

IMPLEMENTASI ALGORITME MODIFIED K-NEAREST NEIGHBOR (MKNN) UNTUK DIAGNOSIS PENYAKIT TANAMAN CENGKEH

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:
Rizaldy Amsyar
NIM: 125150207111022



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

PENGESAHAN

IMPLEMENTASI ALGORITME MODIFIED K-NEAREST NEIGHBOR (MKNN) UNTUK
DIAGNOSIS PENYAKIT TANAMAN CENGKEH

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
Rizaldy Amsyar
NIM: 125150207111022

Skripsi ini telah diuji dan dinyatakan lulus pada
2 Agustus 2018
Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

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

Dosen Pembimbing II

Rizal Setya Perdana, S.Kom, M.Kom
NIK: 201603 910118 1 001



Mengetahui
Ketua Jurusan Teknik Informatika

Iri 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, 2 Agustus 2018

METERAI TEMPEL
1A7C7AFF198903665
6000
ENAM RIBU RUPIAH



Rizaldy Amsyar
NIM: 125150207111022

KATA PENGANTAR

Segala puji dan syukur kehadirat Allah SWT yang tak henti-hentinya memberikan karunia dan kasih sayang-Nya kepada penulis dan semua umatnya, shalawat serta salam juga tak luput untuk dipanjatkan kepada junjungan besar kita Nabi Muhammad SAW yang menjadi suri tauladan sehingga penulis dapat menyelesaikan skripsi dengan judul “Implementasi Algoritme Modified K-Nearest Neighbor (MKNN) Untuk Diagnosis Penyakit Tanaman Cengkeh”. Skripsi ini disusun untuk memenuhi salah satu syarat memperoleh gelar Sarjana Komputer di Fakultas Ilmu Komputer Universitas Brawijaya Malang.

Dengan kata pengantar ini penulis ingin mengucapkan rasa terimakasih kepada berbagai pihak yang membantu penulis dalam menyusun dan mengerjakan skripsi ini. Oleh karena itu penulis sangat berterimakasih kepada:

1. Bapak Nurul Hidayat , S.Pd, M.Sc selaku dosen pembimbing satu yang telah membimbing dan memberikan saran-saran serta nasihat agar penulis dapat merampungkan skripsi ini.
2. Bapak Rizal Setya Perdana, S.Kom, M.Kom selaku dosen pembimbing dua yang telah membimbing dan memberikan saran-saran serta nasihat agar penulis dapat merampungkan skripsi ini.
3. Kedua Orang tua yang tidak kenal lelah dan terus menyemangati serta memberikan bantuan serta doa. Tak lupa adik-adik dan keluarga besar yang terus memberikan dukungan.
4. Seluruh Bapak dan Ibu dosen serta staff FILKOM UB yang telah banyak memberikan Pengetahuan yang berharga selama penulis menempuh Pendidikan.
5. Segenap kerabat teman teman ori'94 , 11PT dan TIF UB 2012, yang lulus lebih dulu atau yang masih kuliah dan masih bersemangat. Terimakasih atas pelajaran pelajaran kehidupan yang diberikan.

Oleh karena itu penulis berharap skripsi ini dapat bermanfaat di keesokan hari baik untuk penulis atau yang memerlukan.

Tak lupa juga penulis menyadari akan banyaknya hal yang kurang dalam pengerjaan skripsi ini. Untuk hal-hal yang kurang tadi penulis dengan senang hati akan mendengarkan kritik serta saran yang diberikan agar dapat menjadi perbaikan di masa yang akan datang.

Akhir kata pada kesempatan ini penulis ingin memohon maaf yang sebesar-besarnya jika selama proses pengerjaan skripsi penulis melakukan kesalahan baik yang dikenal atau tidak, disengaja atau tidak. Sekali lagi penulis ingin sampaikan ucapan terima kasih.

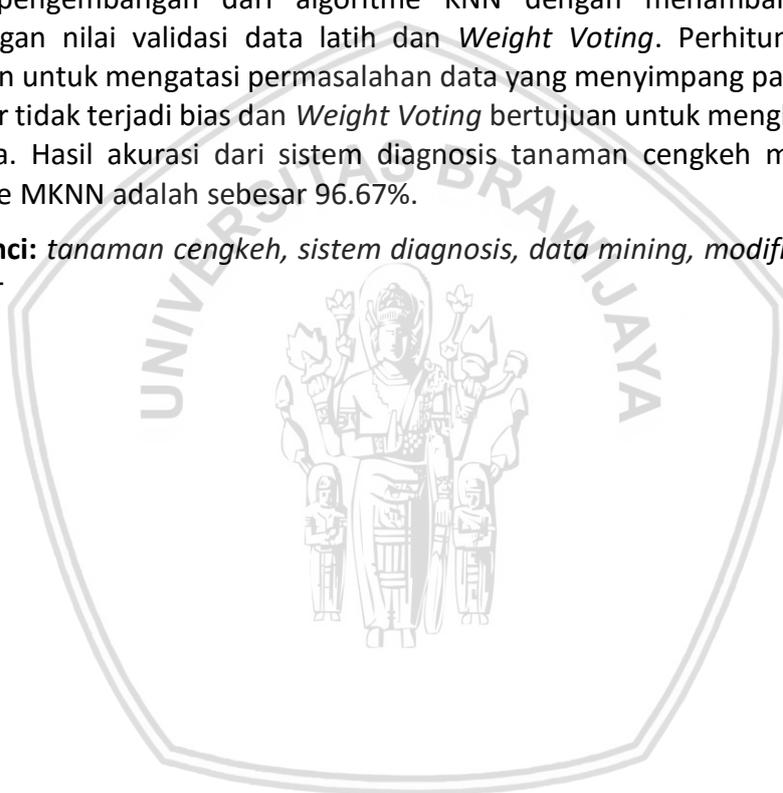
Malang, 2 Agustus 2018

Penulis
Rizaldy.amsyar@gmail.com

ABSTRAK

Tanaman cengkeh memiliki nilai ekonomi tinggi dan merupakan salah satu komoditas ekspor hasil perkebunan Indonesia, di daerah Wonosalam Kabupaten Jombang terdapat kebun cengkeh yang kurang terawat karena pemilik yang tidak setiap saat ada di kebun, dan dengan demikian tanaman rentan terkena penyakit dan berkurangnya hasil panen dari cengkeh tersebut. dibutuhkan sebuah cara untuk membantu petani untuk mengetahui jenis penyakit yang menyerang tanaman cengkeh tersebut, maka dibuatlah sistem diagnosis tanaman cengkeh dengan menggunakan algoritme *Modified K – Nearest Neighbor* (MKNN). Sistem diagnosis akan memberikan informasi penyakit tanaman cengkeh berdasarkan masukan berupa gejala-gejala yang dapat diamati dari tanaman. Algoritme MKNN adalah pengembangan dari algoritme KNN dengan menambahkan proses perhitungan nilai validasi data latih dan *Weight Voting*. Perhitungan validasi bertujuan untuk mengatasi permasalahan data yang menyimpang pada algoritme KNN agar tidak terjadi bias dan *Weight Voting* bertujuan untuk menghitung bobot dari data. Hasil akurasi dari sistem diagnosis tanaman cengkeh menggunakan algoritme MKNN adalah sebesar 96.67%.

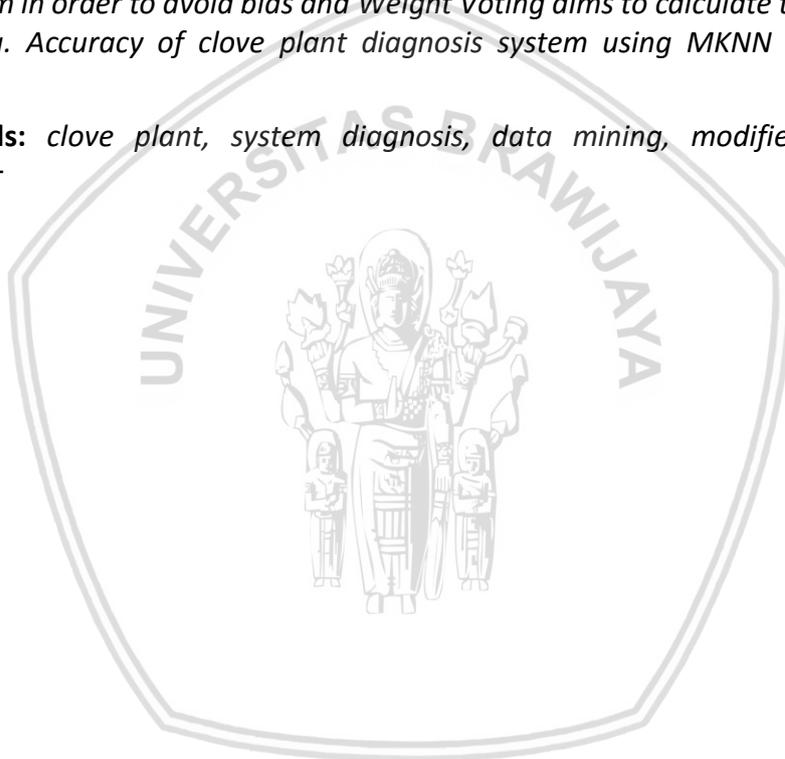
Kata Kunci: *tanaman cengkeh, sistem diagnosis, data mining, modified k-nearest neighbor*



ABSTRACT

Clove plant have high economic value and one of many export commodity of Indonesian plantation product, in Wonosalam region Jombang Regency there are less well groomed clove farm because the owners are not at all the times in the farm, and thus the plant susceptible to disease and reduced yields from the clove harvest. Needed a way to help farmers to know the types of diseases that attack the clove plants, then made a clove plant diagnosis system using the algorithm Modified K - Nearest Neighbor (MKNN). The diagnostic system will provide clove plant disease information based on inputs of observable symptoms of the plant. MKNN algorithm is the development of KNN algorithm by adding calculation process of validation data validation value and Weight Voting. Validation calculation aims to overcome the problem of data that deviates on the KNN algorithm in order to avoid bias and Weight Voting aims to calculate the weight of the data. Accuracy of clove plant diagnosis system using MKNN algorithm is 96.67%.

Keywords: *clove plant, system diagnosis, data mining, modified k-nearest neighbor*



DAFTAR ISI

IMPLEMENTASI ALGORITME MODIFIED K-NEAREST NEIGHBOR (MKNN) UNTUK DIAGNOSIS PENYAKIT TANAMAN CENGKEH.....	i
PENGESAHAN	i
PERNYATAAN ORISINALITAS	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	v
ABSTRACT	vi
DAFTAR ISI.....	vii
DAFTAR GAMBAR.....	x
DAFTAR TABEL.....	xi
BAB 1 PENDAHULUAN.....	1
1.1 Latar belakang.....	1
1.2 Rumusan masalah.....	2
1.3 Tujuan	2
1.4 Manfaat.....	3
1.5 Batasan Masalah.....	3
1.6 Sistematika Pembahasan.....	3
BAB 2 Landasan kepustakaan	5
2.1 Kajian Pustaka	5
2.2 Cengkeh	6
2.2.1 Penyakit Cengkeh.....	7
2.3 Klasifikasi.....	8
2.4 Algoritme K-NN	8
2.5 Algoritme Modified K-NN.....	9
2.5.1 Validasi Data Latih.....	9
2.5.2 Weight Voting	9
2.6 Mengukur akurasi	10
BAB 3 Metodologi	11
3.1 Studi Literatur	12
3.2 Analisis Kebutuhan	12
3.3 Pengumpulan Data	12



3.4 Perancangan sistem	13
3.5 Implementasi	13
3.6 Pengujian Sistem.....	14
3.7 Kesimpulan dan Saran	14
BAB 4 perancangan	15
4.1 Deskripsi Sistem	16
4.2 <i>Preprocessing</i> data.....	16
4.3 Perancangan perangkat lunak	17
4.3.1 Perancangan proses	17
4.4 Perhitungan Manual	22
4.5 Perancangan Desain Antarmuka	27
4.5.1 Desain Antarmuka Halaman Utama.....	27
4.5.2 Desain Antarmuka Halaman Data Latih	28
4.5.3 Desain Antarmuka Halaman Diagnosis	28
4.5.4 Desain Antarmukan Halaman Jarak <i>Euclidean</i>	29
4.5.5 Desain Antarmuka Halaman Validitas.....	29
4.5.6 Desain Antarmuka Halaman <i>Weight Voting</i>	30
4.5.7 Desain Antarmuka Halaman Pengujian	30
4.6 Perancangan Pengujian	31
BAB 5 IMPLEMENTASI SISTEM	32
5.1 Implementasi Program	32
5.1.1 Implementasi Proses <i>Euclidean Distance</i>	32
5.1.2 Implementasi Proses Validitas	33
5.1.3 Implementasi Proses <i>Weight Voting</i>	34
5.2 Implementasi Antarmuka	35
5.2.1 Halaman Utama	35
5.2.2 Halaman Data Latih	35
5.2.3 Halaman Diagnosis	36
5.2.4 Halaman <i>Euclidean</i>	36
5.2.5 Halaman Validitas.....	37
5.2.6 Halaman <i>Euclidean Uji</i>	37
5.2.7 Halaman <i>Weight Voting</i>	38



5.2.8 Halaman Pengujian	38
BAB 6 Pengujian Dan analisis	39
6.1 Pengujian	39
6.2 Hasil Pengujian.....	39
6.2.1 Pengujian 20 Data Latih	39
6.2.2 Pengujian 40 Data Latih	40
6.2.3 Pengujian 60 Data Latih	41
6.3 Analisis Pengujian	43
BAB 7 Kesimpulan Dan Saran.....	44
7.1 Kesimpulan.....	44
7.2 Saran	44
Daftar Pustaka.....	46
Lampiran Data Latih	48

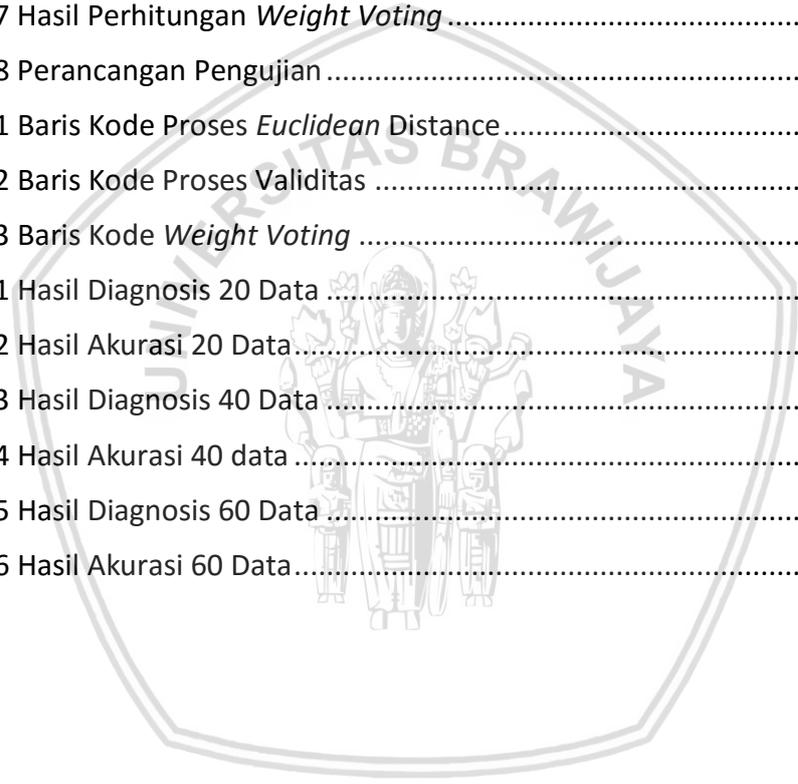


DAFTAR GAMBAR

Gambar 3.1 Diagram Alir Penelitian.....	11
Gambar 4.1 Diagram Perancangan Sistem	15
Gambar 4.2 Diagram alir sistem.....	17
Gambar 4.3 Diagram Alir Proses Klasifikasi MKNN	18
Gambar 4.4 Diagram Alir Perhitungan Jarak <i>Euclidean</i>	19
Gambar 4.5 Diagram Alir Perhitungan Validitas	21
Gambar 4.6 Diagram Alir Perhitungan <i>Weight Voting</i>	22
Gambar 4.7 Perancangan Halaman Utama.....	28
Gambar 4.8 Perancangan Halaman Data Latih	28
Gambar 4.9 Perancangan Halaman Diagnosis	29
Gambar 4.10 Perancangan Halaman Jarak <i>Euclidean</i>	29
Gambar 4.11 Perancangan Halaman Validitas.....	30
Gambar 4.12 Perancangan Halaman <i>Weight Voting</i>	30
Gambar 4.13 Perancangan Halaman Pengujian	31
Gambar 5.1 Tampilan Halaman Utama.....	35
Gambar 5.2 Tampilan Halaman Data Latih	35
Gambar 5.3 Tampilan Halaman Diagnosis	36
Gambar 5.4 Tampilan Halaman <i>Euclidean</i>	36
Gambar 5.5 Tampilan Halaman Validitas.....	37
Gambar 5.6 Tampilan Halaman <i>Euclidean Uji</i>	37
Gambar 5.7 Tampilan Halaman <i>Weight Voting</i>	38
Gambar 5.8 Tampilan Halaman Pengujian	38
Gambar 6.1 Grafik Hasil Pengujian Akurasi	43

DAFTAR TABEL

Tabel 2.1 Tabel Kajian Pustaka.....	6
Tabel 4.1 Gejala dan kode gejala	16
Tabel 4.2 Daftar Data Latih	23
Tabel 4.3 Data Uji.....	23
Tabel 4.4 Hasil Perhitungan Jarak <i>Euclidean</i> Antar Data Latih	24
Tabel 4.5 Hasil Perhitungan Validitas.....	25
Tabel 4.6 Jarak <i>Euclidean</i> Antara Data Uji dan Data Latih.....	26
Tabel 4.7 Hasil Perhitungan <i>Weight Voting</i>	27
Tabel 4.8 Perancangan Pengujian	31
Tabel 5.1 Baris Kode Proses <i>Euclidean Distance</i>	32
Tabel 5.2 Baris Kode Proses Validitas	33
Tabel 5.3 Baris Kode <i>Weight Voting</i>	34
Tabel 6.1 Hasil Diagnosis 20 Data	39
Tabel 6.2 Hasil Akurasi 20 Data.....	40
Tabel 6.3 Hasil Diagnosis 40 Data	40
Tabel 6.4 Hasil Akurasi 40 data	41
Tabel 6.5 Hasil Diagnosis 60 Data	41
Tabel 6.6 Hasil Akurasi 60 Data.....	42



BAB 1 PENDAHULUAN

1.1 Latar belakang

Tanaman cengkeh sejak dulu sudah dibudidayakan di Indonesia, Nilai ekonomi dari produksi cengkeh ini cukup tinggi, hasilnya yang paling banyak digunakan adalah bunga kering sebagai bahan baku rokok kretek dan rempah rempah masakan. Indonesia sendiri merupakan negara produksi cengkeh serta konsumsi cengkeh, serta cengkeh dapat tumbuh subur di Indonesia Karena iklim yang panas dan matahari yang cukup, produksi cengkeh di Indonesia selalu menunjukkan grafik yang naik turun yang tergantung oleh siklus tertentunya (Pertanian.go.id, 2012).

Cengkeh juga memiliki kerumitan dalam penanamannya dan sering terkena penyakit, penyakit cengkeh juga beragam dan memiliki kriteria serta penanganan yang berbeda-beda, serta tidak semua masalah dapat diatasi secara praktis oleh petani (Badan Litbang Pertanian, 2011). Di kecamatan Wonosalam Kabupaten Jombang terdapat perkebunan cengkeh yang rata-rata dimiliki oleh orang luar daerah, dan mereka tidak secara langsung ikut merawat perkebunan cengkeh tersebut.

Permasalahan yang muncul yaitu ketika pemilik perkebunan tidak selalu ada di perkebunan, banyak ditemukan tanaman cengkeh yang terjangkit penyakit dan rusak karena kurangnya perawatan dan penanganan dari penyakit. Hal itu dapat menyebabkan kerugian bagi pemilik perkebunan cengkeh. Tentunya sebelum terkena penyakit tanaman cengkeh menunjukkan gejala-gejala yang dapat diamati dan langsung dilakukan penanganan. Maka dari itu dibutuhkan seorang ahli yang dapat memberikan pengetahuan mengenai hal-hal yang menyangkut dengan tanaman cengkeh. Namun pengetahuan tentang penyakit cengkeh memiliki kendala seperti perlunya konsultasi kepada seorang ahli untuk setiap masalah yang menjadikannya kurang efektif serta dapat berdampak pada kualitas produksi sampai gagal panen. Pada hal-hal seperti inilah para pemilik perkebunan cengkeh memerlukan sebuah alat bantu atau sistem yang dapat memberikan informasi diagnosis mengenai penyakit tanaman cengkeh.

Klasifikasi merupakan salah satu bagian dari data mining yang bertujuan menggali nilai-nilai atau informasi dari sekumpulan data (Nikam, 2015). Dengan melakukan klasifikasi suatu jenis penyakit dapat diketahui polanya berdasarkan data yang ada, yaitu gejala gejala penyakit cengkeh yang terlihat pada tanaman tersebut seperti bagaimana kondisi daun, batang, akar dan buahnya. Dari situ dapat dilakukan proses klasifikasi menggunakan computer dengan menggunakan algoritme seperti *fuzzy tsukamoto*, *Naïve bayes*, *K-Nearest Neighbor* atau *Modified K-Nearest Neighbor*(MKNN).

Algoritme *Modified K-Nearest Neighbor* (MKNN) adalah salah satu cara untuk melakukan klasifikasi dan memberikan hasil diagnosis, alasan menggunakan algoritme ini karena memiliki keunggulan untuk mengklasifikasikan data gejala penyakit dengan data uji yang ada. MKNN memiliki penambahan proses

perhitungan baru untuk melakukan klasifikasi, dengan adanya perhitungan validitas antar data latih dan perhitungan bobot untuk masing-masing tetangga terdekatnya. Dengan adanya 2 proses baru diharapkan dapat memperbaiki kekurangan pada algoritme K-NN (Wafiyah, 2017). Algoritme MKNN masih lebih unggul dibandingkan dengan algoritme KNN, banyaknya data latih mempengaruhi hasil kinerja dari algoritme tersebut, apabila data semakin banyak maka hasilnya semakin baik (Mutrofin, 2016).

Pada penelitian yang dilakukan oleh Hananto dalam mendiagnosis penyakit cengkeh dengan metode *Forward Chaining* menggunakan 27 penyakit dan 89 gejala yang diimplementasikan, pengujian dilakukan oleh pihak ketiga yang tidak memiliki pengetahuan tentang cengkeh dan hasilnya dievaluasi oleh pakar. pada pengujian menghasilkan bahwa sistem yang dibangun memberikan hasil sesuai dengan harapan dan disetujui oleh pakar (Hananto, 2012).

Pada penelitian sebelumnya dengan judul “Implementasi Algoritme *Modified K-Nearest Neighbor* (MKNN) Untuk menentukan Tingkat Resiko Penyakit Lemak Darah (profil Lipid) menghasilkan rata rata akurasi sebesar 85,81% (Kumalasari, 2014). KNN adalah suatu algoritme pengklasifikasian data yang bekerja didasari dengan jarak terpendek antara data uji dan data latih untuk mendapatkan hasil dari K-NN nya, maka dari itu dikembangkanlah algoritme MKNN yang merupakan salah satu algoritme klasifikasi yang bisa mengklasifikasikan data baru yang belum diketahui kelasnya.

Berdasarkan paparan diatas, penulis mengusulkan judul skripsi “Implementasi Algoritme *Modified K-Nearest Neighbor* (MKNN) untuk diagnosis penyakit tanaman cengkeh”.

1.2 Rumusan masalah

Dari latar belakang yang sudah dijelaskan, maka rumusan masalah yang didapatkan adalah sebagai berikut:

1. Bagaimana implementasi *Modified K-NN* untuk mendiagnosis penyakit tanaman cengkeh.
2. Berapa akurasi dari sistem diagnosis dalam menentukan jenis penyakit dari berbagai kriteria gejala yang ada.

1.3 Tujuan

Adapun tujuan dari penelitian ini adalah :

1. Menerapkan algoritme *Modified K-NN* pada sistem diagnosis penyakit tanaman cengkeh.
2. Menguji tingkat akurasi sistem diagnosis penyakit tanaman cengkeh menggunakan algoritme *Modified K-NN*.

1.4 Manfaat

Diharapkan dengan dibuatnya sistem diagnosis ini dapat membantu petani dan pemilik kebun cengkeh sehingga tidak kesulitan dalam perawatan tanaman cengkeh, serta dapat mengetahui jenis jenis penyakit yang menyerang tanaman cengkeh dan bagaimana cara mengatasinya.

1.5 Batasan Masalah

Adapun batasan-batasan masalah pada penelitian ini antara lain sebagai berikut:

1. Data yang diperoleh didapat dari wawancara dengan pakar tentang penyakit tanaman cengkeh di Dinas Pertanian Kecamatan Wonosalam Kabupaten Jombang.
2. Data yang digunakan sebagai input meliputi 11 jenis macam gejala dan 3 jenis penyakit.
3. Implementasi berbasis *web*.
4. Interaksi dari sistem menggunakan menginputkan gejala dari daftar gejala-gejala yang sudah ada kemudian sistem akan melakukan diagnosis terhadap gejala yang diinputkan tadi.
5. Sistem yang dibangun merupakan implementasi dari algoritme dengan hasil diagnosis penyakit tanaman cengkeh.

1.6 Sistematika Pembahasan

Adapun Sistematika pembahasan dari penelitian ini dijelaskan sebagai berikut:

BAB 1 Pendahuluan

Bab ini berisi latar belakang tentang penelitian, rumusan masalah yang didapat, tujuan yang akan diselesaikan, manfaat dari penelitian, batasan masalah dari penelitian serta sistematika pembahasan.

BAB 2 Landasan Kepustakaan

Bab ini menjelaskan teori, konsep dari penelitian, algoritme dan sistem dari literatur ilmiah yang penulis rujuk berkaitan dengan tema, serta sumber pustaka yang berkaitan dengan judul penelitian.

BAB 3 Metodologi

Bagian ini menjelaskan bagaimana cara menyelesaikan masalah, menggunakan metode dan langkah langkah cara mendapatkan hasil dari penelitian.

BAB 4 Perancangan

Menjelaskan suatu analisis kebutuhan dan perancangan sistem diagnosis penyakit tanaman cengkeh menggunakan *Modified K-NN*.

BAB 5 Implementasi

Bab ini membahas mengenai bagaimana implementasi sistem diagnosis berbasis web yang dibuat.

BAB 6 Pengujian dan Analisis

Bab ini menjelaskan bagaimana cara pengujian yang digunakan serta hasil akurasi dari sistem diagnosis penyakit tanaman cengkeh yang di buat.

BAB 7 Penutup dan Kesimpulan

Berisi tentang kesimpulan yang didapatkan penulis dari pembuatan dan pengujian dari sistem yang telah dikembangkan dan berisi saran untuk membangun pengembangan sistem yang lebih baik lagi.



BAB 2 LANDASAN KEPUSTAKAAN

Secara umum, bab landasan kepustakaan meliputi kajian pustaka dan dasar teori yang difokuskan pada tanaman cengkeh sebagai inti dari penelitian dan *Modified K-NN* untuk algoritme yang digunakan dan diimplementasi ke dalam penelitian ini.

2.1 Kajian Pustaka

Pada penelitian ini kajian pustaka berisi pembahasan mengenai penelitian yang sudah dilakukan sebelumnya dan penelitian yang akan diusulkan. Penelitian sebelumnya yang di kaji memiliki objek yang sama yaitu tanaman cengkeh serta penelitian yang lainnya menggunakan algoritme yang sama. Penelitian yang dibandingkan dilakukan dengan membandingkan objek yang digunakan, algoritme serta hasil yang didapat.

1. Penelitian terdahulu yang berjudul Implementasi Algoritma Modified *K-Nearest Neighbor* (MKNN) untuk Klasifikasi Penyakit Demam yang dilakukan oleh Wafiyah menjelaskan bahwa algoritme yang digunakan adalah pengembangan dari algoritme KNN, algoritme MKNN memiliki penambahan proses perhitungan yang baru ketika melakukan klasifikasi, adapun penambahannya adalah perhitungan validitas dan menghitung *Weight Voting* (Wafiyah, 2017).
2. Penelitian dengan judul Identifikasi Diagnosis Gangguan Autisme Pada Anak Menggunakan Metode Modified *K-Nearest Neighbor* (MKNN) yang ditulis oleh Sianipar menyimpulkan bahwa nilai yang terbaik untuk k adalah 1 dan data latih sebanyak 60 dan dari penelitian tersebut didapat bahwa akurasi dipengaruhi oleh besar kecilnya nilai k dan jumlah dari data latih yang digunakan (Sianipar, 2017).
3. Penelitian yang berjudul Diagnosis Penyakit Kulit Pada Kucing Menggunakan Metode *Modified K-Nearest Neighbor* didapat dari peneliti tingkat akurasi sebesar 86.67%. Penulis menyimpulkan bahwa akurasi dari sistem semakin tinggi dengan semakin banyaknya jumlah dari data latih (Putri, 2017).
4. Penelitian yang dilakukan oleh Hananto dengan judul Sistem Pakar Diagnosis Penyakit Tanaman Cengkih dengan Metode Inferensi *Forward Chaining* mendapatkan hasil yang sesuai dengan pakar dan dapat diterima, metode pengujian menggunakan *black-box* serta menggunakan 89 gejala sebagai kriteria dan 27 jenis penyakit yang dapat didiagnosis, perangkat lunak yang dihasilkan dapat memberikan cara-cara pencegahan yang harus dilakukan untuk penanganan penyakit tanaman cengkeh (Hananto, 2012).
5. Penelitian selanjutnya berjudul Perancangan Sistem Diagnosa Penyakit Hepatitis Menggunakan Metode KNN yang dilakukan oleh Helilintar dengan menggunakan 7 jenis gejala penyakit hepatitis dapat

menyimpulkan bahwa menggunakan metode KNN dinilai sudah cukup akurat untuk menyelesaikan masalah (Helilintar, 2017).

Hasil penelitian-penelitian sebelumnya yang penulis kaji dapat dilihat pada Tabel 2.1.

Tabel 2.1 Tabel Kajian Pustaka

Judul	Metode	Hasil
Implementasi Algoritma Modified <i>K-Nearest Neighbor</i> (MKNN) untuk Klasifikasi Penyakit Demam	<i>Modified K-Nearest Neighbor</i>	Hasil akurasi pengujian nilai k rata-rata didapat akurasi sebesar 88.55%
Identifikasi Diagnosis Gangguan Autisme Pada Anak Menggunakan Metode Modified <i>K-Nearest Neighbor</i> (MKNN)	<i>Modified K-Nearest Neighbor</i>	Nilai terbaik didapat dengan menggunakan nilai k adalah 1 dan jumlah data latih sebanyak 60
Diagnosis Penyakit Kulit Pada Kucing Menggunakan Metode Modified <i>K-Nearest Neighbor</i>	<i>Modified K-Nearest Neighbor</i>	Semakin banyak data latih mempengaruhi besarnya nilai akurasi dari sistem, akurasi yang didapat 89.67%
Sistem Pakar Diagnosis Penyakit Tanaman Cengkeh dengan Metode Inferensi <i>Forward Chaining</i>	<i>Forward Chaining</i>	Sistem dapat memberikan diagnosis yang sesuai dengan pakar.
Perancangan Sistem Diagnosa Penyakit Hepatitis Menggunakan Metode KNN	<i>K-Nearest Neighbor</i>	Dalam penggunaan metode <i>K-Nearest Neighbor</i> sudah dinilai akurat untuk menyelesaikan masalah diagnosis penyakit hepatitis

2.2 Cengkeh

Cengkeh merupakan tanaman tropis dengan nama latin *Eugenia aromatica* yang merupakan tanaman asli Indonesia. Tanaman ini tumbuh optimal pada 300 – 600 mdpal dengan suhu rata-rata 22° - 30°C dan curah hujan sekitar 1500-4500 mm/tahun (Badan Litbang Pertanian, 2011). Persebaran perkebunan cengkeh di Indonesia paling banyak dari Sulawesi, berdasarkan survey oleh Kementerian

Pertanian Republik Indonesia luas lahan yang digunakan untuk perkebunan cengkeh di Indonesia mencapai 510.00 Ha dengan total produksi nasional 122.134 Ton (Kementerian Pertanian RI, 2014).

2.2.1 Penyakit Cengkeh

Tanaman Cengkeh tak luput dari serangan penyakit yang merupakan hambatan dan kendala dalam usaha perkebunan cengkeh, selain itu cengkeh juga mudah terserang penyakit. Berikut ini beberapa contoh penyakit yang paling sering ditemui berdasarkan data yang didapat dari pakar:

1. Bakteri Pembuluh Kayu Cengkeh (BPKC)

Penyakit ini biasanya muncul pada saat pohon cengkeh berbuah untuk pertama kali, merupakan penyakit yang paling merusak tanaman cengkeh. Penyebabnya adalah bakteri *pseudomonas syzjii*. Penularan penyakit BPKC dari pohon sakit ke pohon sehat melalui serangga, pola penyebaran penyakit ini biasanya mengikuti arah angin serta dapat melalui alat-alat pertanian yang terkena infeksi dari pohon yang sakit. Ciri-ciri tanaman yang terkena BPKC daunnya gugur secara mendadak kemudian ranting-ranting pada pucuknya mati. Terkadang percabangan dan seluruh tanaman layu mendadak yang mengakibatkan daun menjadi kering, tahap ini berlangsung sekitar beberapa minggu sampai beberapa bulan. Untuk perawatan dari penyakit ini yang pertama dapat dilakukan sanitasi kebun dan penyemprotan *Hindola fulfa* dengan menggunakan insektisida kontak dengan interval sekitar 6 minggu sekali untuk tanaman yang masih bias diselamatkan. Kedua dapat dilakukan penyuntikan antibiotik oksitetasklin (OTC) sebanyak 6 gr/100 ml air dilakukan setiap 3 – 4 bulan sekali. Ketiga dapat dilakukan pemupukan kombinasi N dan K (DPT Madiun, 2017).

2. Cacar Daun Cengkeh (CDC)

Penyakit ini dikategorikan sebagai penyakit utama selain BPKC karena penyakit ini terdapat di hampir semua tempat produksi cengkeh di Indonesia. Penyakit ini disebabkan oleh *Phyllosticta syszygii* yang menyebabkan berkurangnya kemampuan fotosintesis dari daun, sehingga tanaman tampak tidak subur (Mariana, 2013). Gejala serangan yang diketahui pada permukaan atas daun timbul bercak-bercak yang menonjol seperti cacar. Gejala akan terlihat jelas pada daun yang masih muda, terkadang terdapat bintil-bintil hitam kecil. Pengendalian dari penyakit Cacar Daun Cengkeh dapat dilakukan penyemprotan fungisida dengan interval 7 – 10 hari sekali, selain itu sanitasi kebun juga harus dijaga, daun, ranting dan biji dari tanaman yang terkena penyakit sebaiknya di kumpulkan dan di bakar (DPT IPB, 2010).

3. Jamur Akar Putih (JAP)

Dari pakar didapat bahwa penyebab dari penyakit Jamur Akar Putih disebabkan oleh *Rigidoporus sp.* Gejala yang terlihat jelas yaitu daun tampak kasar dan agak melengkung ke arah bawah dan warna berubah

menjadi coklat kekuningan. Pada awal terkena penyakit perubahan warna terjadi pada satu cabang, tapi kemudian menyebar ke seluruh tanaman. Pada akar dan batang dekat akar terdapat pertumbuhan rhizomorph yang menyerupai kipas berwarna putih, oleh karena itu dinamakan jamur akar putih. Akibatnya akar menjadi lunak dan basah dan jika dibiarkan akar tanaman akan busuk. Untuk penanganan dapat dilakukan kultur teknis dengan membersihkan sisa tanaman, atau melakukan pemupukan dan alternative lain dengan menanam tanaman kacang kacang di sekitar pohon cengkeh (DPT IPB, 2010).

2.3 Klasifikasi

Klasifikasi merupakan sebuah teknik pada data mining yang dapat memproses data yang besar, teknik yang sering digunakan untuk melakukan analisis dari data latih yang sudah ada sehingga kesalahan ketika melakukan klasifikasi bisa berkurang (Nikam, 2015). Klasifikasi umumnya memiliki 2 tahap utama, tahap pertama model dibuat menggunakan algoritma klasifikasi seperti algoritme ID3, C4.5, *fuzzy*, *K-Nearest Neighbor*, *support vector machine* dan *naive bayes* dengan menggunakan data pada data latih. Selanjutnya pada tahap kedua model yang diekstrak di test dengan data latih tersebut untuk mengukur performa dan akurasi dari klasifikasi tersebut. Sehingga klasifikasi menentukan kelas dari sebuah data yang kelasnya tidak diketahui dengan menggunakan data latih sebagai acuan untuk menentukan kelasnya.

2.4 Algoritme K-NN

K-Nearest Neighbor banyak digunakan untuk menyelesaikan masalah klasifikasi data dan merupakan satu dari sekian teknik klasifikasi yang paling sederhana dan mudah dipahami. Algoritme ini memiliki keunggulan karena melaukan pelatihan data yang cepat, tahan terhadap data pelatihan yang memiliki derau dan efektif jika pelatihan dengan data yang besar, sedangkan kekurangannya adalah nilai k bias, komputasi yang kompleks, memori terbatas dan mudah tertipu dengan atribut yang tidak relevan (Mutrofin, 2016). Secara umum algoritme K-NN bertujuan mengklasifikasikan objek dari tetangga terdekatnya, nilai k menentukan sebanyak apa tetangga terdekat yang diambil untuk menentukan kelas dari data yang diuji, data uji akan dicari nilai *Euclidean* atau kedekatannya dengan data latih (Nikam, 2015). Prosesnya dimulai dengan cara mencari jarak dari data latih (x) terhadap data uji dengan menggunakan persamaan 2.1 rumus *Euclidean*.

$$d(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (2.1)$$

Keterangan:

$d(x, y)$: Jarak *Euclidean* dari data latih x terhadap data uji y

x_i : data latih x index ke i

y_i : Data uji y index ke i

2.5 Algoritme Modified K-NN

Modified K-NN merupakan pengembangan dari algoritme KNN dengan penambahan beberapa proses yakni perhitungan nilai validasi data latih dan *Weight Voting* atau pembobotan. Perhitungan nilai validasi bertujuan untuk mengatasi permasalahan data yang menyimpang pada algoritme KNN sehingga kurangnya bias dari data latih (Mutrofin, 2016) dan *Weight Voting* untuk menghitung bobot dari data. Algoritme Modified K-NN melakukan proses klasifikasi pada data uji dengan menggunakan hasil *Weight Voting* terbesar dari kelas yang ada pada data latih yang datanya sudah dilakukan validasi, berbeda dengan algoritme K-NN yang tidak melakukan proses validasi dari data latih. Dengan ditambahkan proses perhitungan validasi data latih ini dapat menghasilkan akurasi dari sistem yang rata-rata lebih tinggi dari penggunaan algoritme K-NN (putri, 2017).

2.5.1 Validasi Data Latih

Pada algoritme Modified K-NN semua data latih harus divalidasi dulu, pada proses perhitungan validasi untuk setiap data latih didapat dari nilai tetangga yang terdekat. Persamaan untuk proses validitas data latih ada pada persamaan 2.2.

$$Validitas(x) = \frac{1}{k} \sum_{i=1}^k S(label(x), (label(Ni(x)))) \quad (2.2)$$

Keterangan

k : Jumlah tetangga terdekat

$label(x)$: Label kelas x

$Ni(x)$: Label kelas terdekat dari x

fungsi dari S digunakan untuk mencari kesamaan dari titik kelas x dan data ke- i dari kelas tetangga yang terdekat. Dapat didefinisikan fungsi S seperti persamaan 2.3.

$$S(a, b) = \begin{cases} 1 & a = b \\ 0 & a \neq b \end{cases} \quad (2.3)$$

Keterangan:

a = kelas a dari data latih

b = kelas yang bukan kelas a dari data latih

2.5.2 Weight Voting

Weight Voting adalah proses perhitungan pada algoritme Modified K-NN, pada setiap data dihitung *Weight* atau bobot menggunakan $1/(d_e + a)$ yang mana d_e adalah jarak *Euclidean* dan a ditambah untuk menghindari penyebut dibagi dengan 0. Kemudian nilai validitas untuk masing-masing data latih yang sudah dilakukan perhitungan sebelumnya dikali dengan bobot berdasarkan jarak *Euclidean*. Untuk persamaannya dapat ditulis seperti persamaan 2.4.

$$W(i) = Validitas(i) \times \frac{1}{d_e + a} \quad (2.4)$$

Keterangan:

$W(i)$: Perhitungan *Weight Voting*

Validasi(i) : Nilai Validasi

d_e : Jarak *Euclidean*

a : 0.5

Weight Voting ini mempunyai pengaruh yang penting terhadap data yang memiliki validitas yang lebih besar dan kedekatan dengan data.

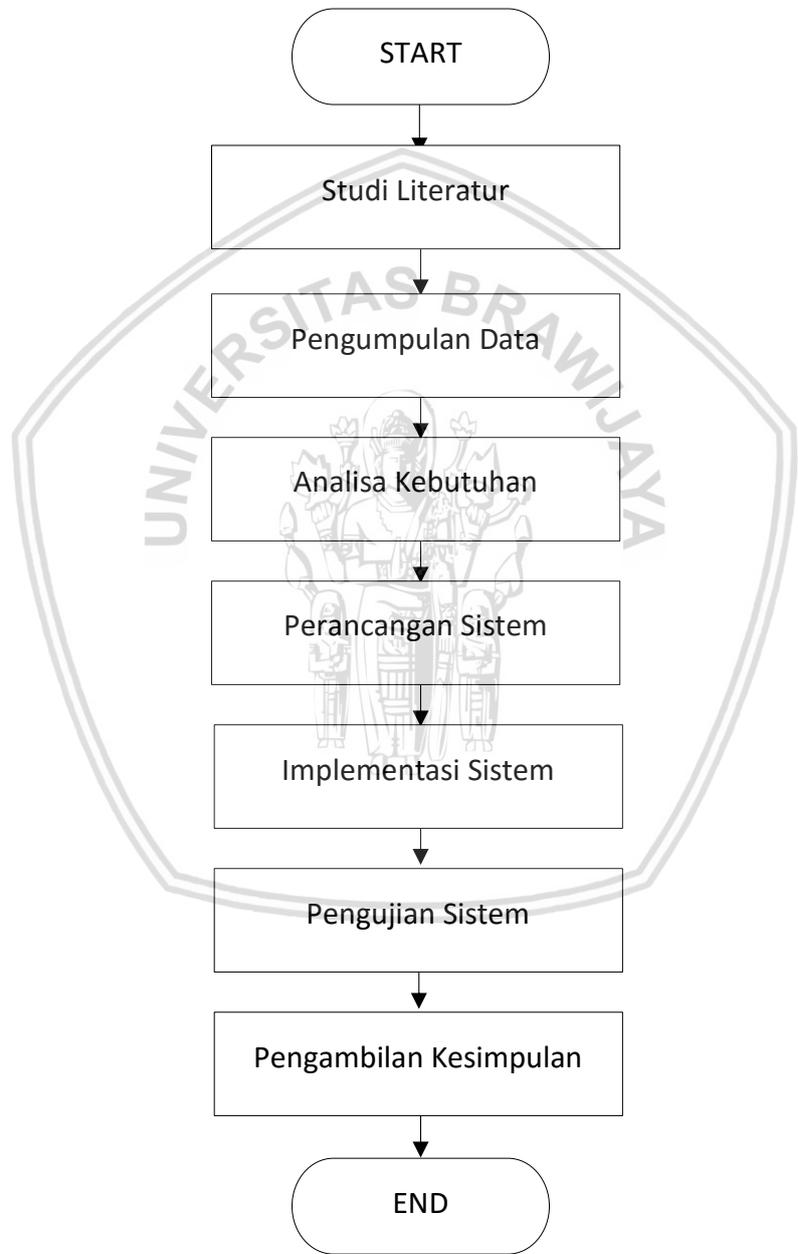
2.6 Mengukur akurasi

Pada sistem diagnosis yang dibuat, setelah didapat hasil diagnosis nya kemudian akan dilakukan mengukur tingkat akurasi dari sistem yang dibangun, dengan cara menyamakan hasil diagnosis dari sistem dengan hasil diagnosis dari data yang didapat dari pakar, adapun cara menghitung persentase akurasi dari hasil diagnosis sistem ada pada persamaan 2.5.

$$\text{Nilai akurasi} = \frac{\text{jumlah } n \text{ data akurat}}{\text{jumlah } n \text{ seluruh data}} \times 100\% \quad (2.5)$$

BAB 3 METODOLOGI

Pada bab ini membahas mengenai metodologi pada penelitian yang dilakukan ini, adapun prosesnya antara lain: studi literatur, pengumpulan data, analisis kebutuhan, perancangan sistem, implementasi sistem, pengujian sistem dan juga pengambilan kesimpulan. Adapun model diagram alur yang menggambarkan proses penelitian ini pada Gambar 3.1.



Gambar 3.1 Diagram Alir Penelitian

3.1 Studi Literatur

Mempelajari mengenai literatur yang akan digunakan dalam menunjang dalam pembuatan sistem diagnosis antara lain:

1. Sistem diagnosis
2. Penyakit pada tanaman cengkeh
3. Solusi pencegahan dan penanganan penyakit
4. Jurnal ilmiah mengenai *Modified K-Nearest Neighbor*
5. Pemrograman berbasis *web* menggunakan PHP dengan framework Code Igniter dan database PHPMyAdmin
6. Proses pengujian sistem

3.2 Analisis Kebutuhan

Analisis kebutuhan dirincikan dengan menyusun kebutuhan yang nantinya digunakan dalam implementasi perangkat lunak sistem diagnosis yang akan dibangun. Berikut adalah rinciannya:

1. Kebutuhan perangkat keras yang diperlukan adalah komputer dengan spesifikasi sebagai berikut:
 - a. Intel(R) Core(TM) i5-4288U CPU @2.60GHz
 - b. Memori 8 GB RAM
 - c. Harddisk kapasitas 500GB
 - d. Perangkat cerdas seluler
2. Kebutuhan perangkat lunak meliputi:
 - a. Sistem operasi Windows 10
 - b. Notepad++
 - c. Microsoft Excel 2016
 - d. Google Chrome
 - e. XAMPP
3. Data yang digunakan antara lain:
 - a. Data gejala penyakit tanaman cengkeh
 - b. Data penyakit dan penanganan tanaman cengkeh

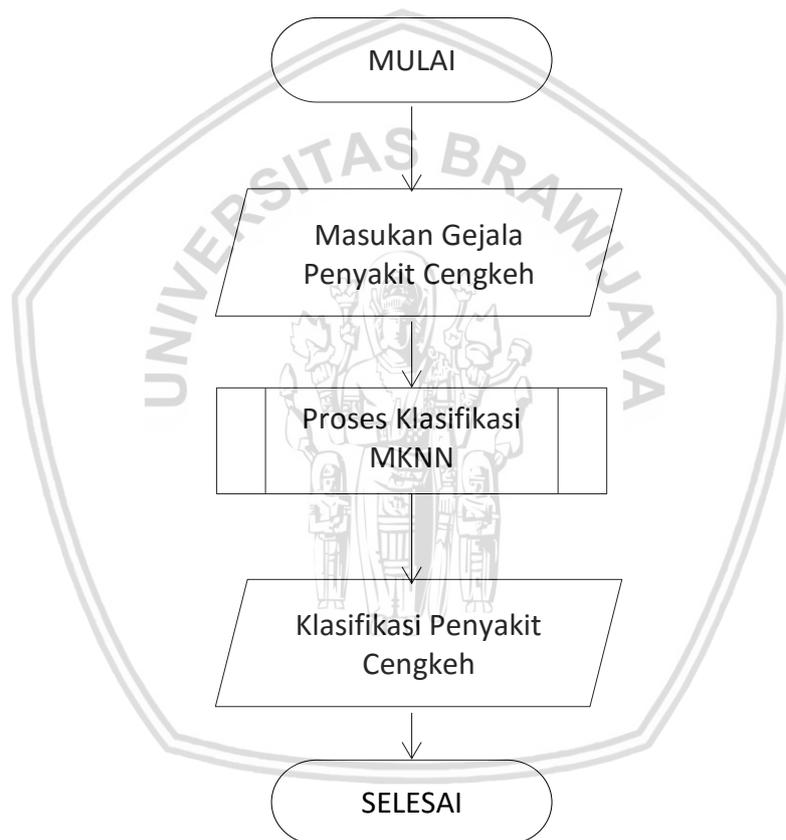
3.3 Pengumpulan Data

Pada tahap pengumpulan data merupakan bagaimana penulis mendapatkan data-data yang akan digunakan dalam penelitian, dimana dalam penelitian ini akan diambil mengenai gejala dan penyakit tanaman cengkeh yang akan

didapatkan melalui jurnal-jurnal serta pakar yang memiliki pengetahuan tentang tanaman cengkeh.

3.4 Perancangan sistem

Sistem yang akan dibangun disini merupakan sebuah sistem yang dapat mendiagnosis penyakit tanaman cengkeh dengan menggunakan algoritme *Modified K-Nearest Neighbor*. Tahap pertama sistem menerima input yang dimasukkan oleh pengguna berupa gejala dari penyakit tanaman cengkeh, kemudian data input tadi akan diolah oleh sistem menggunakan proses dari *Modified K-NN* dan menghasilkan output berupa hasil diagnosis penyakit tanaman cengkeh. Diagram alir sistem ditunjukkan oleh Gambar 3.2.



Gambar 3.2 Diagram Alir Sistem

3.5 Implementasi

Implementasi sistem merupakan sebuah proses membangun sistem dengan acuan perancangan sistem diagnosis dan menerapkan hal-hal yang didapatkan dari hasil proses studi literatur. Adapun tahapan dalam implementasi antara lain:

1. Penerapan algoritme *Modified K-Nearest Neighbor* berbasis web dengan menggunakan *framework Code Igniter*.
2. Pembuatan antarmuka pengguna.

3. Pengolahan data dilakukan.
4. Implementasi menghasilkan hasil diagnosis dari inputan pengguna, dan memberikan bagaimana cara penanganan dan perawatan penyakit tanaman cengkeh.

3.6 Pengujian Sistem

Pengujian sistem dilakukan supaya hasil dari implementasi sistem dapat beroperasi dengan baik dan benar serta sesuai dengan kebutuhan pengguna. Pengujian sistem dilakukan dengan cara membandingkan hasil diagnosis penyakit dari data yang sudah didapat dari ahli dengan hasil diagnosis dari sistem. Uji coba sistem dilakukan untuk mengukur seberapa besar persentase akurasi dari sistem yang telah dibangun.

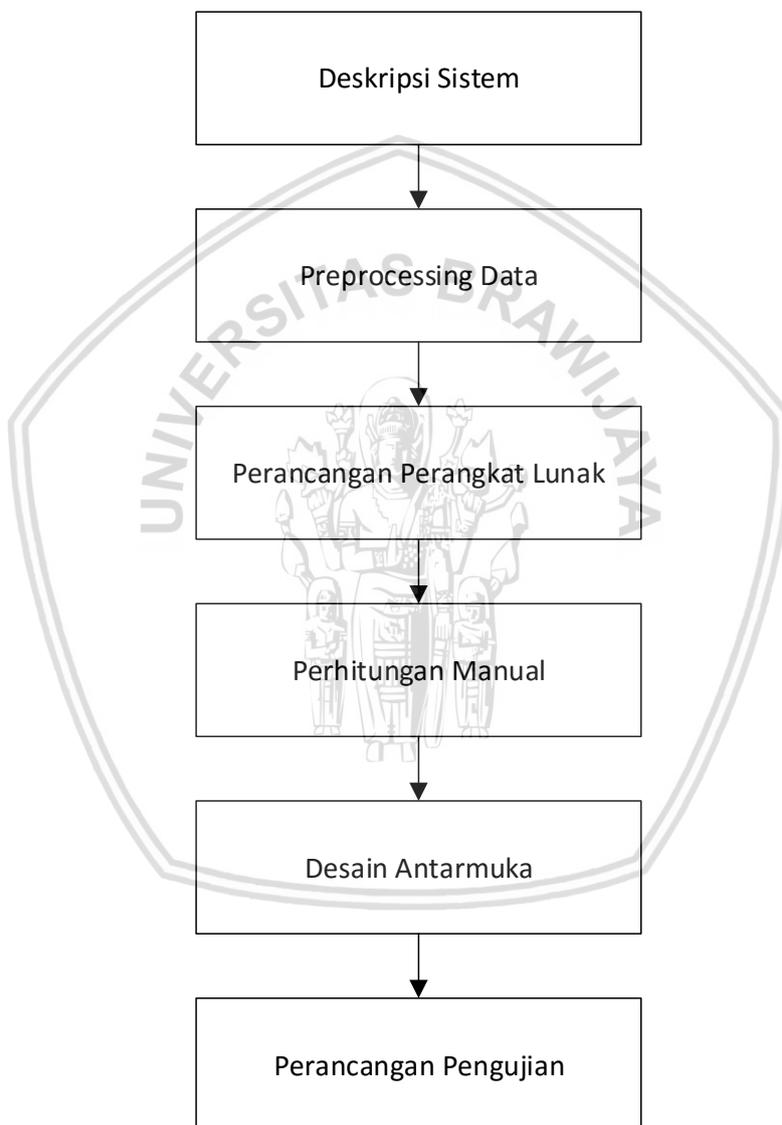
3.7 Kesimpulan dan Saran

Pengambilan kesimpulan akan dilakukan jika seluruh tahap dari perancangan, implementasi dan pengujian sudah selesai, setelah itu dapat dilakukan penarikan kesimpulan dari hasil pengujian dengan algoritme yang telah di terapkan, pada tahap ini akan dilakukan pemberian saran untuk mempertimbangkan pengembangan sistem yang dibangun.



BAB 4 PERANCANGAN

Bab ini akan menjelaskan tentang tahap-tahap perancangan dalam membuat sebuah sistem untuk memberikan informasi dan pengetahuan kepada pengguna mengenai diagnosis penyakit cengkeh. Sistem yang dibangun menggunakan algoritme *Modified K-Nearest Neighbor* (MKNN). Diagram perancangan dapat dilihat pada Gambar 4.1 Diagram Perancangan Sistem.



Gambar 4.1 Diagram Perancangan Sistem

4.1 Deskripsi Sistem

Sistem yang akan dibangun merupakan sebuah sistem diagnosis yang menggunakan algoritme MKNN untuk melakukan diagnosis penyakit pada cengkeh. Sistem ini akan mengolah data dari 11 gejala sebagai inputnya kemudian sistem akan mendiagnosis berdasarkan gejala itu termasuk ke penyakit yang mana dengan menerapkan algoritme *Modified K-Nearest Neighbor* (MKNN). Proses pada sistem ini terdapat pelatihan data dan pengujian data, pada proses pelatihan data, sistem menghitung jarak antar data latih yang kemudian menghasilkan nilai validitas. Setelah itu pada proses pembobotan atau *Weight Voting* yang dihitung dari nilai validitas dan jarak data uji dengan data latih. Pada pengujian data akan dihasilkan keluaran dari sistem diagnosis penyakit cengkeh.

4.2 Preprocessing data

Data yang digunakan pada sistem menggunakan 11 gejala penyakit pada tanaman cengkeh. Sedangkan untuk nilai pembobotan dari gejala tersebut telah di berikan oleh pakar. Nilai pembobotan gejala penyakit tanaman cengkeh yaitu ya dan tidak. Adapun gejala dan kode gejala yang digunakan pada proses perhitungan terdapat pada Tabel 4.1 gejala dan kode gejala.

Tabel 4.1 Gejala dan kode gejala

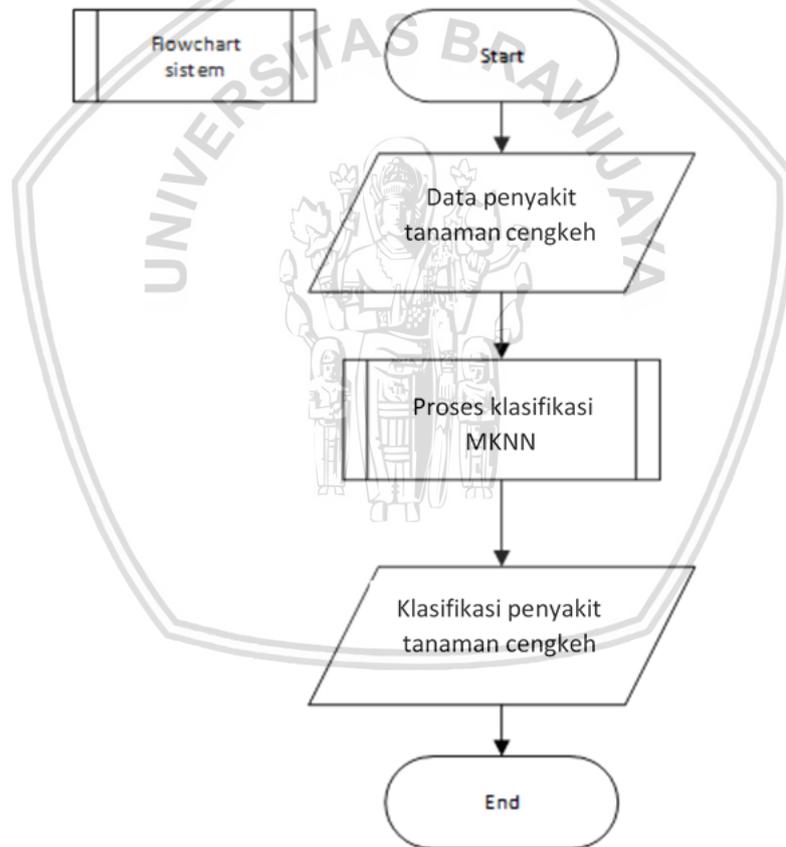
No	Nama gejala	Kode gejala
1	Ranting pada pucuk batang mati	G1
2	Daun gugur dari atas ke bawah	G2
3	Cabang mudah layu secara mendadak	G3
4	Daun yang tua berwarna kekuning-kuningan	G4
5	Batang pohon mengering	G5
6	Pada daun muda yang berwarna merah terdapat bagian melepuh	G6
7	Pada bercak melepuh terdapat titik hitam	G7
8	Daun bercak cacar terdapat di bagian bawah dan semakin ke atas semakin sedikit	G8
9	Daun cengkeh layu tiba-tiba, mengering dan rontok	G9
10	Benang miselium jamur menempel pada akar tanaman	G10
11	Akar membusuk	G11

4.3 Perancangan perangkat lunak

Pada tahap perancangan perangkat lunak akan membahas tentang hubungan dari tiap komponen sistem agar dapat menciptakan sistem yang berfungsi sesuai kebutuhan dari pengguna, adapun proses pada perancangan sistem ini yaitu, proses klasifikasi MKNN, proses *Euclidean* data latih dan data uji, proses validitas, dan proses perhitungan *Weight Voting*.

4.3.1 Perancangan proses

Perancangan proses pada sistem diagnosis penyakit tanaman cengkeh diawali dengan input gejala dari penyakit yang sudah disediakan, selanjutnya sistem akan melakukan klasifikasi MKNN, pada proses klasifikasi MKNN terdapat proses perhitungan nilai validitas, proses perhitungan untuk jarak *Euclidean* antar data latih dan perhitungan *Weight Voting*, kemudian didapatkan hasil diagnosis dari sistem. Model diagram alir dari sistem seperti pada Gambar 4.2.

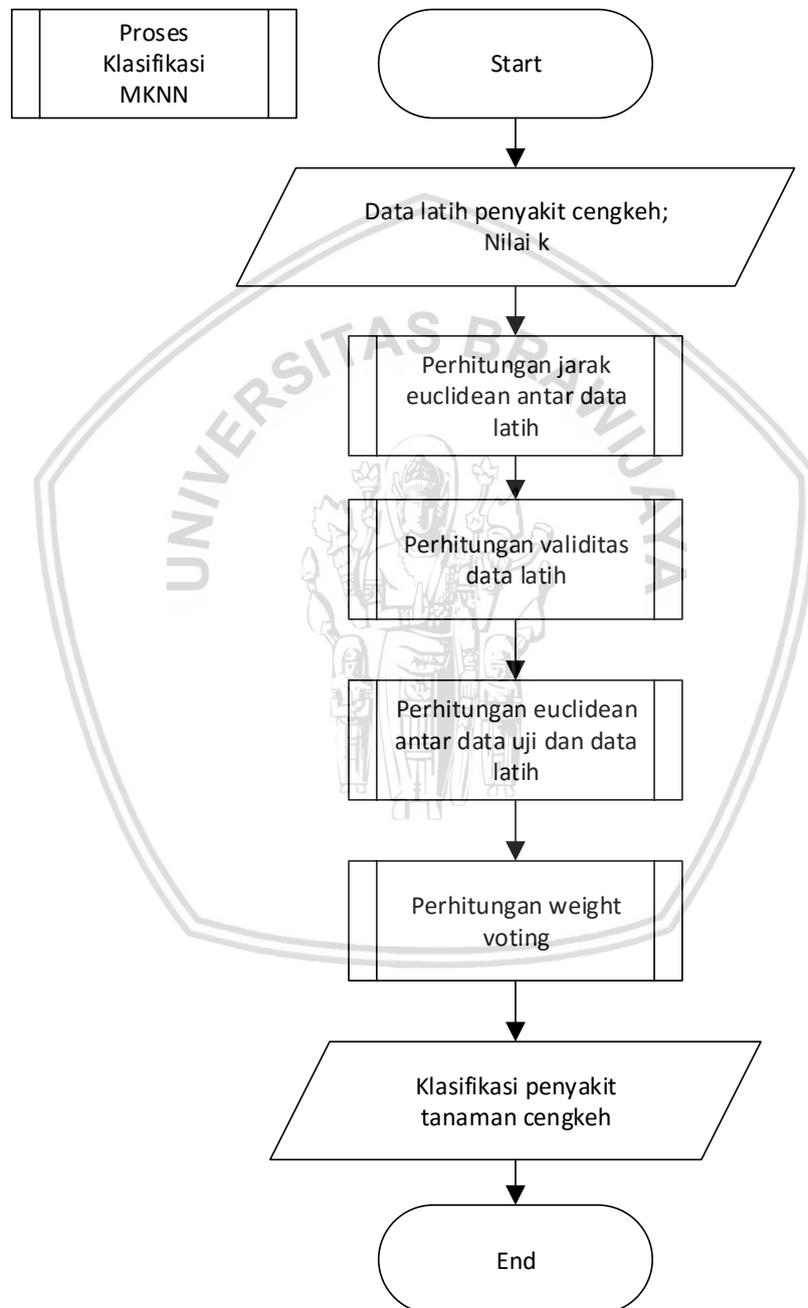


Gambar 4.2 Diagram alir sistem

4.3.1.1 Proses Klasifikasi Modified K-Nearest Neighbor (MKNN)

Proses klasifikasi bertujuan untuk mendapatkan hasil akhir diagnosis dari data yang diinputkan melalui berbagai macam gejala, inputan tadi akan di proses melalui bermacam proses dimulai dari menginputkan nilai k, kemudian data latih menghitung jarak *Euclidean* dari setiap data latih, setelah itu data akan diurutkan dari data terkecil. Setelah itu data terkecil akan diambil sebanyak nilai k kemudian

dilakukan proses validitas, proses validitas akan menentukan apakah data kelas dari nilai k tersebut sama dengan kelas data pada data latih, apabila sama maka bernilai 1 jika tidak maka 0. Setelah itu nilai k akan dijumlah dengan nilai validitas seperti pada persamaan (3). tahap selanjut akan melalui proses *Weight Voting* dan didapatkan hasil klasifikasi mknn, proses klasifikasi ditunjukkan pada Gambar 4.3 Diagram Alir Proses Klasifikasi MKNN.

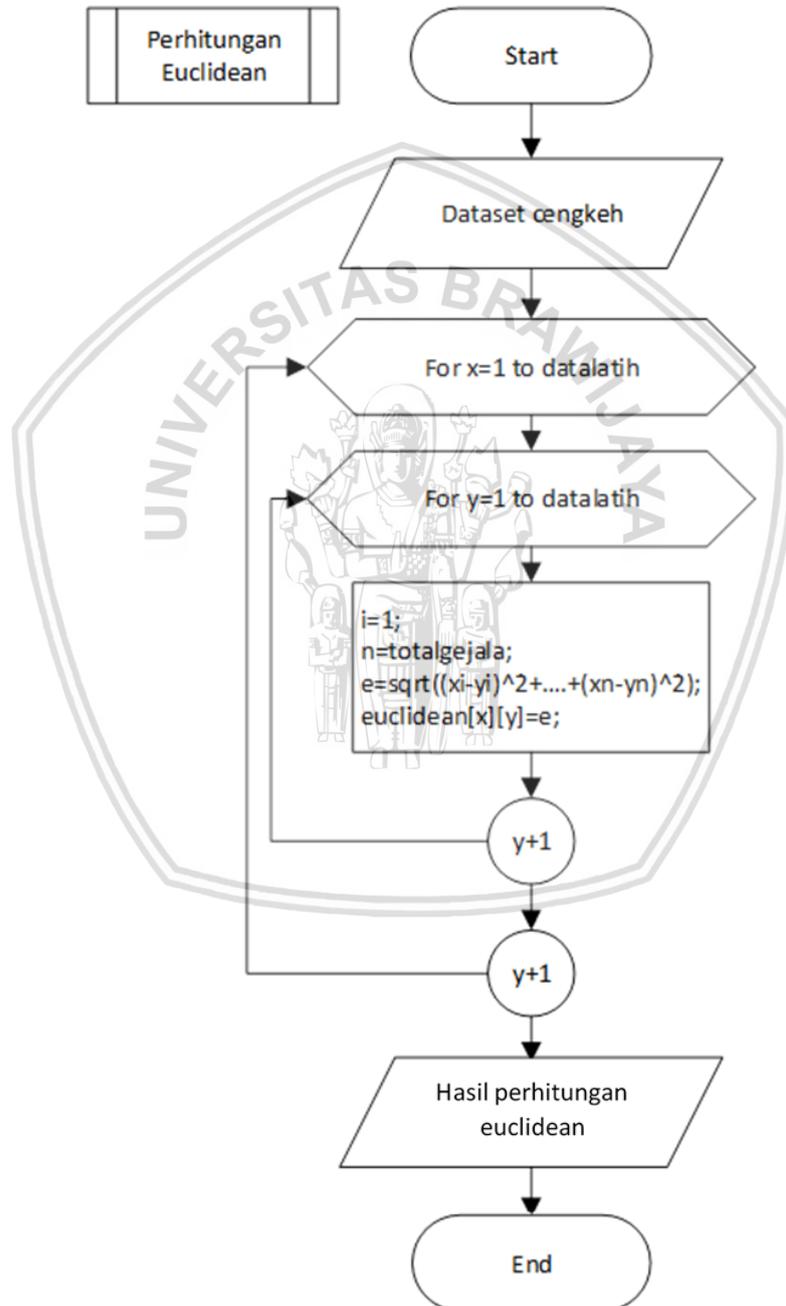


Gambar 4.3 Diagram Alir Proses Klasifikasi MKNN



4.3.1.2 Proses perhitungan jarak Euclidean

Perhitungan untuk jarak *Euclidean* akan dilakukan sebanyak 2 kali, yang pertama pada awal setelah input gejala dan nilai k, yang di hitung *Euclidean*nya adalah antar data latih, dan kedua setelah perhitungan validitas data latih, pada perhitungan *Euclidean* yang ke 2 adalah data uji dihitung dengan data latih. Kemudian untuk proses penghitungan jarak *Euclidean* dapat dilihat pada Gambar 4.4 Diagram Alir Perhitungan Jarak *Euclidean*.

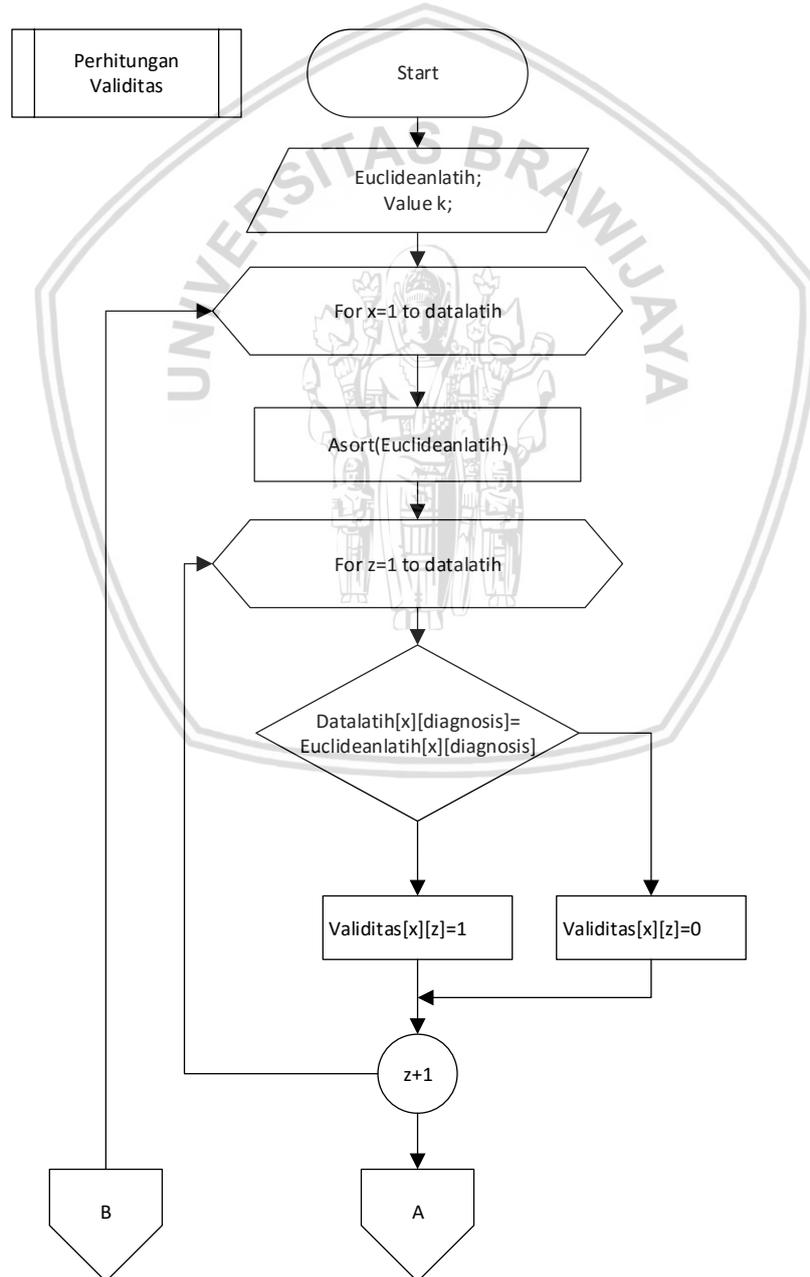


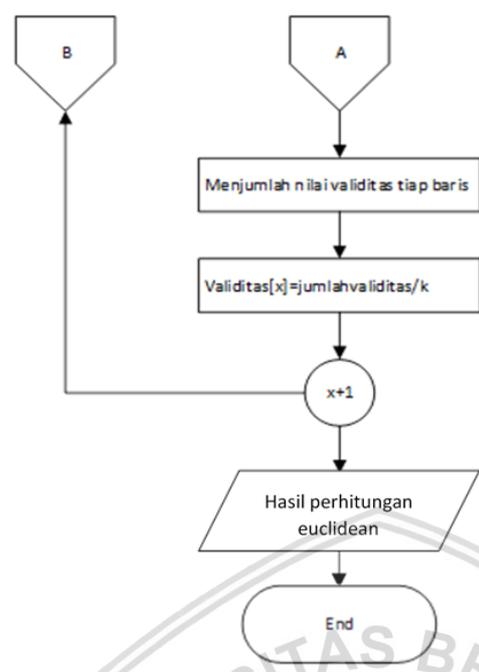
Gambar 4.4 Diagram Alir Perhitungan Jarak *Euclidean*



4.3.1.3 Proses perhitungan validitas

Tahap ini berguna untuk mencari nilai validitas dari data latih dengan menggunakan persamaan (2.2). Proses perhitungan Validitas data latih bertujuan untuk menghasilkan nilai kesamaan antara data latih terhadap tetangga terdekat dari data latih itu berdasarkan nilai dari k. Dilakukan setelah didapat jarak *Euclidean* kemudian di urutkan berdasarkan terkecil yang berarti terdekat sebanyak k, kemudian akan dibandingkan kesamaan dari kelas dari data latih dan data uji, apabila kelasnya sama maka nilainya 1, sedangkan jika tidak sama maka nilainya 0, setelah itu seluruh nilai kesamaan tadi akan di jumlah kemudian dibagi dengan k. Didapatkan nilai validitas, untuk proses validitas dapat dilihat pada diagram 4.5 Diagram Alir Perhitungan Validitas.

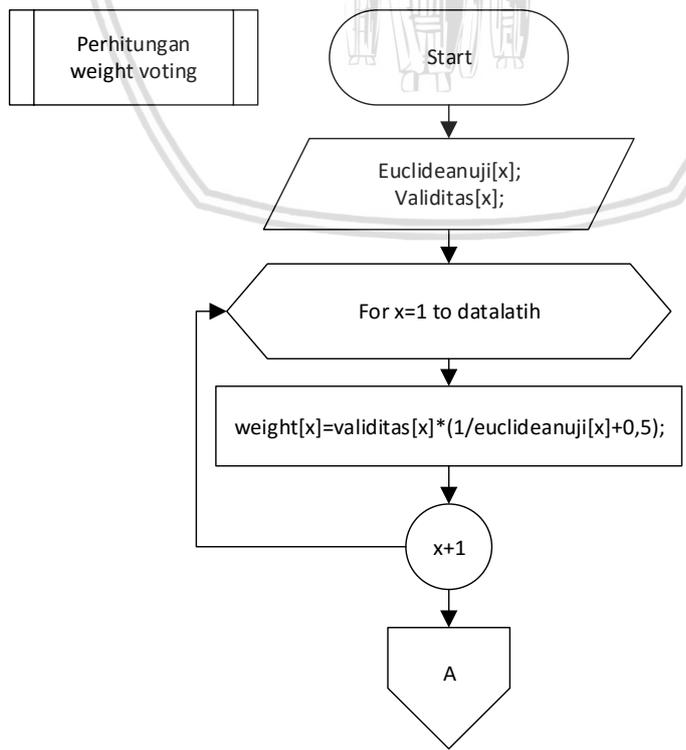


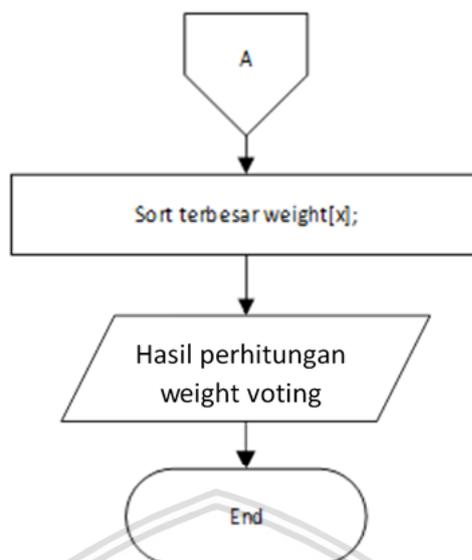


Gambar 4.5 Diagram Alir Perhitungan Validitas

4.3.1.4 Proses Perhitungan Weight Voting

Pada tahap perhitungan *Weight Voting* atau pembobotan berguna untuk mendapatkan hasil akhir dari klasifikasi *Modified K-NN* (MKNN). Klasifikasi didapat dengan mencari *Weight Voting* terbesar dengan menggunakan persamaan (2.4), nilai validitas dan nilai euclidean data latih dan data uji digunakan untuk mendapatkan hasil akhir dari sistem. Proses perhitungan *Weight Voting* dijelaskan pada diagram 4.6 Diagram Alir Perhitungan *Weight Voting*.





Gambar 4.6 Diagram Alir Perhitungan *Weight Voting*

4.4 Perhitungan Manual

Sistem diagnosis yang digunakan untuk mendiagnosis penyakit tanam cengkeh menggunakan algoritme *Modified K-Nearest Neighbor* adapun langkah-langkahnya adalah sebagai berikut:

1. Menentukan nilai k
Untuk perhitungan manual ini nilai k yang digunakan adalah 3.
2. Menghitung jarak *Euclidean* antar data latih
Tahap ini menghitung jarak terdekat terhadap masing masing data latih.
3. Menghitung nilai validitas
Nilai validitas didapatkan dengan membandingkan apakah kelas data uji sama dengan data latih, dilakukan setelah didapatkan jarak *Euclidean*.
4. Menghitung *Euclidean* antar data latih dan data uji
Menghitung *Euclidean* tahap kedua, antara data latih dengan data uji.
5. Menghitung *Weight Voting*
Pada perhitungan *Weight Voting* menggunakan nilai dari validitas, jarak *Euclidean* data uji dan nilai *error* sebesar 0,5.
6. Menentukan kelas data uji
Setelah itu untuk mendapatkan hasil diagnosis dicari nilai terbesar dari perhitungan *Weight Voting*.

Untuk melakukan manualisasi pada penelitian ini diambil data sebanyak 9 dari total dataset 61 dan 1 data uji dari dataset. Daftar data latih dapat dilihat pada Tabel 4.2.

Tabel 4.2 Daftar Data Latih

no	Gejala penyakit cengkeh											Penyakit
	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	
1	1	1	1	0	1	0	0	0	0	0	0	1
2	1	1	1	0	0	0	0	0	0	0	0	1
3	1	1	0	1	1	0	0	0	0	0	0	1
4	0	0	0	0	0	1	1	0	1	0	0	2
5	0	0	0	0	0	1	1	0	0	0	0	2
6	0	0	0	0	0	1	0	1	1	0	0	2
7	0	0	0	0	1	0	0	0	1	1	1	3
8	0	0	0	0	1	0	0	0	1	1	0	3
9	0	0	0	0	1	0	0	0	1	0	1	3

Keterangan
 1 = BPKC
 2 = Cacar Daun
 3 = Jamur Akar

Untuk data uji yang digunakan diambil dari dataset secara acak yang ditunjukkan pada Tabel 4.3.

Tabel 4.3 Data Uji

no	Gejala penyakit cengkeh											Penyakit
	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	
1	0	0	0	0	1	0	0	0	0	1	1	3

Keterangan

G1 = ranting pucuk mati	G7 = bekas melepuh ada titik hitam
G2 = daun gugur dari atas ke bawah	G8 = bercak cacar semakin sedikit
G3 = cabang mudah layu mendadak	G9 = daun layu, kering dan rontok
G4 = daun tua kekuning keuningan	G10 = benang jamur menempel akar
G5 = batang pohon mengering	G11 = akar membusuk
G6 = daun muda ada bagian melepuh	



Langkah 1: Menentukan nilai k

Pada manualisasi ini menggunakan k = 3.

Langkah 2: Menghitung jarak *Euclidean* antar data latih

Disini jarak terdekat dari data latih didapat dengan persamaan (2.1), untuk perhitungan dilakukan dengan membandingkan semua data pada data latih, hasil dari perhitungan ini akan di ranking berdasarkan nilai terkecil setelah itu akan dilanjutkan ke langkah berikutnya mencari nilai validitas.

a) proses perhitungan manual jarak *Euclidean* data 1 dan data 3

$$\begin{aligned}
 d(1,3) &= \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \\
 &= \sqrt{(1-1)^2 + (1-1)^2 + (1-0)^2 + (0-1)^2 + (1-1)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2} \\
 &= 1.414
 \end{aligned}$$

b) proses perhitungan manual jarak *Euclidean* data 3 dan data 5

$$\begin{aligned}
 d(3,5) &= \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \\
 &= \sqrt{(1-0)^2 + (1-0)^2 + (0-0)^2 + (1-0)^2 + (1-0)^2 + (0-1)^2 + (0-1)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2} \\
 &= 2.449
 \end{aligned}$$

hasil perhitungan jarak *Euclidean* antar data latih dapat dilihat pada Tabel 4.4.

Tabel 4.4 Hasil Perhitungan Jarak *Euclidean* Antar Data Latih

x \ y	1	2	3	4	5	6	7	8	9
1	0	1	1.4	2.6	2.4	2.6	2.4	2.2	2.2
2	1	0	1.7	2.4	2.2	2.4	2.6	2.4	2.4
3	1.4	1.7	0	2.6	2.4	2.6	2.4	2.2	2.2
4	2.6	2.4	2.6	0	1	1.4	2.2	2	2
5	2.4	2.2	2.4	1	0	1.7	2.4	2.2	2.2
6	2.6	2.4	2.6	1.4	1.7	0	2.2	2	2



7	2.4	2.6	2.4	2.2	2.4	2.2	0	1	1
8	2.2	2.4	2.2	2	2.2	2	1	0	1.4
9	2.2	2.4	2.2	2	2.2	2	1	1.4	0

Langkah 3: Menghitung nilai validitas

Untuk mendapatkan nilai validitas menggunakan hasil dari perhitungan jarak *Euclidean* dengan mengambil tetang terdekat sebanyak nilai k, karena k yang digunakan ada 3 maka dari Tabel 4.4 diambil 3 nilai terkecil dari setiap data latih, setelah itu ditentukan apakah masing masing 3 nilai terkecil itu merupakan kelas yang sama, jika kelas sama maka nilainya 1 jika tidak sama maka bernilai 0, menghitung validitas menggunakan persamaan (2.2).

a) proses menghitung nilai validitas data ke 1

$$\begin{aligned}
 \text{Validitas}(1) &= \frac{1}{3} \sum_{i=1}^3 S(\text{label}(1), (\text{label}(Ni(1)))) \\
 &= \frac{1}{3} (1 + 1 + 1) \\
 &= 1
 \end{aligned}$$

Nilai dari *label*(1) seperti Tabel 4.2 diketahui kelasnya atau penyakitnya adalah 1, kemudian untuk *label*(*Ni*(1)) adalah kelas dari data dengan jarak *Euclidean* terkecil ke-1 sampai ke-3, diketahui dari Tabel 4.4 nilai *Euclidean* terkecil ke-1 dari data ke 1 bernilai 0 didapat dari data 1, kemudian *Euclidean* terkecil ke-2 bernilai 1 didapat dari data ke 2 dan terkecil ke-3 bernilai 1.4 dari data ke 3. Kemudian di cari nilai S nya dengan membandingkan kelasnya *label*(1) dan *label*(*Ni*(1)), penyakit dari data 1 dan data 1 sama berarti nilai S pertama adalah 1, kemudian data 1 dan data 2 kelasnya sama berarti nilai S kedua adalah 1 dan terakhir data 1 dan data 3 juga sama, hasil S ketiga adalah 1. Kemudian semua nilai S akan di total kemudian dibagi dengan banyaknya nilai k. Hasil dari setiap perhitungan validitas data latih ditampilkan pada Tabel 4.5.

Tabel 4.5 Hasil Perhitungan Validitas

Data	k=1	k=2	k=3	Total validitas	validitas
1	1	1	1	3	1
2	1	1	1	3	1
3	1	1	1	3	1
4	1	1	1	3	1
5	1	1	1	3	1
6	1	1	1	3	1
7	1	1	1	3	1



8	1	1	1	3	1
9	1	1	1	3	1

Langkah 4: Menghitung jarak *Euclidean* antar data latih dan data uji

Proses ini sama seperti langkah 2 tetapi yang dihitung jarak *Euclidean*nya adalah data latih dan data uji, proses perhitungannya tetap menggunakan persamaan (2.1), dan dapat dilihat pada Tabel 4.6.

a) proses perhitungan manual jarak *Euclidean* data latih 1 dan data uji

$$\begin{aligned}
 d(1_{latih}, 1_{uji}) &= \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \\
 &= \sqrt{(1 - 0)^2 + (1 - 0)^2 + (1 - 0)^2 + (0 - 0)^2 + (1 - 1)^2 + (0 - 0)^2 + (0 - 0)^2 + (0 - 0)^2 + (0 - 1)^2 + (0 - 1)^2} \\
 &= 2,236
 \end{aligned}$$

Tabel 4.6 Jarak *Euclidean* Antara Data Uji dan Data Latih

d(a,b)	<i>Euclidean</i>
d(1 _{uji} ,1 _{latih})	2,236
d(1 _{uji} ,2 _{latih})	2,449
d(1 _{uji} ,3 _{latih})	2,236
d(1 _{uji} ,4 _{latih})	2,449
d(1 _{uji} ,5 _{latih})	2,236
d(1 _{uji} ,6 _{latih})	2,449
d(1 _{uji} ,7 _{latih})	1
d(1 _{uji} ,8 _{latih})	1,414
d(1 _{uji} ,9 _{latih})	1,414

Langkah 5: Menghitung *Weight Voting* dan menentukan kelas data uji

Proses *Weight Voting* atau pembobotan dilakukan menggunakan persamaan (2.4), dengan menggunakan nilai validasi dan jarak *Euclidean* data latih dan uji kemudian diambil hasil dengan nilai terbesar untuk menentukan hasil klasifikasi. Hasil perhitungan dapat dilihat pada Tabel 4.7.



a) perhitungan nilai *Weight Voting* data uji 1

$$W(1,1) = \text{Validity}(1) \times \frac{1}{de + 0,5} = 1 \times \frac{1}{2,236 + 0,5} = 0,3654$$

Tabel 4.7 Hasil Perhitungan *Weight Voting*

W(a,b)	<i>Weight Voting</i>	Diagnosis	Urutan
W(1 _{uji} ,1 _{latih})	0,3654	BPKC	4
W(1 _{uji} ,2 _{latih})	0,3390	BPKC	7
W(1 _{uji} ,3 _{latih})	0,3654	BPKC	5
W(1 _{uji} ,4 _{latih})	0,3390	Cacar Daun	8
W(1 _{uji} ,5 _{latih})	0,3654	Cacar Daun	6
W(1 _{uji} ,6 _{latih})	0,3390	Cacar Daun	9
W(1 _{uji} ,7 _{latih})	0,6666	Jamur Akar	1
W(1 _{uji} ,8 _{latih})	0,5224	Jamur Akar	2
W(1 _{uji} ,9 _{latih})	0,5224	Jamur Akar	3

Langkah 6: Menentukan kelas data uji

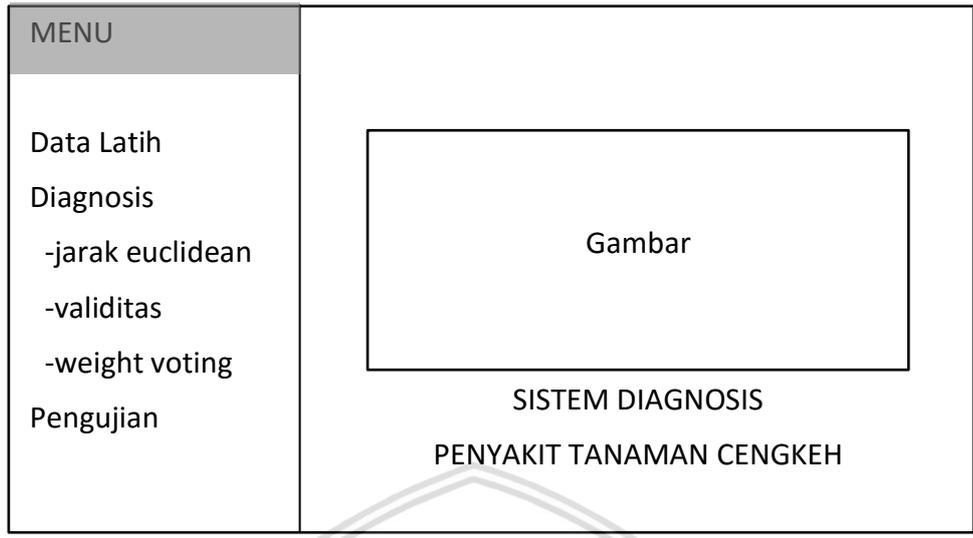
Untuk mendapatkan hasil diagnosis dari kelas data uji, didapatkan dari proses *Weight Voting* dengan mengambil hasil *Weight Voting* yang terbesar, dari perhitungan manual didapatkan nilai terbesar dari *Weight* data uji dan data latihan ke 7 dengan hasil 0,6666 maka didapatkan data uji diagnosis jamur akar, dengan kandidat 2 dan 3 juga jamur akar yang dapat dilihat Tabel 4.7 Hasil Perhitungan *Weight Voting*.

4.5 Perancangan Desain Antarmuka

Pada sistem diagnosis desain antarmuka sangat dibutuhkan sebagai perantara informasi dari sistem ke pengguna agar pengguna dapat menjalankan sistem dan melakukan diagnosis. Desain antarmuka harus mudah dimengerti dan dioperasikan, oleh karena itu proses desain antarmuka sistem diagnosis penyakit tanaman cengkeh akan menampilkan halaman utama, data latihan dan diagnosis.

4.5.1 Desain Antarmuka Halaman Utama

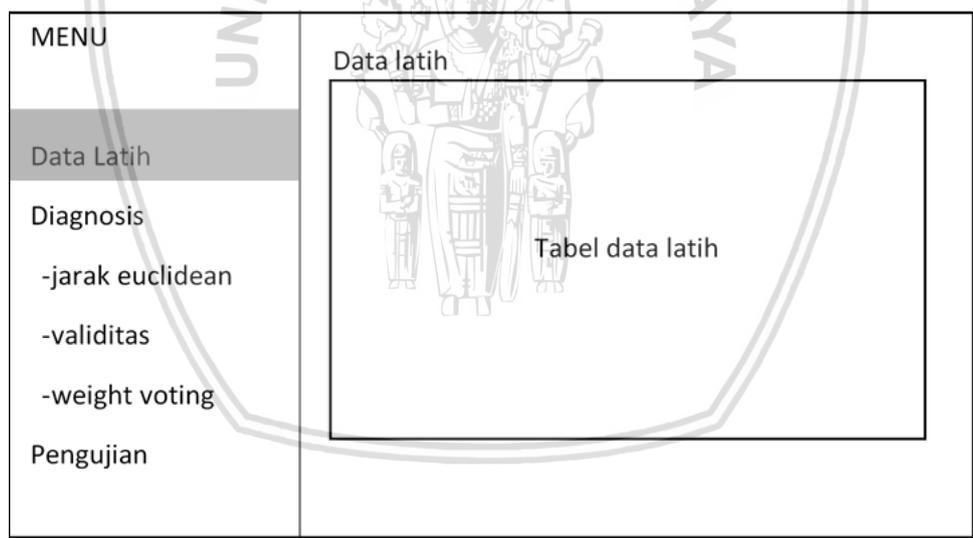
Pada halaman ini terdapat menu utama dari sistem yang diimplementasikan dengan menu yang ada di sebelah kiri, pengguna dapat menemukan menu yang tersedia pada sistem. Desain antarmuka halaman utama seperti pada Gambar 4.7.



Gambar 4.7 Perancangan Halaman Utama

4.5.2 Desain Antarmuka Halaman Data Latih

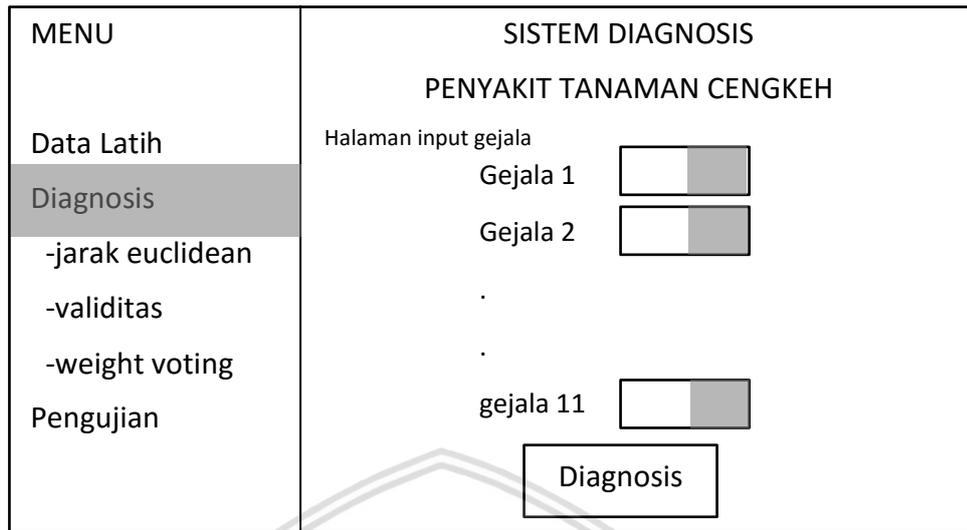
Pada halaman data latih akan digunakan untuk menampilkan semua data yang digunakan pada sistem. Desain antarmuka seperti pada Gambar 4.8.



Gambar 4.8 Perancangan Halaman Data Latih

4.5.3 Desain Antarmuka Halaman Diagnosis

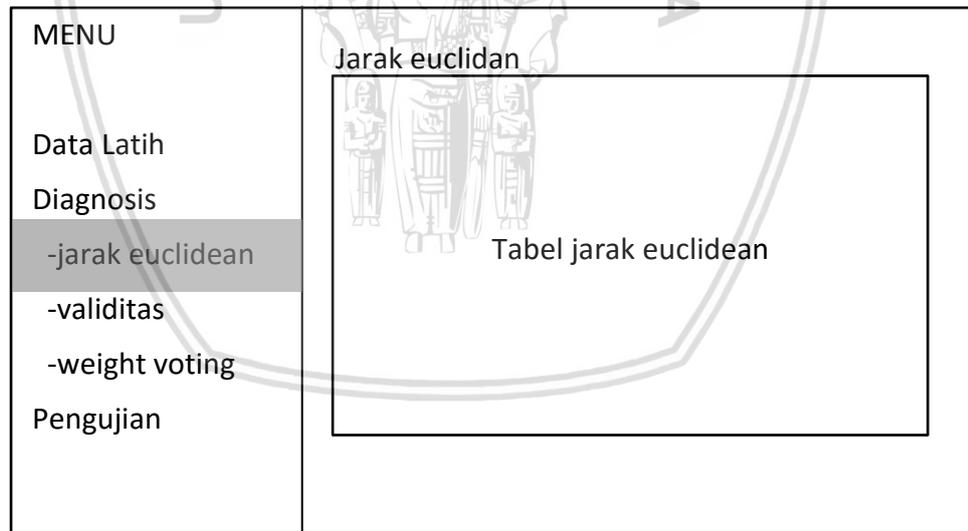
Pada halaman ini pengguna dapat memasukkan gejala gejala pada penyakit tanaman cengkeh, pada halaman diagnosis terdapat submenu antara lain: jarak *Euclidean*, validitas, *Weight Voting* dan hasil. Pengguna terlebih dahulu memasukkan gejala kemudian dapat melakukan proses diagnosis. Desain antarmuka pada halaman diagnosis seperti Gambar 4.9.



Gambar 4.9 Perancangan Halaman Diagnosis

4.5.4 Desain Antarmukan Halaman Jarak *Euclidean*

Halaman jarak *Euclidean* berisi tentang hasil perhitungan sistem tahap pertama, jarak *Euclidean* nanti akan digunakan untuk menghitung nilai dari validitas. Desain antarmuka halaman jarak *Euclidean* seperti pada 4.10.

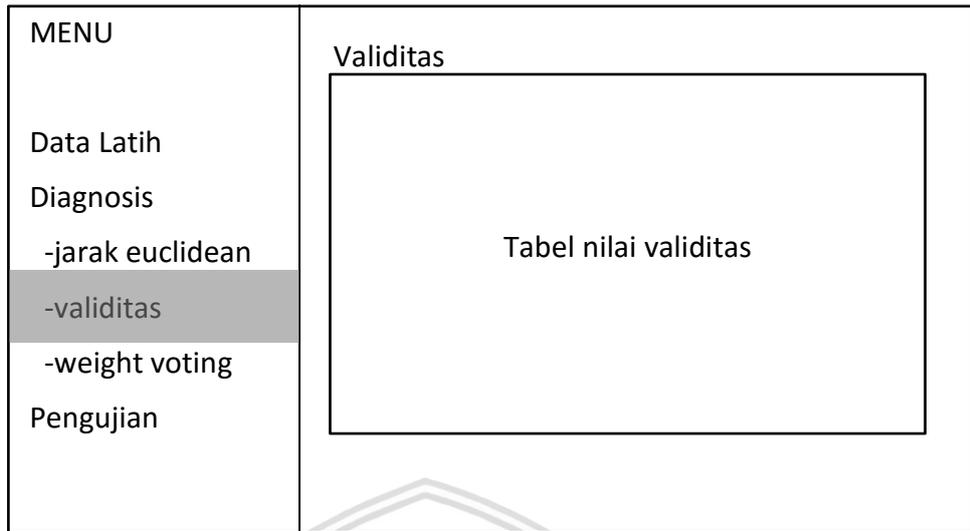


Gambar 4.10 Perancangan Halaman Jarak Euclidean

4.5.5 Desain Antarmuka Halaman Validitas

Halaman validitas berisi hasil perhitungan tahap kedua dari sistem, halaman ini menampilkan nilai validitas dari setiap data latih. Desain antarmuka halaman validitas dilihat pada Gambar 4.11.

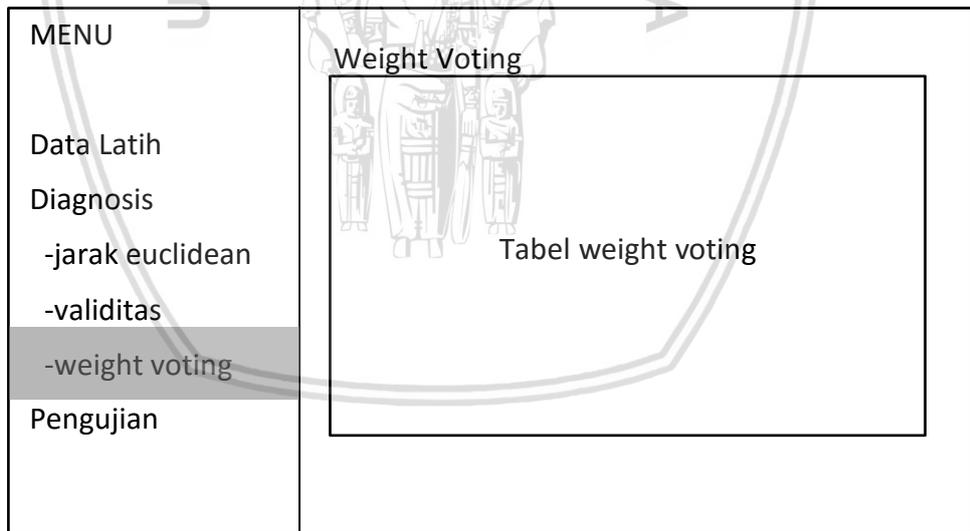




Gambar 4.11 Perancangan Halaman Validitas

4.5.6 Desain Antarmuka Halaman *Weight Voting*

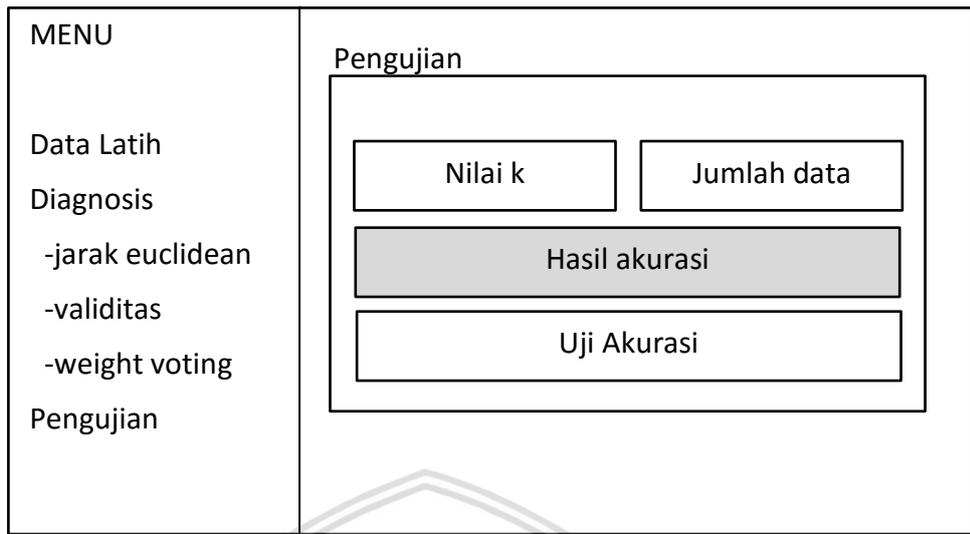
Halaman ini berisi hasil perhitungan dari sistem pada tahap akhir, dengan menampilkan urutan *Weight Voting* dari nilai terbesar yang merupakan hasil akhir dari sistem. Desain antarmuka *Weight Voting* seperti pada Gambar 4.12.



Gambar 4.12 Perancangan Halaman *Weight Voting*

4.5.7 Desain Antarmuka Halaman Pengujian

Halaman Pengujian berisi masukan dari sistem untuk melakukan pengujian akurasi, dengan memilih nilai k dan banyaknya data latih dapat dihasilkan berapa besar akurasi dari sistem. Desain antarmuka pengujian dilihat pada Gambar 4.13.



Gambar 4.13 Perancangan Halaman Pengujian

4.6 Perancangan Pengujian

Pada tahap pengujian data yang digunakan berjumlah 61 data yang berisi gejala penyakit cengkeh yang nantinya akan diambil secara acak berdasarkan berapa jumlah data yang akan digunakan pada sistem. Pengujian yang digunakan pada penelitian ini yakni pengujian akurasi berdasarkan jumlah dari data uji dan kombinasi nilai k. Perancangan yang digunakan seperti pada Tabel 4.8.

Tabel 4.8 Perancangan Pengujian

k	Jumlah data		
	20	40	60
2			
3			
4			
5			
6			
7			



BAB 5 IMPLEMENTASI SISTEM

Pada bab ini akan menjelaskan mengenai implementasi dari sistem yang dibangun serta hasil implementasi dari antarmuka sistem diagnosis penyakit tanaman cengkeh yang dibuat menggunakan algoritme Modified *K-Nearest Neighbor*.

5.1 Implementasi Program

Dari perancangan sistem yang sudah dibuat pada bab 4, maka dibuat implementasi program dari perancangan tersebut, untuk perancangan sistem yang dibangun menggunakan Bahasa pemrograman PHP atau berbasis web.

5.1.1 Implementasi Proses *Euclidean Distance*

Pada proses *Euclidean distance* dilakukan 2 kali, yang pertama antar data latih dengan data latih dan yang kedua antara data latih dengan data uji, nilai *Euclidean distance* ini didapat dengan menggunakan persamaan (2.1), nanti nilai *Euclidean* ini akan dipakai untuk melakukan pembobotan atau *Weight Voting* yang dilakukan pada tahap akhir. Implementasi dari proses *Euclidean distance* ini seperti pada Tabel 5.1.

Tabel 5.1 Baris Kode Proses *Euclidean Distance*

```

34 //mencari Euclidean antar data latih
35 for($i=1;$i<=61;$i++){
36     for($j=1;$j<=61;$j++){
37         //ambil data-1
38         $data['x']=$this->m_data->get_baris($i)-
>result_object();
39         //ambil data-2
40         $data['y']=$this->m_data->get_baris($j)-
>result_object();
41         $data['euc'][0]=$i;
42         //persamaan 2.1
43         $data['euc'][$j]=sqrt(
44     pow(($data['x'][0]->G1)-$data['y'][0]->G1,2)+
45     pow(($data['x'][0]->G2)-$data['y'][0]->G2,2)+
46     pow(($data['x'][0]->G3)-$data['y'][0]->G3,2)+
47     pow(($data['x'][0]->G4)-$data['y'][0]->G4,2)+
48     pow(($data['x'][0]->G5)-$data['y'][0]->G5,2)+
49     pow(($data['x'][0]->G6)-$data['y'][0]->G6,2)+
50     pow(($data['x'][0]->G7)-$data['y'][0]->G7,2)+
51     pow(($data['x'][0]->G8)-$data['y'][0]->G8,2)+
52     pow(($data['x'][0]->G9)-$data['y'][0]->G9,2)+
53     pow(($data['x'][0]->G10)-$data['y'][0]->G10,2)+
54     pow(($data['x'][0]->G11)-$data['y'][0]->G11,2)
55     );
56     }
57 //input data Euclidean ke db
58 $this->m_data->update_e('db_Euclidean_dl',$data['euc'],$i);
59 }
    
```

Penjelasan untuk kode program adalah sebagai berikut:



- Baris 35 & 36 : melakukan perulangan sebanyak data latih
 Baris 38 : mengambil isi dari data pertama
 Baris 40 : mengambil isi dari data kedua
 Baris 41 : mengisi array dengan No untuk id
 Baris 43 : mengisi array ke \$j dengan hasil perhitungan
 Baris 44 – 54 : persamaan 2, melakukan proses *Euclidean distance*
 Baris 58 : memasukkan data hasil perhitungan ke database

5.1.2 Implementasi Proses Validitas

Nilai dari validitas akan digunakan pada proses *Weight Voting*, nilai validitas didapat dengan membandingkan kelas dari nilai *Euclidean* terkecil atau tetangga terdekatnya dengan data latih, baris kode program untuk proses validitas ada pada Tabel 5.2.

Tabel 5.2 Baris Kode Proses Validitas

```

61 //menghitung validitas data latih
62 $data['nilaik']=$this->m_data->get_db()->result();
63 $db="db_Euclidean_dl";
64 $data['baris']=$this->m_data->count();
65 //validasi
66 $this->m_data->hapus('db_validitas');
67 for($x=1;$x<=$data['baris'];$x++){
68     $data['val'][11]="0";
69     $data['tes']="E'.'.$x;
70     $data['rank']=$this->m_data-
>get_kolom($db,$data['tes'],$data['nilaik'][0]->k)-
71 >result();
72     //ambil data pertama
73     $data['a']=$this->m_data->get_penyakit($x)->row();
74     for($y=1;$y<=$data['nilaik'][0]->k;$y++){
75         //ambil data kedua
76         $data['b']=$this->m_data-
>get_penyakit($data['rank'][$y-1]->No)->row();
77         //kelas dari data akan di bandingkan
78         if($data['a']->Penyakit==$data['b']->Penyakit){
79             //kelas sama maka 1
80             $data['val'][$y]="1";
81             $data['val'][11]+=1;
82         }
83         else{
84             //kelas tidak sama maka 0
85             $data['val'][$y]="0";
86         }
87     }
88     //isi total validitas
89     $data['val'][12]=$data['val'][11]/$data['nilaik'][0]-
>k;
90     //input data kedalam database
91     $this->m_data->update_v($data['val'],$x);
}

```

Penjelasan untuk kode program adalah sebagai berikut:

Baris 62	: mengambil nilai k
Baris 67	: perulangan untuk menghitung 1 nilai validitas
Baris 70	: mengambil nilai terkecil pada <i>Euclidean</i> sebanyak k
Baris 72	: mengambil kelas dari data latih
Baris 73	: perulangan sebanyak k, untuk mengisi total validitas
Baris 75	: mengambil kelas dari data <i>Euclidean</i>
Baris 77	: jika kelas kedua data tersebut sama maka
Baris 79	: validitas bernilai 1
Baris 80	: menghitung total validitas
Baris 82	: jika kelas kedua data tidak sama maka
Baris 84	: validitas bernilai 0
Baris 88	: hasil akhir validitas dengan membagi total validitas dengan nilai k
Baris 90	: menyimpan data validitas kedalam database

5.1.3 Implementasi Proses *Weight Voting*

Proses *Weight Voting* merupakan proses akhir dari sistem diagnosis ini dengan menggunakan nilai validitas dan nilai *Euclidean* data uji untuk melakukan perhitungan dengan menggunakan persamaan (2.4). hasil akhir adalah data yang memiliki *Weight Voting* terbesar, adapun implementasinya ada pada Tabel 5.3.

Tabel 5.3 Baris Kode *Weight Voting*

```

157 for($a=1;$a<=61;$a++){
158     $data['validitas']=$this->m_data->get_val($a)->row();
159     $data['Euclidean']=$this->m_data->get_eduno($a)-
160 >row();
161     $data['penyakit']=$this->m_data->get_penyakit($a)-
162 >row();
163     $data['hasil']=(($data['validitas']-
164 >Validitas*(1/($data['Euclidean']->Eu+0.5)));
165     $this->m_data-
166 >input_wv($data['hasil'],$data['penyakit']->Penyakit,$a);
167 };

```

Penjelasan untuk baris kode adalah sebagai berikut:

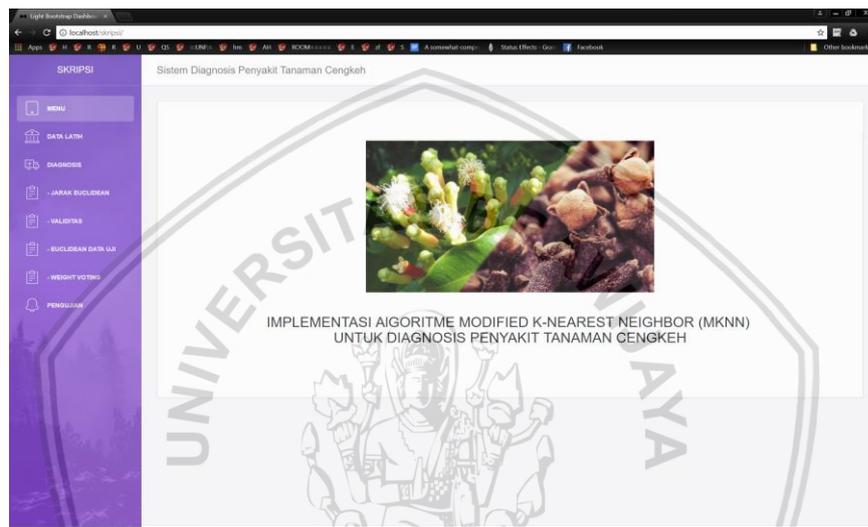
Baris 157	: perulangan untuk menghitung <i>Weight Voting</i>
Baris 158	: mengambil nilai validitas
Baris 159	: mengambil nilai <i>Euclidean</i>
Baris 160	: mengambil nama penyakit
Baris 161	: perhitungan persamaan 2.4 menghitung <i>Weight Voting</i>
Baris 162	: menyimpan hasil <i>Weight Voting</i> kedalam database

5.2 Implementasi Antarmuka

Antarmuka sistem berisi informasi mengenai proses tertentu pada tahap diagnosis, antarmuka sistem antara lain halaman utama, halaman data latih, halaman diagnosis, halaman jarak *Euclidean*, halaman validitas, halaman *Euclidean* data uji, halaman *Weight Voting* dan halaman pengujian. Seperti yang sudah dilampirkan di bab 4.

5.2.1 Halaman Utama

Halaman utama adalah halaman awal dari sistem yang dibuat. Tampilan halaman utama dapat dilihat pada Gambar 5.1.



Gambar 5.1 Tampilan Halaman Utama

5.2.2 Halaman Data Latih

Halaman data latih memberikan informasi mengenai data latih yang digunakan pada sistem diagnosis, terdapat nilai dari gejala masing-masing data dan penyakitnya. Untuk antarmuka dari halaman data seperti Gambar 5.2.

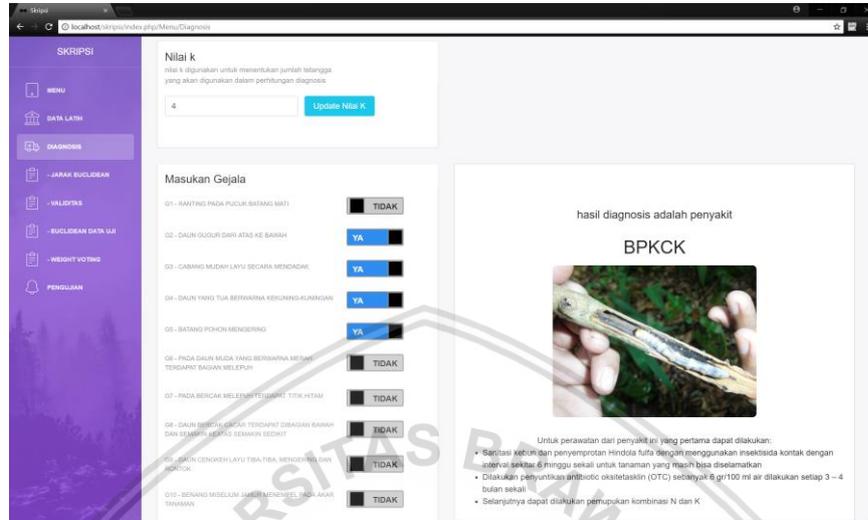
NO	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	PENYAKIT
1	1	1	1	1	1	0	0	0	0	0	0	BPCKK
2	1	1	1	1	0	0	0	0	0	0	0	BPCKK
3	1	1	1	0	1	0	0	0	0	0	0	BPCKK
4	1	1	1	0	0	0	0	0	0	0	0	BPCKK
5	1	1	0	1	1	0	0	0	0	0	0	BPCKK
6	1	1	0	1	0	0	0	0	0	0	0	BPCKK
7	1	1	0	0	1	0	0	0	0	0	0	BPCKK
8	1	1	0	0	0	0	0	0	0	0	0	BPCKK
9	1	0	1	1	1	0	0	0	0	0	0	BPCKK
10	1	0	1	1	0	0	0	0	0	0	0	BPCKK
11	1	0	1	0	1	0	0	0	0	0	0	BPCKK
12	1	0	1	0	0	0	0	0	0	0	0	BPCKK
13	1	0	0	1	1	0	0	0	0	0	0	BPCKK
14	1	0	0	1	0	0	0	0	0	0	0	BPCKK
15	1	0	0	0	1	0	0	0	0	0	0	BPCKK

Gambar 5.2 Tampilan Halaman Data Latih



5.2.3 Halaman Diagnosis

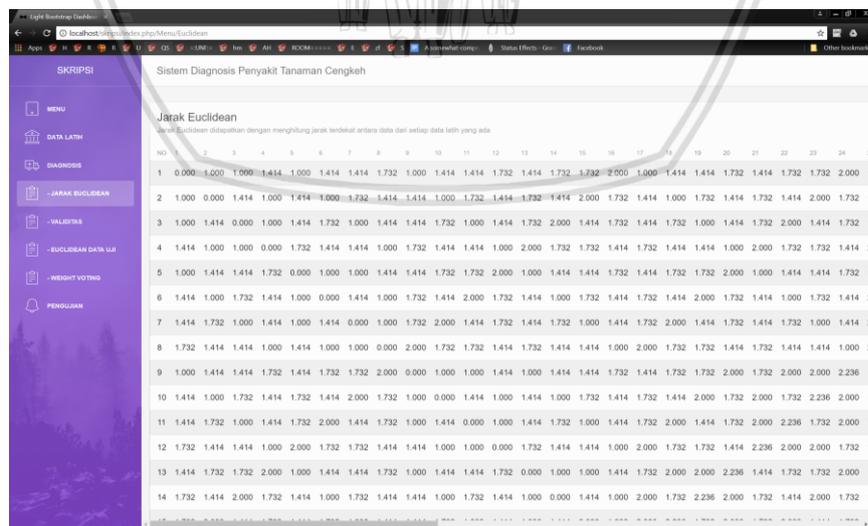
Halaman diagnosis terdapat input nilai k, input gejala, dan hasil diagnosis dari penyakit tanaman cengkeh, antarmuka halaman untuk proses diagnosis seperti pada Gambar 5.3.



Gambar 5.3 Tampilan Halaman Diagnosis

5.2.4 Halaman Euclidean

Untuk proses perhitungan dari sistem diagnosis ini dapat dilihat dengan mengakses halaman selanjutnya pada menu, ada jarak *Euclidean*, validitas, *Euclidean* data uji serta *Weight Voting*, halaman tersebut memberikan informasi mengenai bagaimana perhitungan pada sistem dilakukan. Untuk halaman jarak *Euclidean* dapat dilihat pada Gambar 5.4



Gambar 5.4 Tampilan Halaman Euclidean

5.2.5 Halaman Validitas

Selanjutnya pada proses diagnosis adalah menghitung validitas, hasil perhitungan dari validitas dapat dilihat pada antarmuka validitas pada Gambar 5.5.

NO	K1	K2	K3	K4	K5	K6	K7	K8	K9	K10	TOTAL	VALIDITAS
1	1	1	0	0	0	0	0	0	0	0	2	1
2	1	1	0	0	0	0	0	0	0	0	2	1
3	1	1	0	0	0	0	0	0	0	0	2	1
4	1	1	0	0	0	0	0	0	0	0	2	1
5	1	1	0	0	0	0	0	0	0	0	2	1
6	1	1	0	0	0	0	0	0	0	0	2	1
7	1	1	0	0	0	0	0	0	0	0	2	1
8	1	1	0	0	0	0	0	0	0	0	2	1
9	1	1	0	0	0	0	0	0	0	0	2	1
10	1	1	0	0	0	0	0	0	0	0	2	1
11	1	1	0	0	0	0	0	0	0	0	2	1
12	1	1	0	0	0	0	0	0	0	0	2	1
13	1	1	0	0	0	0	0	0	0	0	2	1
14	1	1	0	0	0	0	0	0	0	0	2	1
15	1	1	0	0	0	0	0	0	0	0	2	1

Gambar 5.5 Tampilan Halaman Validitas

5.2.6 Halaman Euclidean Uji

Setelah mendapatkan nilai dari masukan gejala, maka untuk proses selanjutnya dari sistem adalah menghitung jarak *Euclidean* antar data uji dan data latih yang hasilnya dapat dilihat pada halaman *Euclidean* data uji seperti yang ditunjukkan Gambar 5.6.

NO	G1
31	1
45	1
54	1
15	1.4142135623731
23	1.4142135623731
27	1.4142135623731
29	1.4142135623731
37	1.4142135623731
41	1.4142135623731
44	1.4142135623731
51	1.4142135623731
52	1.4142135623731
53	1.4142135623731
7	1.7320508075689
11	1.7320508075689

Gambar 5.6 Tampilan Halaman Euclidean Uji

5.2.7 Halaman *Weight Voting*

Hasil dari diagnosis didapat dari nilai *Weight Voting* yang terbesar, setelah dilakukan perhitungan akan didapat hasilnya, untuk melihat *Weight Voting* dari tiap data dapat dilihat pada halaman *Weight Voting* seperti Gambar 5.7.

URUTAN	NO	WEIGHT	PENYAKIT
1	45	0.66666666666667	Cacar Daun
2	15	0.52240774992748	BPXCK
3	23	0.52240774992748	BPXCK
4	27	0.52240774992748	BPXCK
5	29	0.52240774992748	BPXCK
6	37	0.52240774992748	Cacar Daun
7	41	0.52240774992748	Cacar Daun
8	44	0.52240774992748	Cacar Daun
9	52	0.52240774992748	Jamur Akar
10	53	0.52240774992748	Jamur Akar
11	7	0.44801847547959	BPXCK
12	11	0.44801847547959	BPXCK
13	13	0.44801847547959	BPXCK
14	16	0.44801847547959	BPXCK
15	19	0.44801847547959	BPXCK

Gambar 5.7 Tampilan Halaman *Weight Voting*

5.2.8 Halaman Pengujian

Selanjutnya untuk mendapatkan hasil pengujian dari sistem dapat mengakses halaman pengujian seperti pada Gambar 5.8.

Gambar 5.8 Tampilan Halaman Pengujian



BAB 6 PENGUJIAN DAN ANALISIS

Pada Bab ini akan menjelaskan mengenai pengujian serta analisis dari sistem yang sudah dibangun. Pada proses pengujian yang dilakukan ini merujuk kepada perancangan pengujian di bab 4.

6.1 Pengujian

Pengujian dilakukan dengan menggunakan kombinasi nilai k dan kombinasi dari jumlah data latih yang digunakan seperti yang dijelaskan pada bab 4, akurasi didapat dengan cara membandingkan hasil diagnosis dari sistem dengan data dengan diagnosis yang sudah ada.

6.2 Hasil Pengujian

Hasil Pengujian dari sistem dibagi menjadi 3 berdasarkan jumlah data yang digunakan pada masing masing pengujian. Adapun hasil dari pengujian yang dilakukan sebagai berikut.

6.2.1 Pengujian 20 Data Latih

Data yang digunakan untuk 20 data latih seperti pada Tabel 6.1.

Tabel 6.1 Hasil Diagnosis 20 Data

Data ke	Data Latih	Hasil Diagnosis
1	BPKC	BPKC
2	BPKC	BPKC
3	BPKC	BPKC
4	BPKC	BPKC
5	BPKC	BPKC
6	BPKC	BPKC
7	Cacar Daun	Jamur Akar
8	Cacar Daun	Cacar Daun
9	Cacar Daun	Cacar Daun
10	Cacar Daun	Cacar Daun
11	Cacar Daun	Cacar Daun
12	Cacar Daun	Cacar Daun
13	Cacar Daun	Cacar Daun
14	Jamur Akar	BPKC
15	Jamur Akar	Jamur Akar

16	Jamur Akar	Jamur Akar
17	Jamur Akar	Jamur Akar
18	Jamur Akar	Jamur Akar
19	Jamur Akar	Jamur Akar
20	Jamur Akar	BPKC

Nilai k yang digunakan pada Tabel adalah 2. Dari data yang dilakukan diagnosis 17 dari 20 data adalah benar, dan 3 dari data tersebut tidak sesuai dengan data pada data latih. Sehingga akurasi dari $k=2$ dengan 20 data adalah 85%. Untuk nilai k yang lainnya dapat dilihat pada Tabel 6.2.

Tabel 6.2 Hasil Akurasi 20 Data

Nilai k	Akurasi
2	85%
3	90%
4	90%
5	90%
6	90%
7	90%

6.2.2 Pengujian 40 Data Latih

Data yang digunakan untuk 40 data latih seperti pada Tabel 6.3.

Tabel 6.3 Hasil Diagnosis 40 Data

Data ke	Data Latih	Hasil Diagnosis
1	BPKC	BPKC
2	BPKC	BPKC
3	BPKC	BPKC
4	BPKC	BPKC
5	BPKC	BPKC
6	BPKC	BPKC
7	BPKC	BPKC
..
13	BPKC	BPKC

14	BPKC	Jamur Akar
15	Cacar Daun	Cacar Daun
..
24	Cacar Daun	Cacar Daun
25	Cacar Daun	Jamur Akar
26	Cacar Daun	Cacar Daun
27	Cacar Daun	Cacar Daun
..
38	Jamur Akar	Jamur Akar
39	Jamur Akar	Jamur Akar
40	Jamur Akar	Jamur Akar

Diagnosis dari sistem menghasilkan 38 data benar dari total 40 data latih, 2 data yang tidak sesuai yaitu data ke-14 dan data ke-25. Oleh karena itu menggunakan persamaan (2.5) untuk mengukur akurasi didapat akurasi untuk 40 data latih sebesar 95%. Untuk nilai k yang lainnya dapat dilihat pada Tabel 6.4.

Tabel 6.4 Hasil Akurasi 40 data

Nilai k	Akurasi
2	95%
3	95%
4	95%
5	95%
6	95%
7	95%

6.2.3 Pengujian 60 Data Latih

Data yang digunakan untuk 60 data latih seperti pada Tabel 6.5.

Tabel 6.5 Hasil Diagnosis 60 Data

Data ke	Data Latih	Hasil Diagnosis
1	BPKC	BPKC
2	BPKC	BPKC
3	BPKC	BPKC
4	BPKC	BPKC



5	BPKC	BPKC
6	BPKC	BPKC
7	BPKC	BPKC
..
30	BPKC	BPKC
31	BPKC	Jamur Akar
32	Cacar Daun	Cacar Daun
33	Cacar Daun	Cacar Daun
34	Cacar Daun	Cacar Daun
35	Cacar Daun	Cacar Daun
36	Cacar Daun	Cacar Daun
..
57	Jamur Akar	Jamur Akar
58	Jamur Akar	Cacar Daun
59	Jamur Akar	Jamur Akar
60	Jamur Akar	Jamur Akar

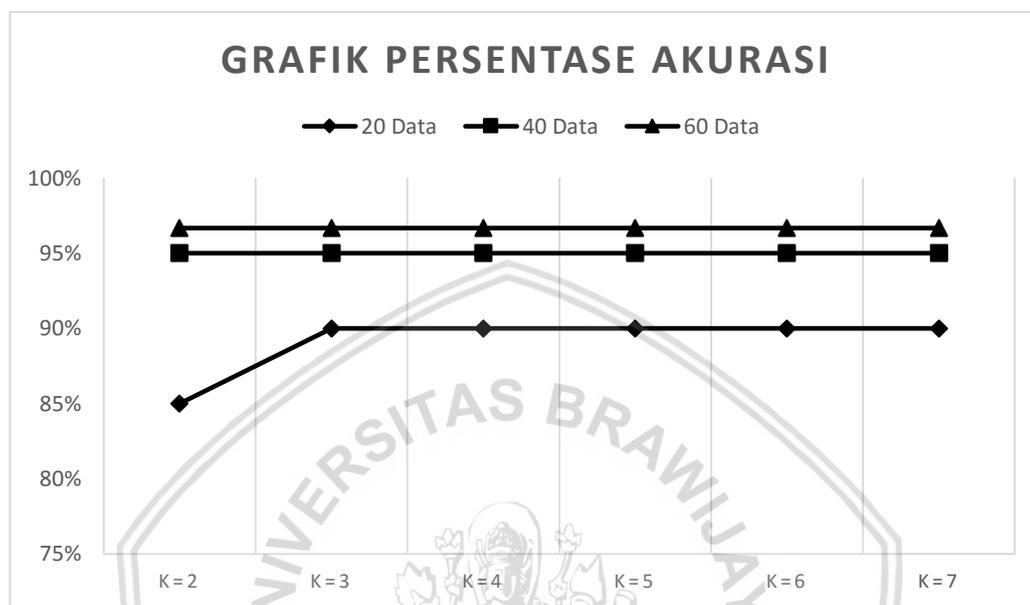
Dengan menggunakan data latih sebanyak 60 diagnosis dari sistem memberikan hasil sebanyak 58 data yang sesuai dengan data latih dan 2 dari data tersebut tidak sesuai yaitu data ke-31 dan data ke-58. Maka didapat akurasi untuk data latih sebanyak 60 dan $k=2$ adalah 96.67%. hasil dari nilai k yang lain dilihat pada Tabel 6.6.

Tabel 6.6 Hasil Akurasi 60 Data

Nilai k	Akurasi
2	96.67%
3	96.67%
4	96.67%
5	96.67%
6	96.67%
7	96.67%

6.3 Analisis Pengujian

Berdasarkan dari pengujian yang sudah dilakukan dengan menggunakan nilai k dan data latih yang berbeda beda dapat dikatakan bahwa jumlah dari data mempengaruhi nilai dari akurasi dari sistem. Pada Gambar 6.1 adalah hasil dari semua pengujian akurasi yang dilakukan.



Gambar 6.1 Grafik Hasil Pengujian Akurasi

Dari Gambar 6.1 dapat dilihat bahwa akurasi terendah sebesar 85% didapat dari pengujian dengan 20 data dan nilai $k = 2$ dari perhitungan sistem didapat 17 dari 20 data yang sesuai, untuk nilai $k = 3$ sampai $k = 7$ akurasi sebesar 90% yang berarti 18 dari 20 data adalah sesuai dengan data latih. Nilai akurasi dari 40 data dan nilai $k = 2$ sampai dengan $k = 7$ tetap sama sebesar 95% yakni 38 dari 40 data yang sesuai. Selanjutnya pada pengujian akurasi dengan 60 data dan nilai $k = 2$ sampai $k = 7$ juga didapat juga besaran akurasi yang stagnan sebesar 96.67% atau 58 data sesuai dari 60 data, data yang tidak sesuai ada pada data ke 31 dan 58 sistem mendiagnosis Jamur Akar dan Cacar Daun namun dari data latih adalah BPKC dan Jamur Akar.

BAB 7 KESIMPULAN DAN SARAN

7.1 Kesimpulan

Kesimpulan diambil berdasarkan hasil perancangan sampai pengujian dan analisis dari penelitian ini yang mengacu kepada rumusan masalah, maka dapat disimpulkan sebagai berikut:

1. Algoritme Modified *K-Nearest Neighbor* (MKNN) digunakan untuk melakukan klasifikasi yang menghasilkan diagnosis penyakit pada tanaman cengkeh dengan masukan yang berupa gejala penyakit yang berjumlah 11 gejala. Proses perhitungan dari MKNN diawali dengan menghitung jarak *Euclidean* antar data latih kemudian hasil dari tiap *Euclidean* tersebut diambil nilai terkecil sebanyak k , kemudian setelah didapat nilai terkecil dari *Euclidean* akan dicari nilai dari validasi yang didapat dengan cara membandingkan kelas dari data latih dengan kelas dari nilai k yang sudah diurutkan tersebut, apabila kelas dari keduanya sama maka nilainya sama dengan 1 dan jika tidak maka nilainya 0, nilai total validitas akan dijumlah dengan semua nilai validasi dan dibagi dengan nilai k dan menghasilkan nilai validitas. Setelah itu masukan berupa gejala diambil untuk dihitung jarak *Euclidean*nya dengan data latih yang ada, setelah didapat akan dihitung *Weight Voting* dengan menggunakan nilai validitas dan *Euclidean* data uji yang sudah dilakukan sebelumnya. Hasil *Weight Voting* yang terbesar adalah hasil diagnosis dari sistem yang dibangun.
2. Tingkat akurasi dari sistem yang didapat dari pengujian dan analisis dapat disimpulkan sebagai berikut:
 - a. Dengan menggunakan jumlah data 60 didapat nilai akurasi terbesar 96.67%, disusul dengan jumlah data 40 yang menghasilkan nilai akurasi rata-rata 95% dan terakhir jumlah data sebanyak 20 menghasilkan nilai akurasi rata-rata terendah sebesar 89.16%.
 - b. Akurasi yang terendah didapat dari nilai $k=2$ dengan jumlah data sebanyak 20 menghasilkan nilai akurasi 85%.

7.2 Saran

Saran untuk mengembangkan sistem diagnosis penyakit tanaman cengkeh jika ingin digunakan untuk penelitian selanjutnya dapat dilakukan sebagai berikut:

1. Karena Penulis menggunakan data gejala yang bernilai 1 dan 0 saja, sebaiknya nilai dari gejala-gejala pada data latih lebih bervariasi dan tidak bernilai 1 dan 0 saja agar diagnosis dari sistem dapat lebih baik.
2. Penulis juga merasa bahwa data yang dimiliki kurang banyak, maka dengan menambahkan jumlah data latih yang digunakan diharapkan menghasilkan akurasi yang lebih baik.

3. Lebih baik lagi jika ada pencarian nilai k optimal dengan menggunakan Algoritma Genetik sehingga nilai k dari sistem akan selalu menghasilkan nilai yang optimal, karena pada penelitian ini nilai k juga kurang begitu berpengaruh banyak terhadap akurasi dari sistem. Diharapkan dengan adanya variasi dari data dan lebih banyaknya data latih yang digunakan dapat memberikan hasil yang berbeda-beda dari tiap k yang digunakan.



DAFTAR PUSTAKA

- Badan Litbang Pertanian, 2011. *Pengendalian Terpadu Hama Penggerek Cengkeh*. [e-book]. Litbang pertanian. Tersedia melalui: website litbang pertanian <<http://www.litbang.pertanian.go.id/download/one/86/file/Pengendalian-Terpadu-Hama-.pdf>> [Diakses 20 Maret 2018]
- Departemen Proteksi Tanaman IPB, 2010. *Cacar Daun Cengkeh*. [online]. Opete - Basisdata Hama dan Penyakit Tanaman. Tersedia di: <<http://www.opete.info/detail2.php?idp=906>> [Diakses 28 Mei 2018]
- Departemen Proteksi Tanaman IPB, 2010. *Penyakit Akar Putih*. [online]. Opete Basisdata Hama dan Penyakit Tanaman. Tersedia di: <<http://www.opete.info/detail.php?idp=147>> [Diakses 28 Mei 2018]
- Dinas Pertanian Kabupaten Madiun, 2017. *Mengenal Gejala dan Penanggulangan Penyakit Bakteri Pembuluh Kayu Cengkeh (BPKC)*. [online]. Dinas Pertanian Tanaman Pangan & Horticoltura Kabupaten Madiun. Tersedia di: <<http://dph.madiunkab.go.id/berita-202-mengenal-gejala-dan-penanggulangan-penyakit-bakteri-pembuluh-kayu-cengkeh-bpkc.html>> [Diakses 28 Mei 2018]
- Hananto, P.E., 2012. Sistem Pakar Diagnosis Penyakit Tanaman Cengkik Dengan Metode Inferensi Forward Chaining. *Journal of Informatics and Technology*. Volume. pp. 1-14.
- Helilintar, R. 2017. Perancangan Sistem Diagnosa Penyakit Hepatitis Menggunakan Metode KNN.
- Sianipar, J.J., 2017. Identifikasi Diagnosis Gangguan Autisme Pada Anak Menggunakan Metode Modified *K-Nearest Neighbor* (MKNN). *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*. pp. 825-831.
- Kementerian Pertanian RI, 2016. Produksi Cengkeh Menurut Provinsi di Indonesia 2013-2017. [e-book]. Website Kementerian Pertanian Indonesia. Tersedia melalui: <[http://www.pertanian.go.id/Data5tahun/Esti2017-BUN\(pdf\)/204-Produksi-Cengkeh.pdf](http://www.pertanian.go.id/Data5tahun/Esti2017-BUN(pdf)/204-Produksi-Cengkeh.pdf)> [Diakses 18 Agustus 2017]
- Kementerian Pertanian RI, 2016. Produksi Cengkeh Nasional. [online]. Website Kementerian Pertanian Indonesia. Tersedia di: <<http://perkebunan.litbang.pertanian.go.id/?p=4219>> [Diakses 29 Agustus 2017]
- Kusumadewi, S., 2003. *Artificial Intelegence (Teknik dan Aplikasinya)*. Yogyakarta: Graha Ilmu.
- Mariana, L., 2013. *Hama dan Penyakit Cengkeh di Wilayah Kabupaten Kediri Jawa Timur*. Bogor: Institut Pertanian Bogor.
- Nikam, S.S., 2015. A Comparative Study of Classification Techniques in Data Mining Algorithms. *Oriental Journal of Computer Science & Technology*. Pp. 13-19.



- Putri, M.B.P., 2017. Diagnosis Penyakit Kulit Pada Kucing Menggunakan Metode Modified *K-Nearest Neighbor*.
- Mutrofin, S., 2016. Optimasi Teknik Klasifikasi Modified K Nearest Neighbor Menggunakan Algoritma Genetika. *Jurnal Gamma*. pp. 1-5.
- Wafiyah, F., 2017. Implementasi Algoritma Modified *K-Nearest Neighbor* (MKNN) untuk Klasifikasi Penyakit demam. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*. Volume 1(10). pp. 1210-1219.

