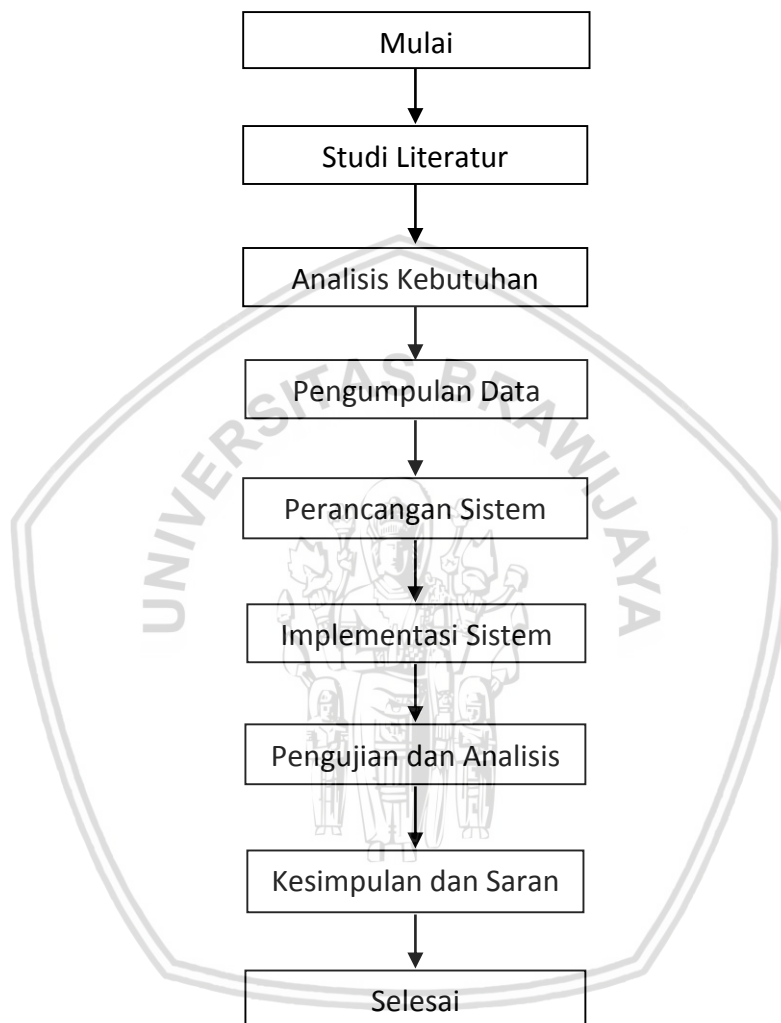
Pengujian akurasi adalah suatu ukuran kedekatan hasil pengukuran terhadap angka sebenarnya (*true value / reference value*). Pada penelitian ini pengujian akurasi dilakukan untuk mengetahui kemampuan sistem dalam membuat keputusan. Akurasi dilakukan dengan menghitung jumlah diagnosis yang tepat dibagi dengan jumlah data. Tingkat akurasi ini dapat diperoleh dengan perhitungan pada persamaan .

$$\text{Tingkat Akurasi} = \frac{\sum \text{data uji benar}}{\sum \text{total data uji}} \quad (2-8)$$



## BAB 3 METODOLOGI

Bab ini menguraikan tahapan-tahapan yang dilakukan dalam penelitian yaitu studi literatur, pengumpulan data, analisis kebutuhan, perancangan, implementasi sistem, pengujian sistem dan kesimpulan. Tahapan-tahapan penelitian ditunjukkan pada Gambar 3.1.



Gambar 3.1 Diagram Alir Metodologi Penelitian

### 3.1 Studi Literatur

Studi literatur yaitu mempelajari pustaka-pustaka yang berasal dari beberapa bidang ilmu yang berhubungan dengan klasifikasi penyakit pada anjing di antaranya adalah kajian pustaka, *data mining*, klasifikasi, metode *Fuzzy K-Nearest Neighbor* (FK-NN) dan penyakit pada anjing. Pustaka-pustaka tersebut bersumber dari penelitian-penelitian sebelumnya, jurnal, e-book, internet, dan dokter hewan.



### 3.2 Analisis Kebutuhan

Analisis kebutuhan dilakukan untuk mengidentifikasi dan memenuhi seluruh kebutuhan yang diperlukan untuk membangun sistem diagnosis penyakit anjing dengan metode *Fuzzy K-Nearest Neighbor* (FK-NN). Kebutuhan-kebutuhan yang perlu dipertimbangkan dalam penelitian ini adalah mengenai kemampuan sistem yang akan dibangun serta kebutuhan apa saja yang diperlukan untuk membangun sistem tersebut.

Dalam penelitian ini, sistem harus dapat melakukan proses perhitungan dengan metode FK-NN dan memberikan hasil klasifikasi berdasarkan perhitungan tersebut. Sistem juga harus memiliki tampilan yang membuat pengguna mudah memakainya, di mana dalam penelitian ini, sistem yang dibuat adalah berbasis Java. Adapun kebutuhan-kebutuhan lainnya yaitu:

1. Kebutuhan perangkat keras dan perangkat lunak:
  - Komputer PC
  - Sistem operasi Microsoft Windows 10
  - Perangkat Lunak Netbeans
2. Kebutuhan data:
  - Data penyakit-penyakit pada anjing
  - Data gejala-gejala penyakit pada anjing
  - Data rekam medis anjing.

### 3.3 Pengumpulan Data

Variabel penelitian ini adalah penyakit-penyakit pada anjing berdasarkan hasil perhitungan menggunakan metode *Fuzzy K-Nearest Neighbor* (FK-NN) terhadap gejala-gejala penyakit yang ada. Data mengenai gejala-gejala penyakit pada anjing didapatkan dengan melakukan konsultasi bersama dokter hewan di klinik hewan Yudisthira, buku, serta literatur lainnya mengenai penyakit pada anjing. Dataset dalam penelitian ini didapatkan dari rekam medis pasien anjing yang ada di klinik hewan Yudisthira. Jumlah total data yang diambil sebanyak 250 dengan 46 gejala penyakit dan 10 penyakit anjing yaitu *Demodicosis* sebanyak 42 data, *Scabiosis* sebanyak 34 data, *Otitis* sebanyak 21 data, *Ringworm* sebanyak 14 data, *Helminthiasis* sebanyak 29 data, *Canine Parvovirus* sebanyak 25 data, *Canine Distemper* sebanyak 16 data, *Ehrlichiosis* sebanyak 23 data, *Coccidiosis* sebanyak 27 data, dan *Urolithiasis* sebanyak 19 data. Rincian pengumpulan data dapat dilihat pada Tabel 3.1.

### 3.4 Perancangan Sistem

Perancangan sistem dilakukan untuk merancang sistem diagnosis penyakit anjing menggunakan algoritme *Fuzzy K-Nearest Neighbor* (FK-NN) yang akan dibangun. Tahapan perancangan meliputi deskripsi sistem, *preprocessing* data, perancangan sistem, perhitungan manual, desain antarmuka, dan perancangan pengujian. Langkah kerja sistem merupakan penerapan dari algoritma FK-NN untuk klasifikasi penyakit anjing. *Preprocessing* data berisi data-data yang

diperlukan dalam implementasi sistem, antara lain data penyakit anjing, data gejala-gejala penyakit dan bobot masing-masing gejala, dan data rekam medis. Adapun perancangan sistem itu sendiri terdiri atas deskripsi umum sistem, perancangan model dan proses FK-NN dengan diagram alir, contoh perhitungan manual, perancangan antarmuka, dan perangan pengujian sistem.

**Tabel 3.1 Pengumpulan Data**

No	Kebutuhan Data	Sumber Data	Metode	Kegunaan Data
1	Gejala-gejala penyakit anjing	Dokter hewan, buku, internet	Konsultasi dengan dokter hewan serta mencari pustaka yang sesuai di buku dan internet	Merupakan data pengetahuan tentang penyakit-penyakit pada anjing
2	Data rekam medis pasien anjing	Klinik hewan Yudisthira	Observasi	Untuk digunakan dalam perhitungan menggunakan algoritme <i>Fuzzy K-Nearest Neighbor</i> (FK-NN)

### 3.5 Implementasi Sistem

Implementasi sistem adalah tahap membangun sistem dengan menerapkan hasil studi literatur dan analisis kebutuhan, serta dilakukan berdasarkan perancangan sistem yang dibuat. Sistem klasifikasi penyakit anjing yang akan dibangun adalah berbasis Java. Tahapan-tahapan implementasi sistem yaitu pembuatan antarmuka, pembuatan basis data untuk data latih dan penerapan metode *Fuzzy K-Nearest Neighbor* (FK-NN). Keluaran sistem berupa hasil klasifikasi penyakit anjing berdasarkan gejala-gejala yang dimasukkan oleh pengguna.

### 3.6 Pengujian dan Analisis

Agar dapat diketahui apakah sistem berjalan dengan baik dan sesuai dengan kebutuhan yang telah ditetapkan, maka dilakukan pengujian sistem. Pengujian yang dilakukan adalah pengujian akurasi, yaitu untuk mengukur tingkat akurasi sistem diagnosis penyakit anjing yang dibangun. Pengujian tingkat akurasi dilakukan dengan membandingkan hasil keluaran dari sistem dengan data sebenarnya, yaitu data rekam medis.

### 3.7 Kesimpulan dan Saran

Penarikan kesimpulan dan pemberian saran dilakukan setelah semua proses yaitu perancangan sistem, implementasi sistem dan pengujian sistem. Kesimpulan didapatkan berdasarkan pada hasil analisis dari pengujian sistem. Sedangkan pemberian saran bertujuan untuk memperbaiki kekurangan yang ada pada sistem yang dibuat serta untuk dijadikan sebagai referensi dan bahan pertimbangan pada penelitian berikutnya.



## BAB 4 PERANCANGAN

Bab ini menjelaskan tentang perancangan yang diperlukan untuk membangun sistem diagnosis penyakit anjing menggunakan FK-NN.

### 4.1 Struktur Fuzzy K-Nearest Neighbor (FK-NN)

Fuzzy K-NN merupakan salah satu metode klasifikasi dengan menggabungkan teknik Fuzzy dan K-NN. Metode ini tidak seperti metode lain yang mana pada metode ini akan secara tegas memprediksi kelas yang diikuti oleh data uji berdasarkan perbandingan K terdekat. Dasar dari algoritma FK-NN adalah untuk menetapkan nilai keanggotaan sebagai fungsi jarak vektor dari K-NN dan keanggotaan tetangga mereka di kelas-kelas yang memungkinkan.

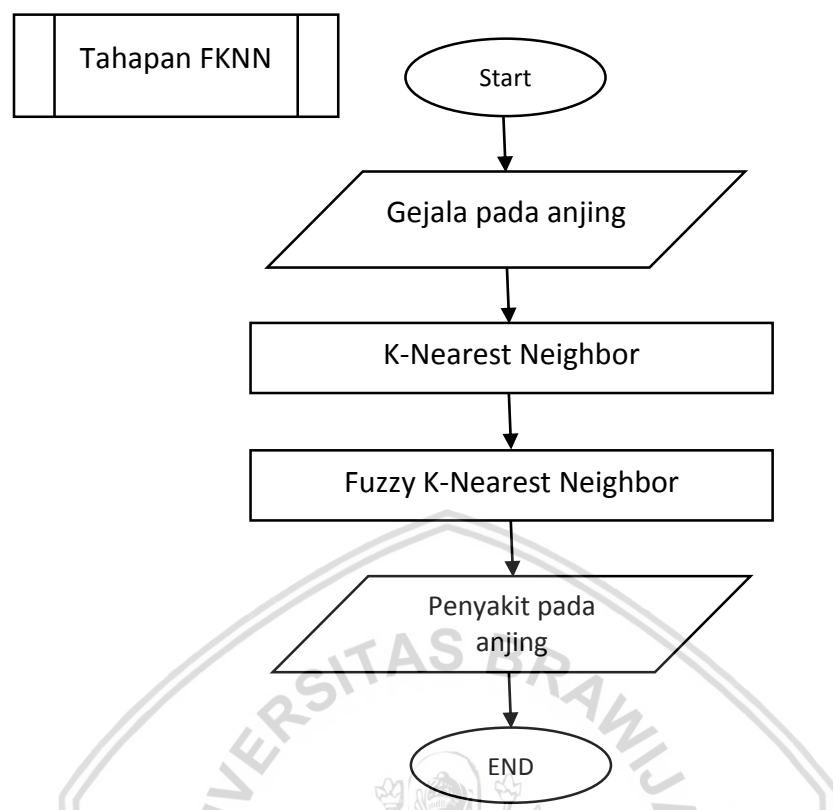
Metode ini berperan penting dalam menghilangkan ambiguitas dalam klasifikasi. Selain itu, sebuah instance akan memiliki derajat nilai keanggotaan pada setiap kelas sehingga akan lebih memberikan kekuatan atau kepercayaan suatu instance berada pada suatu kelas.

Pada tahapan ini dijelaskan mengenai alur proses sistem dalam melakukan pemrosesan data uji serta data latih penyakit anjing dengan menggunakan algoritma *Fuzzy K-Nearest Neighbor*. Adapaun tahapan dari proses tersebut yaitu ditunjukkan oleh gambar 4.1

Berdasarkan Gambar 4.1 proses klasifikasi menggunakan algoritma *Fuzzy K-Nearest Neighbor* terdiri atas dua proses utama yaitu antara lain :

1. Proses *K-Nearest Neighbor*

Adalah suatu proses pengklasifikasian terhadap data uji untuk dimasukkan ke dalam suatu kelas penyakit. Proses pengklasifikasian tersebut menggunakan jarak *euclidean distance* antara data latih dan data uji. Dan selanjutnya mayoritas kelas pada sejumlah k (tetangga terdekat) digunakan untuk menentukan data uji masuk ke dalam salah satu kelas prediksi.



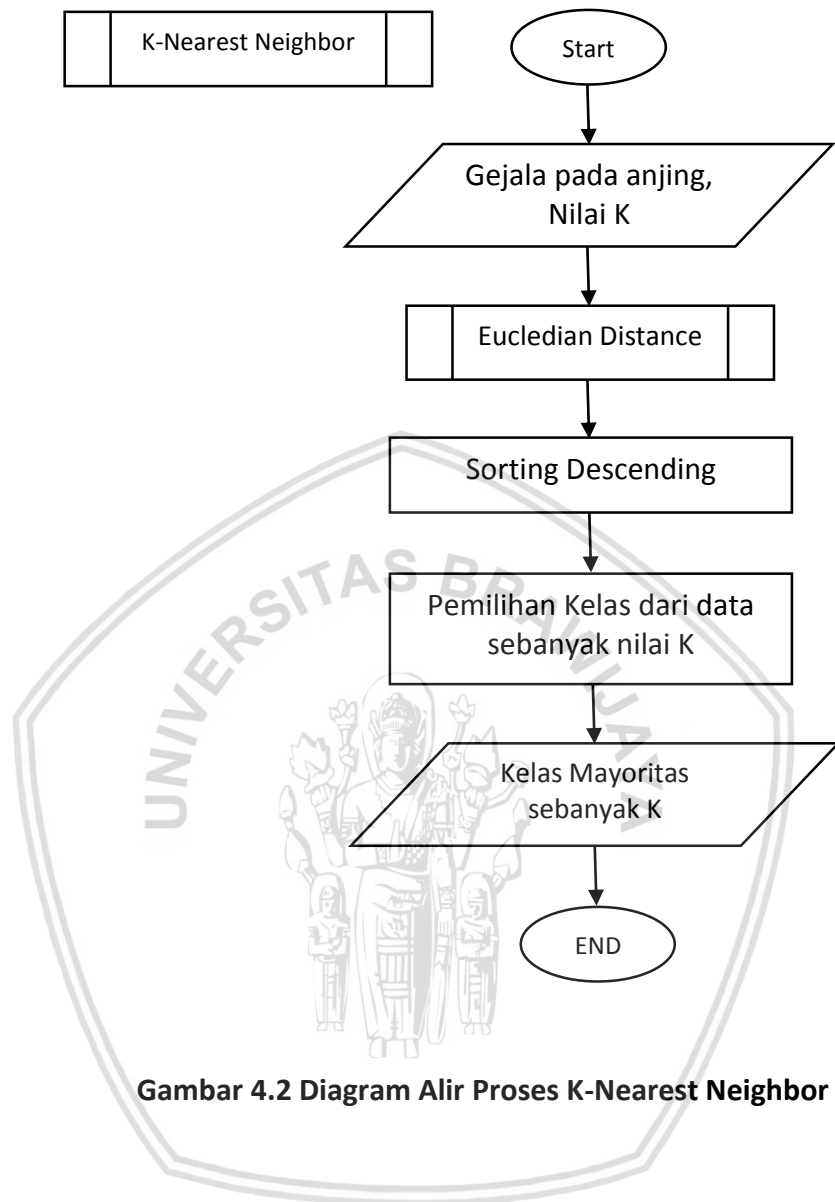
Gambar 4.1 Diagram Alir Tahapan Fuzzy K-NN

2. Proses Fuzzy K-Nearest Neighbor

Pada tahapan ini dilakukan proses pemberian nilai keanggotaan pada data uji setelah diproses menggunakan metode K-NN sehingga setiap data uji mempunyai nilai relatif pada setiap kelas dimana kelas dengan nilai keanggotaan yang lebih besar ialah yang menjadi kelas hasil pada proses FK-NN.

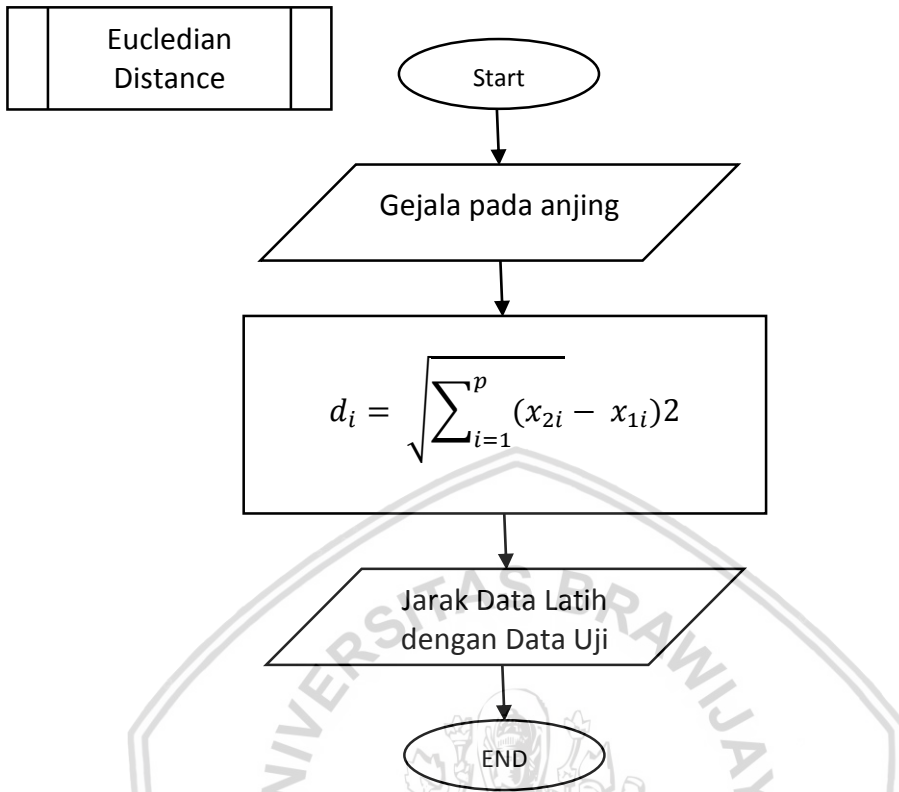
4.1.1 Proses K-Nearest Neighbor (K-NN)

Pada tahapan ini dilakukan klasifikasi dengan menggunakan algoritma *K-Nearest Neighbor* (K-NN) dimana nantinya akan dilakukan proses klasifikasi dengan menempatkan data uji pada salah satu kelas penyakit. Dalam menentukan data ke dalam sebuah kelas, pertama – tama digunakan persamaan 2-3 untuk menentukan jarak antar data latih dan data uji hingga kemudian diurutkan mulai dari nilai jarak terkecil hingga terbesar dan selanjutnya ditentukan jumlah k (tetangga terdekat) untuk didapat jumlah mayoritas kelas yang ada pada sejumlah k tersebut. Flowchart untuk proses klasifikasi *K-Nearest Neighbor* (K-NN) ditunjukkan oleh Gambar 4.2. dan di dalam proses K-NN terdapat pencarian *euclidean distance* yang ditunjukkan pada gambar 4.3.



Gambar 4.2 Diagram Alir Proses K-Nearest Neighbor





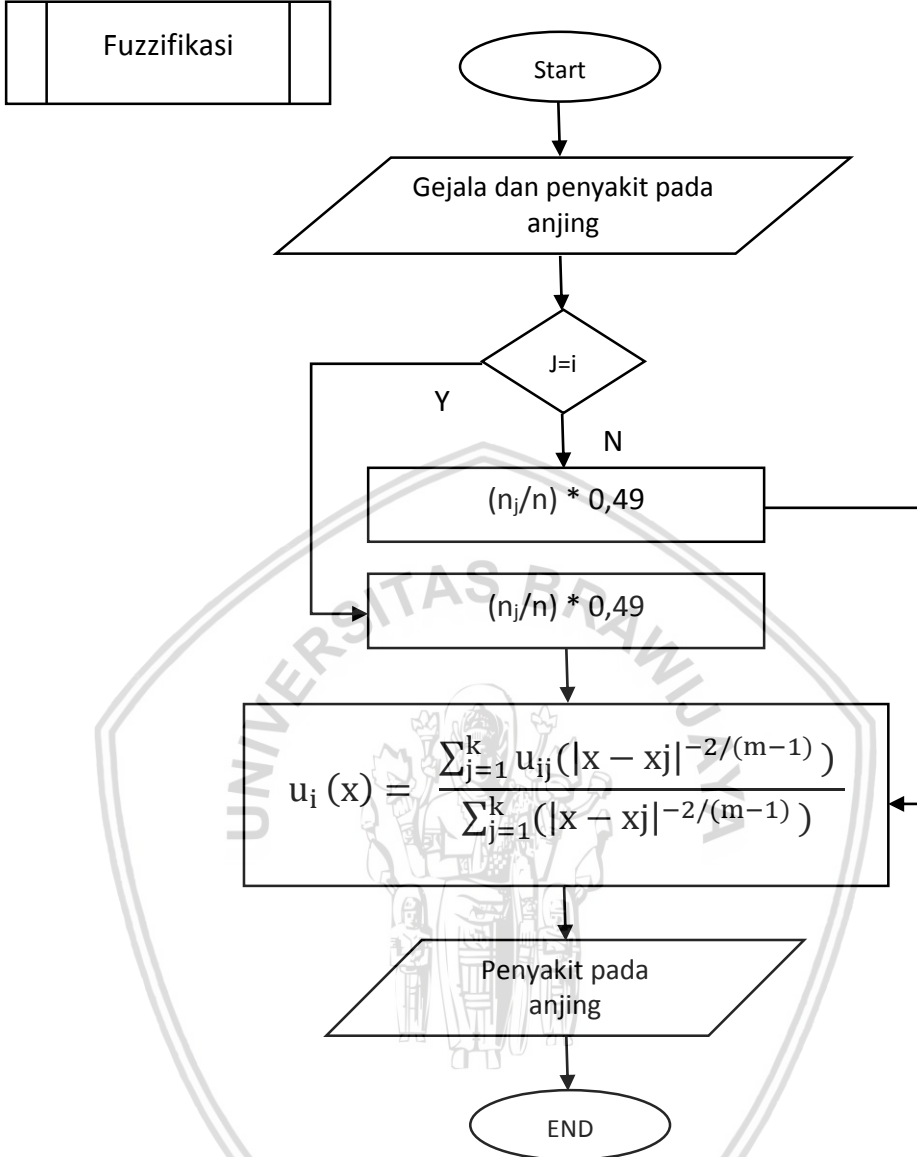
Gambar 4.3 Diagram Alir Eucledian Distance

#### 4.1.2 Proses Fuzzy K-Nearest Neighbor (FKNN)

Pada proses ini pada awal prosesnya sama dengan algoritma *K-Nearest Neighbor*, tetapi setelah proses penentuan nilai k selanjutnya *Fuzzy K-NN* memberikan nilai derajat keanggotaan pada tiap kelas untuk menentukan kelas data yang baru. Pada algoritma ini nilai keanggotaan tiap kelas dihitung dan diambil keanggotaan terbesar yang digunakan untuk menentukan kelas prediksi fuzzy. Hingga akhirnya output berupa hasil kelas prediksi dari data uji dan nilai keanggotaan tiap kelas. Untuk proses klasifikasi *Fuzzy K-Nearest Neighbor* (FK-NN) ditunjukkan oleh Gambar 4.4.







Gambar 4.4 Diagram Alir Fuzzyfikasi FKNN



## 4.2 Contoh Perhitungan Manual

Pada proses ini dilakukan perhitungan manual mulai dari awal hingga akhir. Tidak seperti perhitungan dengan menggunakan program yang menggunakan puluhan data latih. Pada perhitungan manual hanya digunakan 15 data latih agar proses perhitungan simpel dan mudah dipahami. Selain itu digunakan pula data uji sebanyak satu data untuk menguji data tersebut masuk ke dalam kelas penyakit mana. Contoh data latih ditunjukkan pada tabel 4.1.

**Tabel 4.1 Contoh data latih manualisasi**

PENYAKIT	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14	G15	G16	G17
P1	8	0	6	0	8	0	0	0	0	0	0	0	0	0	0	0	0
P1	8	7	6	8	0	5	0	0	0	0	0	0	0	0	0	0	0
P1	8	7	0	8	8	0	0	0	0	0	0	0	0	0	0	0	0
P2	0	0	0	8	0	0	0	6	8	0	0	0	0	0	0	0	0
P2	7	0	0	8	0	0	7	6	8	0	0	0	0	0	0	0	0
P2	6	0	0	8	0	0	0	6	8	5	0	0	0	0	0	0	0
P3	0	0	0	0	0	0	0	0	0	0	6	7	0	0	0	0	0
P3	0	0	0	0	0	0	0	0	0	0	0	6	7	0	0	0	0
P3	0	0	0	0	0	0	0	0	0	0	8	0	7	0	0	0	0
P4	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	7
P4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7	0	8
P4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8
P5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8
P5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7	8
P5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8	0	8

Pada contoh manualisasi ini hanya akan digunakan satu data uji saja, data uji yang digunakan ditunjukkan oleh table 4.2.

**Tabel 4.2 Contoh data uji manualisasi**

PENYAKIT	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14	G15	G16	G17
??	7	7	0	6	0	0	0	8	0	0	0	0	0	0	0	0	0

### 4.2.1 Hitung Euclidian Distance Data Latih Terhadap Data Uji

Tahap pertama dalam perhitungan manual diagnosis penyakit tanaman cabai adalah penghitungan jarak setiap data latih terhadap data uji menggunakan persamaan 2-3.

Contoh :



Data latih :

1	8	0	6	0	8	0	0	0	0	0	0	0	0	0	0	0
2	8	7	6	8	0	5	0	0	0	0	0	0	0	0	0	0
3	8	7	0	8	8	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	8	0	0	0	6	8	0	0	0	0	0	0	0
5	7	0	0	8	0	0	7	6	8	0	0	0	0	0	0	0
6	6	0	0	8	0	0	0	6	8	5	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	6	7	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	6	7	0	0	0	0
9	0	0	0	0	0	0	0	0	0	8	0	7	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	7
11	0	0	0	0	0	0	0	0	0	0	0	0	0	7	0	8
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7	8
15	0	0	0	0	0	0	0	0	0	0	0	0	0	8	0	8

Data Uji :

??	7	7	0	6	0	0	0	8	0	0	0	0	0	0	0	0
----	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Hitung :

$$D_1 = \sqrt{(8-7)^2 + (0-7)^2 + (6-0)^2 + (0-6)^2 + (8-0)^2 + (0-0)^2 + (0-0)^2 + (0-8)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2}$$

$$D_1 = 16$$

$$D_2 = \sqrt{(8-7)^2 + (7-7)^2 + (6-0)^2 + (8-6)^2 + (8-0)^2 + (0-0)^2 + (5-0)^2 + (0-8)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2}$$

$$D_2 = 11$$

- 
- 
- 
- 
- 
- 
- 
- 
- 
- 
- 
- 
- 
- 
- 
- 
-

$$D_{15} = \sqrt{(0-7)^2 + (0-7)^2 + (0-0)^2 + (0-6)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-8)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (8-0)^2 + (0-0)^2 + (8-0)^2}$$

$$D_{15} = 18$$

Hasil Perhitungan eucledian distance ditunjukkan oleh table 4.3.

**Tabel 4.3 Hasil perhitungan eucledian distance**

PENYAKIT	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14	G15	G16	G17	Jarak
P1	8	0	6	0	8	0	0	0	0	0	0	0	0	0	0	0	0	16
P1	8	7	6	8	0	5	0	0	0	0	0	0	0	0	0	0	0	11
P1	8	7	0	8	8	0	0	0	0	0	0	0	0	0	0	0	0	12
P2	0	0	0	8	0	0	0	6	8	0	0	0	0	0	0	0	0	13
P2	7	0	0	8	0	0	7	6	8	0	0	0	0	0	0	0	0	13
P2	6	0	0	8	0	0	0	6	8	5	0	0	0	0	0	0	0	12
P3	0	0	0	0	0	0	0	0	0	0	6	7	0	0	0	0	0	17
P3	0	0	0	0	0	0	0	0	0	0	0	6	7	0	0	0	0	17
P3	0	0	0	0	0	0	0	0	0	0	8	0	7	0	0	0	0	18
P4	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	7	17
P4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7	0	8	18
P4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8	16
P5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8	16
P5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7	8	18
P5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8	0	8	18

#### 4.2.2 Sorting Berdasarkan Eucledian Distance

Tahap Kedua adalah dilakukan sorting data latih mulai dari nilai eucledian distance yang terkecil hingga yang terbesar. Data sebelum disorting ditunjukkan oleh tabel 4.4 dan Hasil Sorting ditunjukkan oleh tabel 4.5.

Contoh :

Data sebelum dilakukan sorting :



**Tabel 4.4 Data sebelum disorting**

PENYAKIT	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14	G15	G16	G17	Jarak
P1	8	0	6	0	8	0	0	0	0	0	0	0	0	0	0	0	0	16
P1	8	7	6	8	0	5	0	0	0	0	0	0	0	0	0	0	0	11
P1	8	7	0	8	8	0	0	0	0	0	0	0	0	0	0	0	0	12
P2	0	0	0	8	0	0	0	6	8	0	0	0	0	0	0	0	0	13
P2	7	0	0	8	0	0	7	6	8	0	0	0	0	0	0	0	0	13
P2	6	0	0	8	0	0	0	6	8	5	0	0	0	0	0	0	0	12
P3	0	0	0	0	0	0	0	0	0	0	6	7	0	0	0	0	0	17
P3	0	0	0	0	0	0	0	0	0	0	0	6	7	0	0	0	0	17
P3	0	0	0	0	0	0	0	0	0	0	8	0	7	0	0	0	0	18
P4	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	7	17
P4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7	0	8	18
P4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8	16
P5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8	16
P5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7	8	18
P5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8	0	8	18

**Tabel 4.5 Data yang telah disorting**

PENYAKIT	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14	G15	G16	G17	Jarak
P1	8	7	6	8	0	5	0	0	0	0	0	0	0	0	0	0	0	11
P1	8	7	0	8	8	0	0	0	0	0	0	0	0	0	0	0	0	12
P2	6	0	0	8	0	0	0	6	8	5	0	0	0	0	0	0	0	12
P2	0	0	0	8	0	0	0	6	8	0	0	0	0	0	0	0	0	13
P2	7	0	0	8	0	0	7	6	8	0	0	0	0	0	0	0	0	13
P1	8	0	6	0	8	0	0	0	0	0	0	0	0	0	0	0	0	16
P4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8	16
P5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8	16
P3	0	0	0	0	0	0	0	0	0	0	6	7	0	0	0	0	0	17
P3	0	0	0	0	0	0	0	0	0	0	0	6	7	0	0	0	0	17
P4	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	7	17
P3	0	0	0	0	0	0	0	0	0	0	8	0	7	0	0	0	0	18
P4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7	0	8	18



Tabel 4.5 (lanjutan)

P5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7	8	18
P5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8	0	8	18

**4.2.3 Pemilihan data sebanyak nilai K**

Setelah data disorting maka akan masuk ke tahapan selanjutnya yaitu pemilihan dan pengambilan data sejumlah k, pada contoh perhitungan manual kali ini nilai k adalah 5. Tabel data sebelum diambil sebanyak nilai K ditunjukkan oleh tabel 4.6 dan tabel data setelah diambil sebanyak nilai K ditunjukkan oleh tabel 4.7.

Tabel 4.6 Data sebelum diambil sejumlah K

PENYAKIT	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14	G15	G16	G17	Jarak
P1	8	7	6	8	0	5	0	0	0	0	0	0	0	0	0	0	0	11
P1	8	7	0	8	8	0	0	0	0	0	0	0	0	0	0	0	0	12
P2	6	0	0	8	0	0	0	6	8	5	0	0	0	0	0	0	0	12
P2	0	0	0	8	0	0	0	6	8	0	0	0	0	0	0	0	0	13
P2	7	0	0	8	0	0	7	6	8	0	0	0	0	0	0	0	0	13
P1	8	0	6	0	8	0	0	0	0	0	0	0	0	0	0	0	0	16
P4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8	16
P5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8	16
P3	0	0	0	0	0	0	0	0	0	0	6	7	0	0	0	0	0	17
P3	0	0	0	0	0	0	0	0	0	0	0	6	7	0	0	0	0	17
P4	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	7	17
P3	0	0	0	0	0	0	0	0	0	0	8	0	7	0	0	0	0	18
P4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7	0	8	18
P5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7	8	18
P5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8	0	8	18

Tabel 4.7 Data yang diambil sejumlah K

PENYAKIT	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14	G15	G16	G17	Jarak
P1	8	7	6	8	0	5	0	0	0	0	0	0	0	0	0	0	0	11
P1	8	7	0	8	8	0	0	0	0	0	0	0	0	0	0	0	0	12
P2	6	0	0	8	0	0	0	6	8	5	0	0	0	0	0	0	0	12



Tabel 4.7 (lanjutan)

P2	0	0	0	8	0	0	0	6	8	0	0	0	0	0	0	0	0	13
P2	7	0	0	8	0	0	7	6	8	0	0	0	0	0	0	0	0	13

#### 4.2.4 Fuzzyfikasi

Setelah proses penentuan nilai k maka akan dilakukan proses fuzzy dimana proses tersebut bertujuan untuk mencari nilai keanggotaan data uji pada masing - masing kelas penyakit dengan menggunakan persamaan 2-4. Untuk itu perlu diketahui nilai n (jumlah data latih) = 15, kemudian kelas P1 sebanyak 3, kelas P2 sebanyak 3, kelas P3 sebanyak 3, kelas P4 sebanyak 3, kelas P5 sebanyak 3, dimana komponen tersebut diperlukan untuk mendapatkan nilai *membership* pada tiap kelas.

$$P1y = 0,51 + ((3/15)*0,49)$$

$$= 0,6$$

$$P1n = (3/15)*0,49$$

$$= 0,1$$

$$P2y = 0,51 + ((3/15)*0,49)$$

$$= 0,6$$

$$P2n = (3/15)*0,49$$

$$= 0,1$$

$$P3y = 0,51 + ((3/15)*0,49)$$

$$= 0,6$$

$$P3n = (3/15)*0,49$$

$$= 0,1$$

$$P4y = 0,51 + ((3/15)*0,49)$$

$$= 0,6$$

$$P4n = (3/15)*0,49$$

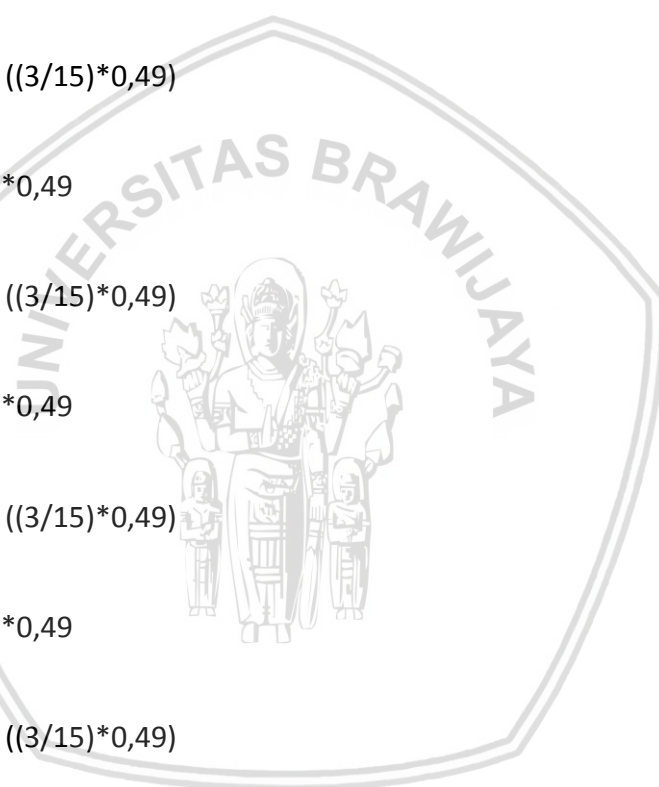
$$= 0,1$$

$$P5y = 0,51 + ((3/15)*0,49)$$

$$= 0,6$$

$$P5n = (3/15)*0,49$$

$$= 0,1$$





### 4.2.5 Defuzzyfikasi

Setelah didapatkan nilai *membership* untuk masing – masing kelas, dilakukan perhitungan nilai keanggotaan data uji pada tiap kelas menggunakan persamaan 2-5, dimana pada perhitungan di sini digunakan nilai  $m = 2$ .

$$P1 = \frac{\left(0,6 \left(11^{\frac{-2}{2-1}}\right)\right) + \left(0,6 \left(12^{\frac{-2}{2-1}}\right)\right) + \left(0,1 + \left(12^{\frac{-2}{2-1}}\right)\right) + \left(0,1 \left(13^{\frac{-2}{2-1}}\right)\right) + \left(0,1 \left(13^{\frac{-2}{2-1}}\right)\right)}{\left(11^{\frac{-2}{2-1}}\right) + \left(12^{\frac{-2}{2-1}}\right) + \left(12^{\frac{-2}{2-1}}\right) + \left(13^{\frac{-2}{2-1}}\right) + \left(13^{\frac{-2}{2-1}}\right)}$$

$$P1 = 0,326$$

$$P2 = \frac{\left(0,1 \left(11^{\frac{-2}{2-1}}\right)\right) + \left(0,1 \left(12^{\frac{-2}{2-1}}\right)\right) + \left(0,6 + \left(12^{\frac{-2}{2-1}}\right)\right) + \left(0,6 \left(13^{\frac{-2}{2-1}}\right)\right) + \left(0,6 \left(13^{\frac{-2}{2-1}}\right)\right)}{\left(11^{\frac{-2}{2-1}}\right) + \left(12^{\frac{-2}{2-1}}\right) + \left(12^{\frac{-2}{2-1}}\right) + \left(13^{\frac{-2}{2-1}}\right) + \left(13^{\frac{-2}{2-1}}\right)}$$

$$P2 = 0,380$$

$$P3 = \frac{\left(0,1 \left(11^{\frac{-2}{2-1}}\right)\right) + \left(0,1 \left(12^{\frac{-2}{2-1}}\right)\right) + \left(0,1 + \left(12^{\frac{-2}{2-1}}\right)\right) + \left(0,1 \left(13^{\frac{-2}{2-1}}\right)\right) + \left(0,1 \left(13^{\frac{-2}{2-1}}\right)\right)}{\left(11^{\frac{-2}{2-1}}\right) + \left(12^{\frac{-2}{2-1}}\right) + \left(12^{\frac{-2}{2-1}}\right) + \left(13^{\frac{-2}{2-1}}\right) + \left(13^{\frac{-2}{2-1}}\right)}$$

$$P3 = 0,098$$

$$P4 = \frac{\left(0,1 \left(11^{\frac{-2}{2-1}}\right)\right) + \left(0,1 \left(12^{\frac{-2}{2-1}}\right)\right) + \left(0,1 + \left(12^{\frac{-2}{2-1}}\right)\right) + \left(0,1 \left(13^{\frac{-2}{2-1}}\right)\right) + \left(0,1 \left(13^{\frac{-2}{2-1}}\right)\right)}{\left(11^{\frac{-2}{2-1}}\right) + \left(12^{\frac{-2}{2-1}}\right) + \left(12^{\frac{-2}{2-1}}\right) + \left(13^{\frac{-2}{2-1}}\right) + \left(13^{\frac{-2}{2-1}}\right)}$$

$$P4 = 0,098$$

$$P5 = \frac{\left(0,1 \left(11^{\frac{-2}{2-1}}\right)\right) + \left(0,1 \left(12^{\frac{-2}{2-1}}\right)\right) + \left(0,1 + \left(12^{\frac{-2}{2-1}}\right)\right) + \left(0,1 \left(13^{\frac{-2}{2-1}}\right)\right) + \left(0,1 \left(13^{\frac{-2}{2-1}}\right)\right)}{\left(11^{\frac{-2}{2-1}}\right) + \left(12^{\frac{-2}{2-1}}\right) + \left(12^{\frac{-2}{2-1}}\right) + \left(13^{\frac{-2}{2-1}}\right) + \left(13^{\frac{-2}{2-1}}\right)}$$

$$P5 = 0,098$$



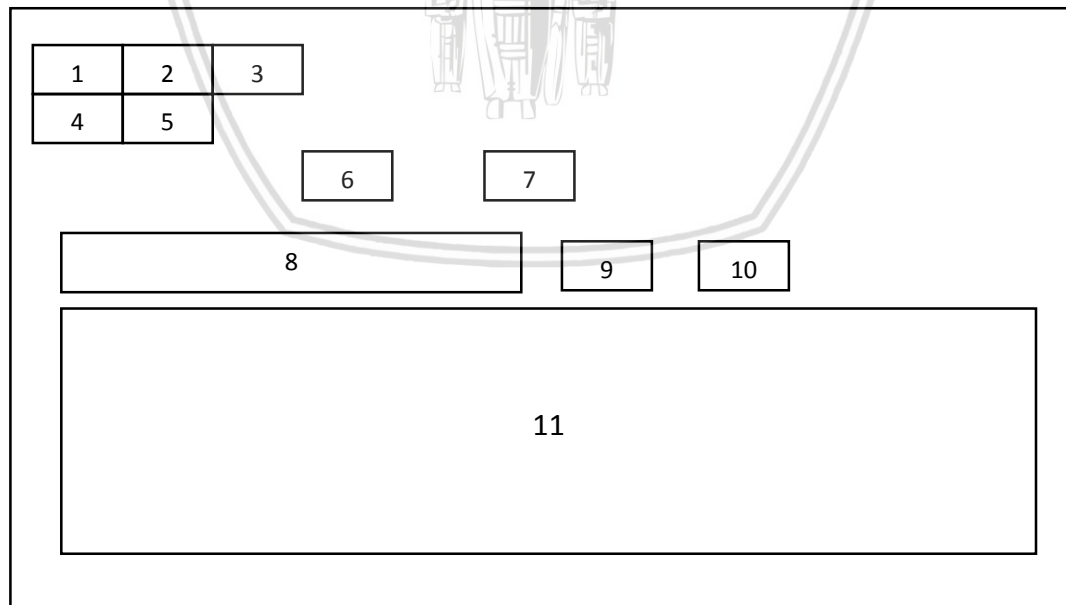
Dari perhitungan tersebut didapatkan nilai keanggotaan dari tiap kelas. Dimana akhirnya didapatkan untuk nilai keanggotaan data uji (x) untuk kelas P1 bernilai = 0,326. Sedangkan untuk untuk nilai keanggotaan data uji (x) pada kelas P2 didapatkan nilai 0,380. Sedangkan untuk untuk nilai keanggotaan data uji (x) pada kelas P3 didapatkan nilai 0,098. Sedangkan untuk untuk nilai keanggotaan data uji (x) pada kelas P4 didapatkan nilai 0,098. Sedangkan untuk untuk nilai keanggotaan data uji (x) pada kelas P5 didapatkan nilai 0,098. Dari perhitungan tersebut dapat disimpulkan bahwa nilai peluang terbesar dimiliki oleh P2, sehingga dapat disimpulkan bahwa data uji yang dimasukkan termasuk dalam P2.

### 4.3 Perancangan Antarmuka

Perancangan antarmuka dilakukan untuk memudahkan pembuatan antarmuka di mana antarmuka sendiri merupakan penghubung antara pengguna dengan sistem. Sistem diagnosis penyakit anjing yang akan dibangun merupakan sistem menggunakan bahasa pemrograman java. Sistem akan memberikan keluaran berupa hasil perhitungan-perhitungan proses FK-NN dan hasil diagnosis penyakit anjing berdasarkan masukan yang diberikan yang berasal dari pengguna. Masukan-masukan tersebut berupa gejala penyakit. Masukan dari pengguna kemudian dibaca oleh sistem dan dilakukan perhitungan menggunakan FK-NN.

#### 4.3.1 Perancangan Antarmuka Data Latih

Perancangan antarmuka data latih ini adalah halaman awal yang muncul ketika perangkat lunak dijalankan, pada halaman ini pengguna diminta memasukkan data latih yang tersimpan di file .txt. Perancangan antarmuka data latih ditunjukkan oleh gambar 4.5.



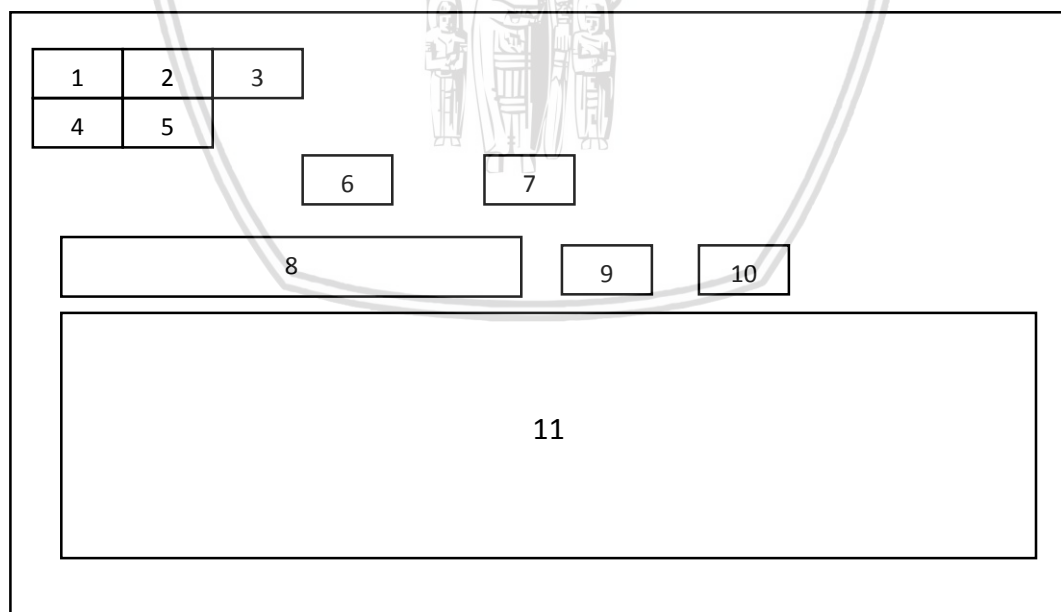
Gambar 4.5 Perancangan Antarmuka Data Latih

Keterangan :

1. Data
2. Proses FKNN
3. Hasil Klasifikasi dan Akurasi
4. Data Latih
5. Data Uji
6. Jumlah Data Latih
7. Jumlah Gejala
8. Lokasi Data Latih
9. Browse
10. Ambil Data
11. List Data Latih

#### 4.3.2 Perancangan Antarmuka Data Uji

Perancangan antarmuka data uji ini adalah halaman yang digunakan untuk memanggil data uji. Pada halaman ini pengguna diminta memasukkan data uji yang tersimpan di file .txt. Perancangan antarmuka data uji ditunjukkan oleh gambar 4.6.



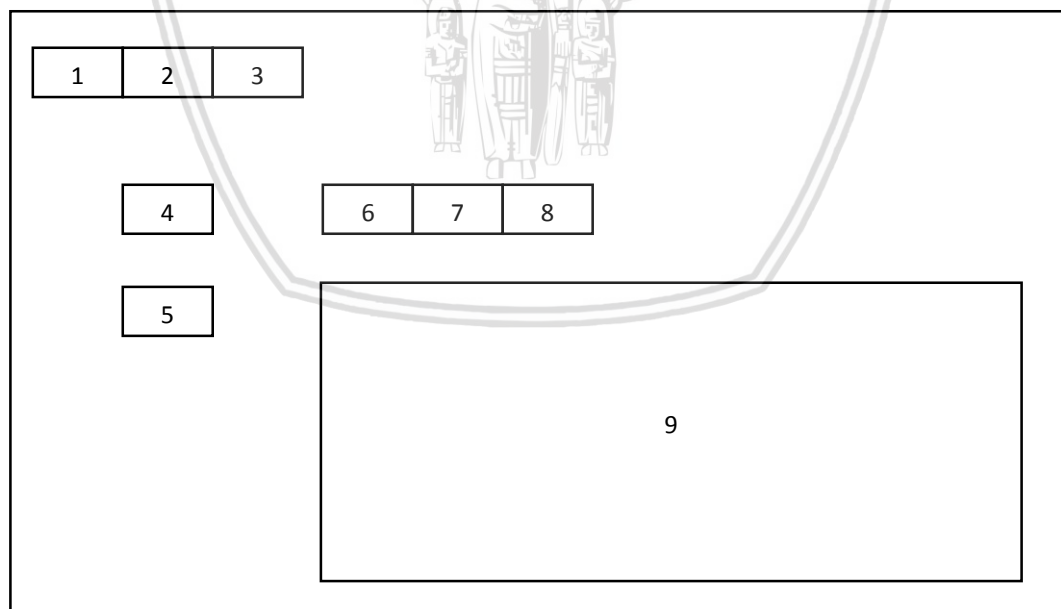
Gambar 4.6 Perancangan Antarmuka Data Uji

Keterangan :

1. Data
2. Proses FKNN
3. Hasil Klasifikasi dan Akurasi
4. Data Latih
5. Data Uji
6. Jumlah Data Latih
7. Jumlah Gejala
8. Lokasi Data Uji
9. Browse
10. Ambil Data
11. List Data Uji

#### 4.3.3 Perancangan Antarmuka Proses FKNN

Perancangan antarmuka FKNN ini adalah halaman dimana akan ditampilkan hasil proses perhitungan FKNN setelah pengguna memasukkan data latih dan data uji disertai dengan nilai K yang dimasukkan pada halaman ini. Halaman antarmuka proses FKNN ditunjukkan oleh Gambar 4.7.



Gambar 4.7 Perancangan Antarmuka Proses FKNN

Keterangan :

1. Data
2. Proses FKNN
3. Hasil Klasifikasi dan Akurasi
4. Nilai K
5. Proses
6. Eucledian
7. K-Data
8. Proses Fuzzy
9. Hasil Perhitungan



## BAB 5 IMPLEMENTASI

Berdasarkan analisis kebutuhan dan perancangan yang telah dilakukan pada sebelumnya, bagian ini akan membahas tentang implementasi algoritma FK-NN untuk diagnosis penyakit anjing. Bahasan-bahasan dalam bab ini terdiri dari spesifikasi sistem, batasan implementasi, implementasi algoritma, dan implementasi antarmuka.

### 5.1 Implementasi sistem

Implementasi sistem memerlukan dua aspek penting agar sistem dapat berjalan dengan baik sesuai fungsinya. Dua aspek tersebut adalah perangkat keras dan perangkat lunak yang digunakan untuk membangun sistem.

#### 5.1.1 Spesifikasi Perangkat Keras

Tabel 5.1 menunjukkan daftar spesifikasi perangkat keras yang digunakan untuk membangun sistem.

**Tabel 5.1 Spesifikasi Perangkat Keras**

Nama Komponen	Spesifikasi
Prosesor	Intel(R) Core(TM) i3-3217U CPU @ 1,80GHz
Memori	4 GB
Kartu grafis	Intel(R) HD Graphics 4000
Hardisk	500 GB

#### 5.1.2 Spesifikasi Perangkat Lunak

Tabel 5.2 menunjukkan daftar spesifikasi perangkat lunak yang digunakan untuk membangun sistem.

**Tabel 5.2 Spesifikasi Perangkat Lunak**

Nama	Spesifikasi
Sistem operasi	Windows 10
Bahasa pemrograman	Java
<i>Tool</i> pemrograman	Netbeans 8.0.1, JDK

### 5.2 Batasan Implementasi

Adapun batasan-batasan dalam implementasi algoritma FK-NN untuk diagnosis penyakit anjing yaitu:

1. Sistem yang dibangun menggunakan bahasa pemrograman java.
2. Data yang digunakan dalam sistem disimpan dalam file txt.
3. Data yang digunakan berupa data-data gejala penyakit anjing,.
4. Keluaran dari sistem adalah salah satu dari 10 penyakit anjing.



5. Metode yang digunakan adalah fuzzy K-NN.

### 5.3 Implementasi Algoritma

Implementasi algoritme FK-NN untuk diagnosis penyakit anjing terdiri dari, perhitungan jarak Euclidean antara data latih dengan data uji, sorting hasil perhitungan jarak euclidian antara data latih dan data uji, pengambilan hasil perhitungan sebanyak k, fuzzifikasi dan defuzzifikasi.

#### 5.3.1 Implementasi Perhitungan Jarak Euclidean Data Latih dan Data Uji

Langkah berikutnya yaitu menghitung jarak Euclidean antara data latih dengan data uji yang ditunjukkan pada Source Code 5.1.

```
1 public void hitung_jarak (double data_uji[][], double
  data_latih[][], String kelas[]){
2   this.jarak = new
  double[data_uji.length][data_latih.length];
3   this.kelas = new
  String[data_uji.length][data_latih.length];
4
5   for (int i=0; i < jarak.length; i++){
6     for (int j=0; j < jarak[0].length; j++){
7       double tot = 0.0;
8       for (int k=0; k < data_latih[0].length; k++){
9         tot += Math.pow(data_uji[i][k]-data_latih[j][k], 2);
10      }
11     this.jarak[i][j] = Math.sqrt(tot);
12     this.kelas[i][j] = kelas[j];
13   }
14 }
15 }
16
17 public double[][] getJarak() {
18   return jarak;
19 }
20
21 public String[][] getKelas() {
22   return kelas;
23 }
```

**Source Code 5.1 Hitung Euclidian Distance**

Tahap pertama pada implementasi algoritma adalah perhitungan euclidian distance antara data latih dengan data uji, dimana dari perhitungan ini akan didapatkan nilai jarak data uji dengan setiap data latih.



### 5.3.2 Implementasi Pengurutan Jarak Euclidian

Langkah berikutnya yaitu pengurutan jarak Euclidean antara data latih dengan data uji yang ditunjukkan pada Source Code 5.2.

```
public void urut_jarak (double jarak[][], String
1 kelas[][]){
2     this.jarak_terurut = copy_jarak(jarak);
3     this.kelas_terurut = copy_kelas(kelas);
4     double temp_jrk = 0.0;
5     String temp_kls = " ";
6
7     for (int i=0; i < jarak_terurut.length; i++){
8         for (int j=0; j < jarak_terurut[0].length-
9 1; j++){
10            for (int k=0; k <
11 jarak_terurut[0].length-1-j; k++){
12                if (this.jarak_terurut[i][k] >
13 this.jarak_terurut[i][k+1]){
14                    temp_jrk =
15 this.jarak_terurut[i][k];
16                    this.jarak_terurut[i][k] =
17 this.jarak_terurut[i][k+1];
18                    this.jarak_terurut[i][k+1] =
19 temp_jrk;
20                    temp_kls =
21 this.kelas_terurut[i][k];
22                    this.kelas_terurut[i][k] =
23 this.kelas_terurut[i][k+1];
24                    this.kelas_terurut[i][k+1] =
25 temp_kls;
26                }
27            }
28        }
29    }
30
31    public double[][] getJarak_terurut() {
32        return jarak_terurut;
33    }
34
35    public String[][] getKelas_terurut() {
36        return kelas_terurut;
37    }
38}
```

### Source Code 5.2 Pengurutan Jarak Euclidian

Setelah sebelumnya dilakukan penghitungan jarak euclidian data latih terhadap data uji, kemudian dilakukan pengurutan hasil perhitungan jarak euclidian dari nilai terkecil hingga terbesar.

#### 5.3.3 Implementasi Pengambilan Hasil Penghitungan Jarak Euclidian Sebanyak K

Langkah berikutnya yaitu Pengambilan hasil penghitungan jarak euclidian sebanyak K yang ditunjukkan pada Source Code 5.3.

```

public void ambil_kdata (int K, double
1 jarak_terurut[][] , String kelas_terurut[][]){
    this.k_jarak = new
2 double[jarak_terurut.length][K];
    this.k_kelas = new
3 String[jarak_terurut.length][K];
4
5     for (int i=0; i < k_jarak.length; i++){
6         for (int j=0; j < k_jarak[0].length; j++){
7             this.k_jarak[i][j] =
8 jarak_terurut[i][j];
9             this.k_kelas[i][j] =
10 kelas_terurut[i][j];
11         }
12     }
13     public double[][] getK_jarak() {
14         return k_jarak;
15     }
16     public String[][] getK_kelas() {
17         return k_kelas;
18     }
19
20     public void cek_nilai_jarak_sama(String
21 k_kelas[][]){
22         this.sama_semua = new String[k_kelas.length];
23
24         for (int i=0; i < sama_semua.length; i++){
25             for (int j=1; j < k_kelas[0].length; j++){
26                 if (k_kelas[i][j-
27 1].equalsIgnoreCase(k_kelas[i][j])){
28                     this.sama_semua[i] = "Sama Semua";
29                 } else {
30                     this.sama_semua[i] = "Tidak Sama";
31                     break;

```

```

31         }
32     }
33 }
34
35 }
36
37 public String[] getSama_semua() {
38     return sama_semua;
39 }

```

**Source Code 5.3 Proses Pengambilan Hasil Penghitungan Jarak Euclidian Sebanyak K**

Setelah sebelumnya dilakukan pengurutan jarak euclidian data latih terhadap data uji, kemudian diambil jarak terkecil sebanyak K.

### 5.3.4 Implementasi Fuzzifikasi

Langkah berikutnya yaitu proses fuzzifikasi yang ditunjukkan pada Source Code 5.4.

```

public void fuzzyfikasi (String [] kelas_data_latih,
1 int jlh_dt_latih){
    this.jlh_tiap_kls =
2 hitung_jlh_tiap_kls(kelas_data_latih);
    this.fuzzyfikasi_y = hitung_py(jlh_tiap_kls,
3 jlh_dt_latih);
    this.fuzzyfikasi_n = hitung_pn(jlh_tiap_kls,
4 jlh_dt_latih);
5 }
6
7 public double[] getJlh_tiap_kls() {
8     return jlh_tiap_kls;
9 }
10
11 public double[] getFuzzyfikasi_y() {
12     return fuzzyfikasi_y;
13 }
14
15 public double[] getFuzzyfikasi_n() {
16     return fuzzyfikasi_n;
17 }
18 private double[] hitung_jlh_tiap_kls(String[]
19 kelas_data_latih) {
20     double jlh_dt_latih[] = new double[10];
21     for (int i=0; i < kelas_data_latih.length; i++){
22         //jumlah penyakit Coccidiosis
23         if
(kelas_data_latih[i].equalsIgnoreCase("Coccidiosis")){

```

```
24         jlh_dt_latih[0] += 1;
25     } else
26         if
27 (kelas_data_latih[i].equalsIgnoreCase("Demodicosis")){
28         jlh_dt_latih[1] += 1;
29     } else
30         if
31 (kelas_data_latih[i].equalsIgnoreCase("Distemper")){
32         jlh_dt_latih[2] += 1;
33     } else
34         if
35 (kelas_data_latih[i].equalsIgnoreCase("Helminthiasis")){
36         jlh_dt_latih[4] += 1;
37     } else
38         if
39 (kelas_data_latih[i].equalsIgnoreCase("Otitis")){
40         jlh_dt_latih[5] +=
41 1;
42     } else
43         if
44 (kelas_data_latih[i].equalsIgnoreCase("Parvo")){
45         jlh_dt_latih[6]
46 +=1;
47     } else
48         if
49 (kelas_data_latih[i].equalsIgnoreCase("Ringworm")){
50         jlh_dt_latih[7] +=1;
51     } else
52         if
53 (kelas_data_latih[i].equalsIgnoreCase("Scabiosis")){
54         jlh_dt_latih[8] +=1;
55     } else
56         if
57 (kelas_data_latih[i].equalsIgnoreCase("Urolithiasis")){
58         jlh_dt_latih[9] +=1;
59     }
60 }
61 }
62
63     return jlh_dt_latih;
64 }
```

```

        private double[] hitung_py(double[] jlh_tiap_kls,
58 int jlh_dt_latih) {
            double tmp_py[] = new
59 double[jlh_tiap_kls.length];
60
61         for (int i=0; i < jlh_tiap_kls.length; i++){
                tmp_py[i] =
62 0.51+((jnh_tiap_kls[i]/jnh_dt_latih)*0.49);
63         }
64
65         return tmp_py;
66     }
67
        private double[] hitung_pn(double[] jlh_tiap_kls,
68 int jlh_dt_latih) {
            double tmp_pn[] = new
69 double[jlh_tiap_kls.length];
70
71         for (int i=0; i < jlh_tiap_kls.length; i++){
                tmp_pn[i] =
72 (jnh_tiap_kls[i]/jnh_dt_latih)*0.49;
73         }
74     }
75 }

```

**Source Code 5.4 Implementasi Fuzzifikasi**

Setelah sebelumnya dilakukan pengambilan jarak euclidian data latih terhadap data uji terkecil sebanyak K, kemudian dilakukan proses perhitungan fuzzifikasi setiap kelas.

### 5.3.5 Implementasi Defuzzifikasi

Langkah berikutnya yaitu proses defuzzifikasi yang ditunjukkan pada Source Code 5.5.

```

public void urut_defuzzyfikasi(double defuzzyfikasi[][] , String
1 defuzzyfikasi_kelas[][]){
2     this.def_terurut = copy_defuzzyfikasi(defuzzyfikasi);
3     this.def_kls_terurut= copy_kls_def(defuzzyfikasi_kelas);
4     double temp_jrk = 0.0;
5     String temp_kls = " ";
6
7     for (int i=0; i < def_terurut.length; i++){
8         for (int j=0; j < def_terurut[0].length-1; j++){
9             for (int k=0; k < def_terurut[0].length-1-j; k++){
                if (this.def_terurut[i][k] <
10 this.def_terurut[i][k+1]){
11                 temp_jrk = this.def_terurut[i][k];
                    this.def_terurut[i][k] =
12 this.def_terurut[i][k+1];

```

```
13         this.def_terurut[i][k+1] = temp_jrk;
14
15         temp_kls = this.def_kls_terurut[i][k];
16         this.def_kls_terurut[i][k+1] =
17         this.def_kls_terurut[i][k+1] = temp_kls;
18     }
19 }
20 }
21 }
22 }
23
24 public double[][] getDef_terurut() {
25     return def_terurut;
26 }
27
28 public String[][] getDef_kls_terurut() {
29     return def_kls_terurut;
30 }
31 private double[][] hitung_pembilang(String[][] k_kls, double[][]
32 k_jarak, double[] fuzzyfikasi_y, double[] fuzzyfikasi_n, int m) {
33     double tmp_defuzzyfikasi[][] = new
34     double[k_kls.length][10];
35
36     for (int i=0; i < k_kls.length; i++){
37         for (int j=0; j < k_kls[0].length; j++){
38
39             //defuzzyfikasi penyakit 1
40             if (k_kls[i][j].equalsIgnoreCase("Coccidiosis")){
41                 if (k_jarak[i][j] == 0.0){
42                     tmp_defuzzyfikasi[i][0] += 0.0;
43                 } else {
44                     tmp_defuzzyfikasi[i][0] +=
45                     fuzzyfikasi_y[0]*(Math.abs(Math.pow(k_jarak[i][j], (-2/(m-1))))));
46                 }
47             } else {
48                 if (k_jarak[i][j] == 0.0){
49                     tmp_defuzzyfikasi[i][0] += 0.0;
50                 } else {
51                     tmp_defuzzyfikasi[i][0] +=
52                     fuzzyfikasi_n[0]*(Math.abs(Math.pow(k_jarak[i][j], (-2/(m-1))))));
53                 }
54             }
55
56             //defuzzyfikasi penyakit 2
57             if (k_kls[i][j].equalsIgnoreCase("Demodicosis")){
```



```
54         if (k_jarak[i][j] == 0.0){
55             tmp_defuzzyfikasi[i][1] += 0.0;
56         } else {
57             tmp_defuzzyfikasi[i][1] +=
58             fuzzyfikasi_y[1]* (Math.abs(Math.pow(k_jarak[i][j], (-2/(m-1)))));
59         }
60     } else {
61         if (k_jarak[i][j] == 0.0){
62             tmp_defuzzyfikasi[i][1] += 0.0;
63         } else {
64             tmp_defuzzyfikasi[i][1] +=
65             fuzzyfikasi_n[1]* (Math.abs(Math.pow(k_jarak[i][j], (-2/(m-1)))));
66         }
67     }
68     //defuzzyfikasi penyakit 3
69     if (k_kls[i][j].equalsIgnoreCase("Distemper")){
70         if (k_jarak[i][j] == 0.0){
71             tmp_defuzzyfikasi[i][2] += 0.0;
72         } else {
73             tmp_defuzzyfikasi[i][2] +=
74             fuzzyfikasi_y[2]* (Math.abs(Math.pow(k_jarak[i][j], (-2/(m-1)))));
75         }
76     } else {
77         if (k_jarak[i][j] == 0.0){
78             tmp_defuzzyfikasi[i][2] += 0.0;
79         } else {
80             tmp_defuzzyfikasi[i][2] +=
81             fuzzyfikasi_n[2]* (Math.abs(Math.pow(k_jarak[i][j], (-2/(m-1)))));
82         }
83     }
84     //defuzzyfikasi penyakit 4
85     if (k_kls[i][j].equalsIgnoreCase("Ehrlichiosis")){
86         if (k_jarak[i][j] == 0.0){
87             tmp_defuzzyfikasi[i][3] += 0.0;
88         } else {
89             tmp_defuzzyfikasi[i][3] +=
90             fuzzyfikasi_y[3]* (Math.abs(Math.pow(k_jarak[i][j], (-2/(m-1)))));
91         }
92     } else {
93         if (k_jarak[i][j] == 0.0){
94             tmp_defuzzyfikasi[i][3] += 0.0;
95         } else {
96             tmp_defuzzyfikasi[i][3] +=
97             fuzzyfikasi_n[3]* (Math.abs(Math.pow(k_jarak[i][j], (-2/(m-1)))));
98         }
99     }
```

```
94         }
95     }
96
97     //defuzzyfikasi penyakit 5
98     if
99     (k_kls[i][j].equalsIgnoreCase("Helminthiasis")){
100         if (k_jarak[i][j] == 0.0){
101             tmp_defuzzyfikasi[i][4] += 0.0;
102         } else {
103             tmp_defuzzyfikasi[i][4] +=
104             fuzzyfikasi_y[4]*(Math.abs(Math.pow(k_jarak[i][j], (-2/(m-1)))));
105         }
106     } else {
107         if (k_jarak[i][j] == 0.0){
108             tmp_defuzzyfikasi[i][4] += 0.0;
109         } else {
110             tmp_defuzzyfikasi[i][4] +=
111             fuzzyfikasi_n[4]*(Math.abs(Math.pow(k_jarak[i][j], (-2/(m-1)))));
112         }
113     }
114
115     //defuzzyfikasi penyakit 6
116     if (k_kls[i][j].equalsIgnoreCase("Otitis")){
117         if (k_jarak[i][j] == 0.0){
118             tmp_defuzzyfikasi[i][5] += 0.0;
119         } else {
120             tmp_defuzzyfikasi[i][5] +=
121             fuzzyfikasi_y[5]*(Math.abs(Math.pow(k_jarak[i][j], (-2/(m-1)))));
122         }
123     } else {
124         if (k_jarak[i][j] == 0.0){
125             tmp_defuzzyfikasi[i][5] += 0.0;
126         } else {
127             tmp_defuzzyfikasi[i][5] +=
128             fuzzyfikasi_n[5]*(Math.abs(Math.pow(k_jarak[i][j], (-2/(m-1)))));
129         }
130     }
131
132     //defuzzyfikasi penyakit 7
133     if (k_kls[i][j].equalsIgnoreCase("Parvo")){
134         if (k_jarak[i][j] == 0.0){
135             tmp_defuzzyfikasi[i][6] += 0.0;
136         } else {
137             tmp_defuzzyfikasi[i][6] +=
138             fuzzyfikasi_y[6]*(Math.abs(Math.pow(k_jarak[i][j], (-2/(m-1)))));
139         }
140     }
```

```
134         } else {
135             if (k_jarak[i][j] == 0.0){
136                 tmp_defuzzyfikasi[i][6] += 0.0;
137             } else {
138                 tmp_defuzzyfikasi[i][6] +=
139                 fuzzyfikasi_n[6]*(Math.abs(Math.pow(k_jarak[i][j], (-2/(m-1)))));
140             }
141         }
142         //defuzzyfikasi penyakit 8
143         if (k_kls[i][j].equalsIgnoreCase("Ringworm")){
144             if (k_jarak[i][j] == 0.0){
145                 tmp_defuzzyfikasi[i][7] += 0.0;
146             } else {
147                 tmp_defuzzyfikasi[i][7] +=
148                 fuzzyfikasi_y[7]*(Math.abs(Math.pow(k_jarak[i][j], (-2/(m-1)))));
149             } else {
150                 if (k_jarak[i][j] == 0.0){
151                     tmp_defuzzyfikasi[i][7] += 0.0;
152                 } else {
153                     tmp_defuzzyfikasi[i][7] +=
154                     fuzzyfikasi_n[7]*(Math.abs(Math.pow(k_jarak[i][j], (-2/(m-1)))));
155                 }
156             }
157         }
158         //defuzzyfikasi penyakit 9
159         if (k_kls[i][j].equalsIgnoreCase("Scabiosis")){
160             if (k_jarak[i][j] == 0.0){
161                 tmp_defuzzyfikasi[i][8] += 0.0;
162             } else {
163                 tmp_defuzzyfikasi[i][8] +=
164                 fuzzyfikasi_y[8]*(Math.abs(Math.pow(k_jarak[i][j], (-2/(m-1)))));
165             }
166         } else {
167             if (k_jarak[i][j] == 0.0){
168                 tmp_defuzzyfikasi[i][8] += 0.0;
169             } else {
170                 tmp_defuzzyfikasi[i][8] +=
171                 fuzzyfikasi_n[8]*(Math.abs(Math.pow(k_jarak[i][j], (-2/(m-1)))));
172             }
173         }
174         //defuzzyfikasi penyakit 10
175         if (k_kls[i][j].equalsIgnoreCase("Urolithiasis")){
176             if (k_jarak[i][j] == 0.0){
```

```

175         tmp_defuzzyfikasi[i][9] += 0.0;
176     } else {
177         tmp_defuzzyfikasi[i][9] +=
178         fuzzyfikasi_y[9]* (Math.abs(Math.pow(k_jarak[i][j], (-2/(m-1)))));
179     }
180     } else {
181         if (k_jarak[i][j] == 0.0){
182             tmp_defuzzyfikasi[i][9] += 0.0;
183         } else {
184             tmp_defuzzyfikasi[i][9] +=
185             fuzzyfikasi_n[9]* (Math.abs(Math.pow(k_jarak[i][j], (-2/(m-1)))));
186         }
187     }
188 }
189 return tmp_defuzzyfikasi;
190 }
191
192 private double[] hitung_penyebut(double[][] k_jarak, int m) {
193     double tmp_penyebut[] = new double[k_jarak.length];
194
195     for (int i=0; i < k_jarak.length; i++){
196         for (int j=0; j < k_jarak[0].length; j++){
197             if (k_jarak[i][j] != 0.0){
198                 tmp_penyebut[i] +=
199                 Math.abs(Math.pow(k_jarak[i][j], (-2./ (m-1))));
200             }
201         }
202     }
203     return tmp_penyebut;
204 }
205
206 private double[][] hitung_defuzzyfikasi(double[][] pembilang,
207 double[] penyebut) {
208     double tmp_defuzzyfikasi[][] = new
209     double[pembilang.length][pembilang[0].length];
210
211     for (int i = 0; i < tmp_defuzzyfikasi.length; i++){
212         for (int j=0; j < tmp_defuzzyfikasi[0].length; j++){
213             if (pembilang[i][j] == 0.0 || penyebut[i] == 0.0){
214                 tmp_defuzzyfikasi[i][j] = 0.0;
215             } else {
216                 tmp_defuzzyfikasi[i][j] =
217                 pembilang[i][j]/penyebut[i];

```

```
215         }
216     }
217 }
218     return tmp_defuzzyfikasi;
219 }
220
221     private String[][] isi_kelas_defuzzyfikasi(String k_kls[][]) {
222         String tmp_kls_defuzzyfikasi[][] = new
223 String[k_kls.length][10];
224
225         for (int i=0; i < tmp_kls_defuzzyfikasi.length; i++){
226             tmp_kls_defuzzyfikasi[i][0] = "Coccidiosis";
227             tmp_kls_defuzzyfikasi[i][1] = "Demodicosis";
228             tmp_kls_defuzzyfikasi[i][2] = "Distemper";
229             tmp_kls_defuzzyfikasi[i][3] = "Ehrlichiosis";
230             tmp_kls_defuzzyfikasi[i][4] = "Helminthiasis";
231             tmp_kls_defuzzyfikasi[i][5] = "Otitis";
232             tmp_kls_defuzzyfikasi[i][6] = "Parvo";
233             tmp_kls_defuzzyfikasi[i][7] = "Ringworm";
234             tmp_kls_defuzzyfikasi[i][8] = "Scabiosis";
235             tmp_kls_defuzzyfikasi[i][9] = "Urolithiasis";
236         }
237
238         return tmp_kls_defuzzyfikasi;
239     }
240
241     private double[][] copy_defuzzyfikasi(double[][]
242 defuzzyfikasi) {
243         double[][] tmp_deff = new
244 double[defuzzyfikasi.length][defuzzyfikasi[0].length];
245
246         for (int i=0; i < tmp_deff.length; i++){
247             for (int j=0; j < tmp_deff[0].length; j++){
248                 tmp_deff[i][j] = defuzzyfikasi[i][j];
249             }
250         }
251
252         return tmp_deff;
253     }
254
255     private String[][] copy_kls_def(String[][]
256 defuzzyfikasi_kelas) {
257         String[][] def_kls = new
258 String[defuzzyfikasi_kelas.length][defuzzyfikasi_kelas[0].length];
259
260         for (int i=0; i < def_kls.length; i++){
```

```

256         for (int j=0; j < def_kls[0].length; j++){
257             def_kls[i][j] = defuzzyfikasi_kelas[i][j];
258         }
259     }
260
261     return def_kls;
262 }

```

#### Source Code 5.5 Implementasi Defuzzifikasi

Setelah didapatkan nilai jarak euclidian sebanyak K dan nilai fuzzifikasi setiap penyakit, kemudian dilakukan penghitungan defuzzifikasi.

#### 5.3.6 Implementasi Penentuan Kelas

Langkah berikutnya yaitu proses penentuan kelas yang ditunjukkan pada Source Code 5.6.

```

public voidurut_defuzzyfikasi(double defuzzyfikasi[][], String
1 defuzzyfikasi_kelas[][]){
2     this.def_terurut = copy_defuzzyfikasi(defuzzyfikasi);
3     this.def_kls_terurut= copy_kls_def(defuzzyfikasi_kelas);
4     double temp_jrk = 0.0;
5     String temp_kls = " ";
6
7     for (int i=0; i < def_terurut.length; i++){
8         for (int j=0; j < def_terurut[0].length-1; j++){
9             for (int k=0; k < def_terurut[0].length-1-j; k++){
10                if (this.def_terurut[i][k] < this.def_terurut[i][k+1]){
11                    temp_jrk = this.def_terurut[i][k];
12                    this.def_terurut[i][k] = this.def_terurut[i][k+1];
13                    this.def_terurut[i][k+1] = temp_jrk;
14
15                    temp_kls = this.def_kls_terurut[i][k];
16                    this.def_kls_terurut[i][k] = this.def_kls_terurut[i][k+1];
17                    this.def_kls_terurut[i][k+1] = temp_kls;
18                }
19            }
20        }
21    }
22 }
23
24 public double[][] getDef_terurut() {
25     return def_terurut;
26 }

```



```

27
28     public String[][] getDef_kls_terurut() {
29         return def_kls_terurut;
30     }
31
32     //7. Hasil klasifikasi
33     public void klasifikasi(double[][] def_terurut,
34 String def_kls_terurut[][], String[] sama_semua, String
35 K_data[][]) {
36         this.nilai_defuzzyfikasi = new
37 double[def_terurut.length];
38         this.kls_klasifikasi      = new
39 String[def_kls_terurut.length];
40
41         for (int i=0; i < def_terurut.length; i++){
42             if (sama_semua[i].equalsIgnoreCase("Tidak
43 Sama")) {
44                 this.nilai_defuzzyfikasi[i] =
45 def_terurut[i][0];
46                 this.kls_klasifikasi[i] =
47 def_kls_terurut[i][0];
48             } else {
49                 int idx_ = 0;
50                 for (int j=0; j <
51 def_kls_terurut[0].length; j++){
52                     if
53 (def_kls_terurut[i][j].equalsIgnoreCase(K_data[i][0])) {
54                         idx_ = j;
55                         break;
56                     }
57                 }
58                 this.nilai_defuzzyfikasi[i] =
59 def_terurut[i][idx_];
60                 this.kls_klasifikasi[i] =
61 K_data[i][0];
62             }
63         }
64     }

```

**Source Code 5.6 Implementasi Penentuan Kelas**

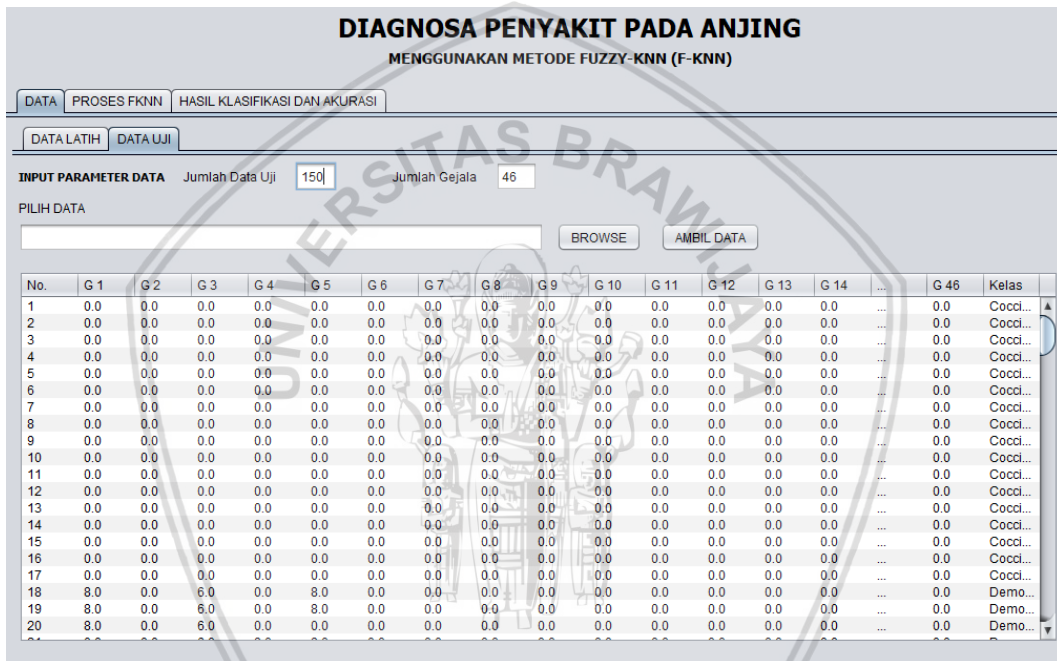
Tahap terakhir dari FKNN adalah penentuan kelas, kelas dengan nilai terbesar yang dijadikan sebagai kelas data uji.

### 5.4 Implementasi Antarmuka

Antarmuka bertujuan untuk memudahkan interaksi pengguna dengan sistem, Implementasi antarmuka terdiri dari tampilan antarmuka awal, data uji, proses FKNN, Penentuan Kelas dan Akurasi.

#### 5.4.1 Implementasi Antarmuka Awal

Tampilan awal merupakan tampilan pertama saat pengguna menjalankan aplikasi. Pada halaman awal pengguna diminta memasukkan file data latih dan data uji yang tersimpan dalam format .txt. Tampilan antarmuka awal ditunjukkan oleh gambar 5.1 dan gambar 5.2.



Gambar 5.1 Antarmuka Halaman Pilih Data Uji



### DIAGNOSA PENYAKIT PADA ANJING MENGUNAKAN METODE FUZZY-KNN (F-KNN)

DATA | PROSES FKNN | HASIL KLASIFIKASI DAN AKURASI

DATA LATIH | DATA UJI

**INPUT PARAMETER DATA**    Jumlah Data Latih     Jumlah Gejala

PILIH DATA

No.	G 1	G 2	G 3	G 4	G 5	G 6	G 7	G 8	G 9	G 10	G 11	G 12	G 13	G 14	...	G 46	Kelas
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	Cocci...
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	Cocci...
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	Cocci...
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	Cocci...
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	Cocci...
6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	Cocci...
7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	Cocci...
8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	Cocci...
9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	Cocci...
10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	Cocci...
11	8.0	7.0	0.0	8.0	8.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	Demo...
12	8.0	0.0	6.0	8.0	0.0	5.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	Demo...
13	8.0	7.0	6.0	8.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	Demo...
14	8.0	7.0	6.0	8.0	0.0	5.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	Demo...
15	8.0	7.0	0.0	8.0	8.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	Demo...
16	8.0	7.0	0.0	8.0	8.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	Demo...
17	8.0	7.0	6.0	8.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	Demo...
18	8.0	0.0	0.0	8.0	0.0	5.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	Demo...
19	8.0	7.0	6.0	8.0	0.0	5.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	Demo...
20	8.0	0.0	6.0	8.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	Demo...

Gambar 5.2 Antarmuka Halaman Pilih Data Latih



### 5.4.2 Implementasi Antarmuka Proses FKNN

Pada implementasi antarmuka proses FKNN pengguna diminta memasukkan nilai K kemudian memilih tombol proses, selanjutnya akan ditampilkan hasil perhitungan fuzzy KNN mulai dari perhitungan jarak Euclidian hingga Defuzzifikasi yang ditunjukkan oleh gambar 5.3 hingga gambar 5.7.

**DIAGNOSA PENYAKIT PADA ANJING**  
MENGUNAKAN METODE FUZZY-KNN (F-KNN)

DATA | PROSES FKNN | HASIL KLASIFIKASI DAN AKURASI

PARAMETER

NILAI K-

JUMLAH DATA UJI

JUMLAH DATA LATIH

D-EUCLIDEAN | K-DATA | PROSES FUZZY

SEBELUM DIURUT | SETELAH DIURUT

[ 128 ]	18,055 ==>	Coccidiosis	17,635 ==>	Coccidiosis	18,815 ==>	Coccidiosis	18,815 ==>	Coccidiosis	20,075 ==>
[ 129 ]	17,692 ==>	Coccidiosis	17,263 ==>	Coccidiosis	18,466 ==>	Coccidiosis	18,466 ==>	Coccidiosis	19,748 ==>
[ 130 ]	17,378 ==>	Coccidiosis	16,941 ==>	Coccidiosis	18,166 ==>	Coccidiosis	18,166 ==>	Coccidiosis	19,468 ==>
[ 131 ]	17,378 ==>	Coccidiosis	16,941 ==>	Coccidiosis	18,166 ==>	Coccidiosis	18,166 ==>	Coccidiosis	19,468 ==>
[ 132 ]	16,643 ==>	Coccidiosis	16,186 ==>	Coccidiosis	17,464 ==>	Coccidiosis	17,464 ==>	Coccidiosis	18,815 ==>
[ 133 ]	18,055 ==>	Coccidiosis	17,635 ==>	Coccidiosis	18,815 ==>	Coccidiosis	18,815 ==>	Coccidiosis	20,075 ==>
[ 134 ]	18,055 ==>	Coccidiosis	17,635 ==>	Coccidiosis	18,815 ==>	Coccidiosis	18,815 ==>	Coccidiosis	20,075 ==>
[ 135 ]	17,378 ==>	Coccidiosis	16,941 ==>	Coccidiosis	18,166 ==>	Coccidiosis	18,166 ==>	Coccidiosis	19,468 ==>
[ 136 ]	18,055 ==>	Coccidiosis	17,635 ==>	Coccidiosis	18,815 ==>	Coccidiosis	18,815 ==>	Coccidiosis	20,075 ==>
[ 137 ]	17,378 ==>	Coccidiosis	16,941 ==>	Coccidiosis	18,166 ==>	Coccidiosis	18,166 ==>	Coccidiosis	19,468 ==>
[ 138 ]	17,692 ==>	Coccidiosis	17,263 ==>	Coccidiosis	18,466 ==>	Coccidiosis	18,466 ==>	Coccidiosis	19,748 ==>
[ 139 ]	18,055 ==>	Coccidiosis	17,635 ==>	Coccidiosis	18,815 ==>	Coccidiosis	18,815 ==>	Coccidiosis	20,075 ==>
[ 140 ]	16,643 ==>	Coccidiosis	16,186 ==>	Coccidiosis	17,464 ==>	Coccidiosis	17,464 ==>	Coccidiosis	18,815 ==>
[ 141 ]	18,055 ==>	Coccidiosis	17,635 ==>	Coccidiosis	18,815 ==>	Coccidiosis	18,815 ==>	Coccidiosis	20,075 ==>
[ 142 ]	17,378 ==>	Coccidiosis	16,941 ==>	Coccidiosis	18,166 ==>	Coccidiosis	18,166 ==>	Coccidiosis	19,468 ==>
[ 143 ]	18,055 ==>	Coccidiosis	17,635 ==>	Coccidiosis	18,815 ==>	Coccidiosis	18,815 ==>	Coccidiosis	20,075 ==>
[ 144 ]	14,967 ==>	Coccidiosis	14,457 ==>	Coccidiosis	15,875 ==>	Coccidiosis	15,875 ==>	Coccidiosis	17,349 ==>
[ 145 ]	17,378 ==>	Coccidiosis	16,941 ==>	Coccidiosis	18,166 ==>	Coccidiosis	18,166 ==>	Coccidiosis	19,468 ==>
[ 146 ]	18,055 ==>	Coccidiosis	17,635 ==>	Coccidiosis	18,815 ==>	Coccidiosis	18,815 ==>	Coccidiosis	20,075 ==>
[ 147 ]	17,378 ==>	Coccidiosis	16,941 ==>	Coccidiosis	18,166 ==>	Coccidiosis	18,166 ==>	Coccidiosis	19,468 ==>
[ 148 ]	18,055 ==>	Coccidiosis	17,635 ==>	Coccidiosis	18,815 ==>	Coccidiosis	18,815 ==>	Coccidiosis	20,075 ==>
[ 149 ]	18,055 ==>	Coccidiosis	17,635 ==>	Coccidiosis	18,815 ==>	Coccidiosis	18,815 ==>	Coccidiosis	20,075 ==>
[ 150 ]	16,643 ==>	Coccidiosis	16,186 ==>	Coccidiosis	17,464 ==>	Coccidiosis	17,464 ==>	Coccidiosis	18,815 ==>

Gambar 5.3 Implementasi Antarmuka Jarak Euclidian Sebelum Diurutkan

**DIAGNOSA PENYAKIT PADA ANJING**  
MENGUNAKAN METODE FUZZY-KNN (F-KNN)

DATA | PROSES FKNN | HASIL KLASIFIKASI DAN AKURASI

PARAMETER

NILAI K-

JUMLAH DATA UJI

JUMLAH DATA LATIH

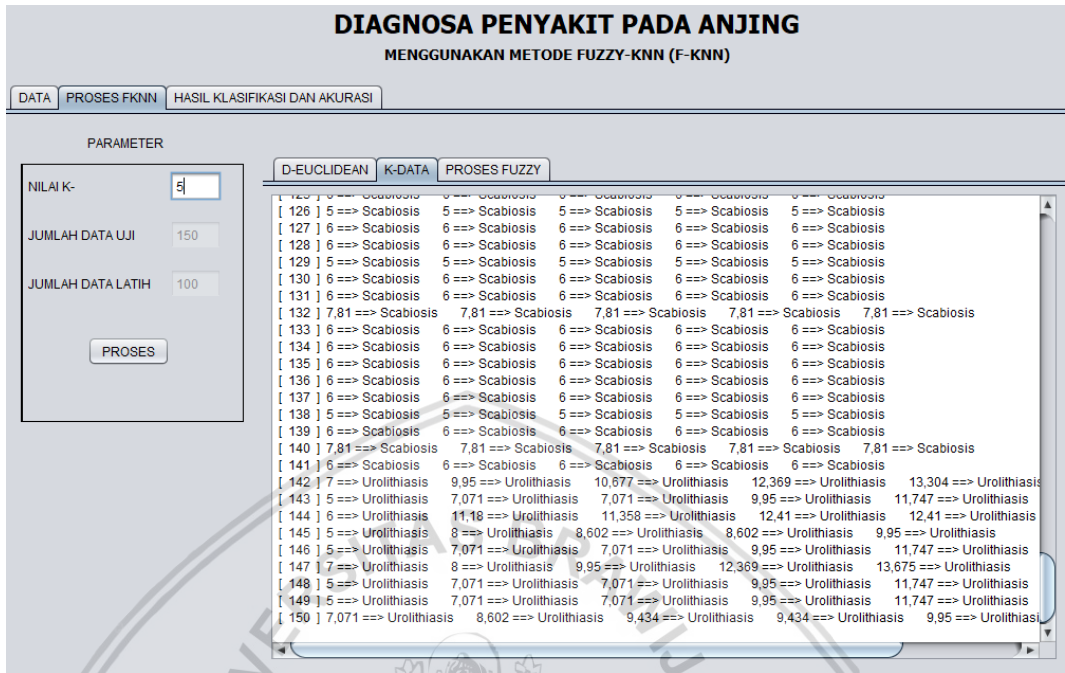
D-EUCLIDEAN | K-DATA | PROSES FUZZY

SEBELUM DIURUT | SETELAH DIURUT

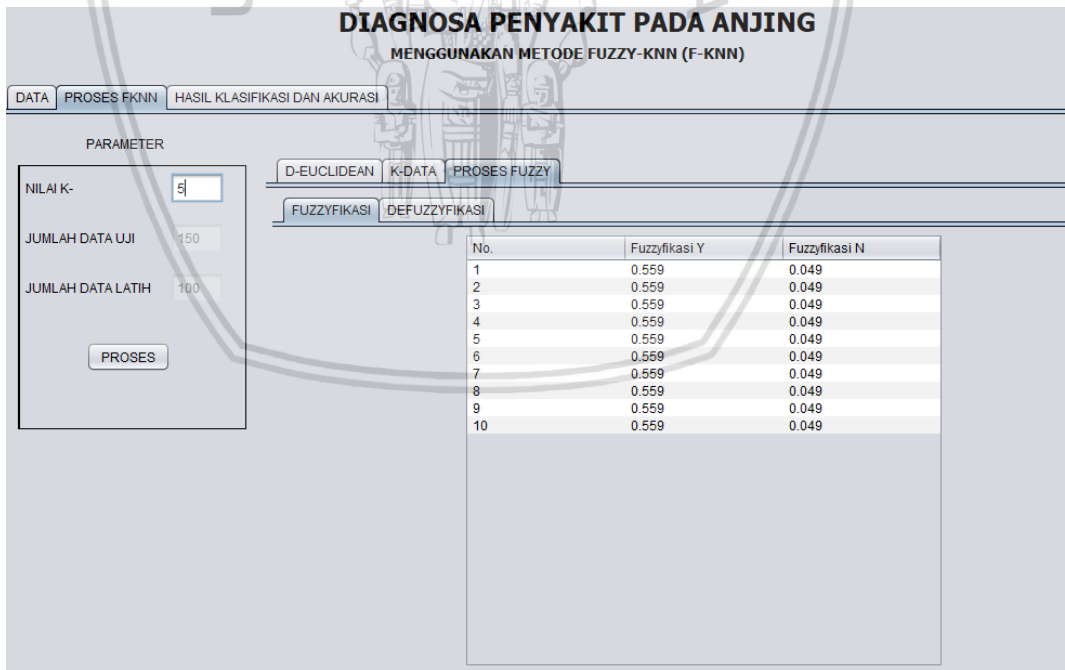
[ 128 ]	6 ==>	Scabiosis	6 ==>	Scabiosis	6 ==>	Scabiosis	6 ==>	Scabiosis	10,488 ==>
[ 129 ]	5 ==>	Scabiosis	5 ==>	Scabiosis	5 ==>	Scabiosis	5 ==>	Scabiosis	7 ==>
[ 130 ]	6 ==>	Scabiosis	6 ==>	Scabiosis	6 ==>	Scabiosis	6 ==>	Scabiosis	10,488 ==>
[ 131 ]	6 ==>	Scabiosis	6 ==>	Scabiosis	6 ==>	Scabiosis	6 ==>	Scabiosis	10,488 ==>
[ 132 ]	7,81 ==>	Scabiosis	7,81 ==>	Scabiosis	7,81 ==>	Scabiosis	7,81 ==>	Scabiosis	9,22 ==>
[ 133 ]	6 ==>	Scabiosis	6 ==>	Scabiosis	6 ==>	Scabiosis	6 ==>	Scabiosis	10,488 ==>
[ 134 ]	6 ==>	Scabiosis	6 ==>	Scabiosis	6 ==>	Scabiosis	6 ==>	Scabiosis	10,488 ==>
[ 135 ]	6 ==>	Scabiosis	6 ==>	Scabiosis	6 ==>	Scabiosis	6 ==>	Scabiosis	10,488 ==>
[ 136 ]	6 ==>	Scabiosis	6 ==>	Scabiosis	6 ==>	Scabiosis	6 ==>	Scabiosis	10,488 ==>
[ 137 ]	6 ==>	Scabiosis	6 ==>	Scabiosis	6 ==>	Scabiosis	6 ==>	Scabiosis	10,488 ==>
[ 138 ]	5 ==>	Scabiosis	5 ==>	Scabiosis	5 ==>	Scabiosis	5 ==>	Scabiosis	7 ==>
[ 139 ]	6 ==>	Scabiosis	6 ==>	Scabiosis	6 ==>	Scabiosis	6 ==>	Scabiosis	10,488 ==>
[ 140 ]	7,81 ==>	Scabiosis	7,81 ==>	Scabiosis	7,81 ==>	Scabiosis	7,81 ==>	Scabiosis	9,22 ==>
[ 141 ]	6 ==>	Scabiosis	6 ==>	Scabiosis	6 ==>	Scabiosis	6 ==>	Scabiosis	10,488 ==>
[ 142 ]	7 ==>	Urolithiasis	9,95 ==>	Urolithiasis	10,677 ==>	Urolithiasis	12,369 ==>	Urolithiasis	13,304 ==>
[ 143 ]	5 ==>	Urolithiasis	7,071 ==>	Urolithiasis	7,071 ==>	Urolithiasis	9,95 ==>	Urolithiasis	11,747 ==>
[ 144 ]	6 ==>	Urolithiasis	11,18 ==>	Urolithiasis	11,358 ==>	Urolithiasis	12,41 ==>	Urolithiasis	12,41 ==>
[ 145 ]	5 ==>	Urolithiasis	8 ==>	Urolithiasis	8,602 ==>	Urolithiasis	8,602 ==>	Urolithiasis	9,95 ==>
[ 146 ]	5 ==>	Urolithiasis	7,071 ==>	Urolithiasis	7,071 ==>	Urolithiasis	9,95 ==>	Urolithiasis	11,747 ==>
[ 147 ]	7 ==>	Urolithiasis	8 ==>	Urolithiasis	9,95 ==>	Urolithiasis	12,369 ==>	Urolithiasis	13,675 ==>
[ 148 ]	5 ==>	Urolithiasis	7,071 ==>	Urolithiasis	7,071 ==>	Urolithiasis	9,95 ==>	Urolithiasis	11,747 ==>
[ 149 ]	5 ==>	Urolithiasis	7,071 ==>	Urolithiasis	7,071 ==>	Urolithiasis	9,95 ==>	Urolithiasis	11,747 ==>
[ 150 ]	7,071 ==>	Urolithiasis	8,602 ==>	Urolithiasis	9,434 ==>	Urolithiasis	9,434 ==>	Urolithiasis	9,95 ==>

Gambar 5.4 Implementasi Antarmuka Jarak Euclidian Setelah Diurutkan





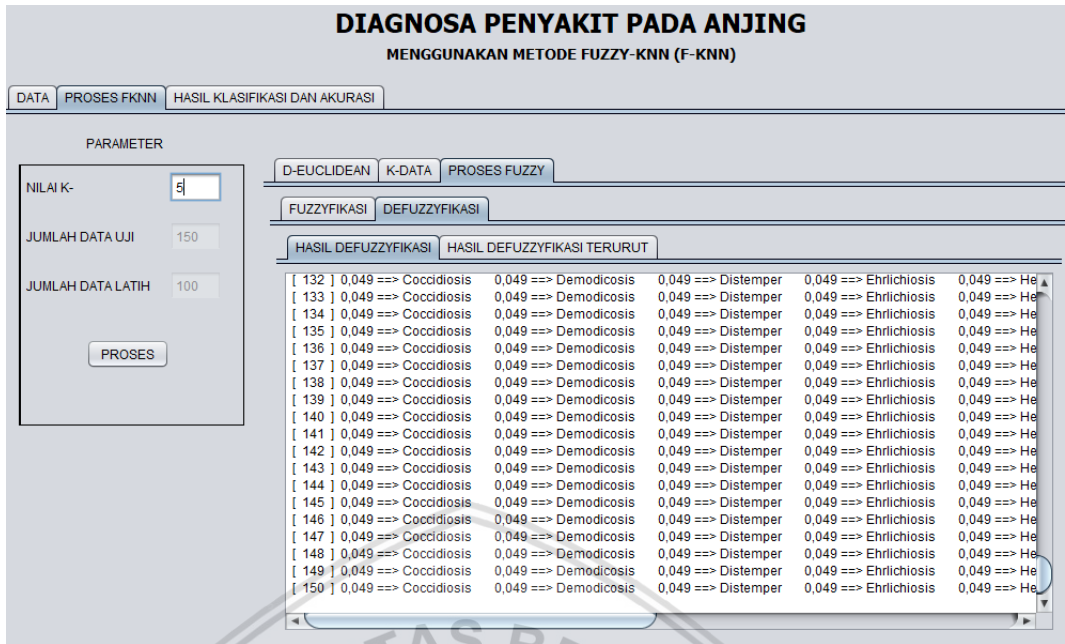
Gambar 5.5 Implementasi Antarmuka Jarak Euclidian Sebanyak K



Gambar 5.6 Implementasi Antarmuka Fuzzifikasi



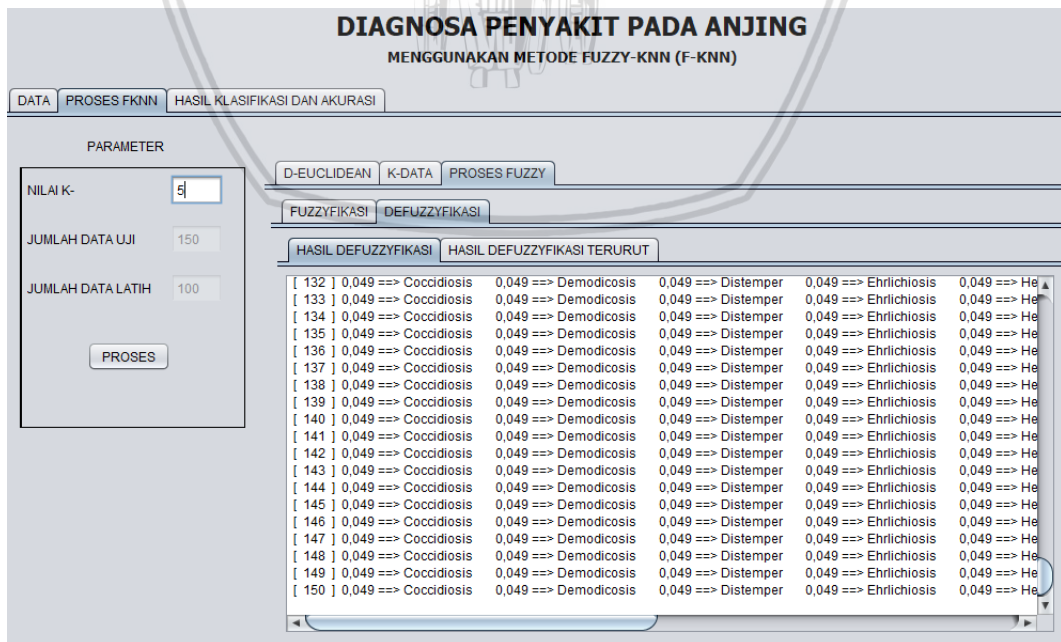




Gambar 5.5 Implementasi Antarmuka Defuzzifikasi

### 5.4.3 Implementasi Hasil Klasifikasi dan Akurasi

Pada implementasi hasil klasifikasi dan data uji ini akan ditampilkan nilai akhir proses FKNN, hasil klasifikasi dan tingkat akurasi sistem berdasarkan data uji yang dimasukkan pengguna. Implementasi antarmuka hasil klasifikasi dan akurasi ditunjukkan oleh gambar 5.8.



Gambar 5.6 Implementasi Antarmuka Hasil Klasifikasi dan Implentasi





## BAB 6 PENGUJIAN DAN ANALISIS

Bagian ini menjelaskan tentang pengujian dan analisis dari implementasi FK-NN untuk diagnosis penyakit pada anjing. Adapun pengujian yang dilakukan adalah pengujian akurasi sistem.

### 6.1 Pengujian Akurasi Sistem

Pengujian akurasi sistem dilakukan dengan memasukan nilai k yang bervariasi dengan nilai 5,10,15 dan 20. Jumlah data uji yang digunakan sebanyak 150 data. Data uji ditunjukkan oleh table 6.1.

Tabel 6.1 Data Uji 1

PENYAKIT	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G46
Coccidiosis	0	0	0	0	0	0	0	0	0	-	-	-	0
Coccidiosis	0	0	0	0	0	0	0	0	0	-	-	-	0
Coccidiosis	0	0	0	0	0	0	0	0	0	-	-	-	0
Coccidiosis	0	0	0	0	0	0	0	0	0	-	-	-	0
Coccidiosis	0	0	0	0	0	0	0	0	0	-	-	-	0
Coccidiosis	0	0	0	0	0	0	0	0	0	-	-	-	0
Coccidiosis	0	0	0	0	0	0	0	0	0	-	-	-	0
Coccidiosis	0	0	0	0	0	0	0	0	0	-	-	-	0
Coccidiosis	0	0	0	0	0	0	0	0	0	-	-	-	0
Coccidiosis	0	0	0	0	0	0	0	0	0	-	-	-	0
Coccidiosis	0	0	0	0	0	0	0	0	0	-	-	-	0
Coccidiosis	0	0	0	0	0	0	0	0	0	-	-	-	0
Coccidiosis	0	0	0	0	0	0	0	0	0	-	-	-	0
Coccidiosis	0	0	0	0	0	0	0	0	0	-	-	-	0
Coccidiosis	0	0	0	0	0	0	0	0	0	-	-	-	0
Coccidiosis	0	0	0	0	0	0	0	0	0	-	-	-	0
Coccidiosis	0	0	0	0	0	0	0	0	0	-	-	-	0
Coccidiosis	0	0	0	0	0	0	0	0	0	-	-	-	0
Coccidiosis	0	0	0	0	0	0	0	0	0	-	-	-	0
Coccidiosis	0	0	0	0	0	0	0	0	0	-	-	-	0
Demodicosis	8	0	6	0	8	0	0	0	0	-	-	-	0
Demodicosis	8	0	6	0	8	0	0	0	0	-	-	-	0
Demodicosis	8	0	6	0	0	0	0	0	0	-	-	-	0
Demodicosis	0	0	6	8	8	0	0	0	0	-	-	-	0
Demodicosis	8	0	6	0	0	0	0	0	0	-	-	-	0
Demodicosis	8	0	0	8	8	0	0	0	0	-	-	-	0
Demodicosis	8	7	6	0	0	0	0	0	0	-	-	-	0
Demodicosis	8	0	0	8	0	0	0	0	0	-	-	-	0
Demodicosis	0	0	6	8	8	0	0	0	0	-	-	-	0
Demodicosis	8	0	6	0	0	0	0	0	0	-	-	-	0

Tabel 6.1 (lanjutan)

Demodicosis	8	0	6	0	8	0	0	0	0	-	-	-	0
Demodicosis	8	7	6	0	0	0	0	0	0	-	-	-	0
Demodicosis	8	7	0	8	0	0	0	0	0	-	-	-	0
Demodicosis	8	0	0	8	8	0	0	0	0	-	-	-	0
Demodicosis	8	0	6	8	0	0	0	0	0	-	-	-	0
Demodicosis	8	0	0	8	0	5	0	0	0	-	-	-	0
Demodicosis	8	7	0	0	8	0	0	0	0	-	-	-	0
Demodicosis	8	0	0	8	0	5	0	0	0	-	-	-	0
Demodicosis	8	7	0	0	8	0	0	0	0	-	-	-	0
Demodicosis	8	0	0	8	0	5	0	0	0	-	-	-	0
Demodicosis	8	0	6	0	0	0	0	0	0	-	-	-	0
Demodicosis	8	7	6	0	0	0	0	0	0	-	-	-	0
Demodicosis	8	7	0	0	8	0	0	0	0	-	-	-	0
Demodicosis	8	0	0	8	8	0	0	0	0	-	-	-	0
Demodicosis	8	0	6	0	8	0	0	0	0	-	-	-	0
Demodicosis	8	0	6	8	0	0	0	0	0	-	-	-	0
Demodicosis	8	7	6	0	0	0	0	0	0	-	-	-	0
Demodicosis	8	0	6	0	8	0	0	0	0	-	-	-	0
Demodicosis	8	0	6	0	0	0	0	0	0	-	-	-	0
Demodicosis	8	0	6	0	0	0	0	0	0	-	-	-	0
Demodicosis	8	0	0	8	0	5	0	0	0	-	-	-	0
Demodicosis	8	7	0	0	8	0	0	0	0	-	-	-	0
Distemper	0	0	0	0	0	0	0	0	0	-	-	-	0
Distemper	0	0	0	0	0	0	0	0	0	-	-	-	0
Distemper	0	0	0	0	0	0	0	0	0	-	-	-	0
Distemper	0	0	0	0	0	0	0	0	0	-	-	-	0
Distemper	0	0	0	0	0	0	0	0	0	-	-	-	0
Distemper	0	0	0	0	0	0	0	0	0	-	-	-	0
Distemper	0	0	0	0	0	0	0	0	0	-	-	-	0
Ehrlichiosis	0	0	0	0	0	0	0	0	0	-	-	-	0
Ehrlichiosis	0	0	0	0	0	0	0	0	0	-	-	-	0
Ehrlichiosis	0	0	0	0	0	0	0	0	0	-	-	-	0
Ehrlichiosis	0	0	0	0	0	0	0	0	0	-	-	-	0
Ehrlichiosis	0	0	0	0	0	0	0	0	0	-	-	-	0
Ehrlichiosis	0	0	0	0	0	0	0	0	0	-	-	-	0
Ehrlichiosis	0	0	0	0	0	0	0	0	0	-	-	-	0
Ehrlichiosis	0	0	0	0	0	0	0	0	0	-	-	-	0
Ehrlichiosis	0	0	0	0	0	0	0	0	0	-	-	-	0
Ehrlichiosis	0	0	0	0	0	0	0	0	0	-	-	-	0
Ehrlichiosis	0	0	0	0	0	0	0	0	0	-	-	-	0
Ehrlichiosis	0	0	0	0	0	0	0	0	0	-	-	-	0
Ehrlichiosis	0	0	0	0	0	0	0	0	0	-	-	-	0



Tabel 6.1 (lanjutan)

Ehrlichiosis	0	0	0	0	0	0	0	0	0	-	-	-	0
Helminthiasis	0	0	0	0	0	0	0	0	0	-	-	-	0
Helminthiasis	0	0	0	0	0	0	0	0	0	-	-	-	0
Helminthiasis	0	0	0	0	0	0	0	0	0	-	-	-	0
Helminthiasis	0	0	0	0	0	0	0	0	0	-	-	-	0
Helminthiasis	0	0	0	0	0	0	0	0	0	-	-	-	0
Helminthiasis	0	0	0	0	0	0	0	0	0	-	-	-	0
Helminthiasis	0	0	0	0	0	0	0	0	0	-	-	-	0
Helminthiasis	0	0	0	0	0	0	0	0	0	-	-	-	0
Helminthiasis	0	0	0	0	0	0	0	0	0	-	-	-	0
Helminthiasis	0	0	0	0	0	0	0	0	0	-	-	-	0
Helminthiasis	0	0	0	0	0	0	0	0	0	-	-	-	0
Helminthiasis	0	0	0	0	0	0	0	0	0	-	-	-	0
Helminthiasis	0	0	0	0	0	0	0	0	0	-	-	-	0
Helminthiasis	0	0	0	0	0	0	0	0	0	-	-	-	0
Helminthiasis	0	0	0	0	0	0	0	0	0	-	-	-	0
Helminthiasis	0	0	0	0	0	0	0	0	0	-	-	-	0
Helminthiasis	0	0	0	0	0	0	0	0	0	-	-	-	0
Helminthiasis	0	0	0	0	0	0	0	0	0	-	-	-	0
Otitis	0	0	0	0	0	0	0	0	0	-	-	-	0
Otitis	0	0	0	0	0	0	0	0	0	-	-	-	0
Otitis	0	0	0	0	0	0	0	0	0	-	-	-	0
Otitis	0	0	0	0	0	0	0	0	0	-	-	-	0
Otitis	0	0	0	0	0	0	0	0	0	-	-	-	0
Otitis	0	0	0	0	0	0	0	0	0	-	-	-	0
Otitis	0	0	0	0	0	0	0	0	0	-	-	-	0
Otitis	0	0	0	0	0	0	0	0	0	-	-	-	0
Otitis	0	0	0	0	0	0	0	0	0	-	-	-	0
Otitis	0	0	0	0	0	0	0	0	0	-	-	-	0
Otitis	0	0	0	0	0	0	0	0	0	-	-	-	0
Otitis	0	0	0	0	0	0	0	0	0	-	-	-	0
Parvo	0	0	0	0	0	0	0	0	0	-	-	-	0
Parvo	0	0	0	0	0	0	0	0	0	-	-	-	0
Parvo	0	0	0	0	0	0	0	0	0	-	-	-	0
Parvo	0	0	0	0	0	0	0	0	0	-	-	-	0
Parvo	0	0	0	0	0	0	0	0	0	-	-	-	0
Parvo	0	0	0	0	0	0	0	0	0	-	-	-	0
Parvo	0	0	0	0	0	0	0	0	0	-	-	-	0
Parvo	0	0	0	0	0	0	0	0	0	-	-	-	0
Parvo	0	0	0	0	0	0	0	0	0	-	-	-	0
Parvo	0	0	0	0	0	0	0	0	0	-	-	-	0



Tabel 6.1 (lanjutan)

Parvo	0	0	0	0	0	0	0	0	0	-	-	-	0
Parvo	0	0	0	0	0	0	0	0	0	-	-	-	0
Parvo	0	0	0	0	0	0	0	0	0	-	-	-	0
Parvo	0	0	0	0	0	0	0	0	0	-	-	-	0
Parvo	0	0	0	0	0	0	0	0	0	-	-	-	0
Ringworm	0	0	0	0	0	0	0	0	0	-	-	-	0
Ringworm	0	0	0	0	0	0	0	0	0	-	-	-	0
Ringworm	0	0	0	0	0	0	0	0	0	-	-	-	0
Ringworm	0	0	0	0	0	0	0	0	0	-	-	-	0
Ringworm	0	0	0	0	0	0	0	0	0	-	-	-	0
Scabiosis	0	0	0	8	0	0	0	6	8	-	-	-	0
Scabiosis	0	0	0	8	0	0	0	0	8	-	-	-	0
Scabiosis	0	0	0	8	0	0	0	0	8	-	-	-	0
Scabiosis	0	0	0	8	0	0	0	0	8	-	-	-	0
Scabiosis	0	0	0	8	0	0	0	0	8	-	-	-	0
Scabiosis	0	0	0	8	0	0	0	0	8	-	-	-	0
Scabiosis	0	0	0	8	0	0	0	0	8	-	-	-	0
Scabiosis	0	0	0	8	0	0	7	0	8	-	-	-	0
Scabiosis	0	0	0	8	0	0	0	6	8	-	-	-	0
Scabiosis	0	0	0	8	0	0	0	0	8	-	-	-	0
Scabiosis	0	0	0	8	0	0	7	0	8	-	-	-	0
Scabiosis	0	0	0	8	0	0	0	6	8	-	-	-	0
Scabiosis	0	0	0	8	0	0	0	0	8	-	-	-	0
Scabiosis	0	0	0	8	0	0	0	0	8	-	-	-	0
Scabiosis	0	0	0	8	0	0	7	0	8	-	-	-	0
Scabiosis	0	0	0	8	0	0	7	0	8	-	-	-	0
Scabiosis	0	0	0	8	0	0	0	0	8	-	-	-	0
Scabiosis	0	0	0	8	0	0	7	0	8	-	-	-	0
Scabiosis	0	0	0	8	0	0	0	0	8	-	-	-	0
Scabiosis	0	0	0	8	0	0	7	0	8	-	-	-	0
Scabiosis	0	0	0	8	0	0	0	6	8	-	-	-	0
Scabiosis	0	0	0	8	0	0	7	0	8	-	-	-	0
Scabiosis	0	0	0	8	0	0	0	0	8	-	-	-	0
Scabiosis	0	0	0	8	0	0	7	0	8	-	-	-	0
Urolithiasis	0	0	0	0	0	0	0	0	0	-	-	-	0
Urolithiasis	0	0	0	0	0	0	0	0	0	-	-	-	7
Urolithiasis	0	0	0	0	0	0	0	0	0	-	-	-	7
Urolithiasis	0	0	0	0	0	0	0	0	0	-	-	-	0
Urolithiasis	0	0	0	0	0	0	0	0	0	-	-	-	7
Urolithiasis	0	0	0	0	0	0	0	0	0	-	-	-	0



**Tabel 6.1 (lanjutan)**

Urolithiasis	0	0	0	0	0	0	0	0	0	0	-	-	-	7
Urolithiasis	0	0	0	0	0	0	0	0	0	0	-	-	-	7
Urolithiasis	0	0	0	0	0	0	0	0	0	0	-	-	-	0

**6.1.1 Pengujian dengan nilai K = 5**

Pengujian yang pertama adalah pengujian dengan nilai K = 5, data uji yang digunakan sebanyak 150 data uji. Hasil pengujian dengan nilai K = 5 ditunjukkan oleh tabel 6.2.

**Tabel 6.2 Hasil Uji K = 5**

No	Hasil Sistem	Kelas Asli
1	Coccidiosis	Coccidiosis
2	Coccidiosis	Coccidiosis
3	Coccidiosis	Coccidiosis
4	Coccidiosis	Coccidiosis
5	Coccidiosis	Coccidiosis
6	Coccidiosis	Coccidiosis
7	Coccidiosis	Coccidiosis
8	Coccidiosis	Coccidiosis
9	Coccidiosis	Coccidiosis
10	Coccidiosis	Coccidiosis
11	Coccidiosis	Coccidiosis
12	Coccidiosis	Coccidiosis
13	Coccidiosis	Coccidiosis
14	Coccidiosis	Coccidiosis
15	Coccidiosis	Coccidiosis
16	Coccidiosis	Coccidiosis
17	Coccidiosis	Coccidiosis
18	Demodicosis	Demodicosis
19	Demodicosis	Demodicosis
20	Demodicosis	Demodicosis
21	Demodicosis	Demodicosis
22	Demodicosis	Demodicosis
23	Demodicosis	Demodicosis
24	Demodicosis	Demodicosis
25	Demodicosis	Demodicosis
26	Demodicosis	Demodicosis
27	Demodicosis	Demodicosis
28	Demodicosis	Demodicosis
29	Demodicosis	Demodicosis



Tabel 6.2 (lanjutan)

30	Demodicosis	Demodicosis
31	Demodicosis	Demodicosis
32	Demodicosis	Demodicosis
33	Demodicosis	Demodicosis
34	Demodicosis	Demodicosis
35	Demodicosis	Demodicosis
36	Demodicosis	Demodicosis
37	Demodicosis	Demodicosis
38	Demodicosis	Demodicosis
39	Demodicosis	Demodicosis
40	Demodicosis	Demodicosis
41	Demodicosis	Demodicosis
42	Demodicosis	Demodicosis
43	Demodicosis	Demodicosis
44	Demodicosis	Demodicosis
45	Demodicosis	Demodicosis
46	Demodicosis	Demodicosis
47	Demodicosis	Demodicosis
48	Demodicosis	Demodicosis
49	Demodicosis	Demodicosis
50	Distemper	Distemper
51	Distemper	Distemper
52	Distemper	Distemper
53	Distemper	Distemper
54	Distemper	Distemper
55	Distemper	Distemper
56	Ringworm	Ehrlichiosis
57	Ehrlichiosis	Ehrlichiosis
58	Ehrlichiosis	Ehrlichiosis
59	Ehrlichiosis	Ehrlichiosis
60	Ehrlichiosis	Ehrlichiosis
61	Ehrlichiosis	Ehrlichiosis
62	Ehrlichiosis	Ehrlichiosis
63	Ehrlichiosis	Ehrlichiosis
64	Ehrlichiosis	Ehrlichiosis
65	Ehrlichiosis	Ehrlichiosis
66	Ehrlichiosis	Ehrlichiosis
67	Ehrlichiosis	Ehrlichiosis
68	Ehrlichiosis	Ehrlichiosis
69	Ringworm	Helminthiasis

Tabel 6.2 (lanjutan)

70	Helminthiasis	Helminthiasis
71	Helminthiasis	Helminthiasis
72	Helminthiasis	Helminthiasis
73	Helminthiasis	Helminthiasis
74	Helminthiasis	Helminthiasis
75	Helminthiasis	Helminthiasis
76	Helminthiasis	Helminthiasis
77	Helminthiasis	Helminthiasis
78	Helminthiasis	Helminthiasis
79	Helminthiasis	Helminthiasis
80	Helminthiasis	Helminthiasis
81	Helminthiasis	Helminthiasis
82	Helminthiasis	Helminthiasis
83	Helminthiasis	Helminthiasis
84	Helminthiasis	Helminthiasis
85	Helminthiasis	Helminthiasis
86	Helminthiasis	Helminthiasis
87	Helminthiasis	Helminthiasis
88	Otitis	Otitis
89	Otitis	Otitis
90	Otitis	Otitis
91	Otitis	Otitis
92	Otitis	Otitis
93	Otitis	Otitis
94	Otitis	Otitis
95	Otitis	Otitis
96	Otitis	Otitis
97	Otitis	Otitis
98	Otitis	Otitis
99	Parvo	Parvo
100	Parvo	Parvo
101	Parvo	Parvo
102	Parvo	Parvo
103	Parvo	Parvo
104	Parvo	Parvo
105	Parvo	Parvo
106	Parvo	Parvo
107	Parvo	Parvo
108	Parvo	Parvo
109	Parvo	Parvo



Tabel 6.2 (lanjutan)

110	Parvo	Parvo
111	Parvo	Parvo
112	Parvo	Parvo
113	Ringworm	Ringworm
114	Ringworm	Ringworm
115	Ringworm	Ringworm
116	Ringworm	Ringworm
117	Ringworm	Ringworm
118	Scabiosis	Scabiosis
119	Scabiosis	Scabiosis
120	Scabiosis	Scabiosis
121	Scabiosis	Scabiosis
122	Scabiosis	Scabiosis
123	Scabiosis	Scabiosis
124	Scabiosis	Scabiosis
125	Scabiosis	Scabiosis
126	Scabiosis	Scabiosis
127	Scabiosis	Scabiosis
128	Scabiosis	Scabiosis
129	Scabiosis	Scabiosis
130	Scabiosis	Scabiosis
131	Scabiosis	Scabiosis
132	Scabiosis	Scabiosis
133	Scabiosis	Scabiosis
134	Scabiosis	Scabiosis
135	Scabiosis	Scabiosis
136	Scabiosis	Scabiosis
137	Scabiosis	Scabiosis
138	Scabiosis	Scabiosis
139	Scabiosis	Scabiosis
140	Scabiosis	Scabiosis
141	Scabiosis	Scabiosis
142	Urolithiasis	Urolithiasis
143	Urolithiasis	Urolithiasis
144	Urolithiasis	Urolithiasis
145	Urolithiasis	Urolithiasis
146	Urolithiasis	Urolithiasis
147	Urolithiasis	Urolithiasis
148	Urolithiasis	Urolithiasis
149	Urolithiasis	Urolithiasis
150	Urolithiasis	Urolithiasis

Dari hasil pengujian diatas didapatkan sebanyak 2 data uji yang memiliki hasil klasifikasi berbeda dengan kelas sebenarnya, dari hasil tersebut kemudian akan dihitung akurasiya menggunakan persamaan (2-8) sehingga dihasilkan akurasi sebagai berikut :

$$\begin{aligned} \text{Akurasi} &= \frac{148}{150} \times 100\% \\ &= 98,67\% \end{aligned}$$

### 6.1.2 Pengujian dengan nilai K = 10

Pengujian yang pertama adalah pengujian dengan nilai K = 10, data uji yang digunakan sebanyak 150 data uji. Hasil pengujian dengan nilai K = 10 ditunjukkan oleh tabel 6.3.

**Tabel 6.3 Hasil Uji K = 10**

No	Hasil Sistem	Kelas Asli
1	Coccidiosis	Coccidiosis
2	Coccidiosis	Coccidiosis
3	Coccidiosis	Coccidiosis
4	Coccidiosis	Coccidiosis
5	Parvo	Coccidiosis
6	Coccidiosis	Coccidiosis
7	Coccidiosis	Coccidiosis
8	Coccidiosis	Coccidiosis
9	Coccidiosis	Coccidiosis
10	Parvo	Coccidiosis
11	Coccidiosis	Coccidiosis
12	Coccidiosis	Coccidiosis
13	Parvo	Coccidiosis
14	Coccidiosis	Coccidiosis
15	Coccidiosis	Coccidiosis
16	Coccidiosis	Coccidiosis
17	Coccidiosis	Coccidiosis
18	Demodiosis	Demodiosis
19	Demodiosis	Demodiosis
20	Demodiosis	Demodiosis
21	Demodiosis	Demodiosis
22	Demodiosis	Demodiosis
23	Demodiosis	Demodiosis
24	Demodiosis	Demodiosis
25	Demodiosis	Demodiosis
26	Demodiosis	Demodiosis

Tabel 6.3 (lanjutan)

27	Demodicosis	Demodicosis
28	Demodicosis	Demodicosis
29	Demodicosis	Demodicosis
30	Demodicosis	Demodicosis
31	Demodicosis	Demodicosis
32	Demodicosis	Demodicosis
33	Demodicosis	Demodicosis
34	Demodicosis	Demodicosis
35	Demodicosis	Demodicosis
36	Demodicosis	Demodicosis
37	Demodicosis	Demodicosis
38	Demodicosis	Demodicosis
39	Demodicosis	Demodicosis
40	Demodicosis	Demodicosis
41	Demodicosis	Demodicosis
42	Demodicosis	Demodicosis
43	Demodicosis	Demodicosis
44	Demodicosis	Demodicosis
45	Demodicosis	Demodicosis
46	Demodicosis	Demodicosis
47	Demodicosis	Demodicosis
48	Demodicosis	Demodicosis
49	Demodicosis	Demodicosis
50	Distemper	Distemper
51	Distemper	Distemper
52	Distemper	Distemper
53	Distemper	Distemper
54	Distemper	Distemper
55	Distemper	Distemper
56	Ringworm	Ehrlichiosis
57	Ehrlichiosis	Ehrlichiosis
58	Ehrlichiosis	Ehrlichiosis
59	Ehrlichiosis	Ehrlichiosis
60	Ehrlichiosis	Ehrlichiosis
61	Ehrlichiosis	Ehrlichiosis
62	Ehrlichiosis	Ehrlichiosis
63	Ehrlichiosis	Ehrlichiosis
64	Ehrlichiosis	Ehrlichiosis
65	Ehrlichiosis	Ehrlichiosis
66	Ehrlichiosis	Ehrlichiosis

Tabel 6.3 (lanjutan)

67	Ehrlichiosis	Ehrlichiosis
68	Ehrlichiosis	Ehrlichiosis
69	Ringworm	Helminthiasis
70	Helminthiasis	Helminthiasis
71	Helminthiasis	Helminthiasis
72	Coccidiosis	Helminthiasis
73	Coccidiosis	Helminthiasis
74	Helminthiasis	Helminthiasis
75	Parvo	Helminthiasis
76	Helminthiasis	Helminthiasis
77	Helminthiasis	Helminthiasis
78	Helminthiasis	Helminthiasis
79	Helminthiasis	Helminthiasis
80	Helminthiasis	Helminthiasis
81	Helminthiasis	Helminthiasis
82	Helminthiasis	Helminthiasis
83	Helminthiasis	Helminthiasis
84	Helminthiasis	Helminthiasis
85	Helminthiasis	Helminthiasis
86	Helminthiasis	Helminthiasis
87	Coccidiosis	Helminthiasis
88	Ringworm	Otitis
89	Otitis	Otitis
90	Otitis	Otitis
91	Otitis	Otitis
92	Otitis	Otitis
93	Otitis	Otitis
94	Otitis	Otitis
95	Otitis	Otitis
96	Otitis	Otitis
97	Otitis	Otitis
98	Otitis	Otitis
99	Parvo	Parvo
100	Parvo	Parvo
101	Parvo	Parvo
102	Parvo	Parvo
103	Parvo	Parvo
104	Parvo	Parvo
105	Parvo	Parvo
106	Parvo	Parvo

Tabel 6.3 (lanjutan)

107	Parvo	Parvo
108	Parvo	Parvo
109	Parvo	Parvo
110	Parvo	Parvo
111	Parvo	Parvo
112	Parvo	Parvo
113	Ringworm	Ringworm
114	Ringworm	Ringworm
115	Ringworm	Ringworm
116	Ringworm	Ringworm
117	Ringworm	Ringworm
118	Scabiosis	Scabiosis
119	Scabiosis	Scabiosis
120	Scabiosis	Scabiosis
121	Scabiosis	Scabiosis
122	Scabiosis	Scabiosis
123	Scabiosis	Scabiosis
124	Scabiosis	Scabiosis
125	Scabiosis	Scabiosis
126	Scabiosis	Scabiosis
127	Scabiosis	Scabiosis
128	Scabiosis	Scabiosis
129	Scabiosis	Scabiosis
130	Scabiosis	Scabiosis
131	Scabiosis	Scabiosis
132	Scabiosis	Scabiosis
133	Scabiosis	Scabiosis
134	Scabiosis	Scabiosis
135	Scabiosis	Scabiosis
136	Scabiosis	Scabiosis
137	Scabiosis	Scabiosis
138	Scabiosis	Scabiosis
139	Scabiosis	Scabiosis
140	Scabiosis	Scabiosis
141	Scabiosis	Scabiosis
142	Urolithiasis	Urolithiasis
143	Urolithiasis	Urolithiasis
144	Urolithiasis	Urolithiasis
145	Urolithiasis	Urolithiasis
146	Urolithiasis	Urolithiasis

**Tabel 6.3 (lanjutan)**

147	Urolithiasis	Urolithiasis
148	Urolithiasis	Urolithiasis
149	Urolithiasis	Urolithiasis
150	Urolithiasis	Urolithiasis

Dari hasil pengujian diatas didapatkan sebanyak 10 data uji yang memiliki hasil klasifikasi berbeda dengan kelas sebenarnya, dari hasil tersebut kemudian akan dihitung akurasiya menggunakan persamaan (2-8) sehingga dihasilkan akurasi sebagai berikut :

$$Akurasi = \frac{140}{150} \times 100\%$$

$$= 93,33\%$$

### 6.1.3 Pengujian dengan nilai K = 15

Pengujian yang pertama adalah pengujian dengan nilai K = 15, data uji yang digunakan sebanyak 150 data uji. Hasil pengujian dengan nilai K = 15 ditunjukkan oleh tabel 6.4.

**Tabel 6.4 Data Uji K = 15**

No	Hasil Sistem	Kelas Asli
1	Coccidiosis	Coccidiosis
2	Coccidiosis	Coccidiosis
3	Coccidiosis	Coccidiosis
4	Parvo	Coccidiosis
5	Parvo	Coccidiosis
6	Coccidiosis	Coccidiosis
7	Coccidiosis	Coccidiosis
8	Coccidiosis	Coccidiosis
9	Coccidiosis	Coccidiosis
10	Parvo	Coccidiosis
11	Coccidiosis	Coccidiosis
12	Coccidiosis	Coccidiosis
13	Parvo	Coccidiosis
14	Coccidiosis	Coccidiosis
15	Coccidiosis	Coccidiosis
16	Coccidiosis	Coccidiosis
17	Parvo	Coccidiosis
18	Demodiosis	Demodiosis
19	Demodiosis	Demodiosis

Tabel 6.4 (lanjutan)

20	Demodicosis	Demodicosis
21	Demodicosis	Demodicosis
22	Demodicosis	Demodicosis
23	Demodicosis	Demodicosis
24	Demodicosis	Demodicosis
25	Demodicosis	Demodicosis
26	Demodicosis	Demodicosis
27	Demodicosis	Demodicosis
28	Demodicosis	Demodicosis
29	Demodicosis	Demodicosis
30	Demodicosis	Demodicosis
31	Demodicosis	Demodicosis
32	Demodicosis	Demodicosis
33	Demodicosis	Demodicosis
34	Demodicosis	Demodicosis
35	Demodicosis	Demodicosis
36	Demodicosis	Demodicosis
37	Demodicosis	Demodicosis
38	Demodicosis	Demodicosis
39	Demodicosis	Demodicosis
40	Demodicosis	Demodicosis
41	Demodicosis	Demodicosis
42	Demodicosis	Demodicosis
43	Demodicosis	Demodicosis
44	Demodicosis	Demodicosis
45	Demodicosis	Demodicosis
46	Demodicosis	Demodicosis
47	Demodicosis	Demodicosis
48	Demodicosis	Demodicosis
49	Demodicosis	Demodicosis
50	Distemper	Distemper
51	Distemper	Distemper
52	Distemper	Distemper
53	Distemper	Distemper
54	Distemper	Distemper
55	Distemper	Distemper
56	Ringworm	Ehrlichiosis
57	Ehrlichiosis	Ehrlichiosis
58	Ehrlichiosis	Ehrlichiosis
59	Ehrlichiosis	Ehrlichiosis
60	Ehrlichiosis	Ehrlichiosis



**Tabel 6.4 (lanjutan)**

61	Ehrlichiosis	Ehrlichiosis
62	Ehrlichiosis	Ehrlichiosis
63	Ehrlichiosis	Ehrlichiosis
64	Ehrlichiosis	Ehrlichiosis
65	Ehrlichiosis	Ehrlichiosis
66	Ehrlichiosis	Ehrlichiosis
67	Ehrlichiosis	Ehrlichiosis
68	Ehrlichiosis	Ehrlichiosis
69	Ringworm	Helminthiasis
70	Helminthiasis	Helminthiasis
71	Helminthiasis	Helminthiasis
72	Coccidiosis	Helminthiasis
73	Parvo	Helminthiasis
74	Helminthiasis	Helminthiasis
75	Parvo	Helminthiasis
76	Helminthiasis	Helminthiasis
77	Helminthiasis	Helminthiasis
78	Helminthiasis	Helminthiasis
79	Ringworm	Helminthiasis
80	Helminthiasis	Helminthiasis
81	Helminthiasis	Helminthiasis
82	Helminthiasis	Helminthiasis
83	Helminthiasis	Helminthiasis
84	Ringworm	Helminthiasis
85	Coccidiosis	Helminthiasis
86	Helminthiasis	Helminthiasis
87	Parvo	Helminthiasis
88	Ringworm	Otitis
89	Otitis	Otitis
90	Otitis	Otitis
91	Otitis	Otitis
92	Otitis	Otitis
93	Otitis	Otitis
94	Otitis	Otitis
95	Otitis	Otitis
96	Otitis	Otitis
97	Otitis	Otitis
98	Otitis	Otitis
99	Parvo	Parvo



Tabel 6.4 (lanjutan)

100	Parvo	Parvo
101	Parvo	Parvo
102	Parvo	Parvo
103	Parvo	Parvo
104	Parvo	Parvo
105	Parvo	Parvo
106	Parvo	Parvo
107	Parvo	Parvo
108	Parvo	Parvo
109	Parvo	Parvo
110	Parvo	Parvo
111	Parvo	Parvo
112	Parvo	Parvo
113	Urolithiasis	Ringworm
114	Urolithiasis	Ringworm
115	Urolithiasis	Ringworm
116	Urolithiasis	Ringworm
117	Urolithiasis	Ringworm
118	Scabiosis	Scabiosis
119	Scabiosis	Scabiosis
120	Scabiosis	Scabiosis
121	Scabiosis	Scabiosis
122	Scabiosis	Scabiosis
123	Scabiosis	Scabiosis
124	Scabiosis	Scabiosis
125	Scabiosis	Scabiosis
126	Scabiosis	Scabiosis
127	Scabiosis	Scabiosis
128	Scabiosis	Scabiosis
129	Scabiosis	Scabiosis
130	Scabiosis	Scabiosis
131	Scabiosis	Scabiosis
132	Scabiosis	Scabiosis
133	Scabiosis	Scabiosis
134	Scabiosis	Scabiosis
135	Scabiosis	Scabiosis
136	Scabiosis	Scabiosis
137	Scabiosis	Scabiosis
138	Scabiosis	Scabiosis
139	Scabiosis	Scabiosis
140	Scabiosis	Scabiosis

**Tabel 6.4 (lanjutan)**

141	Scabiosis	Scabiosis
142	Urolithiasis	Urolithiasis
143	Urolithiasis	Urolithiasis
144	Urolithiasis	Urolithiasis
145	Urolithiasis	Urolithiasis
146	Urolithiasis	Urolithiasis
147	Urolithiasis	Urolithiasis
148	Urolithiasis	Urolithiasis
149	Urolithiasis	Urolithiasis
150	Urolithiasis	Urolithiasis

Dari hasil pengujian diatas didapatkan sebanyak 20 data uji yang memiliki hasil klasifikasi berbeda dengan kelas sebenarnya, dari hasil tersebut kemudian akan dihitung akurasi menggunakan persamaan (2-8) sehingga dihasilkan akurasi sebagai berikut :

$$\begin{aligned} \text{Akurasi} &= \frac{130}{150} \times 100\% \\ &= 86,67\% \end{aligned}$$

#### 6.1.4 Pengujian dengan nilai K = 20

Pengujian yang pertama adalah pengujian dengan nilai K = 20, data uji yang digunakan sebanyak 150 data uji. Hasil pengujian dengan nilai K = 20 ditunjukkan oleh tabel 6.5.

**Tabel 6.5 Hasil Uji K = 20**

No	Hasil Sistem	Kelas Asli
1	Coccidiosis	Coccidiosis
2	Coccidiosis	Coccidiosis
3	Coccidiosis	Coccidiosis
4	Parvo	Coccidiosis
5	Parvo	Coccidiosis
6	Coccidiosis	Coccidiosis
7	Coccidiosis	Coccidiosis
8	Coccidiosis	Coccidiosis
9	Coccidiosis	Coccidiosis
10	Parvo	Coccidiosis
11	Coccidiosis	Coccidiosis
12	Coccidiosis	Coccidiosis

Tabel 6.5 (lanjutan)

13	Parvo	Coccidiosis
14	Coccidiosis	Coccidiosis
15	Coccidiosis	Coccidiosis
16	Coccidiosis	Coccidiosis
17	Parvo	Coccidiosis
18	Demodiosis	Demodiosis
19	Demodiosis	Demodiosis
20	Demodiosis	Demodiosis
21	Demodiosis	Demodiosis
22	Demodiosis	Demodiosis
23	Demodiosis	Demodiosis
24	Demodiosis	Demodiosis
25	Demodiosis	Demodiosis
26	Demodiosis	Demodiosis
27	Demodiosis	Demodiosis
28	Demodiosis	Demodiosis
29	Demodiosis	Demodiosis
30	Demodiosis	Demodiosis
31	Demodiosis	Demodiosis
32	Demodiosis	Demodiosis
33	Demodiosis	Demodiosis
34	Demodiosis	Demodiosis
35	Demodiosis	Demodiosis
36	Demodiosis	Demodiosis
37	Demodiosis	Demodiosis
38	Demodiosis	Demodiosis
39	Demodiosis	Demodiosis
40	Demodiosis	Demodiosis
41	Demodiosis	Demodiosis
42	Demodiosis	Demodiosis
43	Demodiosis	Demodiosis
44	Demodiosis	Demodiosis
45	Demodiosis	Demodiosis
46	Demodiosis	Demodiosis
47	Demodiosis	Demodiosis
48	Demodiosis	Demodiosis
49	Demodiosis	Demodiosis
50	Distemper	Distemper
51	Distemper	Distemper
52	Distemper	Distemper

Tabel 6.5 (lanjutan)

53	Distemper	Distemper
54	Distemper	Distemper
55	Distemper	Distemper
56	Ringworm	Ehrlichiosis
57	Ehrlichiosis	Ehrlichiosis
58	Ehrlichiosis	Ehrlichiosis
59	Ehrlichiosis	Ehrlichiosis
60	Ehrlichiosis	Ehrlichiosis
61	Ehrlichiosis	Ehrlichiosis
62	Ehrlichiosis	Ehrlichiosis
63	Ehrlichiosis	Ehrlichiosis
64	Ehrlichiosis	Ehrlichiosis
65	Ehrlichiosis	Ehrlichiosis
66	Ehrlichiosis	Ehrlichiosis
67	Ehrlichiosis	Ehrlichiosis
68	Ehrlichiosis	Ehrlichiosis
69	Ringworm	Helminthiasis
70	Helminthiasis	Helminthiasis
71	Helminthiasis	Helminthiasis
72	Coccidiosis	Helminthiasis
73	Parvo	Helminthiasis
74	Helminthiasis	Helminthiasis
75	Parvo	Helminthiasis
76	Helminthiasis	Helminthiasis
77	Helminthiasis	Helminthiasis
78	Helminthiasis	Helminthiasis
79	Ringworm	Helminthiasis
80	Helminthiasis	Helminthiasis
81	Helminthiasis	Helminthiasis
82	Helminthiasis	Helminthiasis
83	Helminthiasis	Helminthiasis
84	Ringworm	Helminthiasis
85	Coccidiosis	Helminthiasis
86	Helminthiasis	Helminthiasis
87	Parvo	Helminthiasis
88	Otitis	Otitis
89	Otitis	Otitis
90	Otitis	Otitis
91	Otitis	Otitis
92	Otitis	Otitis

Tabel 6.5 (lanjutan)

93	Otitis	Otitis
94	Otitis	Otitis
95	Otitis	Otitis
96	Otitis	Otitis
97	Otitis	Otitis
98	Otitis	Otitis
99	Parvo	Parvo
100	Parvo	Parvo
101	Parvo	Parvo
102	Parvo	Parvo
103	Parvo	Parvo
104	Parvo	Parvo
105	Parvo	Parvo
106	Parvo	Parvo
107	Parvo	Parvo
108	Parvo	Parvo
109	Parvo	Parvo
110	Parvo	Parvo
111	Parvo	Parvo
112	Parvo	Parvo
113	Helminthiasis	Ringworm
114	Helminthiasis	Ringworm
115	Helminthiasis	Ringworm
116	Helminthiasis	Ringworm
117	Helminthiasis	Ringworm
118	Scabiosis	Scabiosis
119	Scabiosis	Scabiosis
120	Scabiosis	Scabiosis
121	Scabiosis	Scabiosis
122	Scabiosis	Scabiosis
123	Scabiosis	Scabiosis
124	Scabiosis	Scabiosis
125	Scabiosis	Scabiosis
126	Scabiosis	Scabiosis
127	Scabiosis	Scabiosis
128	Scabiosis	Scabiosis
129	Scabiosis	Scabiosis
130	Scabiosis	Scabiosis
131	Scabiosis	Scabiosis
132	Scabiosis	Scabiosis

Tabel 6.5 (lanjutan)

133	Scabiosis	Scabiosis
134	Scabiosis	Scabiosis
135	Scabiosis	Scabiosis
136	Scabiosis	Scabiosis
137	Scabiosis	Scabiosis
138	Scabiosis	Scabiosis
139	Scabiosis	Scabiosis
140	Scabiosis	Scabiosis
141	Scabiosis	Scabiosis
142	Urolithiasis	Urolithiasis
143	Urolithiasis	Urolithiasis
144	Urolithiasis	Urolithiasis
145	Urolithiasis	Urolithiasis
146	Urolithiasis	Urolithiasis
147	Urolithiasis	Urolithiasis
148	Urolithiasis	Urolithiasis
149	Urolithiasis	Urolithiasis
150	Urolithiasis	Urolithiasis

Dari hasil pengujian diatas didapatkan sebanyak 19 data uji yang memiliki hasil klasifikasi berbeda dengan kelas sebenarnya, dari hasil tersebut kemudian akan dihitung akurasi menggunakan persamaan (2-8) sehingga dihasilkan akurasi sebagai berikut :

$$\begin{aligned} \text{Akurasi} &= \frac{131}{150} \times 100\% \\ &= 87,33\% \end{aligned}$$

### 6.1.5 Akurasi Rata-Rata

Setelah dilakukan pengujian pengaruh nilai K terhadap keakuratan hasil, maka kemudian akan dilakukan perhitungan akurasi rata-rata dengan K=5, K=10, K=15, K=20 maka akan didapatkan presentase akurasi sebanyak 91,50%.

## 6.2 Analisis

Setelah dilakukan pengujian pengaruh nilai K terhadap keakuratan hasil, maka kemudian akan dianalisis hasil dari setiap pengujian, dari pengujian yang dilakukan sebanyak empat kali didapatkan hasil pengujian yaitu, pengujian pertama pada dengan nilai K=5 didapatkan akurasi yang tinggi dengan nilai 98,67%, sedangkan untuk pengujian kedua dengan nilai K = 10 akurasi mulai turun menjadi 93,33%, ketika dilakukan pengujian ketiga dengan nilai K = 15 didapatkan akurasi yang lebih rendah lagi yaitu dengan nilai akurasi sebesar 86,67% dan ketika



dilakuka pengujian yang terakhir dengan nilai  $K = 20$  nilai akurasi lebih tinggi daripada pengujian ketika  $K = 15$  meskipun peningkatannya sangat sedikit, nilai akurasi yang dihasilkan yaitu 87,33%. Dari keseluruhan hasil pengujian diketahui bahwa ketika nilai  $K$  semakin tinggi maka terjadi kecenderungan akurasinya semakin rendah.



## BAB 7 KESIMPULAN DAN SARAN

### 7.1 Kesimpulan

Kesimpulan yang didapatkan berdasarkan pengujian yang telah dilakukan terhadap diagnosis penyakit anjing menggunakan *Fuzzy K-Nearest Neighbor* (FK-NN) adalah sebagai berikut:

1. FK-NN dapat diimplementasikan untuk diagnosis penyakit pada anjing dengan beberapa tahap yaitu menghitung jarak antara data latih dengan data uji, mengambil jarak terkecil antara data latih dengan data uji sebanyak  $K$ , Fuzzifikasi dan Defuzzifikasi, Kelas dengan nilai defuzzifikasi tertinggi adalah yang dijadikan sebagai kelas untuk hasil klasifikasi.
2. Akurasi tertinggi yang didapatkan dari hasil pengujian adalah ketika  $K = 5$  yaitu dengan nilai 98,67%.

### 7.2 Saran

Saran yang dapat diberikan dari penelitian ini untuk penelitian selanjutnya, yaitu:

1. Menggunakan data latih dengan komposisi kelas penyakit yang lebih seimbang dan menggunakan nilai  $k$  yang lebih optimal agar didapatkan nilai akurasi yang lebih baik.
2. Melakukan kombinasi metode FK-NN dengan metode lain untuk hasil akurasi yang lebih optimal.

## DAFTAR PUSTAKA

- Allen, K., Blascovich, J., Mendes, W.B., 2002, *Cardiovascular reactivity and the presence of pets, friends, and spouses: the truth about cats and dogs*, *Psychosomatic Medicine*, 64(5), p.727-739.
- Anugerah, A. S. P., Indriati., Dewi, C., 2018., Implementasi Algoritma Fuzzy K-Nearest Neighbor untuk Penentuan Lulus Tepat Waktu (Studi Kasus : Fakultas Ilmu Komputer Universitas Brawijaya).
- Bowman, D.D., Nelson, T., 2014, *Canine Intestinal Helminths*, [pdf] Companion Animal Parasite Council, Tersedia di: <<http://todaysveterinarypractice.navc.com/wp-content/uploads/2016/06/T1401C08.pdf>> [Diakses pada 27 Agustus 2017]
- Brown, S.A., 2016, *Urolithiasis in Small Animals*, [online] Tersedia di: <<http://www.merckvetmanual.com/urinary-system/noninfectious-diseases-of-the-urinary-system-in-small-animals/urolithiasis-in-small-animals#v3296099>> [Diakses pada 10 September 2017]
- Chowdhury, S., 2014, *Did dogs really evolve from wolves? New evidence suggests otherwise*, [online] Tersedia di: <<https://www.csmonitor.com/Science/2014/0117/Did-dogs-really-evolve-from-wolves-New-evidence-suggests-otherwise>> [Diakses pada 10 September 2017]
- Constable, P.D., 2016, *Coccidiosis of Cats and Dogs*, [online] Tersedia di: <<http://www.merckvetmanual.com/digestive-system/coccidiosis/coccidiosis-of-cats-and-dogs>> [Diakses pada 10 September 2017]
- Dryden, M.W., 2016, *Mange in Dogs and Cats*, [online] Tersedia di: <<http://www.merckvetmanual.com/integumentary-system/mange/mange-in-dogs-and-cats>> [Diakses pada 10 September 2017]
- Han, J., Kamber, M., Pei, J., 2012, *Data Mining: Concepts and Techniques Third Edition*, United States of America: Morgan Kaufmann Publisher.
- Hark, D., 2013, *12 Ways Your Pet Can Improve Your Mental Health!*, [online] Tersedia di: <[http://www.huffingtonpost.com/danielle-hark/pet-ownership-health\\_b\\_318796.html](http://www.huffingtonpost.com/danielle-hark/pet-ownership-health_b_318796.html)> [Diakses pada 5 September 2017]
- Iqbal, M., Kustiyo A., Handharyani, E., 2007, *Klasifikasi Pasien Suspect Parvo dan Distemper Pada Data Rekam Medik Rumah Sakit Hewan IPB Menggunakan Voting Feature Intervals*, [pdf] Institut Pertanian Bogor. Tersedia di: <<http://download.portalgaruda.org/article.php?article=85686&val=235>> [Diakses pada 5 Oktober 2017]
- Jacobson, L., 2016, *The History of Dogs as Pets*, [online] Tersedia di: <<http://abcnews.go.com/Lifestyle/history-dogs-pets/story?id=41671149>> [Diakses pada 10 September 2017]

- Mariana, N., Redjeki, R. S., Razaq, J. A., 2015., "Penerapan Algoritma k-NN (Nearest Neighbor) untuk Deteksi Penyakit (Kanker Serviks).
- Meristika, Y. S., 2013., "Perbandingan K-Nearest Neighbor dan Fuzzy K-Nearest neighbor pada Diagnosis Penyakit Diabetes Melitus".
- Mitchell, K.D., 2016, *Canine Parvovirus*, [online] Tersedia di: <<http://www.merckvetmanual.com/digestive-system/diseases-of-the-stomach-and-intestines-in-small-animals/canine-parvovirus>> [Diakses pada 10 September 2017]
- Moriello, K.A., Dryden, M.W., Foil, C.S., Hawkins, W.W., Klei, T.R., Lloyd, J.E., Mignon, B., Rosenkrantz, W., Stiller, D., Talcott, P.A., Villalobos, A.E., White, S.D., White, P.D., 2016. *Ringworm (Dermatophytosis) in Dogs*, [online] Tersedia di: < <http://www.merckvetmanual.com/dog-owners/skin-disorders-of-dogs/ringworm-dermatophytosis-in-dogs>> [Diakses pada 10 September 2017]
- Neer, T.M., Rosenbaum, M.R., 2016, *Otitis Externa in Dogs*, [online] Tersedia di: <<http://www.merckvetmanual.com/dog-owners/ear-disorders-of-dogs/otitis-externa-in-dogs>> [Diakses pada 10 September 2017]
- Nugraha, D. S., Putri, R. R. M., dan Wihandika, R. C., 2017., "Penerapan Fuzzy K-Nearest Neighbor(F-KNN) Dalam Menentukan Status Gizi Balita".
- Putri, Y. C. M., Atasnina, I., dan Yulita, I. N., 2010., "Analisis dan Implementasi teknik Fuzzy K-Nearest Neighbor pada Kasus Imbalance Class.
- Radostits, O.M, Ashford, D.A., Greene, C.E., Janzen, E.D., Stromberg, B.E., Appel, M.J., Barr, S.C., Dubey, J.P., Ettestad, P., Harkin, K.R., Hill, D.E., Hoskins, J.D., Choy, J.L., Rohrbach, B.W., Songer, J.G., Taboada, J., Thoen, C.O., Timoney, J.F., Tizard, I., 2016, *Canine Distemper (Hardpad Disease)*, [online] Tersedia di: <<http://www.merckvetmanual.com/dog-owners/disorders-affecting-multiple-body-systems-of-dogs/canine-distemper-hardpad-disease>> [Diakses pada 10 September 2017]
- Radostits, O.M., Ashford, D.A., Greene, C.E., Janzen, E.D., Stromberg, B.E., Appel, M.J., Barr, S.C., Dubey, J.P., Ettestad, P., Harkin, K.R., Hill, D.E., Hoskins, J.D., Choy, J.L., Rohrbach, B.W., Songer, J.G., Taboada, J., Thoen, C.O., Timoney, J.F., Tizard, I., 2016, *Ehrlichiosis and Related Infections in Dogs*, [online] Tersedia di: <<http://www.merckvetmanual.com/dog-owners/disorders-affecting-multiple-body-systems-of-dogs/ehrlichiosis-and-related-infections-in-dogs>> [Diakses pada 10 September 2017]
- Rialdi, B. R., 2018., "Identifikasi Penyakit pada Kambing Menggunakan Metode Fuzzy K-Nearest Neighbor (F-KNN)., Skripsi Mahasiswa Filkom Universitas Brawijaya.
- Tan, P.N., Steinbach, M., Kumar, V., 2004, *Introduction to Data Mining*, United Kingdom: Pearson Education Limited .

Wisdianto, A., 2013, "Penerapan Metode Fuzzy K-Nearest Neighbor untuk Pengklasifikasian Spam E-mail. Volume 1 no.6.

Zaiane, O.R., 1999, *Principles of Knowledge Discovery in Data*, [pdf] University of Alberta, Tersedia di: <  
<https://webdocs.cs.ualberta.ca/~zaiane/courses/cmput695/F07/slides/ch1-695-F07.pdf>> [Diakses pada 10 Oktober 2017



