

BAB I PENDAHULUAN

1.1 Latar Belakang

Cabai merupakan salah satu jenis sayuran yang memiliki nilai ekonomi yang tinggi. Cabai memiliki berbagai macam vitamin yang bermanfaat bagi kesehatan manusia contohnya vitamin A, vitamin C, dan antioksidan yang berfungsi untuk menjaga tubuh dari serangan radikal bebas. Selain itu, kandungan lain dalam cabai adalah *Lasparaginase* dan *Capsaicin* yang berperan sebagai zat anti kanker .

Salah satu masalah utama tanaman cabai adalah adanya serangan hama dan penyakit yang dapat menyebabkan gagal panen dan merugikan petani. Perubahan pola musim hujan dan kemarau di Wilayah Indonesia juga dapat menyebabkan munculnya penyakit pada tanaman cabai. Munculnya penyakit pada tanaman cabai menimbulkan kerugian bagi para pengusaha adidaya tanaman cabai. Kerugian yang diakibatkan hama, penyakit, dan perubahan musim di Indonesia dapat menyebabkan kerugian 5-65%. Kerugian panen pada tanaman cabai akan berdampak pada tingginya harga cabai di pasaran. Kurangnya jumlah pakar dan pengetahuan petani mengenai cabai menyebabkan penanganan terhadap penyakit cabai menjadi lambat. Berdasarkan masalah tersebut dibutuhkan sistem yang dapat membantu petani dalam melakukan identifikasi penyakit cabai [1].

Cabai merah memiliki banyak manfaat di banyak bidang, selain digunakan di bidang kuliner, buah cabai merah juga mempunyai manfaat di bidang kesehatan. Penelitian sebelumnya mengenai penyakit tanaman cabai telah dilakukan oleh Sasmito yaitu diagnosa penyakit tanaman bawang merah dan cabai menggunakan Forward Chaining [2].

Metode MKNN adalah sebuah metode untuk melakukan klasifikasi terhadap objek berdasarkan data pembelajaran yang jaraknya paling dekat objek tersebut. Kemudian dikelompokkan suatu data baru berdasarkan jarak data baru ke beberapa data/tetangga terdekat [3].

Dengan menggunakan metode *Modified K-Nearest Neighbor* (MK-NN), memiliki beberapa kelebihan yaitu metode ini tangguh terhadap data training yang noisy dan efektif apabila data latihnya besar. Penerapan metode *Modified K-Nearest Neighbor* ini mempunyai keunggulan tingkat akurasi yang lebih tinggi dari *K-Nearest Neighbor* sederhana dengan tambahan perhitungan validitas dan weight voting tingkat akurasi maksimum 92,74% [4].

Terdapat dua pustaka yang dikaji pada penelitian dan berupa pembahasan penelitian sebelumnya yaitu penerapan MKNN pada pengklasifikasian kedelai dan implementasi MKNN pada penentuan keminatan SMA.

Dari proses pengujian pengklasifikasian penyakit tanaman kedelai didapatkan hasil pengujian dengan nilai akurasi rata – rata maksimum terbaik sebesar 92,74%. Dari hasil tersebut sistem yang telah dibuat dan mencerminkan bahwa sistem pakar penyakit tanaman kedelai dalam penelitian pada kajian pustaka yang pertama layak digunakan oleh pengguna dalam menentukan penyakit tanaman cabai dan pengendaliannya[5].

Oleh sebab itu, pada penelitian ini penulis mengusulkan skripsi berjudul “**Diagnosa Penyakit Cabai Merah Menggunakan Metode *Modified K-Nearest Neighbor* (MK-NN)**”. Penelitian ini diharapkan dapat mengidentifikasi jenis penyakit cabai dengan tingkat akurasi yang tinggi. Sehingga dapat mendiagnosa penyakit cabai dengan cepat dan meningkatkan produktivitas cabai.

1.2 Rumusan Masalah

Berdasarkan latar belakang, dapat disimpulkan rumusan masalah dalam penelitian ini adalah :

1. Bagaimana Diagnosa Penyakit Cabai Merah Menggunakan Metode *Modified K-Nearest Neighbor* (MK-NN)?
2. Bagaimana pengujian sistem dilakukan dengan metode *blackbox* dan pengujian pengaruh nilai k pada Diagnosa Penyakit Cabai Merah Menggunakan Metode *Modified K-Nearest Neighbor* (MK-NN)?

1.3 Batasan Masalah

Batasan masalah yang terdapat dalam penelitian ini antara lain:

1. Jumlah gejala penyakit yang dipertimbangkan dalam sistem pakar pada penelitian ini adalah sebanyak 14 butir dan 4 penyakit utama cabai merah.
2. Data tanaman cabai merah dan proses diagnosa yang dilakukan dalam penelitian ini berasal dari Pakar di Balai Pengkajian Teknologi Pertanian Jawa Timur.
3. Program dibuat dengan menggunakan bahasa pemrograman PHP dan *database* MySQL.
4. Terdapat dua tahap pengujian yang dilakukan pada penelitian ini, yaitu pengujian fungsionalitas dan pengujian nilai k .

1.4 Tujuan

Adapun tujuan yang diharapkan oleh penulis dalam penelitian, antara lain:

1. Merancang diagnosa penyakit tanaman cabai merah dengan metode *Modified K-Nearest Neighbor* (MK-NN) yang akurat dan efektif.
2. Mengatasi penyakit tanaman cabai dengan mendiagnosa penyakit tanaman cabai menggunakan diagnosa menggunakan metode *Modified K-Nearest Neighbor* (MK-NN) .

1.5 Manfaat

Manfaat yang diharapkan dari penelitian ini antara lain:

1. Mengatasi penyakit tanaman cabai dengan mendiagnosa dan memberikan solusi pada penyakit tanaman cabai menggunakan diagnose menggunakan metode *Modified K-Nearest Neighbor* (MK-NN)

1.6 Sistematika Penulisan

Sistematika penulisan pada penelitian ini terdiri dari tujuh bab. Berikut sistematika penulisan yang diterapkan:

BAB I : PENDAHULUAN

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

BAB II : TINJAUAN PUSTAKA

Pada bab ini dibahas mengenai dasar teori dan juga referensi penyakit tanaman cabai merah, dan *Modified K-Nearest Neighbor* (MK-NN) .

BAB III : METODOLOGI PENELITIAN

Pada bab ini diuraikan metode yang digunakan dalam penelitian dan langkah-langkah yang diambil pada penelitian.

BAB IV : PERANCANGAN DAN ANALISIS

Pada bab ini diuraikan mengenai perancangan dan analisa kebutuhan sistem.

BAB V : IMPLEMENTASI

Pada bab ini dijelaskan tentang perancangan sistem dan implementasi dari metode yang digunakan pada permasalahan yang dibahas.

BAB VI : PENGUJIAN

Pada bab ini dilakukan pengujian pada metode yang digunakan dengan membandingkan hasil penelitian dengan hasil yang didapat dari pakar dan pengujian fungsionalitas dari sistem yang telah dibangun.

BAB VII : PENUTUP

Pad bab ini diuraikan kesimpulan akhir penelitian dan juga terdapat saran-saran untuk pengembangan penelitian.

BAB II

TINJAUAN PUSTAKA

Bab ini berisi landasan kepustakaan yang membahas kajian pustaka dan dasar teori yang dibutuhkan pada penelitian. Kajian pustaka merupakan pembahasan penelitian yang telah dilakukan dan yang diusulkan. Dasar teori merupakan pembahasan teori yang dibutuhkan untuk menyusun penelitian yang diusulkan.

Terdapat dua pustaka yang dikaji pada penelitian dan berupa pembahasan penelitian sebelumnya yaitu penerapan MKNN pada pengklasifikasian tanaman kedelai dan Implementasi MKNN pada penentuan keminatan SMA.. Dasar teori yang akan dibahas pada bab ini yaitu teori dasar mengenai kecerdasan buatan, algoritma *Modified K-Nearest Neighbor* (MKNN), tinjauan umum tanaman cabai merah, dan jenis penyakit tanaman cabai merah.

2.1 Kajian Pustaka

Pada penelitian ini, kajian pustaka yang pertama membahas penelitian dengan obyek dan metode yang sama dengan penelitian yang diusulkan, sedangkan kajian pustaka yang kedua membahas penelitian dengan obyek yang berbeda dan metode yang sama dengan penelitian yang diusulkan.

Kajian pustaka pertama membahas penerapan MKNN untuk pengklasifikasian penyakit kedelai oleh Zainuddin dengan menguji pengaruh nilai k terhadap tingkat akurasi. Dan pada pengujian ini menggunakan dua parameter yaitu nilai k dan bentuk data uji[5].

Dari proses pengujian didapatkan hasil pengujian dengan nilai akurasi rata – rata maksimum terbaik sebesar 92,74%. Nilai akurasi rata-rata tertinggi 92.74% terdapat pada nilai k bernilai 3, yaitu : 95,59%, 87,90%, dan 94,7%. Dari hasil tersebut sistem yang telah dibuat dan mencerminkan bahwa sistem pakar penyakit tanaman kedelai dalam penelitian pada kajian pustaka yang pertama layak

digunakan oleh pengguna dalam menentukan penyakit tanaman cabai dan pengendaliannya[5].

Kajian pustaka yang kedua yaitu penelitian Hardiyanti yang membahas implementasi MKNN untuk penentuan keminatan SMA dengan menguji nilai $k=1$ sampai $k=20$. Hasil yang didapat adalah rata – rata maksimum akurasi mencapai 86,1% pada saat nilai $k=3$ [6].

Perbedaan yang terdapat pada penelitian yang diusulkan penulis adalah penggunaan metode *Modified K-Nearest Neighbor* yang digunakan untuk identifikasi penyakit pada tanaman cabai merah. Penerapan metode *Modified K-Nearest Neighbor* ini mempunyai keunggulan tingkat akurasi yang lebih tinggi dari *K-Nearest Neighbor* sederhana dengan tambahan perhitungan validitas dan weight voting. Analisa tentang perbandingan kajian pustaka dan penelitian yang diusulkan ditunjukkan pada Tabel 2.1.

Tabel 2.1 Kajian Pustaka

N o	Judul	Objek	Metode	Hasil
1.	Penerapan Algoritma <i>Modified K-Nearest Neighbor</i> pada Pengklasifikasian Penyakit Tanaman Kedelai [6]	Tanaman Kedelai	<i>Modified K-Nearest Neighbor</i>	nilai akurasi rata – rata maksimum terbaik sebesar 92,74%. Nilai akurasi rata rata tertinggi 92.74% terdapat pada nilai k bernilai 3, yaitu : 95,59%, 87,90%, dan 94,7%.

2.	Implementasi metode <i>Modified K-Nearest Neighbor</i> (MKNN) pada Penentuan Keminatan Sekolah Menengah Atas (SMA) (Studi Kasus: SMA Negeri 1 Seririt) [6]	Keminatan SMA	<i>Modified K-Nearest Neighbor</i>	Hasil yang didapat adalah rata – rata maksimum akurasi mencapai 86,1% pada saat nilai $k = 3$
3	Diagnosa Penyakit Tanaman Cabai Merah dengan Metode <i>Modified K-Nearest Neighbor</i> .	Tanaman Cabai	<i>Modified K-Nearest Neighbor</i>	Penyakit yang teridentifikasi dari perhitungan weight voting

Sumber: [4], [6]

2.2 Data Mining

Data mining didefinisikan sebagai satu set teknik yang digunakan secara otomatis untuk mengeksplorasi secara menyeluruh dan membawa ke permukaan relasi-relasi yang kompleks pada set data dimaksud di sini adalah set data yang berbentuk tabulasi, seperti yang banyak diimplementasikan dalam teknologi manajemen basis data relasional. Akan tetapi, teknik-teknik *data mining* dapat juga diaplikasikan pada representasi data yang lain, seperti domain data *spatial*, berbasis text, dan multimedia (citra). *Data mining* dapat juga didefinisikan sebagai “pemodelan dan penemuan pola pola yang tersembunyi dengan memanfaatkan data dalam volume yang besar” [7].

Data mining menggunakan pendekatan *discovery-based* dimana pencocokan pola (*pattern-matching*) dan algoritma-algoritma yang lain digunakan untuk menentukan relasi-relasi kunci di dalam data yang diekplorasi. *Data mining* merupakan komponen baru pada arsitektur sistem pendukung keputusan (DSS) di perusahaan-perusahaan[7].

2.3 Klasifikasi

Suatu teknik dengan melihat pada kelakuan dan atribut dari kelompok yang telah didefinisikan. Teknik ini dapat memberikan klasifikasi pada data baru dengan memanipulasi data yang ada yang telah diklasifikasi dan dengan menggunakan hasilnya untuk memberikan sejumlah aturan. Aturan-aturan tersebut digunakan pada data-data baru untuk diklasifikasi. Teknik ini menggunakan *supervised induction*, yang memanfaatkan kumpulan pengujian dari record yang terklasifikasi untuk menentukan kelas-kelas tambahan. Salah satu contoh yang mudah dan populer adalah dengan Decision tree yaitu salah satu metode klasifikasi yang paling populer karena mudah untuk diinterpretasi. Decision tree adalah model prediksi menggunakan struktur pohon atau struktur berhirarki [8].

Decision tree adalah struktur *flowchart* yang menyerupai *tree* (pohon), dimana setiap simpul internal menandakan suatu tes pada atribut, setiap cabang merepresentasikan hasil tes, dan simpul daun merepresentasikan kelas atau distribusi kelas. Alur pada *decision tree* di telusuri dari simpul akar ke simpul daun yang memegang prediksi kelas untuk contoh tersebut. *Decision tree* mudah untuk dikonversi ke aturan klasifikasi (*classification rules*) [8].

2.4 KNN

Prinsip kerja *K-Nearest Neighbor* (KNN) adalah mencari jarak terdekat antara data yang akan dievaluasi dengan k tetangga (neighbor) terdekatnya dalam data pelatihan [13]. Teknik ini termasuk dalam kelompok klasifikasi non parametric. Disini kita tidak memperhatikan distribusi dari data yang ingin kita kelompokkan. Teknik ini sangat sederhana dan mudah diimplementasikan. Mirip dengan teknik klustering, kita mengelompokkan suatu data baru berdasarkan jarak data baru itu ke beberapa data/tetangga(neighbor) terdekat [14].

2.4.1 Kelebihan dan Kelemahan KNN

KNN mempunyai beberapa kelebihan dan kelemahan dalam sistem analisisnya. Kelebihan-kelebihan KNN adalah [15]:

1. Metode KNN sederhana dalam penggunaannya.

2. Metode KNN dapat menangani data training yang mengandung *noise*.
3. Metode KNN Efektif jika data training besar.

Meskipun memiliki beberapa keuntungan, metode *K-Nearest Neighbor* (KNN) juga memiliki beberapa kelemahan antara lain [15] :

1. *Computation Cost* cukup tinggi karena perlu untuk menghitung jarak setiap *data training*.
2. Metode KNN membutuhkan memori yang besar.
3. Metode KNN memiliki tingkat akurasi yang rendah pada dataset multidimensi.
4. Metode KNN perlu menentukan nilai k parameter, jumlah tetangga terdekat.
5. Metode KNN menggunakan perhitungan jarak, yang belum diketahui pasti jenis jarak yang digunakan.
6. Metode KNN masih belum diketahui atribut yang lebih baik untuk menghasilkan hasil terbaik.

2.4.2 Tahapan KNN

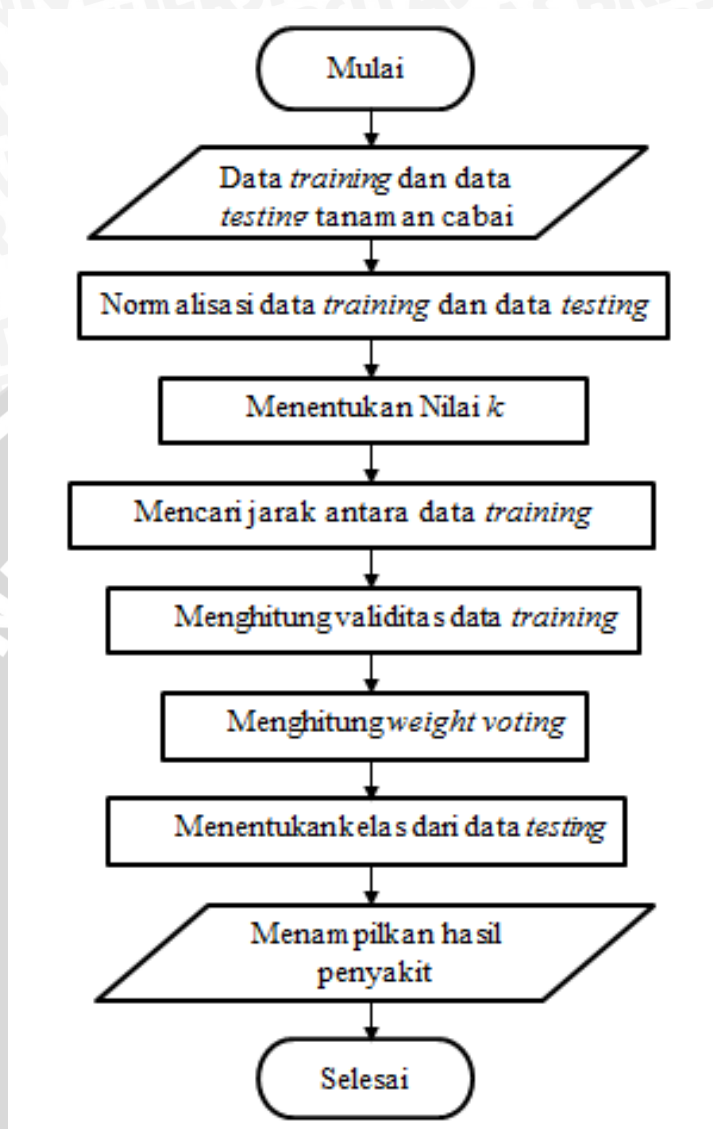
Dalam MKNN dilakukan 8 tahapan sebagai berikut:

1. Normalisasi tabel pada data *training* dan data *testing*
2. Menentukan nilai k atau tetangganya
3. Mencari jarak antar data *training*
4. Urutkan jarak yang terbentuk
5. Tentukan jarak terdekat sampai urutan k
6. Pasangkan kelas yang bersesuaian
7. Cari jumlah kelas dari tetangga yang terdekat dan tetapkan kelas tersebut sebagai kelas data yang akan dievaluasi.
8. Menentukan kelas dari data *testing*

2.5 MKNN

Modified K-Nearest Neighbor (MKNN) adalah menempatkan label kelas data sesuai dengan k divalidasi poin data yang sudah ditetapkan dengan perhitungan *K-Nearest Neighbor*(KNN). Dalam proses algoritma MKNN, terdapat beberapa tambahan proses dibanding dengan algoritma KNN yaitu : validitas *data training* dan *weight voting* [15].

2.5.1 Tahapan MKNN



Gambar 2.2 Tahapan *Modified K-Nearest Neighbor*

Sumber: [Penelitian]

Dalam MKNN pada Gambar 2.2 dilakukan 6 tahapan sebagai berikut:

1. Normalisasi tabel pada data *training* dan data *testing*

Dalam tahap ini normalisasi data bertujuan agar data berada pada *range* [0,1] sehingga sebaran datanya berada pada *range* yang sama. Rumus normalisasi ditunjukkan pada persamaan 2-1 .

$$\text{Normalisasi (x)} = \frac{x - \text{minValue}}{\text{maxValue} - \text{minValue}} \dots\dots\dots(2-1)$$

Keterangan Persamaan 2-1 :

minValue : Nilai terkecil

maxValue : Nilai terbesar

2. Menentukan nilai *k* atau tetangganya.

Dalam tahap ini ditentukan nilai *k* atau tetangganya.

3. Mencari jarak antar data *training*

Dalam tahap ini mencari jarak data *training* dan data *testing* menggunakan rumus *euclidian* pada persamaan 2-2 berikut.

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

Keterangan Persamaan 2.2 :

x = Sampel Data

y = Data Uji/ Testing

i = Variabel Data

d = Jarak

n = Dimensi Data

4. Menghitung validitas data *training*

Validitas data dari setiap data dihitung berdasarkan pada tetangganya. Validitas data dilakukan hanya sekali pada semua data latih. Setelah dilakukan validasi data selanjutnya data tersebut digunakan sebagai informasi tambahan. Rumus yang digunakan untuk menghitung validitas setiap data latih ditunjukkan pada persamaan 2-3[15]:

$$\text{Validitas (x)} = \frac{1}{k} \sum_{i=1}^k S(\text{label}(N_i(x))) \dots\dots\dots(2-3)$$

Keterangan persamaan 2-3 :

k = jumlah titik terdekat

label (*x*) = kelas *x*

label *N_i*(*x*) = label kelas titik terdekat *x*.

Fungsi S sendiri menyamakan kelas data x dengan kelas data terdekat ke $-i$.
Fungsi S ditunjukkan pada persamaan 2-4:

$$S(a,b) = \begin{cases} 1 & a = b \\ 0 & a \neq b \end{cases} \dots\dots\dots(2-4)$$

Keterangan persamaan 2-4:

a = Kelas pada data *training*

b = Kelas selain a pada data *training*

S akan bernilai 1, jika label kategori a sama dengan label kategori b . S bernilai 0, jika label kategori a tidak sama dengan label kategori b .

5. Menghitung *weight voting*

Weight Voting adalah salah satu variasi metode MKNN yang menggunakan k tetangga terdekat, terlepas dari kelas data, akan tetapi menggunakan *weight voting* dari masing – masing data pada data *training*. Masing – masing data diberikan *weight voting* dengan persamaan 2-5 sebagai berikut[15]:

$$W_{(i)} = \frac{1}{d + \alpha} \dots\dots\dots(2-5)$$

Dimana d adalah jarak dan α adalah merupakan nilai *regulator smoothing*. *Weight Voting* ini kemudian dijumlahkan setiap kelasnya dan kelas dengan jumlah terbesar yang akan dipilih menjadi sebuah keputusan. Kemudian validitas dari tiap data pada data *training* dikalikan dengan *weight* berdasarkan pada jarak. Dalam metode MKNN, *weight voting* tiap tetangga seperti pada persamaan 2-6 berikut :

$$W_{(i)} = Validitas(x) \times \frac{1}{d + 0.5} \dots\dots\dots(2-6)$$

Dimana :

$W_{(i)}$: Perhitungan *Weight Voting*

Validitas (x) : Nilai Validitas

d : Jarak Euclidian

6. Menentukan kelas dari data testing

Setelah didapatkan nilai weight voting dari semua data training maka dilakukan pencarian nilai weight voting yang terbesar sebanyak nilai k yang telah ditentukan.

7. Hasil Diagnosis

Tahap ini merupakan tahap terakhir yaitu menampilkan hasil diagnosa penyakit.

2.6 Akurasi

Akurasi adalah ukuran kedekatan suatu hasil pengukuran dengan angka sebenarnya. Pengujian akurasi pada penelitian ini dilakukan untuk mengetahui kemampuan sistem dalam menghasilkan kesimpulan. Perhitungan akurasi dapat menggunakan rumus seperti pada persamaan 2-7[4].

$$Akurasi = \frac{\sum \text{data uji benar}}{\sum \text{data uji}} \times 100\% \dots \dots \dots (2-7)$$

2.7 Pengujian

Testing merupakan pengujian terhadap sistem yang dibuat. *Testing* yang digunakan pada penelitian ini menggunakan *blackbox testing*. *Black Box Testing* adalah pengujian yang dilakukan dengan hanya mengamati hasil eksekusi melalui data uji dan memeriksa fungsional dari perangkat lunak.

Pengujian perangkat lunak dilakukan untuk mengetahui bahwa program dapat melakukan apa yang seharusnya dilakukan dan untuk menemukan sebuah kesalahan pada program sebelum program tersebut mulai digunakan. Proses pengujian perangkat lunak memiliki dua tujuan :

- Menunjukkan kepada pengembang dan pengguna bahwa perangkat lunak memenuhi spesifikasi kebutuhan.
- Menemukan situasi di mana perilaku dari perangkat lunak terdapat sebuah kesalahan, hal yang tidak diinginkan atau tidak sesuai dengan spesifikasi kebutuhan. Perilaku tersebut seperti *systemcrashes*, interaksi antar sistem, kesalahan dalam komputasi dan perubahan data

Black box testing merupakan pengujian yang fokus pada persyaratan fungsional perangkat lunak. *Blackbox testing* menggunakan pendekatan komplementer yang memungkinkan untuk mengungkap kesalahan[16].

2.8 Tanaman Cabai Merah

Cabai ialah jenis tanaman yang termasuk *genus Capsicum*, yang pada umumnya mempunyai rasa pedas. Cabai mempunyai banyak kegunaan , diantaranya untuk sambal, bumbu masak, untuk obat-obatan dan sebagainya [17].

Cabai mempunyai banyak varietas, antara lain cabai besar, cabai kecil, cabai pendek, dan cabai panjang. Sebenarnya cabai bukan asli tanaman Indonesia, walaupun hampir setiap hari penduduk Indonesia mengkonsumsi cabai. Cabai sendiri berasal dari Mexico, Peru, dan Bolivia, tetapi sekarang sudah tersebar di seluruh dunia [17].

Colombus adalah yang pertama kali menemukan cabai dan membawa bijinya ke Spanyol pada tahun 1493, yang kemudian banyak di tanam di Eropa. Cabai yang ditanam di Amerika Serikat sekarang ini berasal dari Eropa, padahal Cabai berasal dari Amerika Selatan dan Tengah. Cabai yang sekarang ini banyak di tanam di Indonesia mungkin yang membawanya adalah orang Eropa[17].

Cabai dapat ditanam di dataran rendah maupun dataran tinggi, tergantung varietasnya. Tanah yang cocok untuk tanaman cabai adalah tanah yang gembur dan subur. Apabila ditanam di tempat yang kurang cocok, cabai mudah sekali terserang hama dan penyakit[17].

Penyakit yang paling merugikan adalah penyakit keriting atau mosaik. Penyakit ini disebabkan oleh virus. Tetapi ada juga penyakit lain yang merugikan, misalnya penyakit akar, penyakit bercak daun, penyakit busuk buah dan lain-lainnya. Hama yang sering menyerang tanaman cabai yaitu lalat buah, apis, thrips, tungau, nematoda, dll. Hama-hama ini biasanya menyerang pada waktu musim kemarau. Sedangkan penyakit biasanya menyerang cabai pada musim hujan. Penyakit tanaman cabai tidak hanya cendawan, bakteri atau virus saja. Penyakit cabai bisa karena kekurangan atau kelebihan unsur-unsur makanan[17].

2.8.1 Morfologi Tanaman Cabai Merah

Cabai merupakan tanaman yang tumbuh tegak dengan batang berkayu, banyak cabang, serta ukuran yang mencapai tinggi 120cm dan lebar tajuk tanaman hingga 90 cm. Umumnya, daun cabai berwarna hijau muda sampai hijau gelap, tergantung varietasnya. Daun cabai yang ditopang oleh tangkai daun mempunyai tulang menyirip. Daun cabai berbentuk bulat telur, lonjong, ataupun oval dengan ujung yang meruncing, tergantung spesies dan varietasnya[18].

Bunga cabai keluar dari ketiak daun dan berbentuk seperti terompet. Sama halnya dengan tanaman dari keluarga Solanaceae lainnya. Bunga cabai merupakan bunga lengkap yang terdiri atas kelopak bunga, mahkota bunga, benang sari, dan putik. Bunga cabai juga berkelamin dua, karena benang sari dan putik terdapat dalam satu tangkai[18].

Bentuk buah cabai berbeda-beda, dari cabai keriting, cabai besar yang lurus dan bisa mencapai ukuran sebesar ibu jari, cabai rawit yang kecil-kecil tapi pedas, cabai paprika yang berbentuk seperti buah apel, dan bentuk-bentuk cabai hias yang banyak ragamnya[18].

2.8.2 Klasifikasi Tanaman Cabai Merah

Klasifikasi tanaman cabai merah adalah sebagai berikut:

1. Kingdom : *Plantae*
2. Divisi : *Spermatophyta*
3. Subdivisi : *Angiospermae*
4. Kelas : *Icotyledoneae*
5. Subkelas : *Sympetaleae*
6. Ordo : *Tubiflorae (Solanales)*
7. Famili : *Solanaceae*
8. Genus : *Capsicum*
9. Spesies : *Capsicum annum L.*[18]

Tanaman cabai termasuk ke dalam family Solanaceae. Tanaman lain yang masih sekerabat dengan cabai., diantaranya kentang (*Solanum tuberosum L.*),

terung (*Solanum melongena* L.), leunca (*Solanum nigrum* L.), akokak (*Solanum torvum* L.), dan tomat (*Solanum lycopersicum*)[18].

2.8.3 Anatomi Cabai Merah

Anatomi tanaman cabai merah adalah sebagai berikut:

1. Akar Cabai

Perakaran cabai termasuk akar tunggang yang terdiri atas akar utama (primer) dan akar lateral (sekunder). Dari akar lateral keluar serabut-serabut akar (akar tersier). Panjang akar primer tanaman cabai berkisar 35–50 cm. Akar lateral tanaman cabai menyebar dengan panjang berkisar 35–45 cm.

2. Batang Cabai

Batang utama tanaman berbentuk tegak lurus dan kokoh, tinggi sekitar 30–40 cm, dan diameter batang sekitar 1,5–3,0 cm. Batang utama tanaman cabai berkayu dan berwarna cokelat kehijauan. Pada budidaya cabai intensif, pembentukan kayu pada batang utama mulai terjadi pada umur 30–40 hari setelah ditanam (HST). Pada setiap ketiak daun cabai tumbuh tunas baru yang dimulai pada umur 10–15 HST. Namun pada budidaya cabai intensif, tunas-tunas baru harus dirempel.

3. Daun Cabai

Daun cabai berwarna hijau muda hingga hijau gelap. Daun ditopang oleh tangkai daun. Bentuk tulang daun cabai menyirip. Secara keseluruhan bentuk daun cabai merah ialah lonjong dengan ujung daun tanaman meruncing.

4. Bunga dan Buah Cabai

Seperti umumnya pada famili *Solanaceae*, bunga cabai berbentuk terompet (*hyporcrateriformis*). Bunga cabai tergolong bunga yang lengkap (*completus*) karena terdiri atas kelopak bunga (*calyx*), mahkota bunga (*corrola*), benang sari (*stamen*), dan putik (*pistillium*). Alat kelamin jantan (benang sari) dan alat kelamin betina (putik) cabai terletak di dalam satu bunga sehingga disebut berkelemin dua (*hermaphroditus*). Bunga cabai tumbuh di percabangan (ketiak daun), terdiri dari enam helai kelopak bunga berwarna hijau dan lima helai mahkota bunga berwarna putih.

Tangkai putik berwarna putih dan kepala putik berwarna kuning kehijauan. Satu bunga cabai terdiri dari satu putik dan enam benang sari. Tangkai sari

berwarna putih dan kepala sari berwarna biru keunguan. Pada saat pembentukan buah cabai, mahkota bunga akan rontok tetapi kelopak bunga tetap menempel pada buah cabai [11].

2.8.4 Penyakit Utama Tanaman Cabai Merah

Penyakit- penyakit tanaman cabai dapat dibagi menjadi dua kelompok, yaitu penyakit parasit dan nonparasit. Penyakit parasit disebabkan oleh bakteri, cendawan, virus, riketsia, mikoplasma, viroid, ganggang, benalu, dan tali pateri. Sedangkan penyakit nonparasit disebabkan oleh kekurangan unsur N, K, P, Ca, Mg, Cn, Fe, Mn, air, sinar matahari, dan sebagainya. Penyakit nonparasit sering disebut juga penyakit fisiologis[18].

Penyakit-penyakit tanaman cabai merah yang dibahas dalam penelitian ini merupakan penyakit tanaman cabai merah yang bersifat utama. Berdasarkan hasil wawancara dengan pakar, hanya dibatasi beberapa jenis penyakit yang pada umumnya menyerang tanaman cabai saja.

Penjelasan mengenai penyakit-penyakit utama pada tanaman cabai merah adalah sebagai berikut:

1. Bercak Daun *Cercospora*

Tanaman cabai sering terjangkit penyakit bercak daun *Cercospora*. Cabai yang mudah terserang penyakit ini adalah jenis cabai paprika dan cabai besar. Gejala *Cercospora* ditandai dengan munculnya bagian tengah berwarna kelabu-muda (sedikit putih),sedangkan tepinya berwarna coklat tua. Bercak juga dapat muncul pada batang, tangkai daun atau tangkai buah dan dapat menyebabkan daun berguguran.Penyakit *cercospora* ditunjukkan pada gambar 2.3.



Gambar 2.3 Bercak daun *cercospora***Sumber:** [16]**2. Antraknosa**

Penyebab penyakit ini adalah cendawan *Colletotrichum Capsici*. Penyakit ini bisa timbul di lapangan ataupun di cabai yang sudah panen. Mula-mula pada buah yang sudah masak kelihatan bercak kecil cekung-kebasahan yang berkembang sangat cepat. Bercak cekung itu berwarna merah-tua sampai coklat-muda dan kelihatan ada jaringan cendawan berwarna hitam dan buah berubah menjadi busuk-lunak[17].

Solusi yang dapat dilakukan antara lain biji didesinfeksi, kemudian jangan menanam biji dari buah yang sakit, dan yang terakhir disemprot dengan fungisida. Penyakit antraknosa ditunjukkan pada Gambar 2.4.

**Gambar 2.4** Antraknosa**Sumber:** [Penelitian]**3. Virus Gemini / Penyakit Kuning (Yellowing)**

Penyebab penyakit ini adalah lalat putih (*Bemisia Tabaci*) yang menyebarkan virus gemini. Gejala virus ini ditandai dengan daun cabai menjadi keriting dan menguning. Cabai yang terjangkit penyakit ini dapat mengalami penurunan pertumbuhan dan menjadi kerdil. Cabai yang terserang ruas-ruasnya menjadi

pendek, daun mengecil, dan tepi daun melengkung keatas. Warna tepi daun menjadi hijau-pucat sampai kuning-cerah. Serangan yang berat dapat menyebabkan daun menjadi kuning[17].

Solusi yang dapat dilakukan yaitu :

1. Sebelum terserang semprotlah dengan insektisida secara rutin.
2. Cabai yang terserang penyakit ini harus dicabut dan dibakar.
3. Gulma harus selalu dibersihkan.
4. Belilah pupuk organik yang higienis.

Penyakit ini ditunjukkan pada Gambar 2.5.



Gambar 2.5 Virus Gemini

Sumber: [Penelitian]

4. Penyakit Kerupuk

Penyakit ini disebabkan oleh virus dari grup luteo. Tanaman menunjukkan gejala berupa daun menggulung kearah bawah. Penyebab penyakit kerupuk yaitu *Chili Puckery Stunt Virus* sejak tanaman masih di persemaian. Pada tanaman muda tampak daun melengkung ke bawah, pada usia-usia selanjutnya, lengkungan bertambah parah disertai kerutan. Daun berwarna hijau pekat mengkilat dengan permukaan yang tidak merata seperti kerupuk, pertumbuhan terhambat, daun menumpuk dan menggumpal, dan juga berguguran sehingga yang tersisa hanya daun-daun yang menggulung pada pucuk. Buah dan bunga berguguran sehingga produksi buah lebih sedikit. Solusi untuk penyakit ini adalah pemupukan secara

berimbang dengan 150-200 kg Urea, 100-150 kg TSP, 450-500 kg Za, 100-150 KCL, dan 20-30 ton pupuk organik untuk tiap hektar [19].

Penyakit kerupuk ditunjukkan pada Gambar 2.6.



Gambar 2.6 Penyakit Kerupuk

Sumber: [Penelitian]



BAB III

METODOLOGI PENELITIAN

Pada bab ini dijelaskan langkah-langkah yang akan dilakukan dalam pengerjaan penelitian ini, yaitu studi literatur, pengumpulan data, analisa dan perancangan sistem, implementasi sistem, pengujian sistem, dan pengambilan kesimpulan. Alur penelitian secara umum ditunjukkan pada gambar 3.1.



Gambar 3.1 Diagram Blok Metodologi Penelitian

Sumber : [Penelitian]

3.1. Studi Literatur

Pada tahap ini dilakukan untuk mendapatkan dasar teori sebagai sumber acuan untuk penulisan skripsi dan pengembangan diagnosa penyakit pada tanaman cabai merah, antara lain :

1. Algoritma *Modified K-Nearest Neighbor*.
2. Identifikasi penyakit pada tanaman cabai merah.

Literatur dan pendukung penelitian ini antara lain buku, jurnal, *paper*, laporan penelitian, dan bantuan mesin pencari (*search engine*) internet.

3.2. Pengumpulan Data

Lokasi penelitian skripsi ini terletak di Balai Pengkajian Teknologi Pertanian (BPTP) Jawa Timur, Jl. Raya Karangploso, Km. 4, Kota Malang, Jawa Timur. Variabel penelitian pada skripsi ini adalah berbagai macam penyakit yang menyerang tanaman cabai merah dan bagaimana cara pengendaliannya berdasarkan hasil perhitungan nilai probabilitas gejala untuk setiap penyakit dengan metode *Modified K-Nearest Neighbor*.

Metode pengumpulan data untuk penelitian terbagi menjadi dua jenis data, yaitu data primer dan data sekunder. Data primer adalah data yang didapat langsung dari responden penelitian. Metode pengumpulan data primer bersifat kuantitatif, dengan arti dapat menggunakan data dari hasil wawancara dengan pakar. Data sekunder adalah data yang dikumpulkan oleh sumber lain dan tidak dipersiapkan untuk penelitian tetapi dapat digunakan untuk penelitian seperti melalui jurnal literatur.

Berdasarkan Tabel 3.1, penentuan kebutuhan data penelitian adalah sebagai berikut :

1. Data mengenai penyakit pada tanaman cabai merah. Sumber data didapat dari pakar tanaman cabai merah dengan metode wawancara.
2. Data kasus tanaman cabai merah yang terserang penyakit. Sumber data didapat dari BPTP Jawa Timur melalui observasi keperkebunan. Data digunakan dalam proses perhitungan dengan metode *Modified K-Nearest Neighbor*.

Tabel 3.1 Penentuan Kebutuhan Data Penelitian

No.	Kebutuhan Data	Sumber Data	Metode	Kegunaan Data
1.	Data penyakit tanaman cabai merah.	Pakar	Wawancara	Dasar pengetahuan dan pembuatan <i>rule</i> mengenai penyakit tanaman cabai merah.
2.	Data kasus tanaman cabai merah yang terserang penyakit.	BPTP Jawa Timur, Kota Malang	Observasi	Digunakan pada proses perhitungan dengan metode <i>Modified K-Nearest Neighbor</i>

Sumber : [Penelitian]

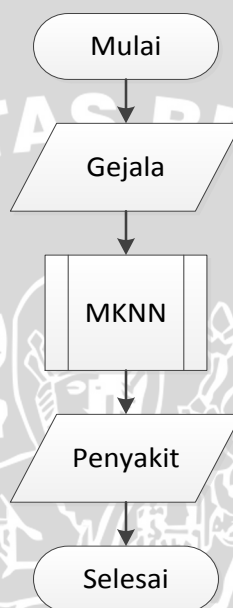
3.3. Analisa Kebutuhan

Analisa kebutuhan adalah tahapan yang dilakukan untuk menganalisa kebutuhan pembangunan sistem diagnosa. Berikut adalah kebutuhan dalam diagnosa penyakit tanaman cabai merah:

1. Kebutuhan *Hardware*: *Laptop* dengan spesifikasi prosesor *Intel® Core™ i3-2310M CPU@2.10GHz*, memori 2 GB, kapasitas *HDD* 512 GB
2. Kebutuhan *Software*:
 - a. Sistem Operasi *Windows 7 64-bit*
 - b. *Internet Browser (Mozilla/ Chrome)*
 - c. *Netbeans IDE 7.20*
 - d. *XAMPP Control Panel v.3.1.0 beta 6*
 - e. *Adobe Photoshop CS6*
3. Data yang diperlukan meliputi:
 - a. Data penyakit tanaman cabai merah
 - b. Data setiap gejala penyakit tanaman cabai merah
 - c. Data nilai dan kriteria

3.4. Perancangan Sistem

Pada perancangan sistem ini akan dijabarkan mengenai proses-proses dalam membangun suatu sistem. Sistem ini dibangun untuk mengklasifikasi penyakit tanaman cabai merah. Sistem akan mengolah data *input* dari user berupa gejala tanaman cabai merah. Kemudian sistem akan menghasilkan *output* berupa hasil diagnosa jenis penyakit dan pengendaliannya dengan menggunakan metode MK-NN. Diagram alir sistem ditunjukkan pada Gambar 3.1.

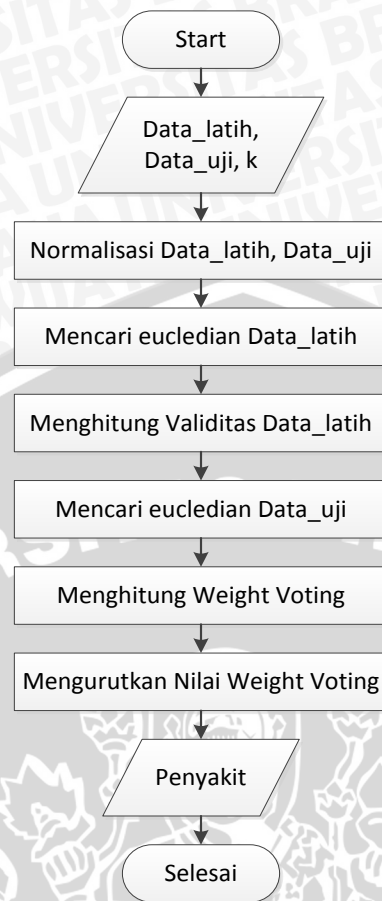


Gambar 3.2 Diagram Alir Sistem

Sumber : [Perancangan]

3.4.1 Diagram Alir Metode Modified K-Nearest Neighbor (MK-NN)

Model perancangan sistem menjabarkan mengenai kinerja sistem secara terstruktur, dimulai dari *input* hingga *output* yang dihasilkan. Diagram alir perancangan sistem dapat dilihat pada Gambar 3.2.



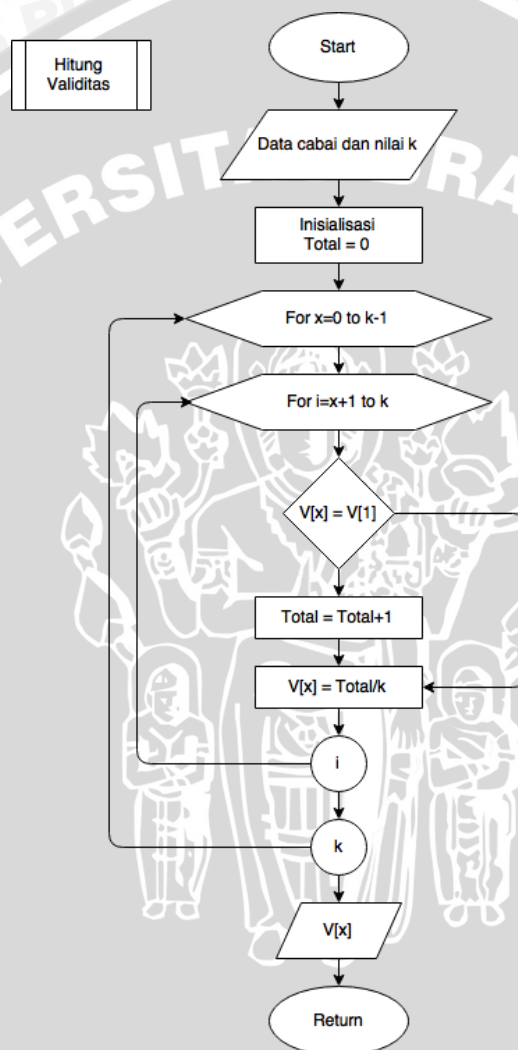
Gambar 3.2 Diagram Alir Perhitungan Metode *Modified K-Nearest Neighbor*
Sumber : [Perancangan]

Perhitungan *Modified K-Nearest Neighbor* berdasarkan gambar 3.2 dilakukan melalui tahapan-tahapan sebagai berikut :

1. Tentukan data *training*, data *testing*, dan nilai k .
2. Normalisasi data *training* dan data *testing* yang didapat dari pakar dengan menggunakan nilai terbesar dan terkecil pada setiap atribut gejala.
3. Mencari jarak euclidian antara data *training* dari setiap atribut gejala.
4. Menghitung validitas data *training*
5. Mencari jarak euclidian antara data *testing* dengan data *training* dari setiap atribut gejala.
6. Menghitung *weight voting* berdasarkan nilai validitas dan euclidian data *testing*.
7. Mengurutkan *weight voting* dari terbesar hingga terkecil.
8. Mendapatkan hasil diagnosa penyakit cabai merah dari nilai terbesar yang didapat dari nilai *weight voting* yang telah diurutkan.

3.4.2 Diagram Alir Proses Perhitungan Validitas

Langkah-langkah dalam proses perhitungan validitas antara lain memberikan input data training berupa data tanaman cabai, menentukan nilai k , melakukan perhitungan validitas sesuai dengan persamaan 2-3, dan memberikan keluaran berupa hasil validitas. Diagram alir perhitungan validitas ditunjukkan pada gambar 3.3 berikut.



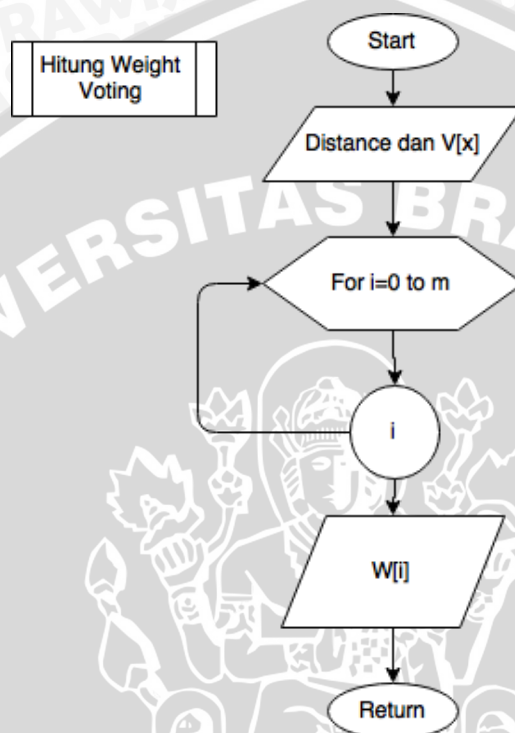
Gambar 3.3 Diagram Alir Perhitungan Validitas
Sumber : [Perancangan]

Pada proses validitas yang dilakukan adalah melakukan input data tanaman cabai kemudian dengan membandingkan kelas-kelas pada data latih sesuai dengan ketentuan sebelumnya. Dengan ketentuan jika kelasnya sama maka $X[x] = X[x] + 1$ nilainya 1, dan jika kelasnya tidak sama maka $V[x] = V[x]$ bernilai 0, dan dilakukan perbandingan kelas sebanyak k data yang diinputkan. Kemudian $v[x]$ nya akan dijumlah dan dibagi sejumlah k data yang sudah

dimasukkan. Sehingga diperoleh nilai validitas tiap data uji pada proses klasifikasi metode *Modified K-Nearest Neighbor* (MK-NN).

3.4.3 Diagram Alir Proses Perhitungan Weight Voting

Langkah-langkah dalam perhitungan weight voting adalah memasukkan nilai *euclidian* dan nilai validitas, kemudian dihitung menggunakan persamaan 2-5. Diagram alir dari proses perhitungan weight voting ditunjukkan pada gambar 3.4 berikut.



Gambar 3.4 Diagram Alir Perhitungan Weight Voting
Sumber : [Perancangan]

3.5. Implementasi

Implementasi adalah tahap pembentukan sistem. Implementasi diagnosa penyakit tanaman cabai menggunakan metode *Modified K-Nearest Neighbor* ini mengacu pada perancangan sistem. Pengembangan sistem dilakukan dengan bahasa pemrograman *HTML* dan *PHP*. Tahapan-tahapan dalam implementasi antara lain:

1. Normalisasi data latih dan data uji.
2. Perhitungan metode *Modified K-Nearest Neighbor* dengan tujuan memperoleh hasil pembobotan.
3. *Output* sistem berupa informasi hasil diagnosa pada tanaman cabai beserta solusi pengendaliannya.

4. Pembuatan antarmuka.

3.6. Pengujian dan Analisa Sistem

Pengujian sistem pada penelitian ini dilakukan agar dapat menunjukkan bahwa sistem telah mampu berjalan sesuai dengan spesifikasi dari kebutuhan yang melandasinya. Pengujian dilakukan dengan dua tahap yaitu :

1. Pengujian pertama yaitu pengujian fungsionalitas. Pengujian ini dilakukan dengan metode *blackbox testing* untuk mengetahui fungsionalitas sistem apakah telah berjalan sesuai harapan yang telah ditentukan. Pengujian ini dilakukan oleh peneliti dan calon pengguna.

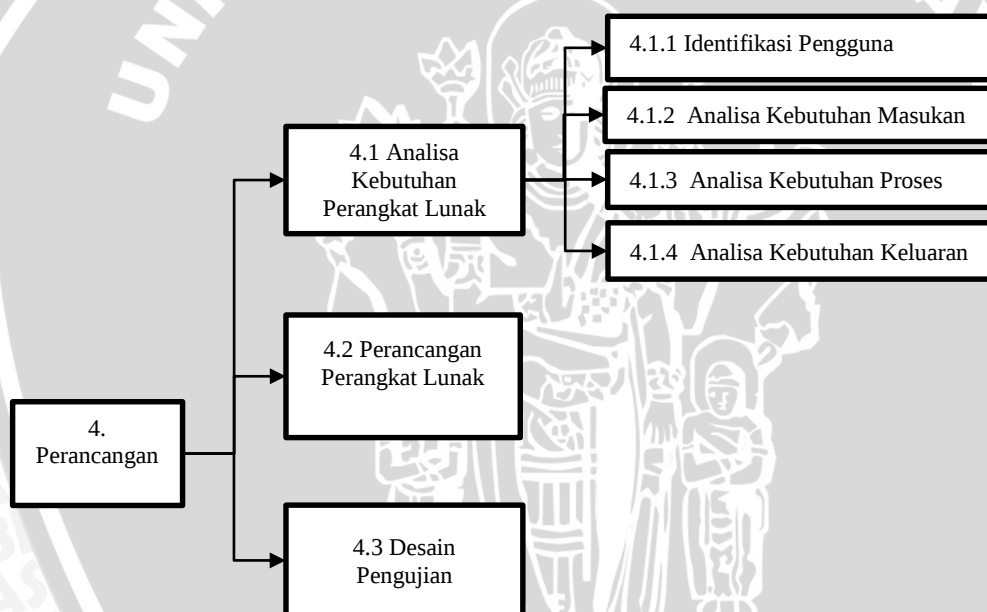
2. Pengujian pertama yaitu pengujian akurasi sistem dengan tujuan membandingkan hasil diagnosa sistem dengan hasil diagnosa pakar untuk mendapatkan tingkat akurasi sistem.

3.7. Pengambilan Kesimpulan

Evaluasi dilakukan dengan cara memeriksa apakah sistem sudah berjalan dengan baik dan tidak ada *error* yang terjadi. Selain itu, pengujian juga dilakukan dengan cara membandingkan hasil identifikasi dari sistem dengan hasil identifikasi yang dilakukan oleh pakar. Pengujian *black box* juga akan dilakukan, hal ini sebagai indikator keberhasilan pada setiap fungsionalitas yang ada pada sistem diagnosa.

BAB IV PERANCANGAN

Bab ini membahas tentang analisis kebutuhan perangkat lunak dan perancangan aplikasi “Diagnosa Penyakit Tanaman Cabai Merah Menggunakan Metode *Modified K-Nearest Neighbor*”. Perancangan sistem ini terdiri dari tiga tahapan, yaitu analisa kebutuhan perangkat lunak, perancangan perangkat lunak, dan desain pengujian. Analisa kebutuhan perangkat lunak meliputi identifikasi aktor, analisa kebutuhan masukan, analisa kebutuhan proses, dan analisa kebutuhan keluaran. Perancangan perangkat lunak meliputi diagram alir sistem. Alur perancangan dari sistem yang akan dibuat ditunjukkan pada gambar 4.1.



Gambar 4.1 Pohon Perancangan
Sumber: [Perancangan]

4.1.1. Analisa Kebutuhan Perangkat Lunak

Analisa kebutuhan ini mencakup kegiatan dalam menentukan kebutuhan-kebutuhan atau kondisi yang harus dipenuhi untuk membuat sistem. Analisa kebutuhan ini dimulai dengan identifikasi aktor-aktor yang terlibat di dalam sistem, penjelasan kebutuhan masukan, proses dan keluaran.

4.1.2. Identifikasi Pengguna

Pada tahap ini direpresentasikan *user* yang berinteraksi dengan sistem. Sistem pakar ini dibangun untuk dapat diakses oleh *user* biasa tanpa perlu tingkatan *user*. *User* biasa dapat melakukan proses diagnosa penyakit tanaman cabai, melihat *history* penyakit, dan membaca informasi-informasi yang dimasukkan peneliti pada sistem mengenai tanaman cabai merah.

4.1.3. Analisa Kebutuhan Masukan

Pakar memberikan masukan berupa :

1. Data gejala tanaman cabai.
2. Data jenis penyakit tanaman cabai merah.
3. Bobot tiap gejala penyakit tanaman cabai merah.

Tiga masukan di atas berfungsi sebagai basis pengetahuan dari sistem dalam mendiagnosa penyakit tanaman cabai. Selain masukan juga terdapat kebutuhan fungsional dan non fungsional. Berikut ini adalah daftar kebutuhan fungsional dijabarkan pada tabel 4.1.

Tabel 4.1 Daftar Kebutuhan Fungsional

No.	Kebutuhan
1.	Sistem mampu menampilkan halaman utama.
2.	Sistem mampu menampilkan halaman informasi <i>history</i> penyakit tanaman cabai .
3.	Sistem mampu menampilkan halaman informasi diagnosa penyakit tanaman cabai ..
4.	Halaman formulir diagnosa dapat menampilkan gejala-gejala tanaman cabai merah beserta bobot dari tiap gejala penyakit.
5.	Tombol <i>submit</i> pada formulir diagnosa mampu mengarahkan user menuju halaman hasil diagnosa.
6.	Sistem mampu menampilkan hasil perhitungan, hasil diagnosa gejala, dan cara penanggulangannya dengan benar.

7.	Sistem mampu menampilkan data <i>training</i> pada sistem.
----	--

Sumber: [Perancangan]

Daftar kebutuhan non-fungsional dijabarkan untuk mengetahui spesifikasi kebutuhan sistem. Daftar kebutuhan non-fungsional aplikasi sistem pakar ini dapat dilihat pada Tabel 4.2.

Tabel 4.2 Daftar Kebutuhan Non-Fungsional

No.	Kebutuhan	Deskripsi
1.	<i>Performance</i>	Sistem dapat beroperasi secara optimal.
2.	<i>Response Time</i>	Sistem dapat melakukan pemrosesan data secara tepat waktu.

Sumber: [Perancangan]

4.1.4. Analisa Kebutuhan Proses

Prioritas proses yang paling penting pada sistem diagnosa ini ialah bagaimana sistem dapat melakukan diagnosa penyakit tanaman cabai berdasarkan gejala yang diinputkan *user* kemudian dicocokkan dengan basis pengetahuan.

4.1.5. Analisa Kebutuhan Keluaran

Keluaran dari sistem ini menampilkan hasil dari diagnosa penyakit tanaman cabai menggunakan metode *Modified K-Nearest Neighbor*. Hasil diagnosa penyakit berdasarkan pada gejala-gejala penyakit yang telah dimasukkan pengguna pada proses diagnosa. Tampilan *output* dari sistem adalah nama penyakit tanaman cabai merah, cara penanggulangannya, dan hasil perhitungan proses diagnosa penyakit.

4.2. Perancangan Perangkat Lunak

Perancangan perangkat lunak bertujuan untuk menjelaskan relasi antar komponen-komponen pembangun sistem. Sistem pakar dirancang berbasis *web* dengan komponen-komponen berbentuk halaman *web* yang memiliki tombol navigasi untuk menghubungkan satu halaman menuju halaman lain sesuai kebutuhan sistem.

Perancangan komponen-komponen perangkat lunak pada sistem ini menggunakan diagram alir yang dijelaskan pada Tabel 4.3.

Tabel 4.3 Perancangan Perangkat Lunak

Nama Halaman	Keterangan	Tombol Navigasi
Halaman Utama	Halaman awal yang diakses <i>user</i>	1. Halaman Utama 2. Halaman Diagnosa 3. Halaman History 4. Halaman Informasi
Halaman History	Berisi informasi tentang <i>output</i> yang pernah diproses oleh sistem	1. Halaman Utama 2. Halaman Diagnosa
Halaman Diagnosa	Berisi formulir diagnosa penyakit cabai merah untuk diproses dalam perhitungan <i>Modified K-Nearest Neighbor</i>	1. Halaman Awal 2. Halaman Diagnosa 3. Halaman Histori 4. Halaman Informasi

Sumber: [Perancangan]

4.2.1 Database Diagnosa Penyakit Cabai Merah Menggunakan MKNN

Pada database diagnosa penyakit tanaman cabai merah terdapat 10 entitas yang digunakan yaitu, *datalatih*, *datarandem*, *datatest*, *eucladian*, *gejala*, *jarak*, *ndatalatih*, *ndatatest*, *opsiskor*, *penyakit*. Berikut penjelasan struktur database aplikasi diagnosa penyakit tanaman cabai merah antara lain :

1. Tabel *datalatih*

Tabel *datalatih* digunakan untuk menempatkan data *training*, berisi atribut G1 sampai G14, ID, dan *diagnosis*. Tabel *datalatih* dijelaskan pada tabel 4.4 berikut.

Tabel 4.4 Tabel *Datalatih*

Nama Kolom	Type data (Length)
ID	Int(11) (Primary Key)
G1	Int(11)
G2	Int(11)
G3	Int(11)
G4	Int(11)
G5	Int(11)
G6	Int(11)
G7	Int(11)
G8	Int(11)
G9	Int(11)

G10	Int(11)
G11	Int(11)
G12	Int(11)
G13	Int(11)
G14	Int(11)
Diagnosis	Int(11)

Sumber: [Perancangan]

2. Tabel datarandem

Tabel datarandem digunakan untuk mengambil data *training* secara acak, berisi atribut G1 sampai G14, ID, dan diagnosis. Tabel datarandem dijelaskan pada tabel 4.5 berikut.

Tabel 4.5 Tabel Datarandom

Nama Kolom	Type data (Length)
ID	Int(11) (Primary Key)
G1	Int(11)
G2	Int(11)
G3	Int(11)
G4	Int(11)
G5	Int(11)
G6	Int(11)
G7	Int(11)
G8	Int(11)
G9	Int(11)
G10	Int(11)
G11	Int(11)
G12	Int(11)
G13	Int(11)
G14	Int(11)
Diagnosis	Int(11)

Sumber: [Perancangan]

3. Tabel datatest

Tabel datatest digunakan untuk data *testing*, berisi atribut G1 sampai G14, ID, dan diagnosis. Tabel datatest dijelaskan pada tabel 4.6 berikut.

Tabel 4.6 Tabel Datatest

Nama Kolom	Type data (Length)
ID	Int(11) (Primary Key)
G1	Int(11)
G2	Int(11)
G3	Int(11)
G4	Int(11)
G5	Int(11)

G6	Int(11)
G7	Int(11)
G8	Int(11)
G9	Int(11)
G10	Int(11)
G11	Int(11)
G12	Int(11)
G13	Int(11)
G14	Int(11)
Diagnosis	Int(11)

Sumber: [Perancangan]

4. Tabel eucledian

Tabel eucledian digunakan untuk perhitungan jarak eucledian data *training*, berisi atribut IDparent, IDchild, Hasil, DiagParent, DiagChild dan poin. Tabel eucledian dijelaskan pada tabel 4.7 berikut.

Tabel 4.7 Tabel eucladian

Nama Kolom	Tipe data (Length)
IDparent	Int(11) (Primary Key)
IDchild	Int(11) (Primary Key)
Hasil	double
DiagParent	Int(11)
DiagChild	Int(11)
Poin	Int(2)

Sumber: [Perancangan]

5. Tabel gejala

Tabel gejala digunakan untuk nama gejala pada setiap kode gejala. berisi atribut kode, dan nama_gejala. Tabel gejala dijelaskan pada tabel 4.8 berikut.

Tabel 4.8 Tabel gejala

Nama Kolom	Tipe data (Length)
kode	varchar(4) (Primary Key)
nama_gejala	varchar(70)

Sumber: [Perancangan]

6. Tabel jarak

Tabel jarak digunakan untuk perhitungan jarak eucledian data *testing*, berisi atribut ID, Hasil, Validitas, Voting, dan diagnosis. Tabel jarak dijelaskan pada tabel 4.9 berikut.

Tabel 4.9 Tabel jarak

Nama Kolom	Tipe data (Length)
ID	Int(11) (Primary Key)
Hasil	double
Validitas	double
Voting	double
Diagnosis	Int(11)

Sumber: [Perancangan]

7. Tabel ndatalatih

Tabel ndatalatih digunakan untuk normalisasi data *training*, berisi atribut G1 sampai G14, ID, dan diagnosis. Tabel ndatalatih dijelaskan pada tabel 4.10 berikut.

Tabel 4.10 Tabel ndatalatih

Nama Kolom	Tipe data (Length)
ID	decimal(15,4) (Primary Key)
G1	decimal(15,4)
G2	decimal(15,4)
G3	decimal(15,4)
G4	decimal(15,4)
G5	decimal(15,4)
G6	decimal(15,4)
G7	decimal(15,4)
G8	decimal(15,4)
G9	decimal(15,4)
G10	decimal(15,4)
G11	decimal(15,4)
G12	decimal(15,4)
G13	decimal(15,4)
G14	decimal(15,4)
Diagnosis	Int(11)

Sumber: [Perancangan]

8. Tabel ndatatest

Tabel ndatatest digunakan untuk normalisasi data *testing*, berisi atribut G1 sampai G14, ID, dan diagnosis. Tabel ndatatest dijelaskan pada tabel 4.11 berikut.

Tabel 4.11 Tabel ndatatest

Nama Kolom	Tipe data (Length)
ID	decimal(15,4) (Primary Key)

G1	decimal(15,4)
G2	decimal(15,4)
G3	decimal(15,4)
G4	decimal(15,4)
G5	decimal(15,4)
G6	decimal(15,4)
G7	decimal(15,4)
G8	decimal(15,4)
G9	decimal(15,4)
G10	decimal(15,4)
G11	decimal(15,4)
G12	decimal(15,4)
G13	decimal(15,4)
G14	decimal(15,4)
Diagnosis	Int(11)

Sumber: [Perancangan]

9. Tabel opsiskor

Tabel opsiskor digunakan untuk opsi skor pada bobot gejala, berisi gejala, opsi, dan nilai. Tabel opsiskor dijelaskan pada tabel 4.12 berikut.

Tabel 4.12 Tabel opsiskor

Nama Kolom	Tipe data (Length)
gejala	varchar(4) (Primary Key)
opsi	varchar(30) (Primary Key)
nilai	int(11)

Sumber: [Perancangan]

10. Tabel Penyakit

Tabel penyakit digunakan untuk nama penyakit dan solusi dari setiap penyakit. Berisi atribut id_penyakit, nama_penyakit, dan solusi. Tabel penyakit dijelaskan pada tabel 4.13 berikut.

Tabel 4.13 Tabel penyakit

Nama Kolom	Tipe data (Length)
id_penyakit	int(11) (Primary Key)
nama_penyakit	varchar(70)
solusi	text

Sumber: [Perancangan]

4.2.2 Perhitungan Kasus Secara Manual

Perhitungan manual dibutuhkan guna mengetahui proses perhitungan sebenarnya yang terjadi di dalam sistem. Dengan adanya perhitungan manual maka akan mempermudah dalam pengaplikasian dalam pembuatan program.

Contoh kasus perhitungan:

Data diperoleh dari data yang ada di lapangan yang digunakan sebagai data *training* dan data *testing*. Terdapat 8 data *training* dan 1 data *testing*. Tabel 4.14 merupakan tabel data *training* yang didapat dilapangan. Data *training* tersebut diambil dari sebagian record yang berjumlah 450. Tabel 4.15 merupakan tabel data *testing*.

Tabel 4.14 Data training

Data	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14
1	60	0	0	0	20	0	0	20	0	90	50	0	0	0
2	0	0	0	0	0	0	0	20	0	50	90	0	0	0
3	60	90	90	0	20	0	0	10	0	0	0	0	0	0
4	0	50	50	0	10	0	0	0	0	0	0	0	0	0
5	0	0	0	90	20	0	0	0	90	0	0	0	60	50
6	0	0	0	0	10	0	0	0	90	0	0	0	60	50
7	0	0	0	0	0	90	90	20	0	0	0	90	30	0
8	0	0	0	0	0	0	50	0	0	0	0	50	30	0

Sumber: [Penelitian]

Tabel 4.15 Data Testing

Data	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14	Hasil
1	60	0	0	0	20	0	0	20	0	90	90	0	0	0	?

Sumber: [Penelitian]

Tabel 4.16 Range Data

	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14
Min	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Max	60	90	90	90	20	90	90	20	90	90	90	90	60	50
Range	60	90	90	90	20	90	90	20	90	90	90	90	60	50

Sumber: [Penelitian]

Perhitungan manual tahapan inferensi adalah sebagai berikut:

1. Normalisasi tabel pada data *training* dan data *testing*

Dalam tahap ini normalisasi data bertujuan agar data berada pada *range* [0,1] sehingga sebaran datanya berada pada *range* yang sama. Normalisasi dihitung dengan cara mengurangi nilai data latih x dengan nilai minimal dan membaginya dengan *range*. Berikut adalah contoh normalisasi data *training* pada G1 ditunjukkan pada kolom berwarna abu-abu pada tabel 4.17 menggunakan persamaan 2-1.

$$G1 = \frac{G1(x) - \min Value}{\max Value - \min Value}$$

$$G1 = \frac{60-0}{60} = 1$$

Tabel 4.17 dan tabel 4.18 merupakan hasil contoh yang diperoleh dalam melakukan normalisasi data *training* dan data *testing*.

Tabel 4.17 Tabel Normalisasi Data *Training*

Data	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14
1	1	0	0	0	1	0	0	1	0	1	0.5	0	0	0
2	0	0	0	0	0	0	0	1	0	0.5	1	0	0	0
3	1	1	0	0	1	0	0	0.5	0	0	0	0	0	0
4	0	0.5	1	0	0.5	0	0	0	0	0	0	0	0	0
5	0	0	0.5	1	1	0	0	0	1	0	0	0	1	1
6	0	0	0	0	0.5	0	0	0	1	0	0	0	1	1
7	0	0	0	0	0	1	1	1	0	0	0	1	0.5	0
8	0	0	0	0	0	0	0.5	0	0	0	0	0.5	0.5	0

Sumber: [Perancangan]

Tabel 4.18 Normalisasi Data *Testing*

Data	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14
1	1	0	0	0	1	0	0	1	0	1	1	0	0	0

Sumber: [Perancangan]

2. Menentukan nilai k atau tetangganya.

Tahap ini ditentukan nilai k atau tetangganya. K adalah jumlah tetangga terdekat yang digunakan. Dalam kasus ini peneliti mengambil k sebesar 3.

3. Mencari jarak *euclidian* antara data *training*

Dalam tahap ini mencari jarak *euclidian* antar data *training* menggunakan rumus *euclidian*. Menghitung jarak *Euclidian* dapat menggunakan persamaan 2-2. Contoh mencari jarak antara data *training* D(1,2) dalam kasus ini didapatkan dari tabel 4.19.

Tabel 4.19 Data *Training* D(1,2) setelah dinormalisasi

Data	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14
1	1	0	0	0	1	0	0	1	0	1	0.5	0	0	0
2	0	0	0	0	0	0	0	1	0	0.5	1	0	0	0

Sumber: [Perancangan]

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

$D(1,2)$

$$= \sqrt{((1-0)^2) + ((0-0)^2) + ((0-0)^2) + ((0-0)^2) + ((1-0)^2) + ((0-0)^2) + ((0-0)^2) + ((1-1)^2) + ((0-0)^2) + ((1-0.56)^2) + ((0.56-1)^2) + ((0-0)^2) + ((0-0)^2) + ((0-0)^2)}$$

$$= 1.5475$$

Contoh jarak *Euclidian* yang telah dihitung dalam kasus ini ditunjukkan pada Tabel 4.20.

Tabel 4.20 Contoh Hasil Perhitungan Jarak *Euclidian* Data *Training*

	D1	D2	D3	D4	D5	D6	D7	D8
D1	0	1.5475	1.8864	2.0435	2.7034	2.5609	2.5609	2.2750
D2	1.5475	0	2.3576	1.7821	2.7034	2.3576	2.1350	1.7821
D3	1.8864	2.3576	0	1.3766	2.6925	2.5465	2.7386	2.2621
D4	2.0435	1.7821	1.3766	0	2.2061	1.9019	2.2621	1.3170
D5	2.7034	2.7034	2.6925	2.2061	0	1.1180	2.8722	2.2061
D6	2.5609	2.3576	2.5465	1.9019	1.1180	0	2.5495	1.765
D7	2.5609	2.1350	2.7386	2.2621	2.8722	2.5495	0	1.5475
D8	2.2750	1.7821	2.2621	1.3170	2.2061	1.765	1.5475	0

Sumber: [Perancangan]

4. Menghitung validitas data *training*

Nilai validitas dihitung berdasarkan rata-rata dari nilai kedekatan data satu dengan tetangganya sebanyak k . Nilai kedekatan bernilai 0 dan 1. Nilai 0 diberikan jika data terdekat tidak dalam kategori, dan sebaliknya diberi nilai 1 jika dalam 1 kategori. Jarak terdekat pada data 1 adalah data 2 dalam 1 kategori yang sama, maka diberi nilai 1. Berikut ini adalah contoh perhitungan validitas pada D1 menggunakan persamaan 2-3.

$$\begin{aligned}\text{Validitas } (x=1) &= \frac{1}{k} \sum_{i=1}^k S(\text{label}(N_i(x))) \\ &= \frac{1}{3} \sum_{i=1}^3 S(\text{label}(N_i(x))) \\ &= \frac{1}{3} \times (1+0+1) \\ &= 0,66666\end{aligned}$$

Hasil lengkap ditunjukkan pada tabel 4.21.

Tabel 4.21 Tabel Validitas Data *Training*

No	K=1	K=2	K=3	Validitas
1	1	0	1	0.66666
2	1	0	0	0.33333
3	1	1	0	0.66666
4	0	1	0	0.33333
5	1	0	0	0.33333
6	1	0	0	0.33333
7	1	0	0	0.33333
8	0	1	0	0.33333

Sumber: [Perancangan]

5. Mencari jarak *euclidian* antara data *testing*

Jarak *euclidian* data *testing* didapat dengan menghitung jarak data *testing* tersebut dengan semua data *training*, berikut ini contoh perhitungan jarak *euclidian* data *testing* D(1,2) pada tabel 4.22 menggunakan persamaan 2-2.

Tabel 4.22 Data *Testing* untuk *Euclidian* Data *Testing* D(1,2)

Data	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14
1	1	0	0	0	1	0	0	1	0	1	1	0	0	0

Sumber: [Perancangan]

Tabel 4.23 Data Training untuk Euclidian Data Testing D(1,2)

Data	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14
2	1	0	0	0	1	0	0	1	0	1	0.5	0	0	0

Sumber: [Perancangan]

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

$$D(1,2) =$$

$$\sqrt{(((1-1)^2) + ((0-0)^2) + ((0-0)^2) + ((0-0)^2) + ((1-1)^2) + ((0-0)^2) + ((0-0)^2) + ((1-1)^2) + ((0-0)^2) + ((1-0.5)^2) + ((0-0)^2) + ((0-0)^2) + ((0-0)^2)}$$

$$= 0.44444444$$

Lakukan perhitungan yang sama untuk semua data *training*. Hasil perhitungan *euclidian data testing* 1 ditunjukkan pada tabel 4.24 berikut.

Tabel 4.24 Jarak Euclidian data testing 1

Data	Euclidian
1,2	0.444444444
1,196	1.482407118
1,218	2.061552813
1,402	2.206192184
1,434	2.828427125
1,602	2.692582404
1,668	2.692582404
1,878	2.42224776

Sumber: [Perancangan]

6. Menghitung Weight Voting

Nilai hasil *weight voting* digunakan untuk menentukan kelas dari data *testing*. Nilai *weight voting* dihitung dengan menggunakan persamaan 2-5. Berikut adalah contoh perhitungan *weight voting* data *testing* dengan menggunakan nilai validitas data 1:

$$W_{(i)} = Validitas(x) \times \frac{1}{d + 0.5}$$

$$\begin{aligned}
 W_{(1)} &= Validitas(1) \times \frac{1}{d + 0.5} \\
 &= 0.666666 \times \frac{1}{0.4 + 0.5} \\
 &= 0.70588
 \end{aligned}$$

Berikut ini adalah hasil perhitungan weight voting Data Testing 1 dijelaskan pada tabel 4.27.

Tabel 4.25 Perhitungan *Weight Voting*

Data	Weight Voting	Diagnosis
1,2	0.70588	Antraknosa
1,196	0.16814	Antraknosa
1,218	0.26025	Cercospora
1,402	0.123174	Cercospora
1,434	0.100147	Gemini
1,602	0.1044086	Gemini
1,668	0.1044086	Kerupuk
1,878	0.1140674	Kerupuk

Sumber: [Perancangan]

7. Mengurutkan Nilai *Weight Voting*

Setelah menghitung *weight voting*, hasil dari *weight voting* diurutkan dari yang terbesar ke terkecil dan karena nilai $k=3$ maka hanya diambil 3 nilai terbesar saja. Hasil pengurutan *weight voting* dijelaskan pada tabel 4.26.

Tabel 4.26 *Weight Voting* setelah Diurutkan

Data	Weight Voting	Diagnosis
1,2	0.70588	Antraknosa
1,218	0.26025	Cercospora
1,196	0.16814	Antraknosa

Sumber: [Perancangan]

Dari tabel 4.26 dapat diambil kesimpulan bahwa data testing 1 termasuk dalam kelas Antraknosa.

8. Hasil Diagnosis

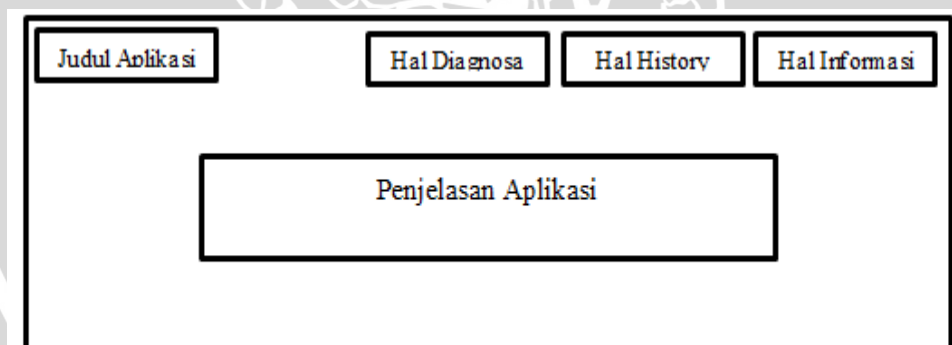
Tahap ini merupakan tahap terakhir yaitu menampilkan hasil diagnosa penyakit. Dalam kasus ini terdiagnosa Penyakit 1 atau Antraknosa yang menyerang cabai merah tersebut.

4.2.3 Antarmuka

Antarmuka merupakan mekanisme yang digunakan oleh *user* dan sistem pakar untuk berkomunikasi seperti melihat informasi yang ada di dalam sistem, menampilkan halaman utama, halam informasi, halaman formulir diagnosa, dan halaman hasil diagnosa, dan halaman history.

4.2.3.1 Antarmuka Tampilan Utama Aplikasi

Antarmuka digunakan oleh pengguna dan sistem untuk berkomunikasi seperti melihat informasi yang ada didalam sistem, menampilkan gejala-gejala yang nantinya akan dipilih oleh pengguna, kemudian sitem akan melakukan diagnosa sesuai gejala yang telah dipilih. Gambar 4.2 merupakan tampilan utama aplikasi.



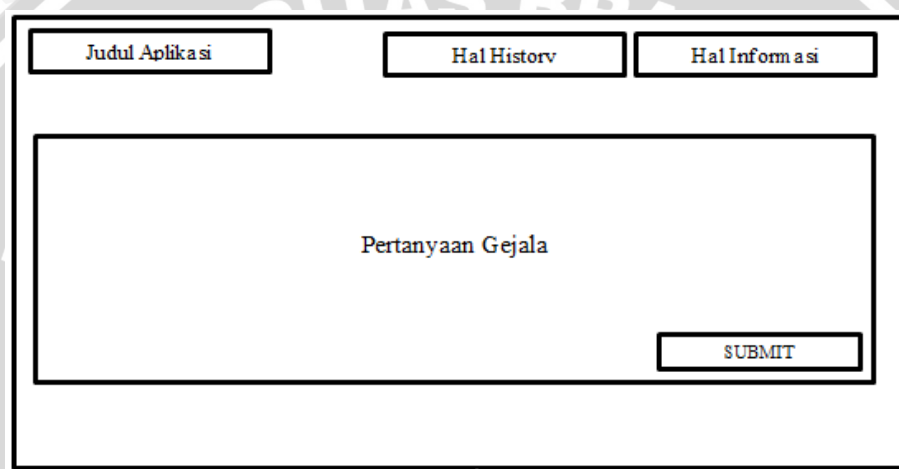
Gambar 4.2 Antarmuka Sistem
Sumber: [Perancangan]

Pada gambar 4.2 adalah rancangan antarmuka yang terdiri dari judul aplikasi, halaman utama, halaman diagnosa, halaman history, dan halaman informasi. Halaman utama berisi penjelasan aplikasi sistem diagnosa, halaman diagnosa berisi pertanyaan gejala-gejala yang nantinya akan dipilih oleh pengguna dan akan ditampilkan pada halaman diagnosa, halaman history berisi tentang

diagnosa yang telah di inputkan oleh pengguna sebelumnya, dan halaman informasi berisi informasi tentang penyakit tanaman cabai merah.

4.2.3.2 Antarmuka Tampilan Halaman Diagnosa

Antarmuka halaman diagnosa berisi pertanyaan-pertanyaan berupa gejala-gejala penyakit yang nantinya akan dipilih oleh pengguna. Pertanyaan-pertanyaan tersebut memiliki 3 bobot penilaian yaitu tidak, sedikit, dan banyak. Setelah pengguna selesai memilih jawaban maka akan langsung muncul hasil diagnosa penyakitnya. Tampilan halaman diagnosa digambarkan pada gambar 4.3.



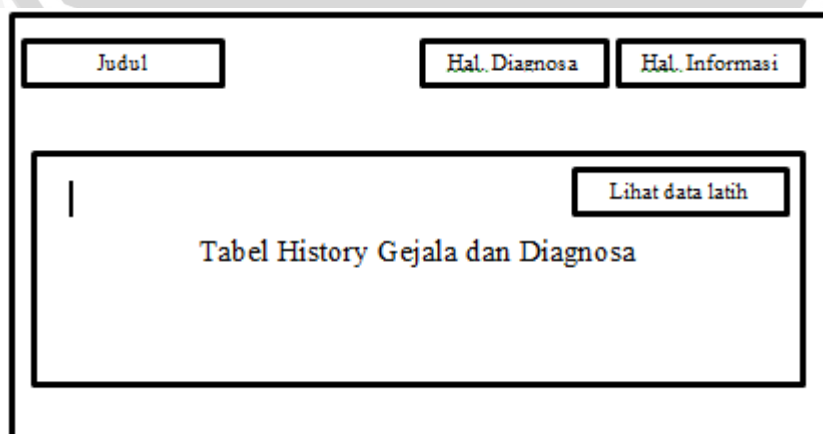
The screenshot shows a web interface for a diagnosis application. At the top, there are three navigation buttons: "Judul Aplikasi", "Hal Histori", and "Hal Informasi". Below these is a large rectangular area labeled "Pertanyaan Gejala". In the bottom right corner of this area is a button labeled "SUBMIT".

Gambar 4.3 Halaman Diagnosa

Sumber: [Perancangan]

4.2.3.3 Antarmuka Halaman History

Halaman history berisi data gejala dan hasil diagnosa yang pernah di inputkan oleh user sebelumnya. Halaman history berfungsi merekam hasil-hasil diagnosa. Tampilan halaman history digambarkan pada gambar 4.4.

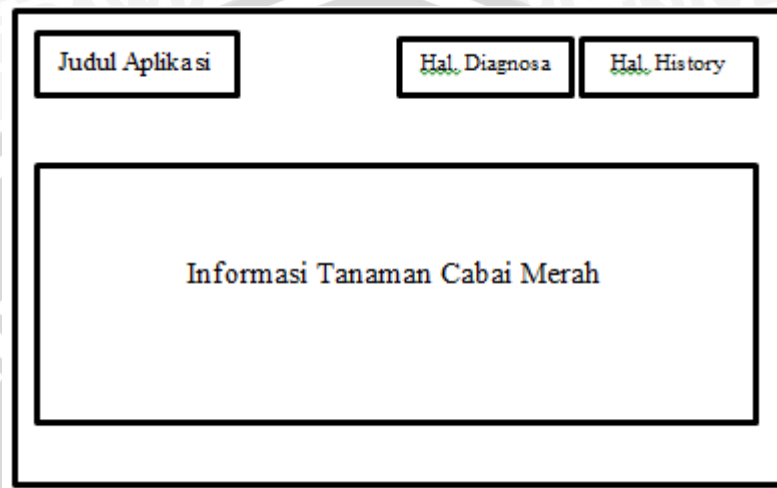


The screenshot shows a web interface for a history page. At the top, there are three navigation buttons: "Judul", "Hal. Diagnosa", and "Hal. Informasi". Below these is a large rectangular area labeled "Tabel History Gejala dan Diagnosa". In the top right corner of this area is a button labeled "Lihat data latih".

Gambar 4.4 Antarmuka Halaman History**Sumber:** [Perancangan]

4.2.3.4 Antarmuka Halaman Informasi

Halaman Informasi berisi tentang informasi tanaman cabai merah dan penyakit tanaman cabai merah. Tampilan halaman informasi digambarkan pada gambar 4.5.

**Gambar 4.5** Antarmuka Halaman Informasi**Sumber:** [Perancangan]

4.3 Desain Pengujian

Pada penelitian ini dilakukan dua buah pengujian, yakni pengujian fungsionalitas dan akurasi. Pengujian fungsionalitas adalah pengujian yang untuk mengetahui apakah sistem yang dibangun telah sesuai dengan kebutuhan-kebutuhan sistem yang telah ditentukan. Terdapat empat skenario pengujian fungsionalitas pada desain pengujian ini. Skenario pengujian fungsionalitas ditunjukkan pada Tabel 4.27.

Tabel 4.27 Skenario Pengujian Fungsionalitas

No	Skenario Pengujian	Hasil yang Diharapkan
1.	Mengklik tombol home	Sistem menampilkan halaman home.
2.	Mengklik tombol menu diagnosa	Sistem menampilkan halaman diagnosa.
3.	Mengisi formulir diagnosa dan mengklik tombol	Sistem menampilkan halaman hasil diagnosa dengan menyertakan gejala penyakit.

	diagnosa pada halaman diagnosa.	
4.	Menampilkan hasil diagnosa	Sistem dapat menampilkan hasil diagnosa dengan hasil perhitungan yang dilakukan dengan metode <i>Modified K-Nearest Neighbor</i> serta solusi pengendalian penyakit.
5.	Mengklik tombol menu History	Sistem menampilkan halaman riwayat yang berisi data yang pernah dimasukkan oleh pengguna.
6.	Mengklik tombol menu about	Sistem dapat menampilkan info tentang tanaman cabai dan about penulis.

Sumber: [Perancangan]

Tujuan dari pengujian akurasi adalah mengetahui jumlah kecocokan data antara hasil diagnosa sistem dengan hasil diagnosa pakar. Contoh pengujian akurasi ditunjukkan pada Tabel 4.28.

Tabel 4.28 Contoh Pengujian Akurasi

No	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14	Hasil Sistem	Hasil Pakar
1	Sedikit	Tidak	Tidak	Tidak	Sedikit	Tidak	Tidak	Sedikit	Tidak	Sedikit	Sedikit	Tidak	Tidak	Tidak	Antrak nosa	Antrak nosa
2	Sedikit	Tidak	Tidak	Tidak	Sedikit	Tidak	Tidak	Sedikit	Tidak	Sedikit	Tidak	Tidak	Tidak	Tidak	Cerco spora	Cerco spora
3	Sedikit	Tidak	Tidak	Tidak	Sedikit	Tidak	Tidak	Sedikit	Tidak	Tidak	Banyak	Tidak	Tidak	Tidak	Penyakit kerupuk	Penyakit kerupuk
4	Sedikit	Tidak	Tidak	Tidak	Sedikit	Tidak	Tidak	Sedikit	Tidak	Tidak	Sedikit	Tidak	Tidak	Tidak	Virus gemini	Virus gemini

Pengujian nilai k bertujuan untuk mengetahui presentase akurasi metode *Modified K-Nearest Neighbor* dengan mengganti nilai k. Contoh pengujian nilai k ditunjukkan pada tabel 4.29.

Tabel 4.29 Contoh Pengujian Nilai k

k	Akurasi (%)									Rata-Rata Akurasi (%)
	Kelompok Data Uji 1 (1-50)	Kelompok Data Uji 2 (51-100)	Kelompok Data Uji 3 (101-150)	Kelompok Data Uji 4 (151-200)	Kelompok Data Uji 5 (201-250)	Kelompok Data Uji 6 (251-300)	Kelompok Data Uji 7 (301-350)	Kelompok Data Uji 8 (351-400)	Kelompok Data Uji 9 (401-450)	
1	100%	85%	80%	100%	95%	85%	95%	80%	85%	89.4%

2	90%	85%	95%	90%	90%	75%	90%	95%	90%	89%
3	85%	95%	95%	85%	95%	90%	90%	80%	95%	91,111%
4	85%	90%	95%	80%	95%	100%	85%	85%	80%	87%
5	90%	85%	95%	90%	90%	95%	90%	90%	90%	90%

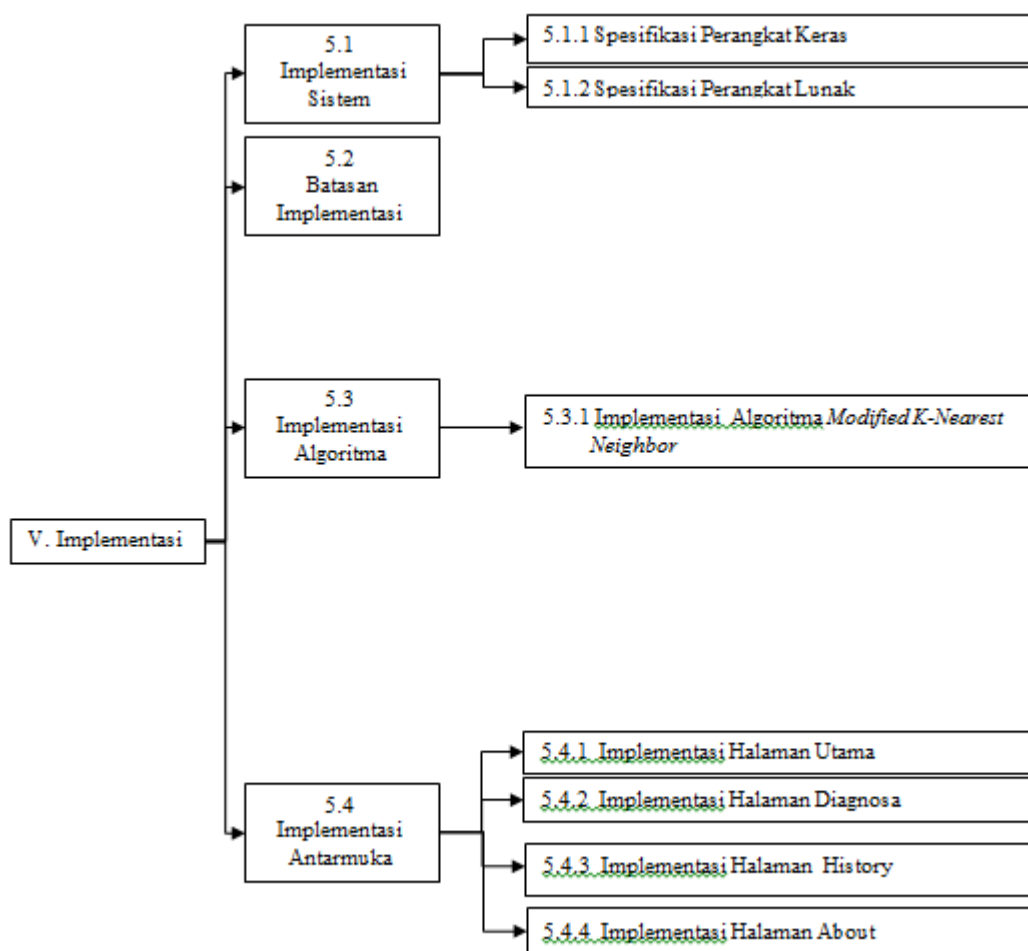
Sumber: [Perancangan]



BAB V

IMPLEMENTASI

Bab ini menjelaskan tentang implementasi pemodelan diagnosa penyakit tanaman cabai merah dengan Metode *Modified K-nearest neighbor*. Pembahasan pada bab ini terdiri dari penjelasan mengenai spesifikasi sistem, batasan-batasan implementasi, implementasi algoritma pada program, dan implementasi antarmuka. Tahapan-tahapan implementasi ditunjukkan pada Gambar 5.1.



Gambar 5.1 Pohon Implementasi

Sumber: [Implementasi]

5.1 Implementasi Sistem

Hasil analisa kebutuhan dan perancangan perangkat lunak akan menjadi acuan dalam melakukan implementasi sistem. Dalam implementasi sistem dibutuhkan spesifikasi perangkat yang memadai agar sistem yang dirancang dapat berfungsi sesuai kebutuhan. Spesifikasi perangkat sistem yang dibutuhkan terdiri atas spesifikasi perangkat keras dan spesifikasi perangkat lunak.

5.1.1 Spesifikasi Perangkat Keras

Pengembangan Diagnosa Penyakit Tanaman Cabai Merah Menggunakan Metode *Modified K-Nearest neighbor* menggunakan spesifikasi perangkat keras yang ditunjukkan pada Tabel 5.1.

Tabel 5.1 Spesifikasi Perangkat Keras

Nama Komponen	Spesifikasi
Prosesor	Intel® Core™ i5-2350M CPU @ 2.30 GHz
Memori	8192 MB
Kartu Grafis	AMD RADEON Graphics 2 GB
Harddisk	640 GB

Sumber : [Implementasi]

5.1.2 Spesifikasi Perangkat Lunak

Pengembangan Diagnosa Penyakit Tanaman Cabai Merah Menggunakan Metode *Modified K-Nearest Neighbor* menggunakan spesifikasi perangkat lunak yang ditunjukkan pada Tabel 5.2.

Tabel 5.2 Spesifikasi Perangkat Lunak

Nama	Spesifikasi
Sistem operasi	Microsoft Windows 7 Ultimate 64-bit
Bahasa pemrograman	HTML dan PHP
Tools Pemrograman	1. NetBeans IDE 7.2.1 2. XAMPP Control Panel v3.1.0 beta 6 3. Web Browser

Sumber : [Implementasi]

5.2 Batasan Implementasi

Batasan-batasan implementasi dari Diagnosa Penyakit Tanaman Cabai Merah Menggunakan Metode *Modified K-nearest neighbor* adalah sebagai berikut:

1. Sistem dirancang dengan ruang lingkup *Web Application* menggunakan bahasa pemrograman HTML dan PHP.
2. Data-data yang digunakan dalam sistem disimpan di *database* MySQL, namun di dalam sistem ini tidak dapat menambah data seperti gejala dengan alasan yang didasarkan pada pernyataan pakar Ferry Ardiansyah, S.P., yakni bahwa minimal semenjak pakar mulai menanam tanaman cabai merah pada tahun 1996, tidak lagi ditemukan penyakit utama baru pada tanaman cabai, sehingga fitur penambahan gejala tidak dibutuhkan di dalam sistem demi efisiensi penelitian. Pernyataan pakar ini dilampirkan pada halaman lampiran.
3. *Input* yang digunakan dalam sistem merupakan data-data gejala pada tanaman cabai merah yang dimasukkan oleh pengguna umum.
4. *Output* yang diterima pengguna adalah jenis penyakit, cara penanggulangan, dan hasil perhitungan dengan metode *Modified K-Nearest Neighbor*.
5. Kriteria yang digunakan dalam perhitungan system diagnose ini adalah gejala-gejala penyakit pada tanaman cabai merah.

5.3 Implementasi Algoritma

Implementasi Algoritma pada Diagnosa Penyakit Tanaman Cabai Merah Menggunakan Metode *Modified K-Nearest Neighbor*.

5.3.1 Implementasi Algoritma *Modified K-Nearest Neighbor*

Implementasi algoritma *Modified K-Nearest Neighbor* pada sistem ini yaitu proses normalisasi data, mencari euclidian data *training*, menghitung validitas data *training*, menghitung euclidian data *testing*, menghitung *weight voting*, mengurutkan nilai *weight voting*.

5.3.1.1 Implementasi Perhitungan Normalisasi Data

Implementasi perhitungan normalisasi data dibagi menjadi dua yaitu normalisasi data *training* dan normalisasi data *testing*. Perhitungan normalisasi

data yaitu mengurangi salah satu record dengan nilai *minimum* dan dibagi dengan nilai *range*. Nilai *range* yaitu nilai maksimum dikurangi dengan nilai minimum. Implementasi normalisasi data latih ditunjukkan pada gambar 5.2 dan implementasi normalisasi data uji ditunjukkan pada gambar 5.3.

```

1  mysql_query(" DELETE FROM pt_ndatalatih;");
2  mysql_query(" INSERT INTO pt_ndatalatih (
3  SELECT d.ID, (d.G1-m.G1)/(mx.G1-m.G1) G1, (d.G2-m.G2)/(mx.G2-m.G2)
4  G2, (d.G3-m.G3)/(mx.G3-m.G3) G3, (d.G4-m.G4)/(mx.G4-m.G4) G4, (d.G5-
5  m.G5)/(mx.G5-m.G5) G5, (d.G6-m.G6)/(mx.G6-m.G6) G6, (d.G7-
6  m.G7)/(mx.G7-m.G7) G7, (d.G8-m.G8)/(mx.G8-m.G8) G8, (d.G9-
7  m.G9)/(mx.G9-m.G9) G9, (d.G10-m.G10)/(mx.G10-m.G10) G10, (d.G11-
8  m.G11)/(mx.G11-m.G11) G11, (d.G12-m.G12)/(mx.G12-m.G12)
9  G12, (d.G13-m.G13)/(mx.G13-m.G13) G13, (d.G14-m.G14)/(mx.G14-m.G14)
10 G14, d.Diagnosis
11 FROM pt_datarandem d, (SELECT min(G1) G1,min(G2) G2,min(G3)
12 G3,min(G4) G4,min(G5) G5,min(G6) G6,min(G7) G7,min(G8) G8,min(G9)
13 G9,min(G10) G10,min(G11) G11,min(G12) G12,min(G13) G13,min(G14)
14 G14 FROM pt_datalatih ) m,
15 (SELECT max(G1) G1,max(G2) G2,max(G3) G3,max(G4) G4,max(G5)
16 G5,max(G6) G6,max(G7) G7,max(G8) G8,max(G9) G9,max(G10)
17 G10,max(G11) G11,max(G12) G12,max(G13) G13,max(G14) G14 FROM
18 pt_datalatih ) mx
); ");

```

Gambar 5.2 Implementasi Proses Perhitungan Normalisasi Data Latih

Sumber: [Implementasi]

Penjelasan *source code* pada gambar 5.2 yaitu:

1. Baris 1 yaitu untuk menghapus isi tabel *ndata_latih*.
2. Baris 2-18 yaitu proses perhitungan normalisasi data latih dan disimpan dalam tabel *ndata_latih*.

```

1  if($iddata==0) $kondisi = "ORDER BY ID DESC LIMIT 0,1";
2  else $kondisi = "WHERE ID = ".$iddata;
3  mysql_query(" DELETE FROM pt_ndatatest;");
4  mysql_query(" INSERT INTO pt_ndatatest (
5  SELECT d.ID, (d.G1-m.G1)/(mx.G1-m.G1) G1, (d.G2-m.G2)/(mx.G2-m.G2)
6  G2, (d.G3-m.G3)/(mx.G3-m.G3) G3, (d.G4-m.G4)/(mx.G4-m.G4) G4, (d.G5-
7  m.G5)/(mx.G5-m.G5) G5, (d.G6-m.G6)/(mx.G6-m.G6) G6, (d.G7-
8  m.G7)/(mx.G7-m.G7) G7, (d.G8-m.G8)/(mx.G8-m.G8) G8, (d.G9-
9  m.G9)/(mx.G9-m.G9) G9, (d.G10-m.G10)/(mx.G10-m.G10) G10, (d.G11-
10 m.G11)/(mx.G11-m.G11) G11, (d.G12-m.G12)/(mx.G12-m.G12)
11 G12, (d.G13-m.G13)/(mx.G13-m.G13) G13, (d.G14-m.G14)/(mx.G14-m.G14)
12 G14, d.Diagnosis FROM (select * FROM pt_datatest ".$kondisi.")
13 d, (SELECT min(G1) G1,min(G2) G2,min(G3) G3,min(G4) G4,min(G5)
14 G5,min(G6) G6,min(G7) G7,min(G8) G8,min(G9) G9,min(G10)
15 G10,min(G11) G11,min(G12) G12,min(G13) G13,min(G14) G14 FROM
16 pt_datalatih ) m, (SELECT max(G1) G1,max(G2) G2,max(G3)
17 G3,max(G4) G4,max(G5) G5,max(G6) G6,max(G7) G7,max(G8) G8,max(G9)
18 G9,max(G10) G10,max(G11) G11,max(G12) G12,max(G13) G13,max(G14)
19 G14 FROM pt_datalatih ) mx); ");

```

19	}
20	
	function setJarak(){
	mysql_query("DELETE FROM pt_jarak;");
	mysql_query(" INSERT INTO pt_jarak (SELECT dl.ID,
	SQRT(POW(dt.G1-dl.G1,2)+POW(dt.G2-dl.G2,2)+POW(dt.G3-
	dl.G3,2)+POW(dt.G4-dl.G4,2)+POW(dt.G5-dl.G5,2)+POW(dt.G6-
	dl.G6,2)+POW(dt.G7-dl.G7,2)+POW(dt.G8-dl.G8,2)+POW(dt.G9-
	dl.G9,2)+POW(dt.G10-dl.G10,2)+POW(dt.G11-dl.G11,2)+POW(dt.G12-
	dl.G12,2)+POW(dt.G13-dl.G13,2)+POW(dt.G14-dl.G14,2)) Hasil,
	0,0,dl.Diagnosis FROM pt_ndatalatih dl, pt_ndatatest dt);");

Gambar 5.3 Implementasi Proses Perhitungan Normalisasi Data Testing

Sumber: [Implementasi]

Penjelasan *source code* pada gambar 5.3 yaitu:

1. Baris 1 yaitu untuk pengecekan apakah data didalam tabel kosong.
2. Baris 2-20 yaitu perhitungan normalisasi data *testing*.

5.3.1.2 Implementasi Perhitungan Euclidian Data Training

Implementasi perhitungan *euclidian* data *training* ditunjukkan pada gambar 5.4.

1	mysql_query("DELETE FROM pt_eucladian;");
2	mysql_query(" INSERT INTO pt_eucladian (SELECT r1.ID, r2.ID,
3	SQRT(POW(r1.G1-r2.G1,2)+POW(r1.G2-r2.G2,2)+POW(r1.G3-
4	r2.G3,2)+POW(r1.G4-r2.G4,2)+POW(r1.G5-r2.G5,2)+POW(r1.G6-
5	r2.G6,2)+POW(r1.G7-r2.G7,2)+POW(r1.G8-r2.G8,2)+POW(r1.G9-
6	r2.G9,2)+POW(r1.G10-r2.G10,2)+POW(r1.G11-r2.G11,2)+POW(r1.G12-
7	r2.G12,2)+POW(r1.G13-r2.G13,2)+POW(r1.G14-r2.G14,2)) hasil,
8	r1.Diagnosis, r2.Diagnosis, if(r1.Diagnosis=r2.Diagnosis,1,0)
9	poin
10	FROM pt_ndatalatih r1, pt_ndatalatih r2);");
11	}
12	

Gambar 5.4 Implementasi Perhitungan Euclidian Data Training

Sumber: [Implementasi]

Penjelasan *source code* pada gambar 5.4 yaitu:

1. Baris 1 untuk mengosongkan tabel *pt_eucladian*.
2. Baris 2-11 untuk perhitungan *euclidian* data *training* kemudian disimpan di tabel *pt_eucladian*.

5.3.1.3 Implementasi Perhitungan Validitas Data Training

Perhitungan nilai validitas dihitung berdasarkan rata-rata dari nilai kedekatan data satu dengan tetangganya sebanyak *k*. Implementasi perhitungan validitas data *training* ditunjukkan pada gambar 5.5.

```

1 $data = mysql_query(" SELECT ID FROM pt_datarandem;  ");
2 while($row = mysql_fetch_row($data)){
3     $jum =0;
4     $perdata = mysql_query(" select poin from pt_eucladian WHERE
5     IDparent = ".$row[0]. " and Hasil >0 ORDER BY Hasil LIMIT
6     0, ".$k."");
7     while($row1 = mysql_fetch_row($perdata)){
8         $jum += $row1[0];
9     }
10    mysql_query("UPDATE pt_jarak SET Validitas = ".$jum/$k)." WHERE
11    ID = ".$row[0]);
12 }
13 mysql_query("UPDATE pt_jarak SET Voting= Validitas/(Hasil+0.5)");

```

Gambar 5.5 Implementasi Perhitungan Validitas Data *Training*

Sumber: [Implementasi]

Penjelasan *source code* pada gambar 5.5 yaitu:

1. Baris 1 yaitu mengambil ID dari tabel datarandem
2. Baris 2-13 yaitu untuk melakukan perhitungan validitas setiap data training.

5.3.1.4 Implementasi Perhitungan *Euclidian Data Testing*

Perhitungan jarak *euclidian* data *testing* didapat dengan menghitung jarak data *testing* dengan semua data *training*. Implementasi perhitungan jarak *euclidian* data *testing* ditunjukkan pada gambar 5.6.

```

1 mysql_query("DELETE FROM pt_jarak;");
2 mysql_query(" INSERT INTO pt_jarak ( SELECT dl.ID,
3 SQRT(POW(dt.G1-dl.G1,2)+POW(dt.G2-dl.G2,2)+POW(dt.G3-
4 dl.G3,2)+POW(dt.G4-dl.G4,2)+POW(dt.G5-dl.G5,2)+POW(dt.G6-
5 dl.G6,2)+POW(dt.G7-dl.G7,2)+POW(dt.G8-dl.G8,2)+POW(dt.G9-
6 dl.G9,2)+POW(dt.G10-dl.G10,2)+POW(dt.G11-dl.G11,2)+POW(dt.G12-
7 dl.G12,2)+POW(dt.G13-dl.G13,2)+POW(dt.G14-dl.G14,2)) Hasil,
8 0,0,dl.Diagnosis FROM pt_ndatalatih dl, pt_ndatatest dt);");
9

```

Gambar 5.6 Implementasi Perhitungan *Euclidian Data Testing*

Sumber: [Implementasi]

Penjelasan *source code* pada gambar 5.6 yaitu:

1. Baris 1 yaitu mengosongkan tabel jarak.
2. Baris 2-9 yaitu untuk menghitung perhitungan *euclidian* data testing.

5.3.1.5 Implementasi Perhitungan *Weight Voting*

Implementasi perhitungan *weight voting* ditunjukkan pada gambar 5.7.

```

1 $hasil = array();
2 $data = mysql_query(" SELECT a.Diagnosis,SUM(a.Voting) dhasil
3 FROM (SELECT * FROM pt_jarak ORDER BY Voting DESC LIMIT 0, ".$k.")
4 a GROUP BY a.Diagnosis ORDER BY dhasil DESC");
5 while($row = mysql_fetch_row($data)){
6 $hasil[] = array('diag'=>$row[0], 'hasil'=>$row[1]);
7 }
8 return $hasil;
9

```

Gambar 5.7 Implementasi Perhitungan *Weight Voting*

Sumber: [Implementasi]

Penjelasan *source code* pada gambar 5.7 yaitu baris 1-8 yaitu untuk melakukan perhitungan *weight voting*.

5.4 Implementasi Antarmuka

Antarmuka dalam pembuatan Diagnosa Penyakit Tanaman Cabai Merah Menggunakan Metode *Modified K-Nearest Neighbor* berfungsi sebagai sarana interaksi pengguna dengan sistem.

5.4.1 Implementasi Antarmuka Halaman Utama

Halaman utama merupakan halaman awal yang diakses pengguna saat menggunakan sistem diagnosa, pada halaman ini terdapat judul, logo, deskripsi sistem, tombol menuju halaman diagnosa, tombol menuju halaman history, tombol menuju halaman dan tombol menuju halaman about. Tombol menu diagnosa akan mengantarkan ke halaman diagnosa. Halaman history akan mengarahkan ke halaman riwayat diagnosa. Sedangkan halaman about akan mengantarkan ke halaman informasi yang berisi tentang tanaman cabai beserta penyakit dan penanggulangannya. Implementasi antarmuka halaman utama ditunjukkan pada Gambar 5.8.



Gambar 5.8 Implementasi Halaman Utama

Sumber: [Implementasi]

5.4.2 Implementasi Halaman Diagnosa

Halaman diagnosa berisi formulir kuisioner yang diajukan sistem kepada pengguna berupa gejala-gejala penyakit tanaman cabai merah dengan opsi jawaban berupa tidak, sedikit, dan banyak yang nantinya akan diproses menuju halaman hasil diagnosa. Tombol next digunakan untuk ke pertanyaan selanjutnya. Setelah selesai menjawab kuisioner, sistem akan menampilkan hasil diagnosa berdasarkan jawaban kuisioner yang telah dipilih oleh pengguna. Hasil diagnosa menggunakan perhitungan dengan menggunakan metode *modified k-nearest neighbor*. Hasil diagnosa berisi nilai validitas, nilai jarak, nilai weight voting, pengurutan nilai weight voting, jenis penyakit tanaman yang menyerang tanaman cabai merah, dan solusi pengendalian terhadap penyakit tersebut. Implementasi antarmuka halaman diagnosa ditunjukkan pada Gambar 5.9 dan hasil diagnosa ditunjukkan pada gambar 5.10.

Data Gejala Penyakit

Data K

Data Latih Random

Gejala 1 : Bercak kering pada daun

☒ Tidak
☐ Sedikit
☐ Banyak

Gejala 2 : Bercak daun berbentuk bulat dan sobek

☒ Tidak
☐ Sedikit
☐ Banyak

Next

Gambar 5.9 Implementasi Halaman Diagnosa

Sumber: [Implementasi]

CHILILOGY

Diagnosa History About

106	0.27355604899468955	CERCOSPORA	436	0.27873575409573176	PENYAKIT KERUPUK
202	0.3581285802046352	CERCOSPORA	11	0.2737599911748574	ANTRAKNOSA
221	0.28827803527635276	VIRUS GEMINI	106	0.27355604899468955	CERCOSPORA
259	0.2908532987422867	VIRUS GEMINI	439	0.27283251051278425	PENYAKIT KERUPUK
265	0.28905820555501914	VIRUS GEMINI	265	0.28905820555501914	VIRUS GEMINI
287	0.3161235158393548	VIRUS GEMINI	12	0.2679913826984485	ANTRAKNOSA
289	0.260380440696038	VIRUS GEMINI	10	0.2679913826984485	ANTRAKNOSA
376	0.28827803527635276	PENYAKIT KERUPUK	289	0.260380440696038	VIRUS GEMINI
388	0.305020198837558	PENYAKIT KERUPUK	396	0.24979788969697542	PENYAKIT KERUPUK
396	0.24979788969697542	PENYAKIT KERUPUK	127	0.22876514912816848	CERCOSPORA
436	0.27873575409573176	PENYAKIT KERUPUK	159	0.2276945473771653	CERCOSPORA
439	0.27283251051278425	PENYAKIT KERUPUK	16	0.21620852645726454	ANTRAKNOSA

Nilai Diagnosis

Penyakit	Total Weight Voting
CERCOSPORA	0.7312132461330381
VIRUS GEMINI	0.8089768143816415
PENYAKIT KERUPUK	0.305020198837558

Penyakit Tanaman : CERCOSPORA

Solusi

Solusi yang dapat dilakukan antara lain disemprot dengan fungisida, misalnya tiofanat, Velimek, dan Delsene. Kemudian yang kedua, daun yang berguguran di tanah segera diambil dan dibakar supaya tidak menular pada tanaman lain

Gambar 5.10 Implementasi Halaman Hasil Diagnosa

Sumber: [Implementasi]

5.4.3 Implementasi Halaman History

Halaman history berisi data gejala, penyakit yang pernah di masukkan pengguna, dan lihat data *training*. Implementasi halaman history ditunjukkan pada gambar 5.11.

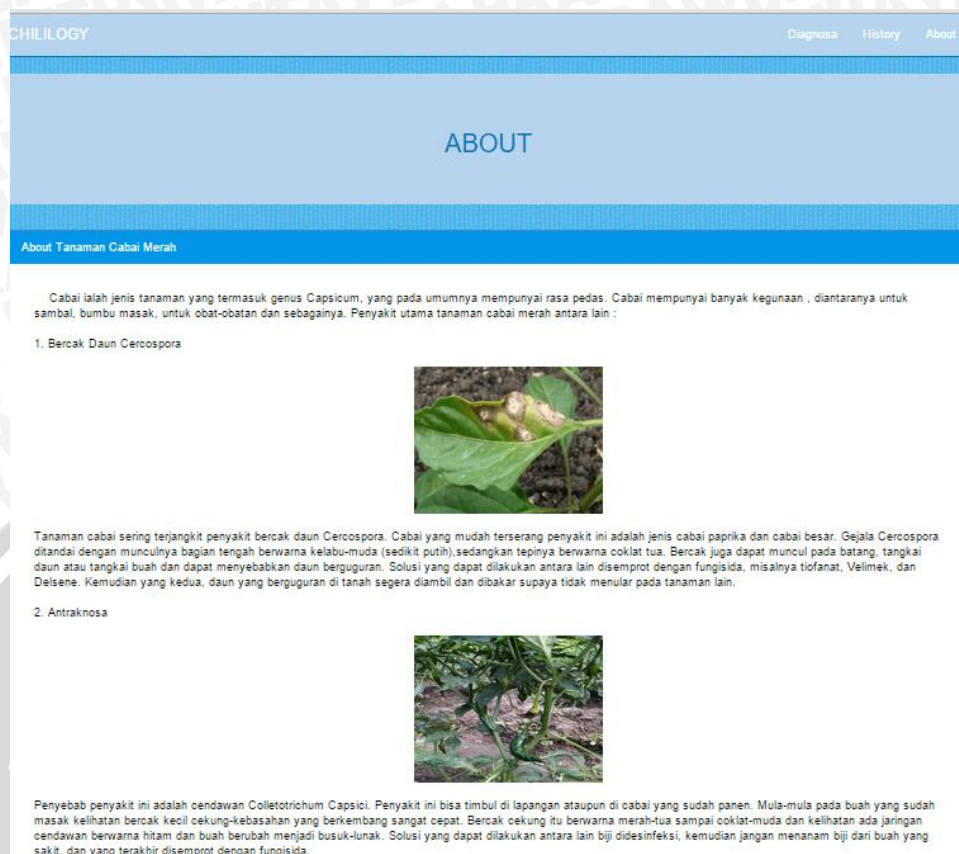
No	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14	Penyakit
1	Tidak	Sedikit	Banyak	Tidak	Tidak	Tidak	Tidak	Banyak	Banyak	Tidak	Tidak	Sedikit	Tidak	Tidak	CERCOSPORA
2	Tidak	Banyak	Sedikit	Sedikit	Tidak	Banyak	Tidak	Banyak	Banyak	Tidak	Tidak	Sedikit	Tidak	Tidak	PENYAKIT KERUPUK
3	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Banyak	Tidak	Tidak	PENYAKIT KERUPUK
4	Tidak	Tidak	Tidak	Tidak	Banyak	Banyak	Banyak	Banyak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	PENYAKIT KERUPUK
5	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Banyak	VIRUS GEMINI
6	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Banyak	VIRUS GEMINI
7	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Banyak	VIRUS GEMINI
8	Tidak	Tidak	Tidak	Tidak	Banyak	Banyak	Banyak	Banyak	Banyak	Tidak	Tidak	Tidak	Tidak	Tidak	PENYAKIT KERUPUK
9	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Banyak	Banyak	Tidak	Tidak	Tidak	ANTRAKNOSA
10	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Banyak	Banyak	Tidak	Tidak	Tidak	ANTRAKNOSA
11	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Banyak	Banyak	Tidak	Tidak	Tidak	ANTRAKNOSA
12	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Banyak	Banyak	Tidak	Tidak	Tidak	ANTRAKNOSA
13	Tidak	Tidak	Tidak	Tidak	Tidak	Banyak	Tidak	Sedikit	Tidak	Banyak	Banyak	Sedikit	Tidak	Banyak	PENYAKIT KERUPUK
14	Tidak	Tidak	Tidak	Tidak	Tidak	Banyak	Tidak	Sedikit	Tidak	Banyak	Banyak	Sedikit	Tidak	Banyak	PENYAKIT KERUPUK
15	Tidak	Banyak	Tidak	Sedikit	Tidak	Sedikit	Tidak	Banyak	Tidak	Sedikit	Tidak	Banyak	Tidak	Tidak	PENYAKIT KERUPUK
16	Banyak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	CERCOSPORA
17	Banyak	Sedikit	Sedikit	Banyak	Sedikit	Banyak	Banyak	Banyak	Sedikit	Tidak	Tidak	Sedikit	Tidak	Tidak	PENYAKIT KERUPUK
18	Banyak	Tidak	Banyak	Tidak	Tidak	Tidak	Tidak	Banyak	Tidak	Tidak	Banyak	Banyak	Sedikit	Sedikit	CERCOSPORA
19	Sedikit	Tidak	Tidak	Tidak	Sedikit	Tidak	Tidak	Sedikit	Tidak	Sedikit	Sedikit	Tidak	Tidak	Tidak	ANTRAKNOSA

Gambar 5.11 Implementasi Halaman History

Sumber: [Implementasi]

5.4.4 Implementasi Halaman About

Halaman about berisi informasi tentang tanaman cabai merah, penyakit cabai merah dan penanggulangan penyakit cabai merah. Implementasi halaman info ditunjukkan pada gambar 5.12.

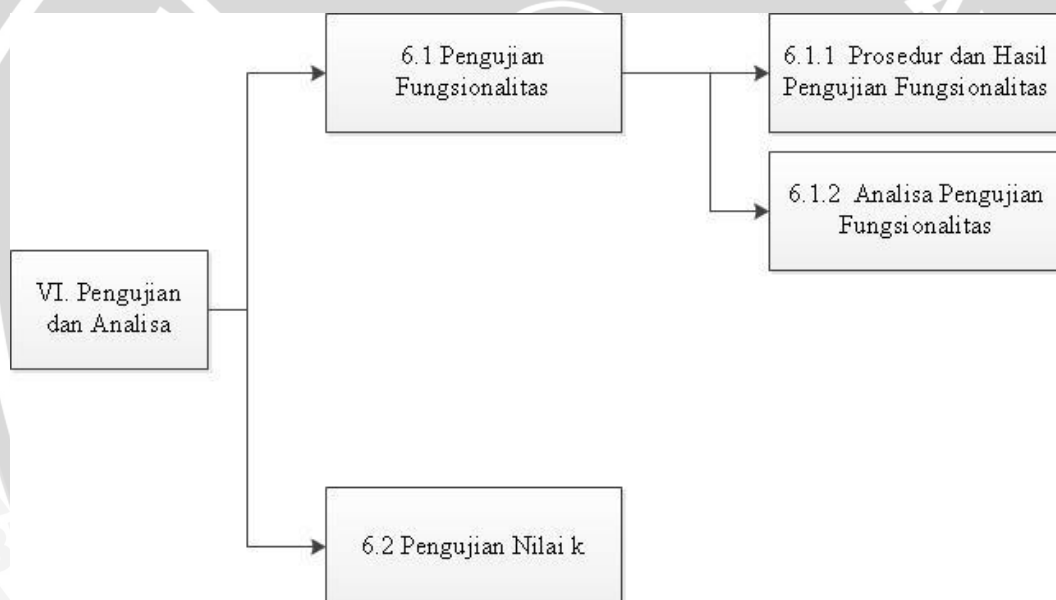


Gambar 5.12 Implementasi Halaman About
Sumber: [Implementasi]

BAB VI

PENGUJIAN DAN ANALISA

Pada bab ini dibahas mengenai prosedur dan hasil pengujian Diagnosa Penyakit Tanaman Cabai Merah Menggunakan Metode *Modified K-Nearest Neighbor*. Proses pengujian dilakukan melalui tiga tahapan, yakni pengujian fungsionalitas, pengujian akurasi, dan pengujian nilai k. Pengujian fungsionalitas dilakukan untuk mengetahui apakah sistem yang dirancang telah sesuai dengan kebutuhan sistem yang diharapkan. Pengujian akurasi dilakukan untuk mengetahui tingkat akurasi sistem dengan cara membandingkan hasil keluaran sistem dengan hasil diagnosa dari pakar. Pohon pengujian dan analisa ditunjukkan pada Gambar 6.1.



Gambar 6.1 Pohon Pengujian dan Analisa

Sumber: [Pengujian dan Analisa]

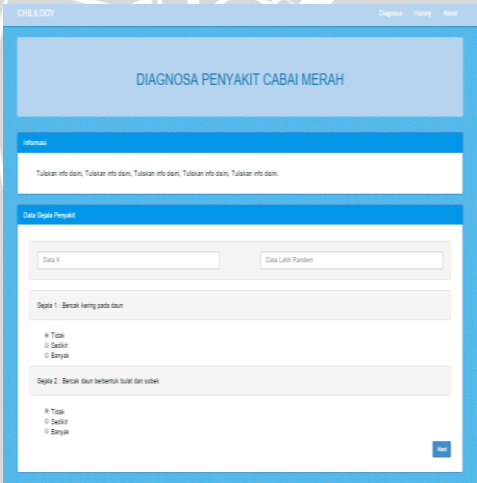
6.1 Pengujian Fungsionalitas

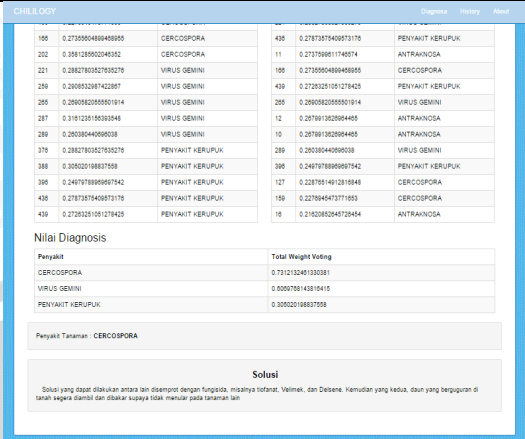
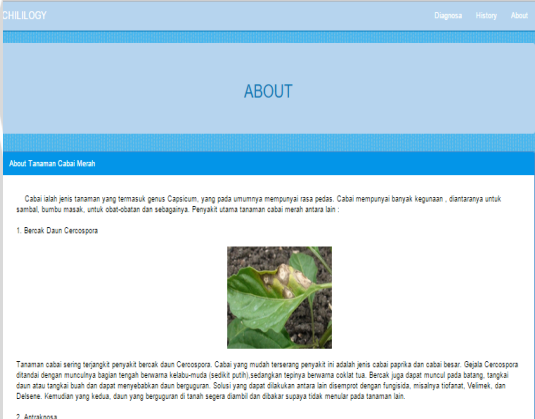
Pengujian fungsionalitas adalah pengujian yang dilakukan terhadap sistem dengan tujuan mengetahui apakah sistem yang dirancang telah memenuhi daftar kebutuhan sistem yang diharapkan.

6.1.1 Prosedur dan Hasil Pengujian Fungsionalitas

Pengujian fungsionalitas dilakukan dengan membuat kasus uji untuk setiap daftar kebutuhan sistem yang telah dirancang. Hasil pengujian fungsionalitas ditunjukkan pada Tabel 6.1.

Tabel 6.1 Hasil Pengujian Fungsionalitas

No	Skenario Pengujian	Hasil yang diharapkan	Hasil yang didapat	Status Validasi
1.	Mengklik tombol home	Sistem mampu menampilkan halaman utama dengan benar.	Sistem berhasil menampilkan halaman utama dengan benar. 	Valid
2.	Mengklik menu diagnosa	Sistem mampu menampilkan halaman diagnosa dengan benar.	Sistem berhasil menampilkan halaman diagnosa dengan benar. 	Valid
3.	Menampilkan hasil diagnosa	Sistem mampu menampilkan hasil	Sistem berhasil menampilkan hasil diagnosa.	Valid

		diagnosa.		
5.	Mengklik tombol History	Sistem mampu menampilkan halaman history.		Valid
6.	Mengklik Menu About	Sistem mampu menampilkan halaman about		Valid

Sumber: [Pengujian]

6.1.2 Analisa Pengujian Fungsionalitas

Analisa hasil pengujian fungsionalitas dilakukan dengan membandingkan kesesuaian antara hasil yang diharapkan dengan hasil yang didapat. Hasil pengujian yang ditunjukkan pada Tabel 6.1 memiliki tingkat kesesuaian 100%, sehingga dapat disimpulkan bahwa fungsionalitas dari Diagnosa Penyakit Tanaman Cabai Merah Menggunakan Metode *Modified K-Nearest Neighbor* berjalan dengan benar.

6.2 Pengujian Nilai k

Pengujian nilai k dilakukan dengan tujuan untuk mengetahui pengaruh nilai k terhadap akurasi metode *Modified K-Nearest Neighbor* dimana data yang digunakan yaitu data kelas seimbang yang dilakukan secara acak dan untuk setiap nilai k dilakukan pengujian sebanyak 10 kali.

6.2.1 Pengujian Pada 20 Data Latih

Pengujian nilai k dilakukan dengan cara mengganti nilai k=10, k=11, k=12, k=13, k=14, k=15, k=16, k=17, k=18 dan k=19, diuji menggunakan 20 data latih dan menggunakan 40 data uji yang bersifat tetap. Hasil rata-rata pengujian untuk setiap nilai k menggunakan 20 data latih ditunjukkan pada tabel 6.2

Tabel 6.2 Hasil Pengujian Nilai k pada 20 Data latih

Data Latih	k	Akurasi
20	10	82.16%
	11	95.83%
	12	96.66%
	13	95.83%
	14	95.83%
	15	96.66%
	16	95.416%
	17	95.416%
	18	95.416%
	19	96.66%
Akurasi Rata-Rata		94.588%

Sumber: [Pengujian]

6.2.2 Pengujian Pada 40 Data Latih

Pengujian nilai k dilakukan dengan cara mengganti nilai k=10, k=11, k=12, k=13, k=14, k=15, k=16, k=17, k=18 dan k=19, diuji menggunakan 40 data latih dan menggunakan 40 data uji yang bersifat tetap. Hasil pengujian nilai k menggunakan 40 data latih ditunjukkan pada tabel 6.3.

Tabel 6.3 Hasil Pengujian Nilai k pada 40 Data latih

Data Latih	k	Akurasi
40	10	95.41%
	11	91.66%
	12	95.41%
	13	95.83%
	14	98.33%
	15	95.41%
	16	94.58%
	17	93.33%
	18	86.67%
	19	99.58%
Akurasi Rata-Rata		94.621%

Sumber: [Pengujian]

6.2.3 Pengujian Pada 60 Data Latih

Pengujian nilai k dilakukan dengan cara mengganti nilai k=10, k=11, k=12, k=13, k=14, k=15, k=16, k=17, k=18 dan k=19, diuji menggunakan 60 data latih dan menggunakan 40 data uji yang bersifat tetap. Hasil pengujian nilai k menggunakan 60 data latih ditunjukkan pada tabel 6.4.

Tabel 6.4 Hasil Pengujian Nilai k pada 60 Data latih

Data Latih	k	Akurasi
	10	94.58%
	11	90.83%
	12	95.83%
	13	93.75%
	14	86.25%
	15	92.5%
	16	95.83%

	17	96.25%
	18	95%
	19	95.83%
Akurasi Rata-Rata		93.665%

Sumber: [Pengujian]

6.2.4 Pengujian Terhadap Nilai K Terbaik

Pengujian nilai k dilakukan dengan menggunakan nilai k terbaik yang didapat pada pengujian data latih sebelumnya, yaitu $k=19$. Untuk data latih yang digunakan adalah 20, 40, dan 60. Pengujian pada data latih 20 menghasilkan nilai akurasi tertinggi sebesar 96.66%, . Pengujian pada data latih 40 menghasilkan nilai akurasi tertinggi sebesar 99.58%, dan . Pengujian pada data latih 60 menghasilkan nilai akurasi tertinggi sebesar 95.93%. Hasil uji coba data latih dengan nilai k terbaik ditunjukkan pada tabel 6.5.

Tabel 6.5 Hasil Pengujian Nilai K Terbaik

Nilai K	Jumlah Data Latih	Akurasi
19	20	96.66%
	40	99.58%
	60	95.93%

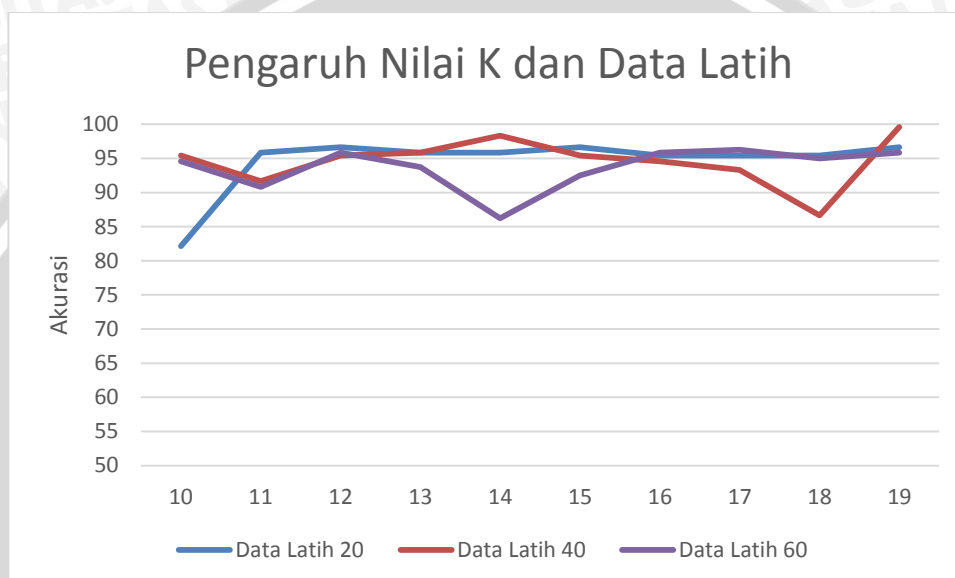
Sumber: [Pengujian]

6.2.5 Analisa Hasil Pengujian Nilai k

Berdasarkan hasil pengujian pada tabel 6.2, 6.3, dan 6.4 menunjukkan perubahan data latih memberikan pengaruh terhadap nilai akurasi yang dihasilkan. Akurasi pada tabel 6.2, 6.3, dan 6.4 cenderung tidak stabil dan akurasi tertinggi terdapat pada $k=19$ dengan akurasi 97.36%. Hal ini dikarenakan adanya kelas yang mendominasi pada saat pengambilan data latih secara acak yang dilakukan sebanyak 10 kali dan data uji memiliki data latih yang berbeda setiap kali dilakukan pengujian. Sehingga nilai k tidak terlalu berpengaruh pada akurasi ketika pengambilan data latih dilakukan secara acak dan setiap pengujian memiliki data latih yang berbeda.

Berbeda dengan hasil analisa pada kajian pustaka yang dikaji pada penelitian ini, yaitu penelitian zainuddin tentang diagnosa penyakit kedelai menggunakan MKNN. Analisa pada penelitian tersebut memiliki tingkat akurasi

yang sangat dipengaruhi k dan jumlah data latih yang digunakan. Semakin besar nilai k maka semakin kecil nilai akurasi dan semakin besar data latih maka semakin besar nilai akurasi. Hal ini dikarenakan data latih yang digunakan adalah data latih kelas seimbang. Berikut adalah hasil pengaruh nilai k dan data latih ditunjukkan pada gambar 6.2.



Gambar 6.2 Grafik Pengaruh Akurasi Nilai k

Sumber: [Pengujian]

Tabel 6.6 menunjukkan nilai k dengan rata-rata tingkat akurasi tertinggi yaitu $k = 19$ dengan tingkat akurasi 99.58% pada 40 data latih dan rata-rata tingkat akurasi terendah yaitu $k = 10$ pada 20 data latih dengan tingkat akurasi sebesar 82.16%. Total rata-rata akurasi pada aplikasi ini yaitu sebesar 94.29% yang menunjukkan bahwa sistem mampu menghasilkan diagnosa yang cukup akurat.

BAB VII

PENUTUP

7.1 Kesimpulan

Berdasarkan hasil penelitian tentang Diagnosa Penyakit Tanaman Cabai Merah dengan Metode *Modified K-Nearest Neighbor* ini, maka dapat diambil kesimpulan bahwa :

1. Instrumen penelitian dengan judul Diagnosa Penyakit Tanaman Cabai Merah dengan Metode *Modified K-Nearest Neighbor* ini dapat memberikan manfaat dalam memperluas pengetahuan tentang penanganan penyakit pada tanaman cabai merah dan dapat memberikan rekomendasi solusi dalam melakukan diagnosa penyakit tanaman cabai merah.
2. Tingkat akurasi yang dihasilkan metode *Modified K-Nearest Neighbor* bergantung pada 3 parameter yaitu:
 - a. Secara umum peningkatan nilai k mempengaruhi penurunan nilai akurasi, dan k optimal yaitu $k=19$ dengan nilai akurasi 99.58%.
 - b. Peningkatan jumlah data latih mempengaruhi peningkatan nilai akurasi hingga 99.58%.
 - c. Pengambilan data latih secara acak mempengaruhi nilai akurasi dengan nilai rata-rata akurasi 94.29% .

7.2 Saran

Diagnosa Penyakit Tanaman Cabai Merah dengan Metode *Modified K-Nearest Neighbor* ini masih memiliki beberapa kekurangan. Saran yang dapat diberikan untuk pengembangan sistem diagnosa ini antara lain:

1. Sistem dapat dikembangkan untuk penelitian lebih lanjut dengan menambah parameter lain yang juga turut mempengaruhi akurasi sistem contohnya lokasi daerah dan penggunaan pupuk.
2. Penentuan nilai k terbaik dapat dibuat secara otomatis oleh sistem.

DAFTAR PUSTAKA

- [1] <http://id.wikipedia.org/wiki/Cabai>, diakses pada tanggal 3 Maret 2015.
- [2] Sasmito, Ginanjar Wiro. 2010. Aplikasi Sistem Pakar untuk Diagnosa Hama dan Penyakit Tanaman Bawang Merah dan Cabai Merah Menggunakan Forward Chaining dan Pendekatan Berbasis. Program Pascasarjana Universitas Diponegoro, Semarang.
- [3] Larose, D.T. 2005. Discovering Knowledge in Data: An Introduction to Data Mining. Willey, Chichester.
- [4] M.Abdul Huda, Candra Dewi, Dian Eka.2013. Implementasi Modified K-Nearest Neighbor untuk Penentuan Kesesuaian Lahan dalam Budidaya Tanaman. Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya, Malang.
- [5] Zainuddin, Sofa. 2014. Penerapan Algoritma Modified K-Nearest Neighbour (MKNN) pada Pengklasifikasian Penyakit Tanaman Kedelai. Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya, Malang.
- [6] Hardiyanti, Siti. 2014. Implementasi metode Modified K-Nearest Neighbour (MKNN) pada Penentuan Keminatan Sekolah Menengah Atas (SMA) (Studi Kasus: SMA Negeri 1 Seririt). Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya, Malang.
- [7] Moertini, Veronika S. 2002. "Data Mining Sebagai Solusi Bisnis".
- [8] Kusnawi (2007). "Pengantar Solusi Data Mining".Seminar Nasional Teknologi 2007. Yogyakarta.
- [9] Hidayat, Syaeful. 2010. Aplikasi Untuk Mendeteksi Jenis Penyakit Pada Tanaman Tebu Dan Cara Penanganannya Berbasis Web. Bandung, Universitas Komputer Indonesia.

- [10] Arif, Son Wicaksana. 2009. Sistem Pakar Identifikasi Hama Dan Penyakit Tanaman Apel Berbasis Web. Malang, Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang.
- [11] <http://ira.lecturer.pens.ac.id/SPK/sistem%20pakar.pdf>, diakses pada tanggal 26 Mei 2015.
- [12] <http://ira.lecturer.pens.ac.id/SPK/sistem%20pakar.pdf>, diakses pada tanggal 22 Mei 2015
- [13] Rismawan, Tedy. Wiedha Irawan, Ardhitya. Prabowo, Wahyu. Kusumadewi, Sri., 2008. "Sistem Pendukung Keputusan Berbasis Pocket PC sebagai Penentu Status Gizi Menggunakan Metode KNN (K-Nearest Neighbor)".
- [14] Santosa, Budi. 2007. Data Mining Teknik Pemanfaatan Data untuk Keperluan Bisnis. Yogyakarta: Graha Ilmu.
- [15] Hamid Parvin, Hoseinali and Behrouz Minati. 2010. Modification on K-Nearest Neighbour Classification. Global Journal of Computer Science and Technology Vol. 10 Issue 14 (Ver.1.0).
- [16] <http://agropedia.iitk.ac.in/content/chilli-cercospora-leaf-spot>, diakses pada tanggal 23 Mei 2015.
- [17] Ir. Pracaya. 1993. Bertanam Lombok. Yogyakarta: Kanisius. Hal 9-
- [18] Redaksi Agromedia. 2008. Panduan lengkap budidaya & bisnis cabai. Jakarta Selatan: Agromedia Pustaka.
- [19] Latiful Ashar, Busyairi. 2009. Sistem Pakar Diagnosa Hama dan Penyakit Tanaman Cabai Besar Merah (*Capcicum annum* L.). Program Studi Hama dan Penyakit Tumbuhan, Fakultas Pertanian, Institut Pertanian Bogor, Bogor.

