

**IMPLEMENTASI ALGORITMA *MODIFIED*
K-NEAREST NEIGHBOR (MKNN) UNTUK
DIAGNOSA PENYAKIT TANAMAN
CENGKEH**

Oleh

DIDANG WIKATAMA

T3120076

SKRIPSI

Untuk Memenuhi Salah Satu Syarat Ujian

Guna Memperoleh Gelar Sarjana



PROGRAM SARJANA

TEKNIK INFORMATIKA

UNIVERSITAS ICHSAN GORONTALO

GORONTALO

2024

PERSETUJUAN SKRIPSI

**IMPLEMENTASI ALGORITMA *MODIFIED*
K-NEAREST NEIGHBOR (MKNN) UNTUK
DIAGNOSA PENYAKIT TANAMAN
CENGKEH**

OLEH
DIDANG WIKATAMA

T3120076

SKRIPSI

Untuk memenuhi salah satu syarat ujian
guna memperoleh gelar Sarjana
Program Studi Teknik Informatika,
ini telah disetujui oleh Tim Pembimbing

Gorontalo, Agustus 2024

Pembimbing I

Irma Surya Kumala Idris, M.kom
NIDN: 0921128801

Pembimbing II

Abd. Rahmat Karim Haba, M.kom
NIDN: 0923118703

PENGESAHAN SKRIPSI

IMPLEMENTASI ALGORITMA *MODIFIED* *K-NEAREST NEIGHBOR (MKNN)* UNTUK DIAGNOSA PENYAKIT TANAMAN CENGKEH

OLEH

DIDANG WIKATAMA

T3120076

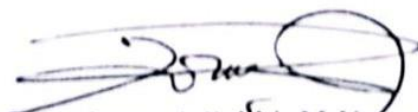
Diperiksa Oleh Panitia Ujian Strata (S1)
Universitas Ichsan Gorontalo

- 1 Ketua Penguji
Sudirman Melangi, M.Kom
- 2 Anggota
Muis Nanja, M.Kom
- 3 Anggota
Yulianty Lasena, M.Kom
- 4 Anggota
Irma Surya Kumala Idris., M.Kom
- 5 Anggota
Abd. Rahmat Karim Haba, M.Kom




Mengetahui,

Dekan Fakultas Ilmu Komputer


Irvan A. Salihi, M.Kom
NIDN. 0928028101

Ketua Program Studi


Sudirman S. Panna, M.Kom
NIDN. 0924038205

PERNYATAAN SKRIPSI

Dengan ini saya menyatakan bahwa :

1. Karya Tulis (Skripsi) saya ini adalah asli dan belum pernah diajukan untuk mendapatkan gelar akademik (Sarjana) baik di Universitas Ichsan Gorontalo maupun diperguruan tinggi lainnya.
2. Karya tulis (Skripsi) saya ini adalah murni gagasan, rumusan, dan penelitian saya sendiri, tanpa bantuan pihak lain, kecuali arahan dari tim pembimbing.
3. Dalam karya tulis (Skripsi) saya ini tidak terdapat karya atau pendapat yang telah dipublikasi orang lain, kecuali secara tertulis dicantumkan sebagai acuan / sitasi dalam naskah dan dicantumkan pada daftar pustaka.
4. Pernyataan ini saya buat dengan sesungguhnya dan apabila kemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini, maka saya bersedia menerima sanksi akademik berupa pencabutan gelar yang telah diperoleh karena karya tulis ini, serta sanksi lainnya sesuai dengan norma-norma yang berlaku di Universitas Ichsan Gorontalo.

Gorontalo, Agustus 2024

myataan,

DIDANG WIKATAMA

ABSTRAK

DIDANG WIKATAMA. T3120076. THE IMPLEMENTATION OF THE MODIFIED K-NEAREST NEIGHBOR (MKNN) ALGORITHM FOR CLOVE PLANT DISEASE DIAGNOSIS

This study discusses the development of a disease diagnosis system in clove plants using the Modified K-Nearest Neighbor (MKNN) method. This study aims to develop a system that accurately diagnoses clove plant diseases based on identified symptoms. The MKNN method is chosen for its ability to handle non-linear and complex data, such as plant disease symptoms. The system is developed to identify three common diseases in clove plants, namely clove wood killer bacteria, clove smallpox on leaves, and white root fungus. The development is by employing the symptom datasets collected. The results show that the system can provide appropriate diagnostics and is consistent with manual calculations performed by plant experts. It also implies the system engineering process for the development of MKNN in the context of clove plant disease diagnosis. The MKNN algorithm implementation in this system produces a Cyclomatic Complexity value of 7, indicating a level of complexity to be well managed in the context of this implementation. This study concludes that the implemented MKNN system can effectively diagnose diseases in clove plants based on the observed symptoms. The implication of this study is the potential use of similar systems for more efficient and accurate management of clove plants in the field.

Keywords: *Modified K-Nearest Neighbor (MKNN), plant disease diagnosis, clove plant, diagnosis system, Cyclomatic Complexity*

ABSTRAK

DIDANG WIKATAMA. T3120076. IMPLEMENTASI ALGORITMA MODIFIED K-NEAREST NEIGHBOR (MKNN) UNTUK DIAGNOSA PENYAKIT TANAMAN CENGKEH

Penelitian ini membahas tentang pengembangan sistem diagnosa penyakit pada tanaman Cengkeh menggunakan metode Modified K-Nearest Neighbor (MKNN). Penelitian ini bertujuan untuk mengembangkan sistem yang dapat secara akurat mendiagnosa penyakit tanaman cengkeh berdasarkan gejala yang teridentifikasi. Metode MKNN dipilih karena kemampuannya dalam menangani data non-linier dan kompleks seperti gejala penyakit tanaman. Sistem yang dikembangkan berhasil mengidentifikasi tiga penyakit umum pada tanaman cengkeh, yaitu penyakit Bakteri pembunuh kayu cengkeh (BPKC), Cacar daun cengkeh (CDC), dan Jamur akar Putih (JAP), dengan memanfaatkan dataset gejala yang telah terkumpul. Hasil penelitian menunjukkan bahwa sistem mampu memberikan diagnosa yang sesuai dan konsisten dengan perhitungan manual yang dilakukan oleh pakar tanaman. Penelitian ini juga mengungkapkan proses rekayasa sistem untuk pengembangan MKNN dalam konteks diagnosa penyakit tanaman cengkeh. Penerapan algoritma MKNN dalam sistem ini menghasilkan nilai Cyclomatic Complexity sebesar 7, menunjukkan tingkat kompleksitas yang dapat dikelola dengan baik dalam konteks aplikasi ini. Kesimpulan dari penelitian ini adalah bahwa sistem MKNN yang diimplementasikan dapat efektif digunakan untuk mendiagnosa penyakit pada tanaman cengkeh berdasarkan gejala yang diamati. Implikasi dari penelitian ini adalah potensi penggunaan sistem serupa dalam mendukung pengelolaan tanaman cengkeh secara lebih efisien dan akurat di lapangan.

Kata kunci: Modified K-Nearest Neighbor (MKNN), diagnosa penyakit tanaman, tanaman cengkeh, sistem diagnosa, Cyclomatic Complexity

KATA PENGANTAR

Alhamdulillah, penulis dapat menyelesaikan usulan penelitian ini dengan judul “**Implementasi Metode *Modified K-Nearest Neighbor* Untuk Diagnosis Penyakit Tanaman Cengkeh**” skripsi ini disusun sebagai salah satu syarat untuk mengerjakan skripsi pada program S1 di jurusan Teknik Informatika Fakultas Ilmu Komputer Universitas Ichsan Gorontalo.

Penulis menyadari sepenuhnya bahwa skripsi ini tidak mungkin terwujud tanpa bantuan dan dorongan dari berbagai pihak, baik bantuan moril maupun materil. Untuk itu, dengan segala keiklasan dan kerendahan hati, penulis mengucapkan banyak terima kasih dan penghargaan yang setinggi-tingginya kepada :

1. Muhammad Ichsan Gaffar, SE M.Ak, selaku ketua Yayasan Pengembang Ilmu Pengetahuan dan Teknologi (YPIPT) Ichsan Gorontalo;
2. Dr. Abdul Gaffar La Tjokke, M.si, selaku Rektor Universitas Ichsan Gorontalo;
3. Irvan Abraham Salihi, M.kom, selaku Dekan Fakultas Ilmu Komputer Universitas Ichsan Gorontalo;
4. Sudirman Melangi, M.kom, selaku Wakil Dekan I Bidang Akademik dan Bidang Kemahasiswaan Fakultas Ilmu Komputer Universitas Ichsan Gorontalo;
5. Irma Surya Kumala Idris, M.Kom, selaku Wakil Dekan II Bidang Administrasi Umum dan Keuangan Fakultas Ilmu Komputer Universitas Ichsan Gorontalo; sekaligus sebagai Pembimbing I yang selalu membantu atau membimbing penulis untuk menyelesaikan usulan penelitian
6. Sudirman S. Panna, S.kom, selaku ketua Jurusan Teknik Informatika Fakultas Ilmu Komputer Universitas Ichsan Gorontalo;

7. Abd. Rahmat Karim Haba, M.kom, selaku Pembimbing II, yang selalu membantu atau membimbing penulis untuk menyelesaikan usulan penelitian.
8. Bapak dan Ibu Dosen Universitas Ichsan Gorontalo yang telah mendidik dan mengajarkan berbagai disiplin ilmu kepada penulis.
9. Kepada Kedua Orang Tua, dan adik tercinta yang selalu memberikan dorongan moral maupun materi yang sangat besar kepada saya;
10. Rekan-rekan seperjuangan yang telah banyak memberikan bantuan dan dukungan sangat besar kepada saya;
11. Kepada semua pihak yang ikut membantu dalam menyelesaikan usulan penelitian ini yang tak sempat penulis sebutkan satu-persatu;

Penulis mengucapkan terima kasih kepada semua pihak yang telah membantu dalam menyusun usulan penelitian ini sehingga usulan penelitian ini dapat terselesaikan. Penulis mengharapkan saran dan kritik sehingga usulan penelitian ini dapat memberikan manfaat bagi bidang pendidikan serta bisa dikembangkan lagi lebih lanjut.

Gorontalo, Agustus 2024

Penulis

DAFTAR ISI

PERSETUJUAN	
SKRIPSI.....	Error! Bookmark not defined.
PENGESAHAN SKRIPSI.....	ii
PERNYATAAN	
SKRIPSI.....	Error! Bookmark not defined.
ABSTRAK.....	v
KATA PENGANTAR.....	vii
DAFTAR ISI.....	ix
DAFTARGAMBAR.....	xii
i	
DAFTARTABEL.....	xiv
DAFTARLAMPIRAN.....	xv
BAB 1 PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Identifikasi masalah.....	3
1.3 Rumusan masalah.....	4
1.4 Tujuan penelitian	4
1.5 Manfaat penelitian	4
BAB II LANDASAN TEORI.....	5
2.1 Tinjauan Studi	5
2.2 Tinjauan Pustaka	6
2.2.1 Cengkeh	6

2.2.2	Penyakit Cengkeh	6
2.2.3	Data Mining	8
2.2.4	Klasifikasi	10
2.2.5	K-Nearest Neighbor (KNN).....	11
2.2.6	Modified K-Nearest Neighbor (MKKN)	13
2.2.7	Pengujian Akurasi	16
2.2.8	Penerapan Algoritma MKNN	16
2.3	Analisis sistem.....	16
2.4	Desain sistem.....	17
2.5	Implementasi Sistem	18
2.6	Pengujian sistem.....	20
2.6.1	White box	20
2.6.2	Black Box.....	23
2.7	Kerangka pemikira	25
BAB III METOLOGI PENELITIAN.....		26
3.1	Jenis, Metode, Subjek, Objek, Waktu, dan Lokasi Penelitian.....	26
3.2	Pengumpulan Data.....	26
3.3	Pemodelan	28
3.3.1	Studi Literatur	29
3.3.2	Analisis Kebutuhan	29
3.3.3	Analisis dan Perancangan Sistem.....	30
3.3.4	Implementasi	30
3.3.6	Pengujian Sistem.....	31
3.3.7	Evaluasi dan Analisis Hasil.....	32
BAB IV HASIL PENELITIAN.....		33

4.1	Hasil Pengumpulan Data	33
4.2	Hasil Pemodelan Data	36
4.3	Hasil Pengembangan Sistem	37
4.3.1	Analisis Sistem.....	37
4.3.2	Diagram Konteks	39
4.3.3	Diagram Berjenjang	40
4.3.4	Diagram Arus Data (DAD)	40
4.3.5	Kamus Data.....	43
4.3.6	Arsitektur Sistem/Kebutuhan <i>Hardware dan Software</i>	45
4.3.7	Desain Secara Umum.....	45
4.3.8	Desain Secara Terinci	46
4.3.8.3	desain Database Secara Terinci.....	48
4.3.9	Pengujian Sistem.....	50
BAB V PEMBAHASAN PENELITIAN.....		55
5.1	Pembahasan Sistem	55
5.1.1	Halaman Menu Utama	55
5.1.2	Halaman Login.....	55
5.1.3	Halaman Menu Gejala	56
5.1.4	Halaman Menu Penyakit.....	57
5.1.5	Halaman Menu Dataset.....	57
5.1.6	Halaman Menu Input User	58
5.1.8	Halaman Menu Diagnosa	59
5.1.9	Halaman Menu Hasil Diagnosa	60
BAB VI KESIMPULAN DAN SARAN.....		61
6.1	KESIMPULAN	61

6.2	SARAN	61
	DAFTAR PUSTAKA.....	62

DAFTAR GAMBAR

Gambar 2. 1 Proses Penemuan Pengetahuan	8
Gambar 2. 2 : Proses klasifikasi	10
Gambar 2. 3 Bagan Air	21
Gambar 2. 4 Kerangka Pemikiran	25
Gambar 3. 1 Pemodelan	28
Gambar 4. 1 Sistem Yang diusulkan.....	38
Gambar 4. 2 Diagram Konteks.....	39
Gambar 4. 3 Diagram Berjenjang	40
Gambar 4. 4 Diagram Arus Data Level 0	41
Gambar 4. 5 Diagram Arus Data Level 1	42
Gambar 4. 6 Desain Input Gejala.....	47
Gambar 4. 7 Desain Input Penyakit	47
Gambar 4. 8 Desain Menu Utama.....	50
Gambar 4. 9 Flowchart Penyakit.....	51
Gambar 5. 1 Halaman Utama.....	55
Gambar 5. 2 Halaman Login.....	56
Gambar 5. 3 Halaman Menu Gejala.....	56
Gambar 5. 4 Halaman Menu Penyakit	57
Gambar 5. 5 Halaman Menu Dataset	57
Gambar 5. 6 Halaman Input User	58
Gambar 5. 7 Halaman Menu Data User	59
Gambar 5. 8 Halaman Menu Diagnosa	59
Gambar 5. 9 Halaman Menu Hasil Diagnosa	60

DAFTAR TABEL

Tabel 1.1 Penyakit Tanaman Cengkeh.....	2
Tabel 2.1 Tinjauan studi.....	5
Tabel 2.2 Tabel Simbol DFD	18
Tabel 3.1 Jenis Gejala	26
Tabel 3.2 Nama Penyakit	27
Tabel 4. 1 Daftar penyakit Tanaman Cengkeh.....	33
Tabel 4. 2 Daftar Gejala Penyakit tanaman Cengkeh	33
Tabel 4. 3 Penyakit dan Gejala Tanaman Cengkeh	34
Tabel 4. 4 Daftar Bobot Dari Setiap Gejala Penyakit	35
Tabel 4. 5 Matarix Tanaman Penyakit Jagung.....	36
Tabel 4. 6 Kamus Data User	43
Tabel 4. 7 Kamus Data Gejala	43
Tabel 4. 8 Kamus Data Penyakit.....	43
Tabel 4. 9 Kamus Data Dataset.....	44
Tabel 4. 10 Kamus Data k.....	44
Tabel 4. 11 Kamus Data Hasil	44
Tabel 4. 12 Kamus Data Testing.....	44
Tabel 4. 13 Kamus Data Laporan	45
Tabel 4. 14 Desain Output Secara Umum.....	46
Tabel 4. 15 Desain Input Secara Umum	46
Tabel 4. 16 Database	48
Tabel 4. 17 Penyakit.....	48
Tabel 4. 18 GejalaTabel 4. 19 Testing	48
Tabel 4. 20 Hasil	49
Tabel 4. 21 Laporan	49
Tabel 4. 22 Pengujian Blackbox	53

DAFTAR LAMPIRAN

Lampiran 1 : Kode Program.....	62
Lampiran 2 : Riwayat Hidup.....	81
Lampiran 3 : Surat Penelitian.....	82
Lampiran 4 : Surat Rekomendasi Penelitian.....	83
Lampiran 5 : Surat Keterangan Bebas Pustaka.....	84
Lampiran 6 : Hasil Turnitin.....	85

BAB I

PENDAHULUAN

1.1 Latar Belakang

Cengkeh (*Syzygium aromaticum* (L.) Merr & Perry) merupakan tanaman tropis asli Indonesia dan dapat tumbuh di berbagai daerah di Indonesia, baik di dataran rendah, dekat pantai maupun daerah pegunungan di ketinggian 900 mdpl. Tanaman cengkeh dapat tumbuh dengan baik jika mendapat cukup air dan sinar matahari langsung, oleh karena itu tanaman cengkeh tumbuh baik psada daerah yang memiliki curah hujan sekitar 2210-3607 mm/tahun serta suhu udara berkisar 24-39 °C[1]. Indonesia adalah negara yang menghasilkan jumlah cengkeh terbesar di dunia. Berdasarkan data FAQ (2021), menunjukkan bahwa sebanyak dua pertiga cengkeh di seluruh dunia diproduksi oleh Indonesia. Pada tahun 2017, produksi cengkeh mencapai 113.178 ton, meningkat menjadi 131.013 ton pada tahun 2018, 140.797 ton pada tahun 2019, 140.806 ton pada tahun 2020, dan 140.997 ton pada tahun 2021. Menurut informasi yang ada, terdapat 34 provinsi di Indonesia yang secara keseluruhan memberikan kontribusi sebesar 7,47 persen terhadap produksi cengkeh dari perkebunan rakyat pada periode tahun 2017 hingga 2021.[2]

Tanaman cengkeh memiliki nilai ekonomi tinggi dan merupakan salah satu komoditas ekspor hasil perkebunan Indonesia, Penanaman cengkeh juga menghadapi tantangan yang kompleks, dengan resiko terkena berbagai penyakit seperti bakteri pembunuh kayu cengkeh (BPKC), cacar daun cengkeh (CDC), dan jamur akar putih (JAP) yang memerlukan metode penanganan yang berbeda-beda. Tentu saja, sebelum tanaman cengkeh terinfeksi penyakit pasti memiliki tanda gejala-gejala yang dapat dilihat oleh mata dan segera memerlukan penanganan, namun petani memiliki keterbatasan dalam pengetahuan mengenai penyakit yang menyerang tanaman cengkeh sehingga mereka menghadapi kesulitan dalam mengambil keputusan dan berpotensi merusak hasil produksi hingga menyebabkan gagal panen. Berikut ini adalah daftar penyakit tanaman cengkeh.

Tabel 1.1 Penyakit Tanaman Cengkeh

Kode gejala	Nama gejala	Kode penyakit	Nama penyakit
G-1	Daun tampak kekuning-kuningan	P-1	Bakteri pembunuh kayu cengkeh (BPKCK)
G-2	Daun mengering dan akhirnya mati	P-1	Bakteri pembunuh kayu cengkeh (BPKCK)
G-3	Kulit batang terlepas	P-1	Bakteri pembunuh kayu cengkeh (BPKCK)
G-4	Batang tanaman kering	P-1	Bakteri pembunuh kayu cengkeh (BPKCK)
G-5	Bercak-bercak cokelat kekuningan dari tepi helai daun	P-2	Cacar daun cengkeh (CDC)
G-6	Tulang-tulang daun tampak terang	P-2	Cacar daun cengkeh (CDC)
G-7	Daun menjadi rontok	P-2	Cacar daun cengkeh (CDC)
G-8	Bercak-bercak berwarna kelabu sampai cokelat yang memenuhi bagian permukaan daun	P-2	Cacar daun cengkeh (CDC)
G-9	Bercak-bercak kecil pada daun berwarna merah	P-2	Cacar daun cengkeh (CDC)
G-10	Bintik-bintik putih pada daun	P-2	Cacar daun cengkeh (CDC)
Sampai
G-26	Akar busuk dan bau tak sedap	P-3	Jamur Akar Putih (JAP)

Sehingga dibutuhkan sebuah cara untuk membantu petani untuk mengetahui jenis penyakit yang menyerang tanaman cengkeh tersebut, maka dibuatlah sistem diagnosa 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. Proses klasifikasi menggunakan komputer yang diterapkan dengan menggunakan metode *Modified K-Nearest Neighbor* (MKNN).

Metode *Modified K-Nearest Neighbor* (MKNN) merupakan pengembangan dari algoritma KNN dengan penambahan beberapa proses yakni perhitungan nilai validasi data latih dan *weight voting* atau pembobotan.[3] Perhitungan nilai validasi bertujuan untuk mengatasi permasalahan data yang

menyimpang pada algoritma KNN sehingga kurangnya bias dari data latih dan weight voting untuk menghitung bobot dari data. Algoritma Modified K-Nearest Neighbor menggunakan nilai perhitungan weight voting terbesar untuk menentukan kelas dari suatu data yang belum diketahui.

penelitian yang menggunakan algoritma MKNN salah satunya untuk melakukan klasifikasi penyakit demam. Penelitian ini dilakukan karena adanya kemiripan gejala-gejala penyakit yang menimbulkan kesulitan dalam mendapatkan diagnosis sementara, yang membuat kurang tepatnya penanganan awal yang didapatkan pasien. Ada 15 gejala penyakit dengan total 133 jumlah data pasien penderita penyakit demam berdarah, tifoid dan malaria. Pengujian terhadap perubahan nilai k menghasilkan rata-rata akurasi sebesar 88,55%. Pengujian terhadap pengaruh variasi jumlah data latih menghasilkan rata-rata akurasi sebesar 92,42%. Pengujian pengaruh komposisi data latih menghasilkan rata-rata akurasi sebesar 87,89%, Sementara pengujian pengaruh komposisi data latih dan data uji terhadap akurasi menghasilkan rata-rata akurasi sebesar 96,35%.[4]

Penelitian lainnya membahas tentang penerapan metode M-KNN untuk mengklasifikasi penyakit tanaman kedelai. Pada penelitian ini menghasilkan tingkat akurasi Rata-rata akurasi maksimum yang dihasilkan pada penelitian ini sebesar 98,83%.[5]

Berdasarkan uraian diatas, maka dianggap perlu untuk melakukan penelitian dengan judul “**Implementasi Metode Modified K-Nearest Neighbor Untuk Diagnosis Penyakit Tanaman Cengkeh**” Diharapkan penelitian ini dapat membantu petani cengkeh lebih cepat dalam mengidentifikasi jenis penyakit yang menyerang tanaman cengkehnya.

1.2 Identifikasi masalah

Berdasarkan latar belakang dapat diidentifikasi masalah sebagai berikut :

1. Petani mengalami kesulitan dalam menentukan jenis penyakit yang menyerang tanaman cengkeh.

2. Penanggulangan keputusan yang tidak tepat dalam penanggulangan penyakit tanaman cengkeh berdampak pada kesehatan dan kegagalan panen.

1.3 Rumusan masalah

Berdasarkan uraian latar belakang tersebut, maka rumusan masalah adalah sebagai berikut :

1. Bagaimana mengimplementasikan metode *Modified K-Nearest Neighbor* untuk diagnosis penyakit pada tanaman cengkeh ?
2. Bagaimana tingkat akurasi metode *Modified K-Nearest Neighbor* untuk diagnosis penyakit tanaman cengkeh ?

1.4 Tujuan penelitian

Penelitian yang dilakukan ini memiliki tujuan:

1. Untuk mengimplementasikan metode *Modified K-nearest Neighbor* Diagnosis penyakit tanaman cengkeh.
2. Membantu petani untuk mendiagnosis penyakit tanaman cengkeh yang diperoleh dari hasil implementasi metode *Modified K-nearest Neighbor* untuk pengklafikasian penyakit tanaman cengkeh.

1.5 Manfaat penelitian

Manfaat yang bisa diperoleh dari penelitian ini yaitu diharapkan bisa memberikan informasi untuk memprediksi penyakit yang menyerang tanaman cengkeh, sehingga memudahkan petani cengkeh dalam mengambil tindakan dalam penanganan penyakit yang menyerang tanaman cengkeh selanjutnya.

BAB II

LANDASAN TEORI

2.1. Tinjauan Studi

Berikut merupakan tabel dari beberapa studi penelitian yang pernah dilakukan sebelumnya yang berkaitan dengan penelitian ini.

Tabel 2.1 Tinjauan studi

N O	Judul	Objek dan Input	Metode(proses)	Hasil (Output)
1	Implementasi Modified K-Nearest Neighbor Dengan Otomatisasi Nilai K Pada Pengklasifikasian Penyakit Tanaman Kedelai[5]	Inputan menggunakan Soybean Disease Data Set yang terdiri dari 266 data latih	<ol style="list-style-type: none"> 1. Menghitung validasi data pada semua data di data latih 2. Menghitung <i>Euclidean</i> 3. Mencari <i>Weight voting</i> pada semua data uji menggunakan validitas data. 	<ul style="list-style-type: none"> - Kesimpulan Penyakit beserta prosentase tingkat keyakinannya dan solusi untuk penanganan/pengendaliannya. - Hasil akurasi tertinggi pada pengujian ini sebesar 100% dengan nilai $k=1$ dan rata-rata akurasi dari 5 percobaan sebesar 98,83%.
2	Implementasi Algoritma Modified K-Nearest Neighbor (MKNN) untuk Klasifikasi Penyakit Demam[4]	Inputan sistem berupa 15 gejala penyakit dengan total 133 jumlah data pasien penderita penyakit demam berdarah, tifoid dan malaria	<ol style="list-style-type: none"> 1. Menghitung validasi data pada semua data di data latih 2. Menghitung <i>Euclidean</i> 3. Mencari <i>Weight voting</i> pada semua data uji menggunakan validitas data. 	<ul style="list-style-type: none"> - pengujian pengaruh komposisi data latih dan data uji terhadap akurasi menghasilkan rata-rata akurasi sebesar 96,35%

N O	Judul	Objek dan Input	Metode(proses)	Hasil (Output)
3	Diagnosis Penyakit Tanaman Cabai Menggunakan Metode Modified KNearest Neighbor (MKNN) (Studi Kasus: BPTP Karang Ploso Malang) .[6]	Inputan sistem berupa pertanyaan tentang gejala penyakit yang menyerang tanaman cabai	<ol style="list-style-type: none"> 1. Menjawab pertanyaan dari sistem yang berupa gejala penyakit tanaman tomat. 2. Normalisasi data. 3. Menghitung nilai validitas. 4. Menghitung nilai <i>Euclidean</i>. 5. Menghitung weighted voting (pembobotan) dari data. 6. Menentukan kelas dari data uji tersebut 	- Hasil pengujian menggunakan K=5 mendapatkan akurasi sebesar 94%, kemudian K=8 akurasi sebesar 92%, K=11 akurasi sebesar 88% dan pengujian K=14 menghasilkan akurasi sebesar 88%

2.2 Tinjauan Pustaka

2.2.1 Cengkeh

Cengkeh (*Syzygium aromaticum* (L.) Merr & Perry) merupakan tanaman tropis asli Indonesia dan dapat tumbuh di berbagai daerah di Indonesia, baik di dataran rendah, dekat pantai maupun daerah pegunungan di ketinggian 900 mdpl. Tanaman cengkeh dapat tumbuh dengan baik jika mendapat cukup air dan sinar matahari langsung, oleh karena itu tanaman cengkeh tumbuh baik pada daerah yang memiliki curah hujan sekitar 2210-3607 mm/tahun serta suhu udara berkisar 24-39 °C[1].

2.2.2 Penyakit Cengkeh

Penyakit yang menyerang tanaman cengkeh menyebabkan tanaman cengkeh kurang produktif bahkan dapat menyebabkan gagal panen bagi petani.

Serta cengkeh juga rentan terkena penyakit dan penanganan untuk masing – masing penyakit berbeda, adapum beberapa penyakit yang sering ditemui berdasarkan data yang didapatkan dari pakar:

1. Bakteri pembunuh Kayu Cengkeh (BPKC)

Penyakit yang menyerang tanaman cengkeh selama fase awal berbuah atau saat tanaman masih muda dan siap untuk menghasilkan buah, bisa dikategorikan sebagai penyakit yang sangat merusak. Penyakit yang ditimbulkan oleh *pseudomonas syzgia* ini menular melalui serangga yang membantu penyebarannya dari tanaman yang terinfeksi ke tanaman yang belum terinfeksi. Ada beberapa tanda-tanda yang dapat menunjukkan bahwa tanaman telah terinfeksi penyakit ini, seperti daun yang tiba-tiba rontok dan banyak cabang yang mati di bagian pucuknya. Penyakit ini dapat berlangsung mulai dari satu minggu hingga beberapa bulan. Untuk mengatasi penyakit ini, langkah pertama adalah melakukan perawatan kebun yang meliputi sanitasi dan penyemprotan Hindola fulfa dengan insektisida kontak setiap 6 minggu sekali. Hal ini diperlukan terutama untuk tanaman yang masih dapat diselamatkan. Penyuntikan antibiotik oksitetrasiklin (OTC) sejumlah 6 gram per 100 ml air dapat dilakukan secara rutin setiap 3 hingga 4 bulan sekali. Pemupukan kombinasi N dan K dapat dilakukan dengan melakukan ketiga langkah tersebut.

2. Cacar Daun Cengkeh(CDC)

Penyakit cacar daun dapat dianggap sebagai salah satu penyakit yang berbahaya, selain penyakit pembusukan batang dan kuncup cengkeh(BPBK). Cacar daun pada tanaman cengkeh disebabkan oleh *phyllosticta syzgia*, oleh karena itu jika tanaman cengkeh terinfeksi cacar daun akan terlihat tidak berkembang dengan baik dan terdapat bercak mencolok pada daunnya. Hal ini mengakibatkan tanaman tidak mendapatkan nutrisi yang cukup karena kemampuan tanaman dalam melakukan fotosintesis berkurang. Untuk mengatasi penyakit Cacar daun cengkeh, dapat dilakukan pengendalian dengan cara menyemprotkan fungisida setiap 7-10 hari sekali. Selain itu, penting untuk

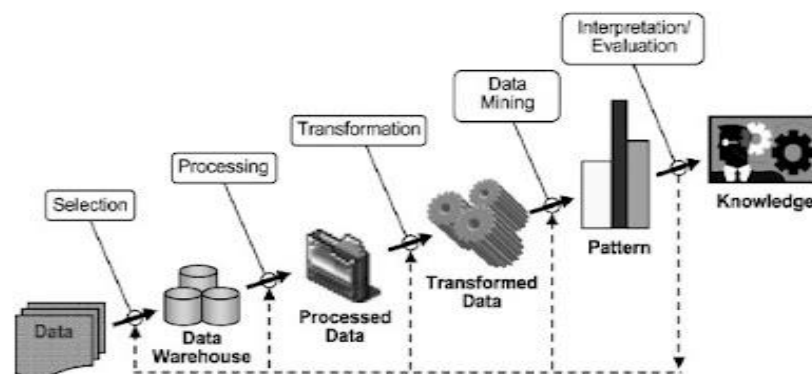
menjaga kebersihan kebun dengan melakukan sanitasi. Daun, ranting, dan biji dari tanaman yang terinfeksi sebaiknya dikumpulkan dan dibakar.

3. Jamur Akar Putih (JAP)

Menurut pakar, penyakit jamur akar putih disebabkan oleh jenis *Rigidoporus* sp. Salah satu gejalanya adalah adanya pertumbuhan rhizomorph yang menyerupai kipas berwarna putih pada akar dan batang dekat akar. Karena itulah penyakit ini dinamakan jamur akar putih. Jika akar dibiarkan, akibatnya akar akan menjadi lembek dan berair, dan ini dapat menyebabkan pembusukan akar tanaman. Dan bagian atas daun memiliki tekstur kasar dan melengkung ke bawah yang tidak biasa bagi daun cengkeh, kemudian berubah menjadi warna coklat. Beberapa metode perawatan yang dapat dilakukan ketika tanaman terkena serangan jamur akar putih adalah dengan melakukan perawatan budidaya di perkebunan atau memberikan pemupukan yang tepat.

2.2.3 Data Mining

Data Mining, atau biasa disebut sebagai *Knowledge Discovery in Databases* (KDD), mengacu pada ekstraksi, penggalian, atau penemuan suatu pengetahuan (*Knowledge*) yang berasal dari data yang berjumlah besar. Adapun beberapa yang menganggap bahwa data mining itu sendiri hanya merupakan suatu tahap penting dalam proses penemuan pengetahuan.[7] **GAMBAR 2.1** menunjukkan proses penemuan pengetahuan.



Gambar 2. 1 Proses Penemuan Pengetahuan

Proses penemuan pengetahuan terdiri dari urutan iterasi langkah-langkah berikut.[7]

1. **Data Cleansing**, Proses di mana data-data yang tidak lengkap, mengandung error dan tidak konsisten di buang dari koleksi data
2. **Data Integration**, Proses integrasi data di mana yang berulang akan di kombinasikan.
3. **Selection**, Proses seleksi atau pemilihan data yang relevan terhadap analisis untuk di terima dari koleksi data yang ada.
4. **Data Transformation**, Proses transformasi data yang sudah di pilih ke dalam bentuk mining procedure melalui cara dan agresi data.
5. **Data Mining**, Proses yang paling penting dimana akan dilakukan berbagai teknik yang di aplikasikan untuk mengekstrak berbagai pola-pola potensial untuk mendapatkan data yang berguna.
6. **Pattern Evolution**, Sebuah proses di mana pola-pola menarik yang sebelumnya sudah di temukan dengan identifikasi berdasarkan measure yang telah di berikan
7. **Knowledge Presentation**, Merupakan proses tahap terakhir, Dalam hal ini di gunakan teknik visualisasi yang bertujuan membantu user dalam mengerti dan menginterpretasikan hasil dari penambangan data.

Secara garis besar metode pada Data mining dapat dikelompokkan menjadi 2 kategori utama, yaitu:

1. *Descriptive mining*

Deskriptive mining adalah sebuah proses untuk mengidentifikasi fitur-fitur yang signifikan dari data di dalam sebuah basis data.

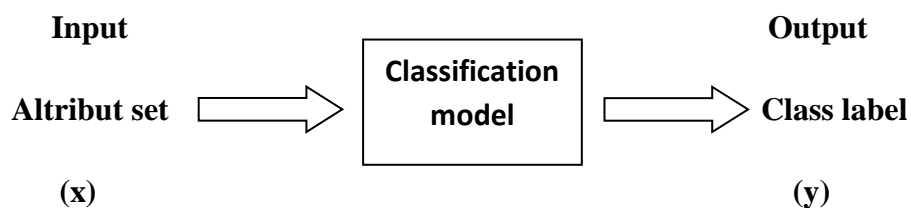
Pada descriptive mining, teknik-teknik data mining yang termasuk di dalamnya adalah clustering, sequential mining, dan association.

2. *Predictive mining*

Metode *predictive mining* digunakan untuk mencari pola pada data dengan memanfaatkan variabel lain di masa yang akan datang. Satu teknik dalam *Predictive mining* adalah menggunakan metode klasifikasi.

2.2.4 Klasifikasi

Klasifikasi merupakan salah satu teknik dalam Data mining yang digunakan untuk menemukan pola atau fungsi yang dapat menjelaskan atau membedakan kelas atau konsep data. Tujuannya adalah untuk dapat memprediksi atau memperkirakan kelas dari suatu objek yang tidak memiliki label[3]. Tujuan klafikasi adalah untuk memperkirakan kategori suatu objek yang tidak memiliki label. Oleh proses klafikasi, dijelaskan pada **Gambar 2.2**



Gambar 2. 2 : Proses klasifikasi

Pada prosses pelatihan, data latih yang memiliki kelas yang sudah diketahui akan dianalisis dan menggunakan algoritme klasifikasi, sebuah model akan dibuat untuk setiap kelas. Proses latihan yang terjadi pada klasifikasi juga dikenal sebagai supervised learning karena setiap kelas dari data latih sudah diketahui. Model hasil penelitian adalah serangkaian aturan-aturan untuk melakukan klasifikasi. Setelah itu, aturan-aturan tersebut diuji menggunakan data uji untuk mengestimasi tingkat keakuratan. Data pengujian adalah data yang tidak tergantung dan tidak digunakan dalam proses pelatihan. Jika akurasi tinggi, maka aturan-aturan tersebut bisa digunakan untuk mengklasifikasikan data baru yang kelasnya tidak diketahui. secara umum, proses klasifikasi sering kali terbagi menjadi dua tahap, yaitu :

1. Learning

Pada fase ini, Sejumlah data yang sudah diketahui kelasnya digunakan untuk membuat model perkiraan.

2. Test

Pada fase test, Model yang telah terbentuk akan diuji menggunakan sebagian data lainnya untuk mengukur sejauh mana keakuratannya. Jika akurasi mencapai tingkat yang memadai, model ini dapat digunakan untuk meramalkan kategori data yang belum diketahui.

2.2.5 K-Nearest Neighbor (KNN)

K-Nearest Neighbor (KNN) suatu metode yang biasa digunakan untuk klafikasi data. Algoritma KNN merupakan suatu teknik untuk mengklasifikasi objek berdasarkan data pembelajaran yang memiliki jarak terdekat dengan objek yang sedang dianalisis. KNN adalah suatu teknik yang menggunakan pendekatan terawasi yang menggunakan algoritma, dimana hasil dari *instance query* yang belum diketahui diklasifikasikan berdasarkan mayoritas kategori yang ada pada KNN. Tujuan dari algoritma ini adalah untuk melakukan klasifikasi pada objek baru dengan mempertimbangkan atribut dan sampel pelatihan [8].

Metode KNN adalah metode yang simpel, mudah diterapkan, dan efektif ketika data pelatihan memiliki volume yang besar. Walaupun ada beberapa keuntungan dalam menggunakan metode KNN, namun terdapat juga beberapa kekurangan sebagai berikut.[8] :

1. Menggunakan fungsi jarak yang belum diketahui pasti dengan menghitung jarak.
2. Biaya komputasi yang tinggi disebabkan oleh perlunya menghitung jarak setiap data training.
3. Diperlukan untuk memutuskan nilai parameter k, yaitu jumlah tetangga terdekat.
4. Diperlukan kapasitas memori yang besar.
5. Saat ini belum diketahui atribut yang dapat menghasilkan hasil terbaik.
6. Ketidak akuratan pada dataset multidimensi sangat rendah.

Prinsip kerja dari algoritma K-Nearest Neighbor adalah dengan membandingkan data uji yang diberikan dengan data pelatihan yang mirip, dan mencari jarak terdekat antara data yang sedang dievaluasi dengan k tetangga terdekat dalam data pelatihan. Jarak atau seringnya interaksi dengan tetangga umumnya diukur dengan menggunakan variabel.

Interval scaled variable adalah metode pengukuran yang mencakup ukuran-ukuran kontinu yang bersifat linier. Salah satu contoh ukuran yang digunakan adalah ukuran jarak, dimana ukuran jarak yang paling umum digunakan adalah jarak Euclidean. Perhitungan Euclidean distance dilakukan dengan menggunakan rumus berikut. Proses KNN ini melibatkan pencarian jarak terdekat antara tetangga menggunakan *Euclidean Distance*. Dalam KNN (K-nearest neighbors) yang berarti k tetangga terdekat, diperlukan nilai k untuk menentukan jumlah tetangga terdekat yang akan digunakan dalam perhitungan. Kemudian, proses berikutnya adalah menghitung nilai Euclidean tersebut hingga menentukan kelas dari data masukan atau data uji tersebut. Dapat dilihat pada persamaan (2-1)[8].

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

Keterangan:

$d(x, y)$: Euclidean Distance dari data Training x dan data uji y .

x_i : Data Training

y_i : Data uji

Dalam konteks ini, x_i adalah sebuah nilai yang mewakili data testing, sedangkan y_i adalah sebuah nilai yang mewakili data training. Data kontinu dapat diolah dengan menggunakan formula normalisasi atau standardisasi sebelum menjalankan proses klasifikasi. Normalisasi dilakukan dengan tujuan menghindari atribut-atribut yang memiliki nilai yang sangat berbeda, yaitu atribut dengan rentang nilai yang besar dan atribut dengan rentang nilai yang kecil. Perhitungan

normalisasi min-max dapat dipakai untuk mengubah atribut A yang memiliki nilai v menjadi v' yang berada dalam rentang (0,1). Normalisasi min-max melibatkan mengubah nilai-nilai dalam kisaran tertentu. Dapat dilihat pada persamaan (2-2)[8]:

$$v' = \frac{v - \min A}{\max A - \min A} \times (\text{new}_{\max A} - \text{new}_{\min A}) + \text{new}_{\min A} \quad (2-2)$$

Dimana :

V' = nilai normalisasi

V = nilai data yang akan dinormalisasi

\min_A = nilai terendah (minimal) data pada atribut A

\max_A = nilai tertinggi (maximal) data pada atribut A

$\text{new}_{\min A}$ = nilai minimum terbaru yang diinginkan

$\text{new}_{\max A}$ = nilai maksimum terbaru yang diinginkan

2.2.6 Modified K-Nearest Neighbor (MKKN)

Algoritma Modified K-nearest Neighbor (MKNN) adalah sebuah perbaikan dari algoritma KNN dengan tambahan beberapa langkah yaitu menghitung nilai validasi data latih dan weight voting atau pembobotan. Perhitungan validasi nilai bertujuan untuk mengatasi masalah data yang tidak konsisten pada algoritma KNN agar tidak ada kecenderungan terhadap data latih dan menggunakan metode Voting untuk menghitung bobot dari data. Algoritma Modified K-NN menggunakan nilai perhitungan *Weight Voting* terbesar untuk memprediksi kelas suatu data yang belum diketahui. Bobot juga terpengaruh oleh nilai validasi. Sehingga, dengan memasukkan perhitungan nilai validasi dan *Weight Voting*, secara efektif dapat meningkatkan keakuratan dalam penggunaan algoritma K-NN [8].

1. Validasi Data *Training*

Dalam Modified K-NN, semua data latih harus diperiksa, dan dalam tahap ini nilai validasi diperoleh dengan membandingkan kelas data training dengan kelas tetangga yang dekat. Kemudian, perbandingan nilai dari kelas sebelumnya akan dijumlahkan dan kemudian dibagi dengan angka k. persamaan untuk validasi data pelatihan diberikan seperti pada persamaan (2-3)[8].

$$Validitas(x) = \frac{1}{k} \sum_{i=1}^k S(label(x), (label(Ni(x)))) \quad (2-3)$$

Keterangan:

k : Banyaknya tetangga

label(x): Kelas dari data x

Ni(x) : Kelas tetangga terdekat ke-i data x

Lalu, dalam konteks ini fungsi S digunakan sebagai perbandingan kelas antara data a dan data b. Jika kedua kelasnya sama, maka nilai fungsi S adalah 1. Namun, jika kelasnya berbeda, maka nilainya adalah 0. Fungsi S didefinisikan dengan persamaan sebagai berikut (2-4)[9].

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

Keterangan:

a = nilai kelas a data training

b = nilai kelas b data training

2. *Weight Voting*

Weight voting KNN merupakan salah satu bentuk KNN yang menggunakan k tetangga terdekat, tanpa mempertimbangkan kelas data, melainkan menggunakan bobot voting dari setiap data dalam data pelatihan. Setiap data diberikan bobot suara yang umumnya sama dengan pengurangan

fungsi jarak dari data yang tidak diketahui. Persamaan weight voting dijelaskan menggunakan persamaan berikut: (2-5) [9] :

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

Dimana d mewakili jarak dan α mewakili nilai pengatur smoothing. Pada penelitian ini, nilai α yang dipergunakan ialah 0,5. Selanjutnya, total suara pemilihan ini dihitung untuk setiap kelasnya, dan kelas dengan jumlah suara terbanyak akan dipilih sebagai keputusan. Kemudian, kita mengalikan bobot dengan setiap data pada data training untuk menentukan keabsahan data tersebut. berdasarkan pada metode pengukuran jarak *Euclidean*. Dalam metode MKNN, perhitungan bobot voting untuk setiap tetangga dilakukan dengan menggunakan persamaan. (2-6)[8].

$$W(i) = Validitas(i) \times \frac{1}{d_e+a} \quad (2-6)$$

Keterangan:

$W_{(i)}$: Perhitungn *weight voting*

Validitas (i) : Hasil Validitas data ke-i

d_e : *Euclidean Distance* dari data uji

a : Nilai *regulator smoothing* (pemulusan)

Perhitungan Weight Voting dalam algoritme Modified K-NN adalah sebuah langkah di mana setiap data akan diberi bobot menggunakan rumus $1/(d_e + a)$, dimana d_e adalah jarak *Euclidean* dan a ditambahkan agar tidak terjadi pembagian dengan 0. Hasil klafikasi Modified K-NN ditentukan dengan mengambil nilai perhitungan W yang terbesar.

2.2.7 Pengujian Akurasi

Pengujian akurasi digunakan untuk mengevaluasi kemampuan sistem dalam mengambil keputusan. Ini dilakukan untuk mengevaluasi seberapa dekat hasil pengukuran dengan angka sebenarnya. Jumlah diagnosa yang benar dibagi dengan jumlah data saat ini untuk menentukan tingkat akurasi. Persamaan ini dapat digunakan untuk menentukan tingkat keakuratan dalam perhitungan persamaan ini. (2-7).

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

2.2.8 Penerapan Algoritma MKNN

Contoh penerapan Algoritma MKNN berikut langkah-langkahnya ;

1. Menentukan nilai k atau tetangganya
2. Melakukan normalisasi data
3. Menghitung validasi data latih
4. Menghitung jarak enclidean
5. Menghitung weight voting (pembobotan)
6. Menentukan kelas dari data uji

2.3 Analisis sistem

Analisis sistem adalah proses mengurai sistem informasi secara keseluruhan menjadi komponen-komponennya. Hal ini bertujuan untuk mengidentifikasi dan mengevaluasi masalah, peluang, hambatan, serta kebutuhan yang muncul atau diharapkan. Dengan melakukan analisis sistem tersebut, diharapkan dapat melakukan peningkatan dan perbaikan yang sesuai.

Dalam fase analisis sistem, ada langkah-langkah utama berikut yang harus dilakukan oleh seorang analisis sistem.

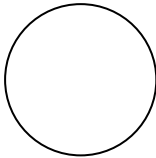
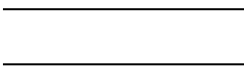
1. Identify, adalah pengidentifikasian permasalahan
2. Understand, adalah pengetahuan kerja sistem yang ada.
3. Analyze, adalah pembuatan laporan analisis

Tahap analisis sistem dilaksanakan selesai perencanaan sistem dan sebelum tahap perencanaan sistem. Fase analisis merupakan fase yang kritis dan sangat penting karena kesalahan pada fase ini juga dapat berakibat kesalahan pada fase selanjutnya.

2.4 Desain sistem

Sesudah analisis sistem terselesaikan dilakukan, maka analisis sistem telah menerima ilustrasi dengan jelas apa yang harus dilakukan. Kini saatnya bagi analisis sistem untuk mengkaji cara membentuk sistem tersebut. Termin ini dianggap menggunakan Desain Sistem (*Systems Design*). Desain sistem bisa dibagi menjadi dua bagian, yaitu Desain Sistem secara umum (*General System Design*) serta Desain Sistem secara naratif (*Detailed Systems Design*). Desain Sistem secara umum (*General Systems Design*) disebut juga dengan desain Konseptual (*Conceptual Design*) atau desain logika (*Logical Design*) atau desain secara macro (*Macro Design*). Desain Sistem secara terinci disebut juga dengan desain sistem secara Fisik (*Physical System Design*) atau desain internal (*Internal desain*). Pada sistem gosip, alat bantu desain dibutuhkan. Di tahapan ini, pengembangan sistem bisa menentukan desain sistem merencanakan gambaran teoritis berasal sistem, merencanakan kumpulan data, merencanakan antar muka, sampai membentuk diagram alur acara. Keliru satu perangkat yang dapat dipergunakan dalam memutuskan sistem bantu keputusan ialah Data flow Diagram (DFD). DFD merupakan contoh nalar atau proses info yang didesain untuk menggambarkan awal info serta tujuan gosip yang muncul berasal sistem, dimana isu disimpan serta proses yang dipaksa di informasi.

Tabel 2.2 Tabel Simbol DFD

NO	SIMBOL	NAMA	KETERANGAN
1		Proses Transformasi	Proses yang membarui data dari input menjadi hasil
2		Sumber dan Tujuan Data	Karyawan dan organisasi yang mengirim data ke dan mendapatkan asal sistem
3		Arus Data	Arus data yang masuk ke dalam serta keluar berasal sebuah proses
4		Peyimpanan Data	Penyimpanan Data

2.5 Implementasi Sistem

Sistem dianalisis dirancang secara rinci dan teknologi telah diselksi dan dipilih. Waktunya telah tiba untuk menerapkan sistem. Tahap implementasi sistem adalah tahap dimana sistem diataur sehingga siap untuk digunakan. Tahap implementasi sistem dapat mencakup langkah-langkah berikut :

1. Rencana implementasiRencana implementasi adalah kegiatan awal dari fase implementasi sistem. Rencana pelaksanaan terutama ditujukan untuk mengolah biaya dan waktu yang diperlukan selama hidup implementasi.

2. Kegiatan implementasi

Kegiatan implementasi dilakukan berdasarkan kegiatan yang direncanakan dalam rencana implementasi. Kegiatan yang dapat dilakukan selama tahap implementasi ini adalah sebagai berikut ;

- a. Persiapan lokasi dan instalasi perangkat keras dan perangkat lunak
jika anda ingin memiliki peralatan baru, anda perlu menyiapkan lokasi atau ruangan untuk peralatan tersebut terlebih dahulu. Keamanan fisik tempat juga harus diperhatikan. Sistem mainframe membutuhkan lingkungan komputasi yang lebih baik. Langkah selanjutnya setelah menyiapkan lokasi atau ruangan untuk peralatan tersebut terlebih dahulu. Keamanan fisik tempat juga harus diperhatikan. Sistem mainframe membutuhkan lingkungan komputasi yang lebih baik. Langkah selanjutnya setelah menyiapkan fasilitas adalah menginstal perangkat keras yang sudah dikirimkan dan menginstal perangkat keras yang sudah dikirimkan dan menginstal perangkat keras yang sudah dikirimkan dan menginstal perangkat lunak yang sudah ada.

- b. Pemrograman dan pengujian sistem

Pemrograman merupakan proses menulis rangkaian instruksi yang akan dieksekusi oleh komputer. Dalam membuat kode program, penting bagi seorang programmer untuk merujuk pada dokumentasi yang diberikan oleh hasil analisis sistem dari desain sistem yang terperinci. Sebelum menjalankan program, penting untuk memastikan bahwa tidak ada kesalahan dalam program tersebut. Karena itu, penting untuk memeriksa program agar dapat mengidentifikasi kesalahan yang mungkin terjadi. Program diuji untuk setiap bagian dan dilanjutkan dengan menguji untuk semua bagian yang telah dikompilasi.

- c. Pengujian sistem

Pengujian sistem secara umum dilaksanakan sesudah pengujian perangkat lunak. Pengujian sistem dikerjakan untuk memvalidasi

apakah komponen sistem yang diterapkan terdapat kesesuaian. Pengujian sistem ini bertujuan utama untuk memverifikasi bahwa bagian-bagian atau komponen beroperasi sesuai dengan keinginan.

2.6 Pengujian sistem

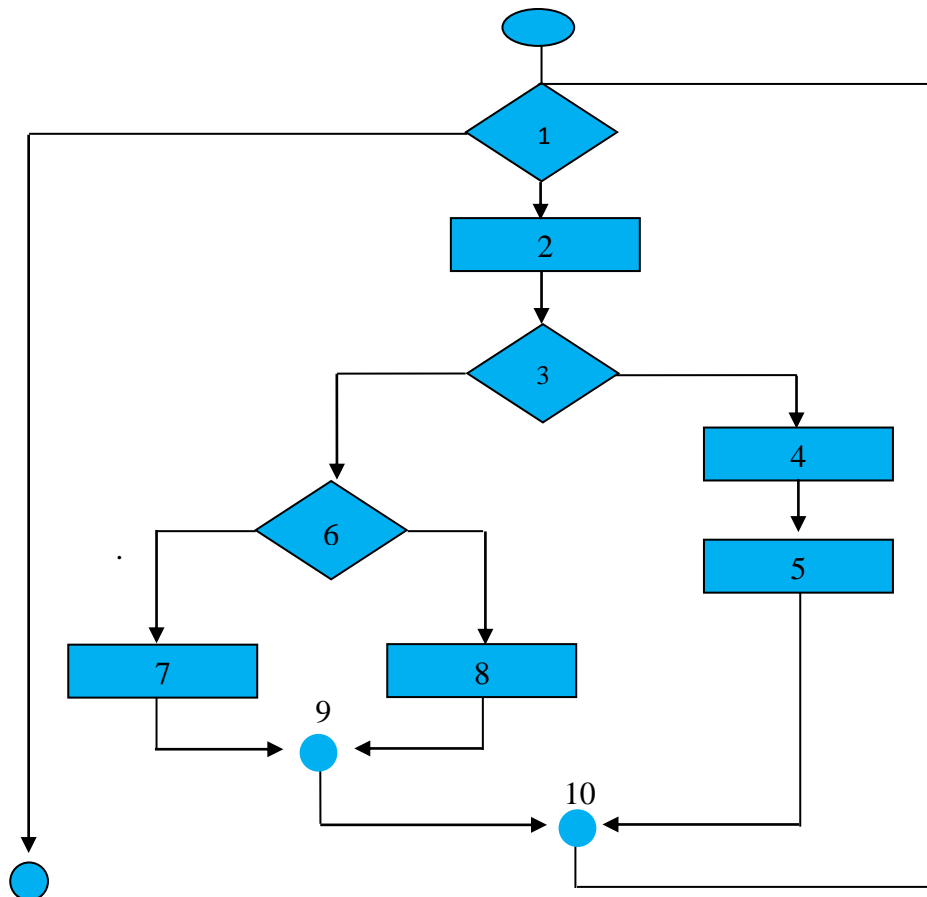
2.6.1 White box

Pengujian *White box* (*glass box*), adalah komponen penting dari jaminan kualitas perangkat lunak dan menyajikan pemeriksaan mendasar tentang spesifikasi, desain dan pengkodean.

Pengujian sistem/perangkat lunak dilakukan dengan mengikuti beberapa aturan yang berperan sebagai target pengujian, termasuk sebagai berikut:

1. Pengujian adalah proses eksekusi suatu program dengan maksud menemukan kesalahan.
2. *Test case* yang baik adalah test case yang memiliki probabilitas tinggi untuk menemukan kesalahan yang belum pernah ditemukan sebelumnya.
3. Pengujian yang sukses adalah pengujian yang mengungkapkan semua kesalahan yang belum pernah ditemukan sebelumnya.

Pengujian *White box*, adalah metode pengujian yang menggunakan proses desain struktur kontrol untuk mendapatkan *test case*. Menggunakan metode *White box*, perekrayan sistem dapat melakukan *test case* yang memastikan bahwa semua jalur independen dalam modul telah digunakan setidaknya sekali, menggunakan semua keputusan logis pada sisi berat dan salah, menjalankan semua loop hingga batasnya membatasi operasi merdeka, dan menggunakan struktur data internal untuk memastikan validitasnya. Pengujian basis path adalah teknik pengujian *White box* yang dikembangkan oleh Tom McCabe dalam pendekatan white-box. Metode basis path memungkinkan perencanaan test case untuk mengukur kompleksitas logis dari desain prosedural dan digunakan sebagai pedoman untuk menentukan serangkaian jalur eksekusi fundamental.



Gambar 2. 3 Bagan Air

Bagan arus digunakan untuk menggambarkan struktur kendali program. Dan untuk menggambarkan diagram alir, harus memperhatikan representasi desain prosedural pada diagram alir. Dalam ilustrasi di atas, diagram alir mengkonversi bagan alir tersebut menjadi diagram alir yang sesuai (dengan mengabaikan kemungkinan kondisi senyawa yang diisi dalam keputusan berbentuk berlian pada bagan alir tersebut). Setiap lingkaran dalam grafik aliran yang disebut simpul mewakili satu atau beberapa pernyataan prosedural. Rangkaian kotak proses dan permata pengambilan keputusan bisa digunakan untuk menghubungkan satu simpul. Anak panah yang disebut edge atau links, menggambarkan mengalirnya

kontrol dan serupa dengan anak panah dalam diagram alir. Edge harus berakhir di suatu titik, meskipun simpul tersebut tidak mewakili pernyataan prosedural.

- Node adalah lingkaran yang mewakili satu atau lebih statemen prosedural
- Tepi adalah anak panah dalam diagram alir
- Wilayah adalah area yang terbatas pada tepi dan node
- Simpul predikat adalah simpul atau node yang berisi kondisi yang ditandai dengan dua atau lebih edge yang berasal darinya.

Dari gambar flowchart di atas didapat :

Path 1 = 1-11

Path 2 = 1-2-3-4-5-10-11

Path 3 = 1-2-3-6-8-9-10-1-11

Path 4 = 1-2-3-6-7-9-10-1-11

Path 1,2,3,4 yang telah didefinisikan diatas merupakan basis test untuk diagram alir.

Cyclomatic complexity digunakan untuk mencari jumlah path dalam satu *flowchart*. Dapat dipergunakan rumus sebagai berikut :

1. Jumlah region grafik alir sesuai dengan *cyclomatic complexity*.

Rumus :

$$V(G) = E - N + 2 \quad (2-8)$$

Dimana :

E = jumlah edge pada grafik alir

N = jumlah node pada grafik alir

Jalur 3 : 1-2-3-6-8-9-10-1-11

Jalur 4 : 1-2-3-6-7-9-10-1-11

Jalur 1,2,3, dan 4 yang didefinisikan di atas membentuk himpunan dasar untuk grafik aliran pada gambar . Bagaimana kita tahu berapa banyak jalan untuk mencari ? menghitung kompleksitas siklis memberikan jawabannya. Dasar dari kompleksitas siklis adalah teori grafik, dan memberi kita indikator perangkat

lunak yang sangat berguna. Kompleksitas siklis dihitung dengan salah satu dari tiga cara berikut :

1. Jumlah area diagram blok sesuai dengan kompleksitas siklik.
2. Kompleksitas siklik, $V(G)$, untuk grafik alir G ditentukan sebagai $V(G) = E - N + 2$ dimana E adalah jumlah edge grafik alir dan N adalah jumlah simpul grafik alir G
3. Kompleksitas siklik, $V(G)$, untuk grafik alir G juga di tentukan sebagai $V(G) = P + 1$, dimana p adalah jumlah simpul predikat yang diisikan dalam grafik alir G

Pada gambar grafik alir, kompleksitas siklik dapat dihitung dengan menggunakan masing-masing dari algoritma yang ditulis di atas :

1. Grafik alir mempunyai 4 region
2. $V(G) = 11 \text{ tepi} - 9 \text{ simpul} + 2 = 4$
3. $V(G) = 3 \text{ simpul yang diperkirakan} + 1 = 4$

Oleh karena itu, kompleksitas siklik dari grafik aliran pada gambar adalah 4. Lebih penting lagi, nilai untuk $V(G)$ memberi kita batas atas untuk jumlah alur independen yang membentuk himpunan basis, dan menyiratkan batas atas jumlah pengujian, yang harus didesain dan dieksekusi untuk menjamin semua statemen program.

2.6.2 Black Box

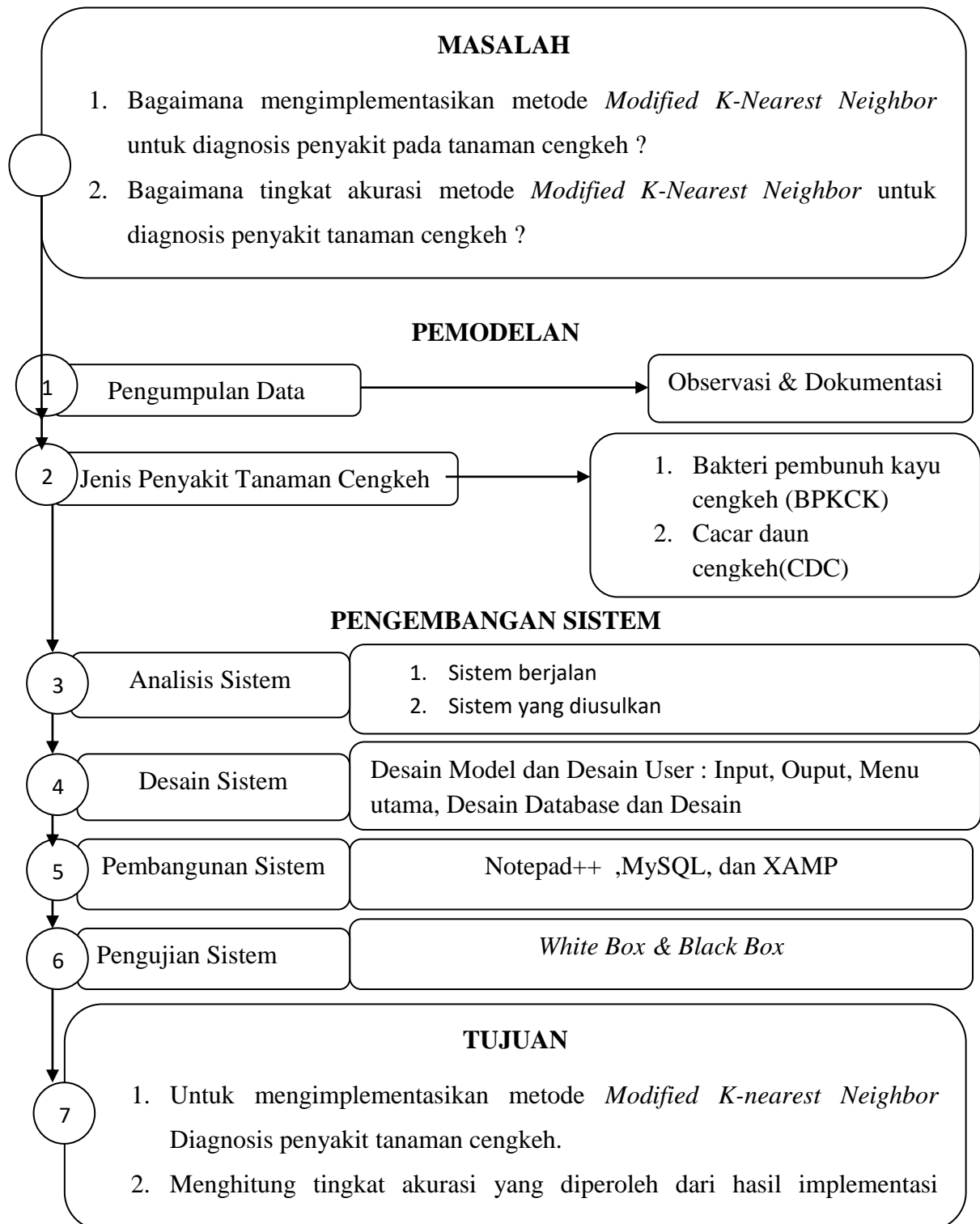
Pengujian *Black Box* bertujuan untuk mendeteksi kesalahan yang terdapat dalam kategori tertentu:

- Pada struktur data (pengaksesan basis data)
- Kesalahan fungsi tidak benar atau hilang
- Kesalahan antar muka
- Kesalahan inisialisasi dan penghentian program.
- Kesalahan kinerja.

Pengujian ini difokuskan pada fungsi-fungsi yang harus terpenuhi oleh perangkat lunak dan berfungsi sebagai tambahan untuk pengujian dengan pendekatan *White Box*. Ini dapat diwujudkan dengan:

- a. Pengujian Graph-based : mulailah dengan membuat grafik dari sekumpulan node yang mewakili objek (misal New File, Layar baru dengan atributnya), link (antar objek), node-weight (misal nilai data tertentu seperti atribut layar, perilaku), dan link-weight (karakteristik suatu link, misal pemilihan menu).
- b. Equivalence partitioning : pisahkan domain input atau menguji kelas kesalahan (misal kelompok data karakter, atau atribut yang lain).
- c. Analisis Nilai Batas : periksa nilai batas dari domain input.
- d. Pengujian Perbandingan : juga dikenal sebagai pengujian back-to-back yang diterapkan pada versi perangkat lunak atau perangkat lunak cadangan untuk memastikan konsistensinya.

2.7 Kerangka pemikira



Gambar 2. 4 Kerangka Pemikiran

BAB III

METOLOGI PENELITIAN

3.1 Jenis, Metode, Subjek, Objek, Waktu, dan Lokasi Penelitian

Dilihat dari tingkat penerapannya, maka penelitian ini termasuk penelitian terapan. Dilihat dari jenis informasi yang diolah, maka penelitian ini bersifat kuantitatif. Dipandang dari perlakuan terhadap data, maka penelitian ini merupakan penelitian konfirmatori.

Penelitian ini menggunakan metode penelitian survey. Oleh karena itu, jenis penelitian ini adalah penelitian deskriptif. Objek penelitian ini adalah mendiagnosa penyakit tanaman cengkeh yang berlokasi di Kecamatan Wawo Kabupaten Kolak Utara Provinsi Sulawesi Tenggara.

3.2 Pengumpulan Data

Data primer ini menggunakan data yang diperoleh dari observasi langsung dan melakukan wawancara kepada petani yang mengelolah kebun cengkeh dan ahli pakar dari tanaman cengkeh di Kecamatan Wawo Kabupaten Kolaka Utara Provinsi Sulawesi Tenggara. Proses pengumpulan data yang dibutuhkan dimulai pada tanggal 11 september sampai 16 september 2023, Didapatkan data 26 jenis gejala penyakit dan 3 jenis penyakit yang sudah diberi nilai untuk tiap parameternya sesuai dengan pengetahuan dari narasumber serta bagaimana penangana untuk masing-masing penyakit. Untuk rincingan gejala yang didapatkan tersebut ada pada tabel berikut ini

Tabel 3.1 Jenis Gejala

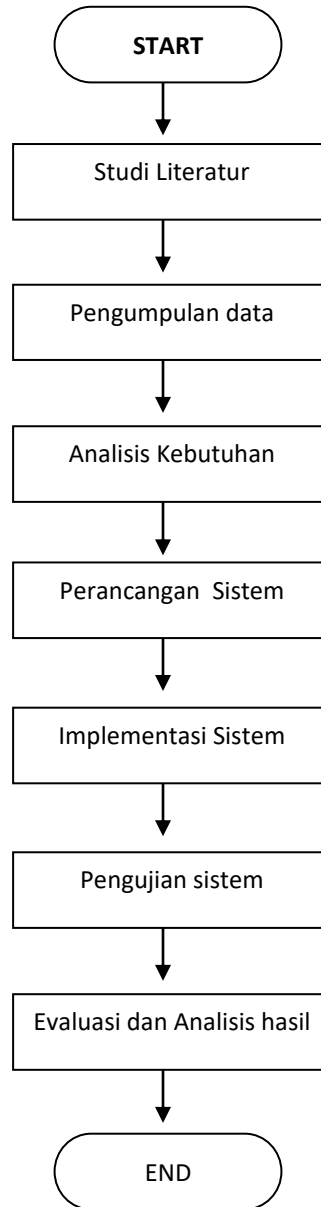
Kode Gejala	JENIS GEJALA
G-1	Daun tampak kekuning-kuningan
G-2	Daun mengering dan akhirnya mati
G-3	Kulit batang terlepas
G-4	Batang tanaman kering
G-5	Bercak-bercak cokelat kekuningan dari tepi helai daun

G-6	Tulang-tulang daun tampak terang
G-7	Daun menjadi rontok
G-8	Bercak-bercak berwarna kelabu sampai coklat yang memenuhi bagian permukaan daun
G-9	Bercak-bercak kecil pada daun berwarna merah
G-10	Bintik-bintik putih pada daun
G-11	Daun yang berada di bawah lebih banyak bercak cacar dari daun yang ada diatas
G-12	Ranting yang ada dipucuk batang kering
G-13	Daun-daun dari beberapa ranting berwarna kekuningan
G-14	Daun layu kemudian rontok
G-15	Ranting-ranting banyak yang mati
G-16	Gugur daun dari puncak pohon
G-17	Daun berwarna kelabu dan hampir gundul
G-18	Daun berwarna kekuning-kuningan dan agak layu
G-19	Daun sebagian tampak kering
G-20	Daun yang lebih tua berwarna kekuning-kuningan dan agak layu
G-21	Daun tampak layu dan akhirnya berubah warna menjadi coklat
G-22	Dalam waktu singkat daun-daun berguguran dari pohon
G-23	Tanaman mendadak layu dan kering
G-24	Ada titik hitam di bagian yang melepuh
G-25	Benang-benang miselium dari jamur ditemui pada bagian akar
G-26	Akar busuk dan bau tak sedap

Tabel 3.2 Nama Penyakit

KODE PENYAKIT	NAMA PENYAKIT
P-1	Bakteri pembunuh kayu cengkeh (BPKC)
P-2	Cacar daun cengkeh (CDC)
P-3	Jamur akar putih (JAP)

3.3 Pemodelan



Gambar 3. 1 Pemodelan

3.3.1 Studi Literatur

Tahapan ini dilakukan untuk mendapatkan dasar-dasar teori dan sumber acuan untuk pembuatan sistem Diagnosis penyakit pada tanaman cengkeh menggunakan metode *Modified K-Nearest Neighbor*, diantaranya sebagai berikut:

1. Teori metode *Modified K-Nearest Neighbor*
2. Berbagai jenis penyakit pada tanaman cengkeh dan pengendaliannya

Literatur-literatur tersebut diperoleh dari buku, jurnal, e-book, penelitian sebelumnya dan situs-situs ilmiah. Cara ini dilakukan untuk mendapatkan dasar-dasar referensi yang kuat bagi peneliti guna menyelesaikan penelitian

3.3.2 Analisis Kebutuhan

Analisis kebutuhan merupakan sebuah fase yang berfungsi untuk merumuskan kebutuhan-kebutuhan yang diperlukan dalam mengembangkan suatu sistem. Analisis kebutuhan harus sesuai dengan lokasi penelitian, variabel penelitian dan mempersiapkan kebutuhan penelitian.

Secara keseluruhan, kebutuhan yang digunakan dalam pembuatan sistem diagnosa penyakit tanaman cengkeh ini meliputi :

1. Spesifikasi kebutuhan hardware, meliputi :
 - Laptop
 - Memory 4GB
 - Hardish 1GB
2. Spesifikasi kebutuhan software, meliputi :
 - Windows 10 64bit sebagai sistem operasi.
 - XAMPP sebagai server localhost, MySQL termasuk didalamnya sebagai database management system (DBMS).

3. Data yang dibutuhkan, meliputi :

- Data gejala penyakit tanaman cengkeh
- Data nilai pembobotan gejala penyakit tanaman cengkeh

3.3.3 Analisis dan Perancangan Sistem

Analisis dan perencanaan sistem membahas mengenai hal-hal yang terkait dengan apa saja yang diperlukan dalam pembuatan sistem. Hal-hal tersebut meliputi deskripsi sistem, perencanaan program aplikasi, perhitungan manual, desain antarmuka dan perancangan pengujian. Hasil yang diharapkan pada perencanaan sistem yaitu dapat membangun sebuah sistem berdasarkan hasil dan analisis yang ada. Secara umum, sistem yang dibangun adalah sebuah sistem untuk mendiagnosis penyakit tanaman cengkeh menggunakan metode *Modified K-Nearest Neighbor* (MKNN).

Pada tahap analisis, pertama menyiapkan data penyakit tanaman cengkeh yang telah berupa angka untuk digunakan dalam proses diagnosis penyakit. Model metode yang akan digunakan dalam sebuah diagram alir, agar permasalahan mudah untuk dipecahkan dan dilakukan perencanaan sistem sesuai dengan analisis yang telah dilakukan.

Tahap perencanaan sistem dilakukan untuk memecahkan permasalahan diagnosis penyakit tanaman cengkeh meliputi deskripsi umum sistem, perancangan model MKNN, perancangan proses MKNN dengan diagram alir, contoh perhitungan manual, perancangan antarmuka dan perancangan pengujian sistem yang disajikan dalam bentuk tabel pengujian.

3.3.4 Implementasi

Implementasi perangkat lunak yang menerapkan algoritma *Modified K-Nearest Neighbor* dilakukan dengan mengacu pada perancangan sistem yang dibuat.

Implementasi sistem tersebut meliputi :

1. Implementasi *Interface*, menggunakan *software* Visual studio.
2. Implementasi basis data, dengan menggunakan sebuah DBMS MySQL pada *server localhost* (XAMPP) yang bertujuan untuk memudahkan manipulasi dan penyimpanan data.
3. Implementasi algoritma, melakukan perhitungan dengan metode *Modified K-Nearst Neighbor* ke dalam bahasa pemrograman *Javascript* dan menggunakan *software* Visual Studio.
4. Implementasi ini akan menghasilkan Diagnosis penyakit yang menyerang tanaman cengkeh melalui inputan yang berupa gejala penyakit.

3.3.6 Pengujian Sistem

1. White Box Testing

Software yang telah direkayasa kemudian diuji dengan metode White Box Testing pada kode program proses penerapan metodenya/modelnya. Kode program tersebut dibuatkan Flowchart programnya, kemudian dipetakan kedalam bentuk flowgraph (bagan air kontrol) yang tersusun dari beberapa node dan edge. Berdasarkan Flowgraph, ditentukan jumlah region dan Cyclomatic Complexity (CC). Apabila $\text{independen path} = V(G) = (CC) = \text{Region}$, dimana setiap path hanya dieksekusi sekali dan sudah benar, maka sistem dinyatakan efisien dari segi kelayakan logika pemrograman.

2. Black Box Testing

Selanjutnya software diuji pula dengan metode Black Box Testing yang fokus pada keperluan fungsional dari software dan berusaha untuk menemukan kesalahan dalam beberapa kategori, diantaranya : (1) Fungsi-fungsi yang salah atau hilang; (2) Kesalahan interface; (3) Kesalahan dalam struktur data atau akses basis data eksternal; (4) Kesalahan performa; (5) Kesalahan inisialisasi dan terminasi. Jika sudah tidak ada kesalahan-kesalahan tersebut, maka sistem dinyatakan efisien dari segi kesalahan komponen-komponen system.

3.3.7 Evaluasi dan Analisis Hasil

Setelah proses pengujian sistem, tahap evaluasi dilakukan. Tujuan evaluasi ini adalah untuk menilai tingkat keberhasilan penerapan metode dalam memecahkan masalah diagnosis penyakit tanaman cengkeh dan apakah kinerja metode MKNN lebih optimal sesuai dengan tujuan penelitian ini. Tahap ini juga merupakan tahap akhir dari proses, yaitu penarikan kesimpulan dari hasil pengujian dan analisis yang telah dilakukan. Kesimpulan dan rekomendasi juga dapat digunakan oleh pembaca sebagai referensi untuk perbaikan dan pengembangan aplikasi berikutnya.

BAB IV

HASIL PENELITIAN

4.1 Hasil Pengumpulan Data

Pada matriks penyakit Tanaman Cengkeh terdiri 3 macam jenis penyakit yang ditunjukkan oleh kode P01,P02 dan P03 yaitu terdiri dari penyakit Bakteri pembunuh kayu cengkeh (BPKCK), Cacar daun cengkeh (CDC), dan Jamur akar putih (JAP). Semua jenis-jenis penyakit terdapat 26 Gejala penyakit yang ditunjukkan oleh kode G01 sampai G26

Penyakit yang terdiri dari 3 jenis penyakit dan 26 gejala penyakit yang dapat diolah, sehingga dapat menghasilkan suatu kesimpulan serta solusi akhir yang dapat dipahami dari permasalahan yang ada.

Tabel 4. 1 Daftar penyakit Tanaman Cengkeh

KODE PENYAKIT	NAMA PENYAKIT
P-01	Bakteri pembunuh kayu cengkeh (BPKC)
P-02	Cacar daun cengkeh (CDC)
P-03	Jamur akar putih (JAP)

Tabel 4. 2 Daftar Gejala Penyakit tanaman Cengkeh

Kode Gejala	JENIS GEJALA
G-01	Daun tampak kekuning-kuningan
G-02	Daun mengering dan akhirnya mati
G-03	Kulit batang terlepas
G-04	Batang tanaman kering
G-05	Bercak-bercak cokelat kekuningan dari tepi helai daun
G-06	Tulang-tulang daun tampak terang
G-07	Daun menjadi rontok
G-08	Bercak-bercak berwarna kelabu sampai cokelat yang memenuhi bagian permukaan daun
G-09	Bercak-bercak kecil pada daun berwarna merah
G-10	Bintik-bintik putih pada daun
G-11	Daun yang berada di bawa lebih banyak bercak cacar dari daun yang ada diatas

G-12	Ranting yang ada dipucuk batang kering
G-13	Daun-daun dari beberapa ranting berwarna kekuningan
G-14	Daun layu kemudian rontok
G-15	Ranting-ranting banyak yang mati
G-16	Gugur daun dari puncak pohon
G-17	Daun berwarna kelabu dan hampir gundul
G-18	Daun berwarna kekuning-kuningan dan agak layu
G-19	Daun sebagian tampak kering
G-20	Daun yang lebih tua berwarna kekuning-kuningan dan agak layu
G-21	Daun tampak layu dan akhirnya berubah warna menjadi cokelat
G-22	Dalam waktu singkat daun-daun berguguran dari pohon
G-23	Tanaman mendadak layu dan kering
G-24	Ada titik hitam di bagian yang melepuh
G-25	Benang-benang miselium dari jamur ditemui pada bagian akar
G-26	Akar busuk dan bau tak sedap

Tabel 4. 3 Penyakit dan Gejala Tanaman Cengkeh

Kode penyakit	Nama Penyakit	Kode gejala	Nama gejala
P-01	Bakteri pembunuh kayu cengkeh (BPKCK)	G-01	Daun tampak kekuning-kuningan
		G-02	Daun mengering dan akhirnya mati
		G-03	Kulit batang terlepas
		G-04	Batang tanaman kering
		G-05	Ranting-ranting banyak yang mati
		G-06	Daun berwarna kekuning-kuningan dan agak layu
		G-07	Daun yang lebih tua berwarna kekuning-kuningan dan agak layu Gugur daun dari puncak pohon
		G-08	Gugur daun dari puncak pohon
P-02	Cacar daun cengkeh (CDC)	G-09	Daun yang lebih tua berwarna kekuning-kuningan dan agak

			layu
		G-10	Tulang-tulang daun tampak terang
		G-11	Daun menjadi rontok
		G-12	Bercak-bercak berwarna kelabu sampai cokelat yang memenuhi bagian permukaan daun
		G-13	Bercak-bercak kecil pada daun berwarna merah
		G-14	Bintik-bintik putih pada daun
		G-15	Daun yang berada di bawah lebih banyak bercak cacar dari daun yang ada diatas
		G-16	Daun-daun dari beberapa ranting berwarna kekuningan
		G-17	Daun berwarna kelabu dan hampir gundul
		G-18	Daun sebagian tampak kering
		G-19	Ada titik hitam di bagian yang melepuh
P-3	Jamur akar putih (JAP)	G-20	Ranting yang ada dipucuk batang kering
		G-21	Daun layu kemudian rontok
		G-22	Daun tampak layu dan akhirnya berubah warna menjadi cokelat
		G-23	Dalam waktu singkat daun-daun berguguran dari pohon
		G-24	Tanaman mendadak layu dan kering
		G-25	Benang-benang miselium dari jamur ditemui pada bagian akar
		G-26	Akar busuk dan bau tak sedap

Tabel 4. 4 Daftar Bobot Dari Setiap Gejala Penyakit

Kode Gejala	NAMA GEJALA PENYAKIT	Bobot
G-01	Daun tampak kekuning-kuningan	0.5
G-02	Daun mengering dan akhirnya mati	0.5
G-03	Kulit batang terlepas	0.5
G-04	Batang tanaman kering	0.5
G-05	Bercak-bercak cokelat kekuningan dari tepi helai daun	0.5
G-06	Tulang-tulang daun tampak terang	0.5
G-07	Daun menjadi rontok	0.5

G-08	Bercak-bercak berwarna kelabu sampai coklat yang memenuhi bagian permukaan daun	0.5
G-09	Bercak-bercak kecil pada daun berwarna merah	0.5
G-10	Bintik-bintik putih pada daun	0.5
G-11	Daun yang berada di bawah lebih banyak bercak cacar dari daun yang ada diatas	0.5
G-12	Ranting yang ada dipucuk batang kering	0.5
G-13	Daun-daun dari beberapa ranting berwarna kekuningan	0.5
G-14	Daun layu kemudian rontok	0.5
G-15	Ranting-ranting banyak yang mati	0.5
G-16	Gugur daun dari puncak pohon	0.5
G-17	Daun berwarna kelabu dan hampir gundul	0.5
G-18	Daun berwarna kekuning-kuningan dan agak layu	0.5
G-19	Daun sebagian tampak kering	0.5
G-20	Daun yang lebih tua berwarna kekuning-kuningan dan agak layu	0.5
G-21	Daun tampak layu dan akhirnya berubah warna menjadi coklat	0.5
G-22	Dalam waktu singkat daun-daun berguguran dari pohon	0.5
G-23	Tanaman mendadak layu dan kering	0.5
G-24	Ada titik hitam di bagian yang melepuh	0.5
G-25	Benang-benang miselium dari jamur ditemui pada bagian akar	0.5
G-26	Akar busuk dan bau tak sedap	0.5

4.2 Hasil Pemodelan Data

Pemodelan Data merupakan proses yang berfungsi untuk mendefinisikan dan menganalisis kebutuhan data yang diperlukan. Berikut adalah hasil pemodelan data dari sistem.

Tabel 4. 5 Matrix Tanaman Penyakit Jagung

P/G	P01	P02	P03
G-01	*		
G-02	*		
G-03	*		
G-04	*		
G-05	*		
G-06	*		
G-07	*		
G-08	*		
G-09		*	

G-10		*	
G-11		*	
G-12		*	
G-13		*	
G-14		*	
G-15		*	
G-16		*	
G-17		*	
G-18		*	
G-19		*	
G-20			*
G-21			*
G-22			*
G-23			*
G-24			*
G-25			*
G-26			*

4.3 Hasil Pengembangan Sistem

4.3.1 Analisis Sistem

Analisis sistem merujuk pada proses menyelidiki, merencanakan, dan menganalisis sistem informasi untuk memastikan bahwa sistem tersebut memenuhi kebutuhan bisnis atau organisasi dengan efisien dan efektif. Tujuan dari analisis sistem adalah untuk memahami secara mendalam bagaimana suatu sistem bekerja, mengidentifikasi masalah atau kekurangan, serta merancang perbaikan atau peningkatan yang diperlukan.

Tahap analisis ini meneliti sistem berjalan untuk mengidentifikasi masalah. Ini memudahkan tahap berikutnya, yaitu tahap perancangan sistem.

4.3.1.1 Analisis Sistem Berjalan

Analisis sistem berjalan digunakan untuk mendeteksi penyakit tanaman cengkeh dengan melihat sistem yang sedang berjalan atau lama digunakan. Karena sistem berjalan saat ini dilakukan secara manual, membutuhkan waktu yang lama karena petani harus bertemu langsung dengan pakar untuk bertanya tentang gejala penyakit. Setelah itu, pakar membuat laporan data tentang penyakit tanaman cengkeh dan dikirim ke Kepala Dinas Pertanian sebagai Laporan Kegiatan. Sistem saat ini dapat menyebabkan tanaman cengkeh lenih mati dengan mudah karena penanganan yang lambat.

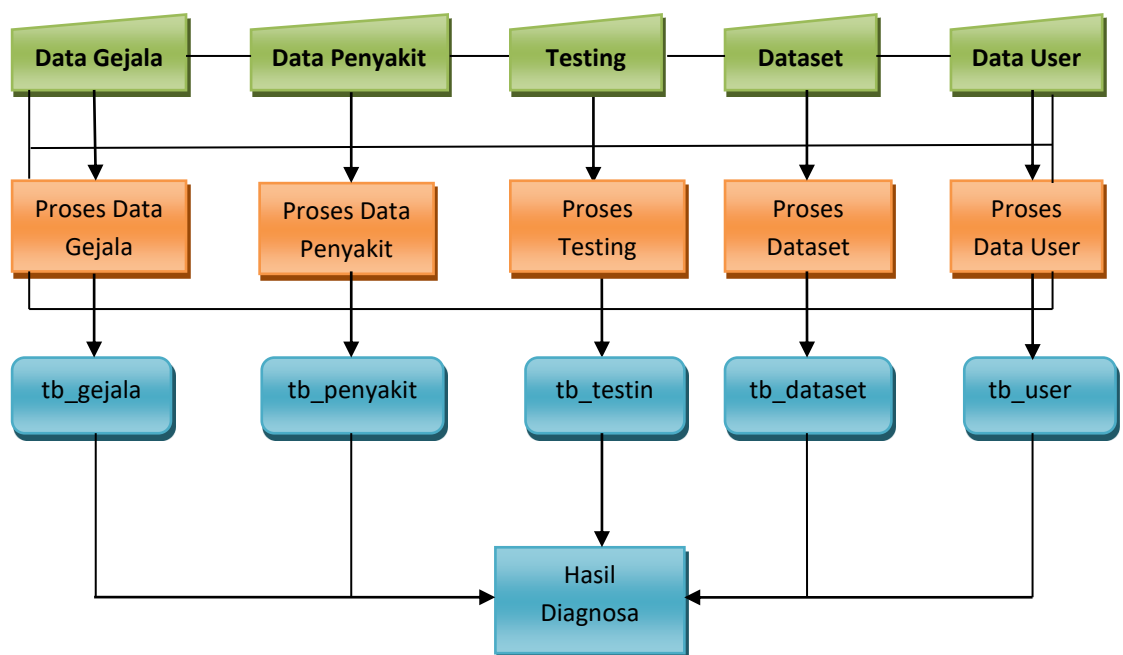
Beberapa penyakit tanaman cengkeh dapat menyebar dengan cepat melalui tanah jika tidak ditangani dengan segera, menyebabkan seluruh tanaman terjangkit, menyebabkan gagal panen dan kerugian bagi petani.

Jika tidak ditangani dengan segera, beberapa penyakit tanaman cengkeh dapat menyebar dengan cepat melalui tanah, menyebabkan seluruh tanaman terjangkit, menyebabkan panen gagal dan kerugian bagi petani.

4.3.1.2 Analisis Sistem Yang Diusulkan

Analisis sistem yang disarankan adalah penguraian sistem untuk setiap bagian komponen. Tujuan dari analisis ini adalah untuk menemukan dan mengevaluasi masalah setiap peluang, tantangan, dan kebutuhan untuk perbaikan.

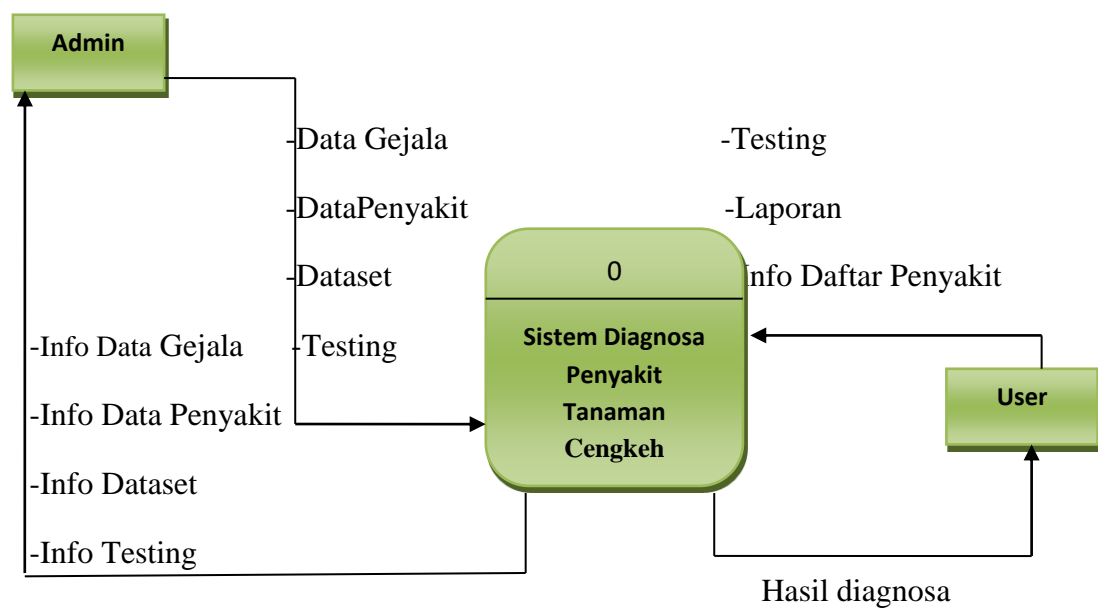
Tujuan dari analisis ini adalah untuk menemukan dan mengevaluasi masalah setiap peluang, kesulitan, dan kebutuhan perbaikan. Analisis sistem yang disarankan adalah penguraian sistem untuk setiap bagian komponennya.



Gambar 4. 1 Sistem Yang diusulkan

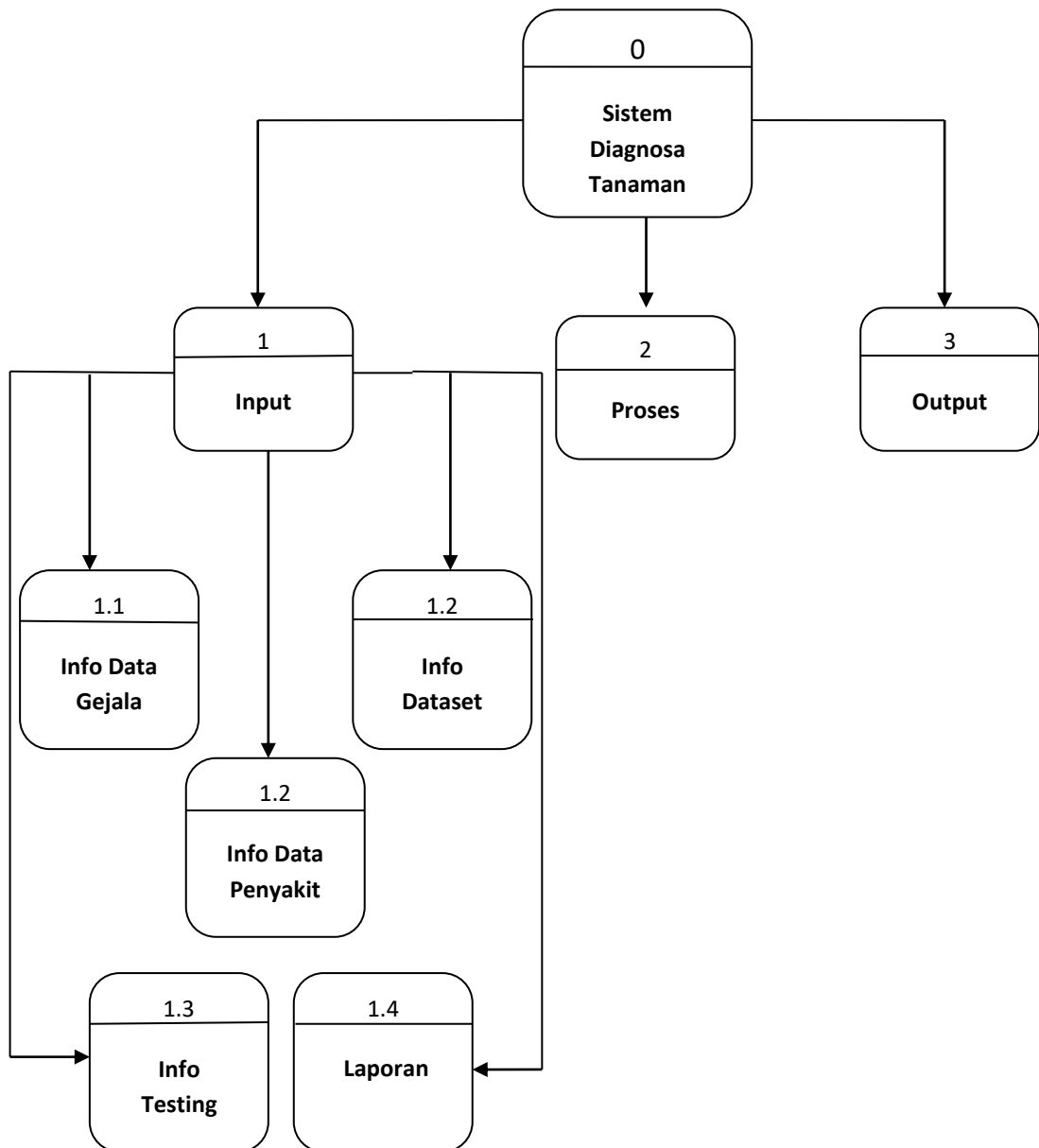
4.3.2 Diagram Konteks

Diagram konteks, yang merupakan tingkat DFD yang lebih tinggi, terdiri dari suatu proses dan menggambarkan ruang lingkup suatu sistem. Diagram konteks terdiri dari dua entitas, yaitu administrator dan pengguna. Dengan memasukkan data tentang penyakit, gejala, dan basis pengetahuan atau kasus yang telah dia peroleh dari set data, pengguna dapat melakukan konsultasi secara langsung tanpa harus menghubungi admin. Selanjutnya, sistem akan mengeluarkan output kepada pengguna berupa hasil diagnosa.



Gambar 4. 2 Diagram Konteks

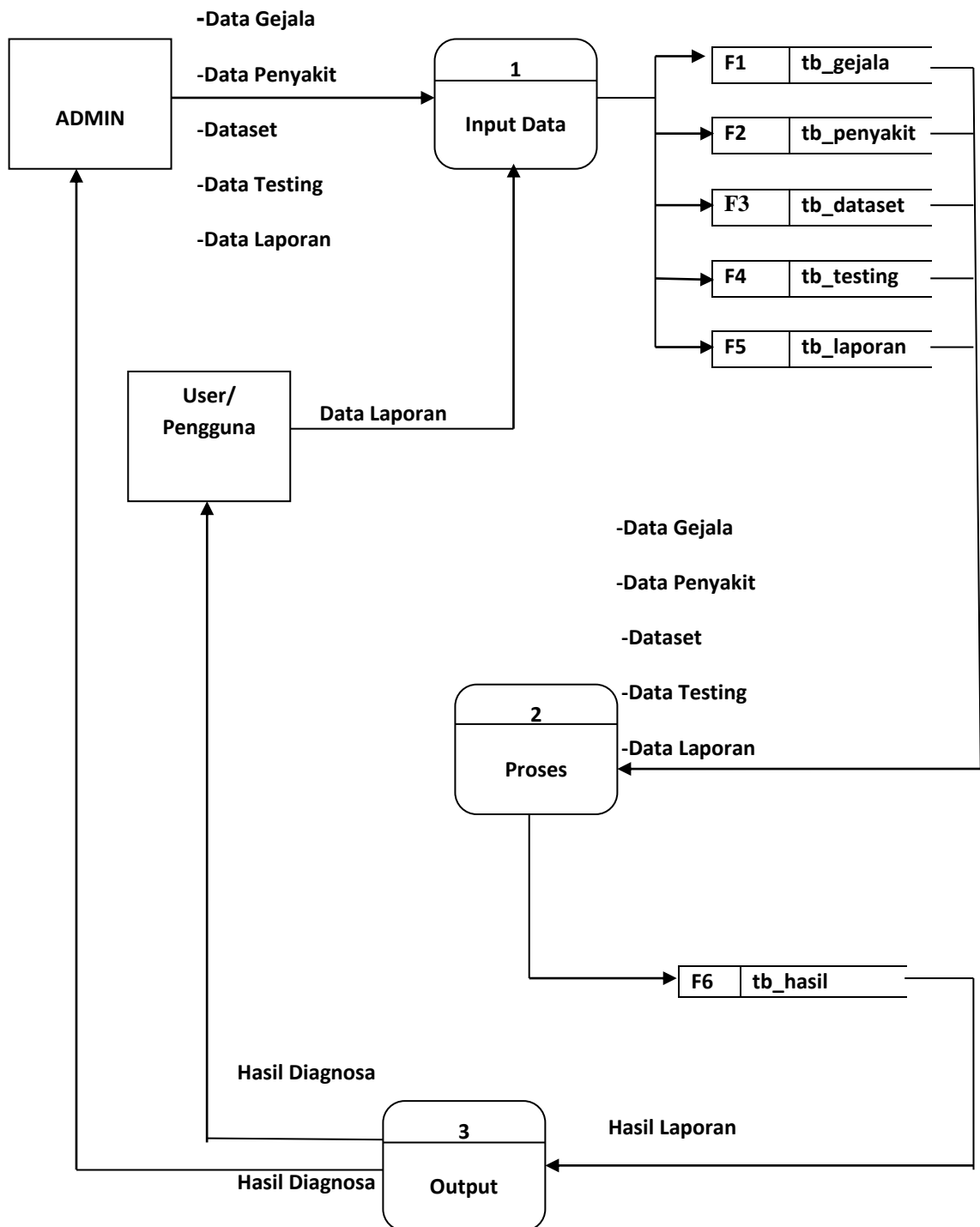
4.3.3 Diagram Berjenjang



Gambar 4. 3 Diagram Berjenjang

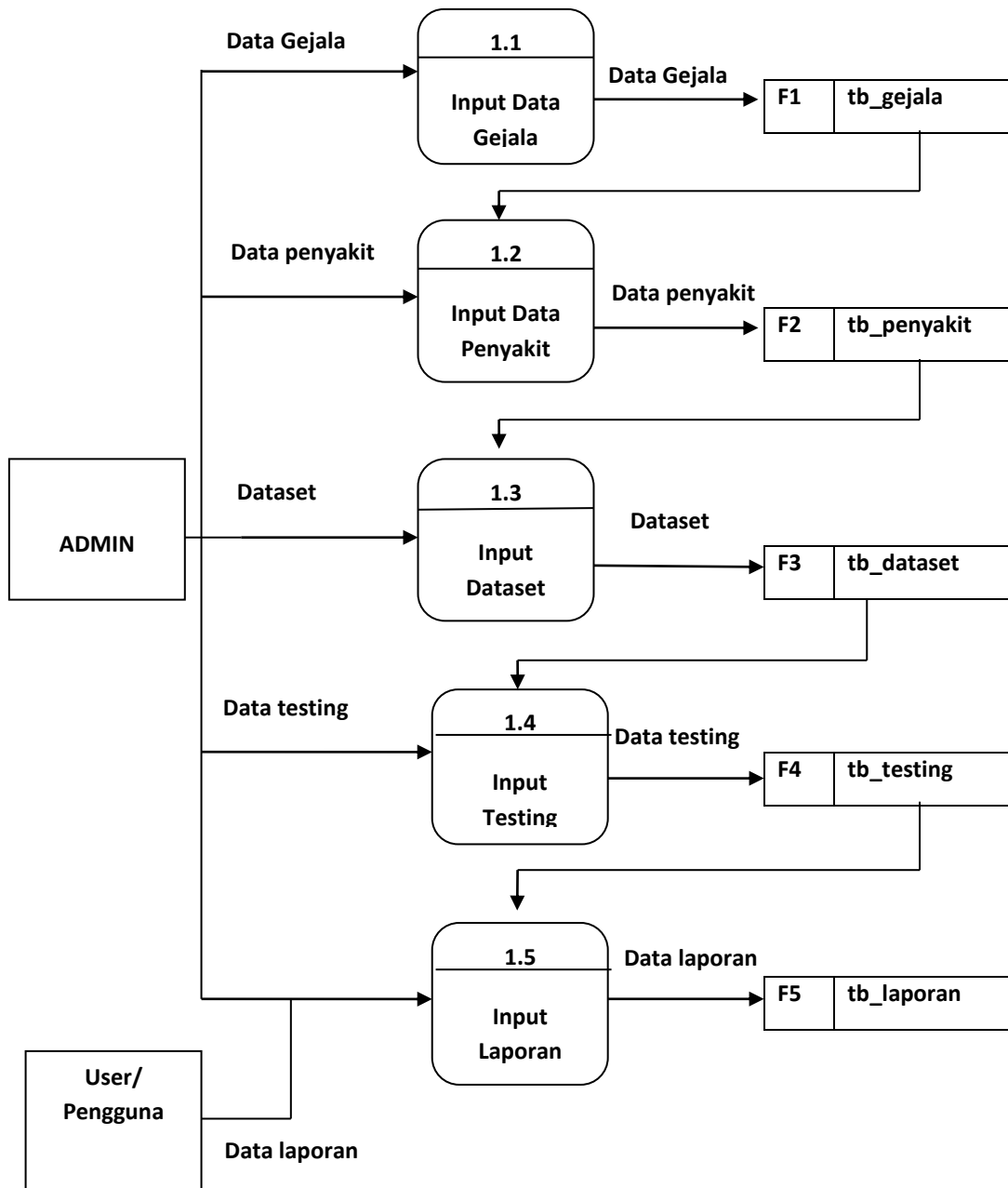
4.3.4 Diagram Arus Data (DAD)

4.3.4.1 Diagram Arus Data (DAD) Level 0



Gambar 4. 4 Diagram Arus Data Level 0

4.3.4.2 Diagram Arus Data (DAD) Level 1



Gambar 4. 5 Diagram Arus Data Level 1

4.3.5 Kamus Data

Tabel 4. 6 Kamus Data User

Nama Database : dbtanaman				
Nama Tabel : tb_user				
No	Field	Type	Size	Keterangan
1	id_user	int	5	id user
2	nama_lengkap	varchar	50	nama lengkap user
3	username	Varchar	30	nama user
4	password	Varchar	15	password user
5	jenis_kelamin	Varchar	15	jenis kelamin user
6	status_admin	varchar	20	admin

Tabel 4. 7 Kamus Data Gejala

Nama Database : dbtanaman				
Nama Tabel : tb_gejala				
Fungsi : Tempat Untuk Menyimpan Data Gejala				
No	Field	Type	Size	Keterangan
1	No	int	5	nomor
2	Kd_g	varchar	5	Kode gejala
3	Kd_p	Varchar	5	Kode penyakit
4	Gejala	Varchar	50	Nama gejala
5	Bobot	float	-	-

Tabel 4. 8 Kamus Data Penyakit

Nama Database : dbtanaman				
Nama Tabel : tb_penyakit				
Fungsi : Tempat Untuk Menyimpan Data Penyakit				
No	Field	Type	Size	Keterangan
1	No	int	5	nomor
2	kd_p	varchar	5	Kode penyakit
3	nama_penyakit	Varchar	50	Nama penyakit

Tabel 4. 9 Kamus Data Dataset

Nama Database : dbtanaman				
Nama Tabel : tb_dataset				
No	Field	Type	Size	Keterangan
1	no	int	5	id user
2	nama_lengkap	varchar	50	nama lengkap user
3	username	Varchar	30	nama user
4	password	Varchar	15	password user
5	jenis_kelamin	Varchar	15	jenis kelamin user
6	status_admin	varchar	20	admin

Tabel 4. 10 Kamus Data k

Nama Database : dbtanaman				
Nama Tabel : tb_k				
No	Field	Type	Size	Keterangan
1	No	int	5	nomor
2	Kode_penyakit	Varchar	10	Kode penyakit
3	k	float	-	Nilai k

Tabel 4. 11 Kamus Data Hasil

Nama Database : dbtanaman				
Nama Tabel : tb_hasil				
No	Field	Type	Size	Keterangan
1	No	int	5	nomor
2	kd_p	Varchar	5	kode penyakit
3	penyakit	Varchar	50	nama penyakit
4	k	float	-	kilai k

Tabel 4. 12 Kamus Data Testing

Nama Database : dbtanaman				
Nama Tabel : tb_testing				
No	Field	Type	Size	Keterangan
1	No	int	5	nomor
2	Nama_tanaman	Varchar	100	Nama tanaman
3	penyakit	Varchar	50	nama penyakit
4	G1	float	-	nilai k
5	G2	float	-	nilai k
6	G3 Samapai	float	-	nilai k
7	G26	float	-	nilai k

Tabel 4. 13 Kamus Data Laporan

Nama Database : dbtanaman				
Nama Tabel : tb_laporan				
No	Field	Type	Size	Keterangan
1	No	int	5	nomor
2	Nama_tanaman	Varchar	100	Nama tanaman
3	penyakit	Varchar	50	nama penyakit
4	G1	float	-	nilai k
5	G2 Samapai	float	-	nilai k
6	G26	float	-	nilai k

4.3.6 Arsitektur Sistem/Kebutuhan *Hardware dan Software*

Penulis menggunakan Notepad++ dan XAMPP Database untuk membuat sistem ini. Untuk menjalankan program aplikasi, Anda harus memiliki perangkat lunak atau perangkat keras dasar berikut:

1. Spesifikasi Hardware dan Software

- a. Prosesor AMD RYZEN 3 7320U atau lebih
- b. RAM (memory) 2 atau lebih
- c. ROM 256 gb atau lebih
- d. Monitor SVGA dengan Resolusi 1024×768
- e. Dan peralatan I/O lainnya
- f. Windows 8, Windows 10, atau lebih
- g. *Google Chrome* untuk membuka *Web*

4.3.7 Desain Secara Umum

4.3.7.1 Desain Output Secara Umum

Untuk : Diagnosa penyakit tanaman Cengkeh

Sistem : Menggunakan Algoritma MKNN untuk diagnosa penyakit tanaman cengkeh

Tahap : Desain Output Secara Umum

Tabel 4. 14 Desain Output Secara Umum

No	Nama	Tipe	Format	Media	alat	Distribusi
1	Daftar Gejala Yang dipilih	Internal/ Eksternal	Tabel	Layar	Monitor	Pengguna, Admin
2	Hasil Diagnosa	Internal/ Eksternal	Tabel	Layar	Monitor	Pengguna, Admin

4.3.7.2 Desain Input Secara Umum

Tujuan Sari Desain Input secara umum adalah untuk memberi pengguna gambaran umum tentang sistem baru atau sistem yang diusulkan. Komponen sistem informasi yang akan didesain secara rinci biasanya diidentifikasi dalam desain sistem. Pemrograman komputer dan ahli implementasi sistem menggunakan desain khusus.

Untuk : Diagnosa Penyakit Tanaman Cengkeh

Sistem : Penerapan Algoritma Modified K-nearest Neighbor (MKNN)

untuk diagnosa penyakit Tanaman Cengkeh

Tahap : Desain Input Secara Umum

Tabel 4. 15 Desain Input Secara Umum

No	Nama	Tipe	Format	Media	Alat	Distribusi
1	Data User	Internal	Tabel	Layar	Monitor	Admin
2	Dataset	Internal	Tabel	Layar	Monitor	Admin
3	Data Gejala	Internal	Tabel	Layar	Monitor	Admin
4	Data Penyakit	Internal	Tabel	Layar	Monitor	Admin

4.3.8 Desain Secara Terinci

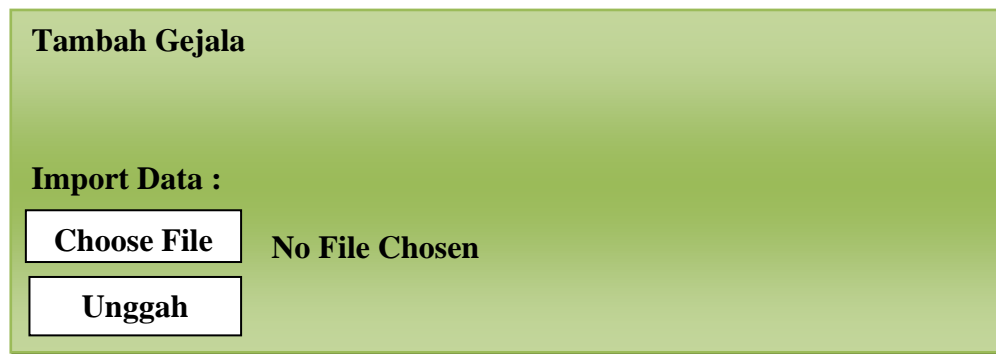
4.3.8.1 Desain Output Secara Terinci

Gambar berikut adalah desain dari hasil analisis yang dimaksudkan untuk menampilkan data pengguna. Ini menunjukkan identifikasi dataset penyakit, gejala, dan penyakit tanaman cengkeh.

4.3.8.2 Desain Input Secara Terinci

1) Input Data Gejala

Berikut adalah gambar desain input gejala ang dibuat untuk menginput data gejala penyakit Tanaman Cengkeh ke tabel tb_gejala database.



Tambah Gejala

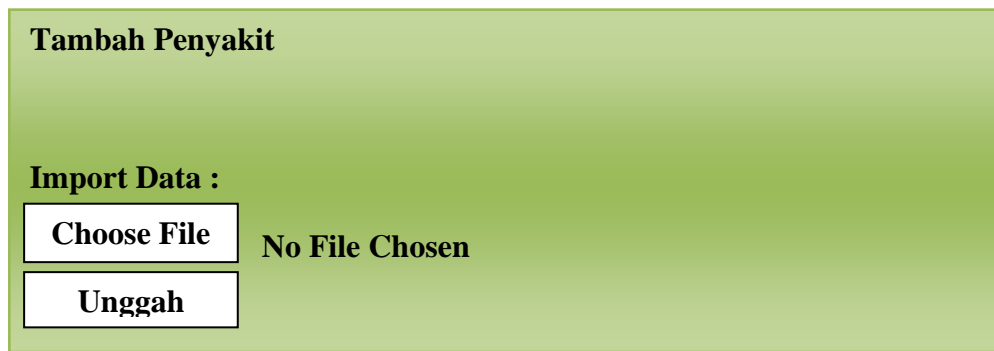
Import Data :

Choose File No File Chosen

Unggah

Gambar 4. 6 Desain Input Gejala

2) Input Data Penyakit



Tambah Penyakit

Import Data :

Choose File No File Chosen

Unggah

Gambar 4. 7 Desain Input Penyakit

4.3.8.3 desain Database Secara Terinci

Tabel 4. 16 Database

No	Field	Type	Size	Keterangan
1	No	int	5	nomor
2	Kd_g	varchar	5	Kode gejala
3	Kd_p	Varchar	5	Kode penyakit
4	Gejala	Varchar	50	Nama gejala
5	Bobot	float	-	Nilai gejala

Tabel 4. 17 Penyakit

No	Field	Type	Size	Keterangan
1	No	int	5	nomor
2	kd_p	varchar	5	Kode penyakit
3	nama_penyakit	Varchar	50	Nama penyakit

Tabel 4. 18 Gejala

No	Field	Type	Size	Keterangan
1	No	int	5	nomor
2	Kd_g	varchar	5	Kode gejala
3	Kd_p	Varchar	5	Kode penyakit
4	Gejala	Varchar	50	Nama gejala
5	Bobot	float	-	Nilai gejala

Tabel 4. 19 Testing

No	Field	Type	Size	Keterangan
1	No	int	5	nomor
2	Nama_tanaman	Varchar	100	Nama tanaman
3	penyakit	Varchar	50	nama penyakit
4	G1	float	-	nilai k
5	G2	float	-	nilai k
6	G3 Samapai	float	-	nilai k
7	G26	float	-	nilai k

Tabel 4. 20 Hasil

No	Field	Type	Size	Keterangan
1	No	int	5	nomor
2	kd_p	Varchar	5	kode penyakit
3	penyakit	Varchar	50	nama penyakit
4	k	float	-	nilai k

Tabel 4. 21 Laporan

No	Field	Type	Size	Keterangan
1	No	int	5	nomor
2	Nama_tanaman	Varchar	100	Nama tanaman
3	penyakit	Varchar	50	nama penyakit
4	G1	float	-	nilai k
5	G2 Samapai	float	-	nilai k
6	G26	float	-	nilai k

4.3.8.4 Desain Menu Utama



Gambar 4. 8 Desain Menu Utama

4.3.9.1 Pengujian White Box

White Box yaitu untuk menguji semua statement program. Penggunaan metode pengujian *White Box* dilakukan untuk:

- Memberikan jaminan bahwa semua jalur *indenpendnt* suatu modul digunakan minimal satu kali.
- Menggunakan semua keputusan logis untuk semua kondisi *true* atau *false*.
- Mengeksekusi semua perulangan pada batasan nilai dan operasional pada setiap kondisi

b. Menghitung Nilai Cyclomatic Complexity

Dari Flowgraph di dapatkan

- Region (R) = 7
- Node (N) = 15
- Edge (E) = 20
- Predicate Node (P) =

Dari Flowgraph di atas, Cyclomatic complexity dari sebuah program dapat dibuat dengan menggunakan rumus dibawah ini :

$V(G)$: Cyclomatic Complexity

E : Total Jumlah edge $V(G) = E - N + 2$

N : Total jumlah node

Pada Flowgraph di atas (Gambar), dapat dihitung Cyclomatic Complexity nya sebagai berikut

$$\begin{aligned} V(G) &= 20 \text{ Edge} - 15 \text{ Node} + 2 \\ &= 7 \end{aligned}$$

Atau

$$\begin{aligned} V(G) &= 6 \text{ Predicate} + 1 \\ &= 7 \end{aligned}$$

c. Menentukan Basic path

Angka 7 dari hasil perhitungan kompleksitas cyclomatik menunjukkan jumlah jalur independen dari jalur pengujian basis, atau jumlah pengujian yang

harus dilakukan untuk memastikan bahwa semua pernyataan pada program dijalankan setidaknya sekali.

Hasil independent path pada contoh di atas dapat dijabarkan sebagai berikut :

Path 1 : 1-2-3-4-5-6-23-7-15

Path 2 : 1-2-3-4-5-2-3-7-15

Path 3 : 1-2-3-7-8-9-10-11-2-3-7-15

Path 4 : 1-2-3-7-8-9-10-11-12-2-3-7-15

Path 5 : 1-2-3-7-8-9-13-14-2-3-7-15

Path 6 : 1-2-3-7-8-9-13-15

Path 7 : 1-2-3-7-15

4.3.9.2 Pengujian Black Box

Di bawah ini adalah pengujian dengan metode Black Box dari perangkat lunak yang telah dibuat; ini adalah pendekatan komplementer dari metode White Box.

Tabel 4. 22 Pengujian Blackbox

NO	Input/Event	Fungsi	Hasil	Hasil
1	Halaman Utama	Menampilkan halaman utama	Halaman Utama Tampilan	Sesuai
2	Klik Login	Menampilkan Menu utama Login admin	Tampilan Menu Utama Admin Tampil	Sesuai
3	Login Input Username yang Salah	Menampilkan Pesan kesalahan	Pesan Kesalahan Username Yang Anda Masukkan Salah	Sesuai
4	Login Input Password yang salah	Menampilkan Pesan kesalahan	Pesan Kesalahan Password Yang anda Masukkan	Sesuai

			Salah	
5	Klik Menu Beranda	Menampilkan Halaman depan Beranda	Halaman Depan Beranda Tampil	Sesuai
6	Klik Menu Gejala	Menampilkan Data Gejala	Data Gejala Tampil	Sesuai
7	Klik Tambah Data Gejala	Menampilkan Data Gejala	Tambahan Data Gejala Di Tampilkan	Sesuai
8	Klik Ubah Pada Data Gejala	Menampilkan Form Ubah Data Gejala	Ubah Data Gejala Di Tampilkan	Sesuai
9	Klik Hapus Pada Data Gejala	Menampilkan Pesan	Tampil Pesan Data Berhasil Dihapus	Sesuai
10	Klik Menu Data penyakit	Menampilkan Data Penyakit	Data Penyakit Tampil	Sesuai
11	Klik Tambah Data Penyakit	Menampilkan Tambahan Data Penyakit	Tambahan data Penyakit Di Tampilkan	Sesuai
12	Klik Menu Informasi Penyakit	Menampilkan Informasi Penyakit	Informasi Penyakit Tampil	Sesuai
13	Klik Sub Data Admin	Menampilkan Data Admin	Data Admin Tampil	Sesuai
14	Klik Sub Ubah Password Pada Menu Administrator	Menampilkan Form Data Ubah Password	Form Data Ubah Password Tampil	Sesuai
15	Klik Sub Logout Pada Menu Administrator	Menampilkan Halaman Utama	Halaman Utama Tampilan	Sesuai

Hasil pengujian menunjukkan bahwa uji Black Box, yang menguji input, proses, dan output perangkat lunak, telah dilakukan dengan benar dan hasil sesuai dengan desain. Program utama dan program pendukung lainnya juga diuji.

BAB V

PEMBAHASAN PENELITIAN

5.1 Pembahasan Sistem

Pada tahap ini dilakukan hasil perancangan antarmuka kedalam sistem yang telah dibangun dengan menggunakan perangkat lunak.

5.1.1 Halaman Menu Utama




Gambar 5. 1 Halaman Utama

Pada tampilan menu utama ini terdapat 6 menu yang bisa digunakan pada sistem, yaitu Beranda, User, Dataset, Proses, Hasil, dan Logout. Sedangkan pada tampilan yang berada dibawa menu terdapat link untuk mempercepat/mempermudah untuk menampilkan penyakit atau gejala dan sebagainya

5.1.2 Halaman Login

LOGIN




Username

admin

Password

.....




LOGIN

Gambar 5. 2 Halaman Login

Pada halaman login, bisa digunakan para user untuk mengakses kedalam menu utama menggunakan username dan password yang sudah dibuat oleh user.

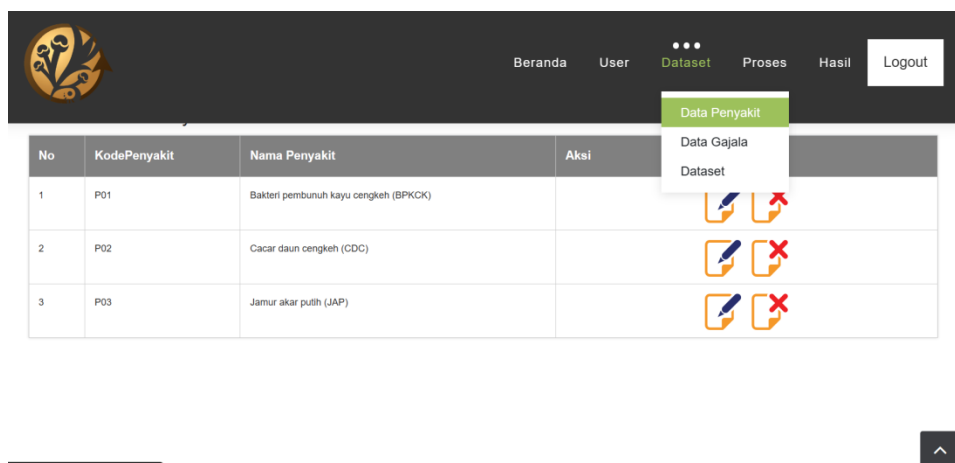
5.1.3 Halaman Menu Gejala

<div>  <div> Beranda User Dataset Proses Hasil Logout </div> </div>					
No	Kode Gejala	Kode Penyakit	Gejala	Bobot	
1	G01	P01	Daun tampak kekuning- kuningan	0.5	<div> Data Penyakit Data Gejala Dataset </div>
2	G02	P01	Daun mengering dan akhirnya mati	0.5	<div> </div>
3	G03	P01	Kulit batang terlepas	0.5	<div> </div>
4	G04	P01	Batang tanaman kering	0.5	<div> </div>
5	G05	P01	Ranting-ranting banyak yang mati	0.5	<div> </div>
6	G06	P01	Gugur daun dari puncak pohon	0.5	<div> </div>
7	G07	P01	Daun berwarna kekuning-kuningan dan agak layu	0.5	<div> </div>

Gambar 5. 3 Halaman Menu Gejala

Pada tampilan menu gejala, bisa digunakan untuk melihat gejala-gejala dari penyakit yang menyerang tanaman Cengkeh, dan pada halaman ini juga bisa digunakan untuk mengubah/menginput, menambah serta menghapus gejala.

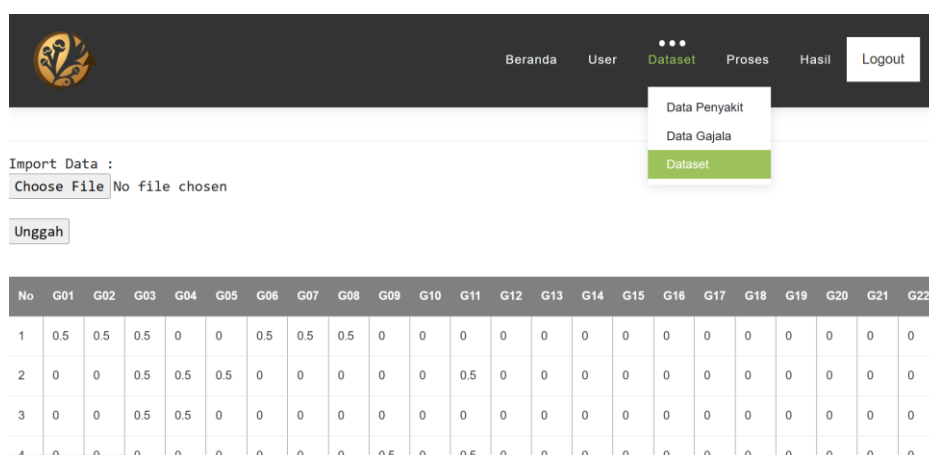
5.1.4 Halaman Menu Penyakit



Gambar 5. 4 Halaman Menu Penyakit

Pada Tampilan menu penyakit ini juga admin bisa melihat penyakit menyerang tanaman Cengkeh, serta admin juga bisa mengubah, menambah, dan menghapus nama penyakit.

5.1.5 Halaman Menu Dataset



Gambar 5. 5 Halaman Menu Dataset

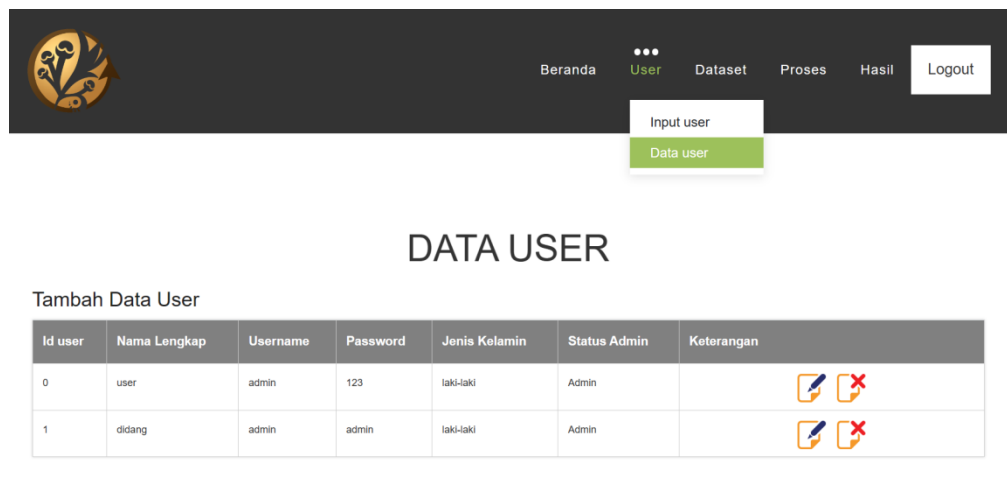
Pada Tampilana Menu Dataset, admin bisa melihat secara detail setiap gejala-gejala, penyakit, dan nilai bobot yang dialami pada setiap tanaman cengkeh, serta bisa untuk mengubah, menambah, dan menghapus data yang telah diunggah.

5.1.6 Halaman Menu Input User

Gambar 5. 6 Halaman Input User

Pada tampilan menu input user, dapat digunakan oleh admin untuk membuat username, password dan data user baru untuk digunakan user agar dapat mengakses halaman utama serta dapat mengubah, menambah, dan menghapus data user.

5.1.7 Halaman Menu Data User



Gambar 5. 7 Halaman Menu Data User

Pada halaman Menu Data User, Admin dapat melihat secara detail data user yang telah diinput pada halaman menu input user serta dapat mengubah dan menghapus data user yang telah ada.

5.1.8 Halaman Menu Diagnosa

1. No 10

2. Nama Tanaman

3. Daun tampak kekuning-kuningan ☐ Ya ☒ Tidak

4. Daun mengering dan akhirnya mati ☐ Ya ☒ Tidak

5. Kulit batang terlepas ☐ Ya ☒ Tidak

6. Batang tanaman kering ☐ Ya ☒ Tidak

7. Ranting-ranting banyak yang mati ☐ Ya ☒ Tidak

8. Gugur daun dari puncak pohon ☐ Ya ☒ Tidak

9. Daun berwarna kekuning-kuningan dan agak layu ☐ Ya ☒ Tidak

10. Gugur daun dari puncak pohon ☐ Ya ☒ Tidak

11. Bercak-bercak cokelat kekuningan dari tepi helai d ☐ Ya ☒ Tidak

12. Tulang-tulang daun tampak terang ☐ Ya ☒ Tidak

13. Daun menjadi rontok ☐ Ya ☒ Tidak

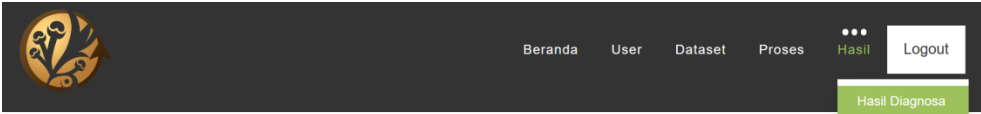
14. Bercak-bercak berwarna kelabu sampai cokelat yang ☐ Ya ☒ Tidak









15. Bercak-bercak kecil pada daun berwarna merah ☐ Ya ☒ Tidak

Gambar 5. 8 Halaman Menu Diagnosa

Pada halaman menu diagnosa, dapat digunakan oleh user untuk mengisi nama tanaman yang terserang penyakit dan pertanyaan mengenai gejala-gejala yang dialami tanaman yang digunakan untuk sebagai data uji agar dapat di diagnosa jenis penyakit yang dialami tanaman tersebut.

5.1.9 Halaman Menu Hasil Diagnosa



No	Nama Tanaman	Hasil Diagnosa	AKSI
1	Cengkeh 1	Cacar daun cengkeh (CDC)	 
2	Cengkeh 2	Bakteri pembunuh kayu cengkeh (BPKCK)	 
3	Cengkeh 3	Jamur akar putih (JAP)	 
4	Cengkeh 4	Cacar daun cengkeh (CDC)	 

localhost/Didang/hasil.php

Gambar 5. 9 Halaman Menu Hasil Diagnosa

Pada Halaman Menu Hasil Diagnosa, dapat digunakan oleh admin untuk melihat secara detail hasil diagnosa nama penyakit yang dialami tiap kasus tanaman cengkeh yang diinput oleh user sebagai data uji serta dapat menghapus hasil diagnosa yang telah diinput oleh user.

BAB VI

KESIMPULAN DAN SARAN

6.1 KESIMPULAN

Berdasarkan hasil penelitian yang dilakukan pada dinas pertanian Kabupaten Kolaka Utara yang berlokasi di Ponggiha, Kec. Lasusua, Kabupaten Kolaka Utara. Berdasarkan hasil analisa dan pengujian sistem diagnosa pada tanaman Cengkeh dengan menggunakan Metode MODIFIED K-NEAREST NEIGHBOR. Maka yang dapat saya simpulkan bahwa :

1. Sistem ini dibangun mampu mendiagnosa 3 penyakit pada tanaman Cengkeh berdasarkan gejala-gejala yang telah dikumpulkan yang sesuai dengan perhitungan manual
2. Dapat diketahui cara merekayasa sistem diagnosa penyakit tanaman Cengkeh
3. Sistem ini dapat diimplementasikan untuk mendiagnosa penyakit tanaman Cengkeh. Hal ini dapat dibuktikan dengan hasil pengujian yang dilakukan dengan Metode MODIFIED K-NEAREST NEIGHBOR yang menghasilkan nilai Cyclomatic Complexity = 7

6.2 SARAN

Bagi siapapun yang ingin memperbaiki atau menyempurnakan sistem ini. Adapun saran yang dapat penulis sampaikan dalam pengembangan sistem ini antara lain sebagai berikut :

- Sistem diagnosa penyakit tanaman Cengkeh ini dapat dikembangkan oleh siapapun dengan menggunakan metode lain agar lebih akurat dan menghasilkan hasil yang tepat

DAFTAR PUSTAKA

- [1] Mh Badrut Tamam, “Cengkeh,” *generasibiologi.com*. Accessed: Feb. 13, 2024. [Online]. Available: <https://generasibiologi.com/2018/04/klasifikasi-ciri-deskripsi-kandungan-manfaat-cengkeh-cengkih.html>
- [2] Dwi Erianto, “Komoditas Cengkeh: Sejarah, Manfaat, Produsen Dunia, dan Produksi Indonesia,” *kompaspedia.kompas.id*. Accessed: Feb. 13, 2024.
- [3] R. Amsyar, N. Hidayat, and R. S. Perdana, “Implementasi Algoritme Modified K-Nearest Neighbor (MKNN) untuk Diagnosis Penyakit Tanaman Cengkeh,” 2018.
- [4] F. Wafiyah, N. Hidayat, and R. S. Perdana, “Implementasi Algoritma Modified K-Nearest Neighbor (MKNN) untuk Klasifikasi Penyakit Demam,” *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 1, no. 10, pp. 1210–1219, 2017.
- [5] T. H. Simanjuntak, W. F. Mahmudy, and Sutrisno Sutrisno, “Implementasi Modified K-Nearest Neighbor Dengan Otomatisasi Nilai K Pada Pengklasifikasian Penyakit Tanaman Kedelai,” *Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 1, No.2, no. 2, pp. 75–79, 2017,
- [6] H. Dwi, A. Hamid, N. Hidayat, and R. K. Dewi, “Diagnosis Penyakit Tanaman Cabai Menggunakan Metode Modified K-Nearest Neighbor (MKNN) (Studi Kasus: BPTP Karang Ploso Malang),” vol. 3, no. 3, pp. 2881–2886, 2019,
- [7] JAGOAN HOSTING TEAM, “Data Mining: Pengertian, Fungsi, Metode & Penerapannya,” *jagoanhosting.com*. Accessed: Feb. 13, 2024.

- [8] S. H. Sutrisno Nurul; Arahman, Mohammad Yusuf, “Diagnosis Hama dan Organisme Pengganggu Tanaman Bawang Merah Metode Algoritma Modified K-Nearest Neighbour (MKNN),” *J. Pengem Tek Inf. dan Ilk.*, vol. 3, no. Vol 3 No 1 (2019), pp. 404–415, 2019.
- [9] N. Silvia, T. Cahyani, N. Hidayat, and E. Santoso, “Klasifikasi Penyakit Tanaman Kacang Tanah menggunakan Metode MKNN (Modified K-Nearest Neighbor),” *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 7, no. 3, pp. 1191–1197, 2023,
- [10] E. A. Aipassa, D. Kusnandar, and N. Imro’ah, “PENERAPAN ALGORITMA MODIFIED K-NEAREST NEIGHBOR (MK-NN) DALAM KLASIFIKASI KELULUSAN MAHASISWA (Studi Kasus: Prodi Statistika Universitas Tanjungpura),” *Bul. Ilm. Math. Stat. dan Ter.*, vol. 13, no. 1, pp. 89–98, 2024.

LAMPIRAN 1 : KODE PROGRAM

```

include_once "koneksi.php";

$query = mysqli_query($kon, "select *from tb_laporan order by
No desc limit 0,1");

while ($q= mysqli_fetch_array($query)) {

    echo $q ['No']+1;

}

?>

"></td></tr>

    <tr><td>2. </td><td>Nama    Tanaman    </td><td><input
type='text' name='nt'></td></tr>

    <tr><td>3. </td><td>

<?php

    include_once "koneksi.php";

    $query = mysqli_query($kon, "select *from tb_gejala where
No = '1'");

    while ($q= mysqli_fetch_array($query)) {

        echo $q ['Gejala'];

    }

?>

```

```

        </td><td><input type='radio' name='g1' value='0.5'> Ya
        <input type='radio' name='g1' value='0' checked> Tidak </td>

    </tr>

    <tr><td>4. </td><td>

```

```
<?php
```

```
    include_once "koneksi.php";
```

```
    $query = mysqli_query($kon, "select *from tb_gejala where
    No = '2'");
```

```
    while ($q= mysqli_fetch_array($query)) {
```

```
        echo $q ['Gejala'];
```

```
    }
```

```
?>
```

```

        </td><td><input type='radio' name='g2' value='0.5'> Ya
        <input type='radio' name='g2' value='0' checked> Tidak</td>

```

```
    </tr>
```

```
    <tr><td>5. </td><td>
```

```
<?php
```

```
    include_once "koneksi.php";
```

```
    $query = mysqli_query($kon, "select *from tb_gejala where
    No = '3'");
```

```

while ($q= mysqli_fetch_array($query)) {

echo $q ['Gejala'];

}

?>

</td><td><input type='radio' name='g3' value='0.5'> Ya
<input type='radio' name='g3' value='0' checked> Tidak</td>

</tr>

<tr><td>6. </td><td>

<?php

include_once "koneksi.php";

$query = mysqli_query($kon, "select *from tb_gejala where
No = '4'");

while ($q= mysqli_fetch_array($query)) {

echo $q ['Gejala'];

}

?>

</td><td><input type='radio' name='g4' value='0.5'> Ya
<input type='radio' name='g4' value='0' checked> Tidak</td>

</tr>

<tr><td>7. </td><td>

```

```
<?php
```

```
    include_once "koneksi.php";
```

```
    $query = mysqli_query($kon, "select *from tb_gejala where  
No = '5'");
```

```
    while ($q= mysqli_fetch_array($query)) {
```

```
        echo $q ['Gejala'];
```

```
    }
```

```
?>
```

```
    </td><td><input type='radio' name='g5' value='0.5'> Ya  
<input type='radio' name='g5' value='0' checked> Tidak</td>
```

```
</tr>
```

```
<tr><td>8. </td><td>
```

```
<?php
```

```
    include_once "koneksi.php";
```

```
    $query = mysqli_query($kon, "select *from tb_gejala where  
No = '6'");
```

```
    while ($q= mysqli_fetch_array($query)) {
```

```
        echo $q ['Gejala'];
```

```
    }
```

```
?>
```

```

        </td><td><input type='radio' name='g6' value='0.5'> Ya
        <input type='radio' name='g6' value='0' checked> Tidak</td>

```

```

    </tr>

```

```

    <tr><td>9. </td><td>

```

```

<?php

```

```

    include_once "koneksi.php";

```

```

    $query = mysqli_query($kon, "select *from tb_gejala where
    No = '7'");

```

```

    while ($q= mysqli_fetch_array($query)) {

```

```

        echo $q ['Gejala'];

```

```

    }

```

```

?>

```

```

        </td><td><input type='radio' name='g7' value='0.5'> Ya
        <input type='radio' name='g7' value='0' checked> Tidak</td>

```

```

    </tr>

```

```

    <tr><td>10. </td><td>

```

```

<?php

```

```

    include_once "koneksi.php";

```

```

    $query = mysqli_query($kon, "select *from tb_gejala where
    No = '8'");

```

```

while ($q= mysqli_fetch_array($query)) {

echo $q ['Gejala'];

}

?>

</td><td><input type='radio' name='g8' value='0.5'> Ya
<input type='radio' name='g8' value='0' checked> Tidak</td>

</tr>

<tr><td>11. </td><td>

<?php

include_once "koneksi.php";

$query = mysqli_query($kon, "select *from tb_gejala where
No = '9'");

while ($q= mysqli_fetch_array($query)) {

echo $q ['Gejala'];

}

?>

</td><td><input type='radio' name='g9' value='0.5'> Ya
<input type='radio' name='g9' value='0' checked> Tidak</td>

</tr>

<tr><td>12. </td><td>

```



```
<?php
```

```
    include_once "koneksi.php";
```

```
    $query = mysqli_query($kon, "select *from tb_gejala where  
No = '10'");
```

```
    while ($q= mysqli_fetch_array($query)) {
```

```
        echo $q ['Gejala'];
```

```
    }
```

```
?>
```

```
    </td><td><input type='radio' name='g10' value='0.5'> Ya  
<input type='radio' name='g10' value='0' checked> Tidak</td>
```

```
    </tr>
```

```
    <tr><td>13. </td><td>
```

```
<?php
```

```
    include_once "koneksi.php";
```

```
    $query = mysqli_query($kon, "select *from tb_gejala where  
No = '11'");
```

```
    while ($q= mysqli_fetch_array($query)) {
```

```
        echo $q ['Gejala'];
```

```
    }
```

```
?>
```

```

        </td><td><input type='radio' name='g11' value='0.5'> Ya
        <input type='radio' name='g11' value='0' checked> Tidak</td>

    </tr>

    <tr><td>14. </td><td>

```

```

<?php

```

```

    include_once "koneksi.php";

```

```

    $query = mysqli_query($kon, "select *from tb_gejala where
    No = '12'");

```

```

    while ($q= mysqli_fetch_array($query)) {

```

```

        echo $q ['Gejala'];

```

```

    }

```

```

?>

```

```

        </td><td><input type='radio' name='g12' value='0.5'> Ya
        <input type='radio' name='g12' value='0' checked> Tidak</td>

```

```

    </tr>

```

```

    <tr><td>15. </td><td>

```

```

<?php

```

```

    include_once "koneksi.php";

```

```

    $query = mysqli_query($kon, "select *from tb_gejala where
    No = '13'");

```

```

while ($q= mysqli_fetch_array($query)) {

echo $q ['Gejala'];

}

?>

</td><td><input type='radio' name='g13' value='0.5'> Ya
<input type='radio' name='g13' value='0' checked> Tidak</td>

</tr>

<tr><td>16. </td><td>

<?php

include_once "koneksi.php";

$query = mysqli_query($kon, "select *from tb_gejala where
No = '14'");

while ($q= mysqli_fetch_array($query)) {

echo $q ['Gejala'];

}

?>

</td><td><input type='radio' name='g14' value='0.5'> Ya
<input type='radio' name='g14' value='0' checked> Tidak</td>

</tr>

```

```
<tr><td>17. </td><td>
```

```
<?php
```

```
include_once "koneksi.php";
```

```
$query = mysqli_query($kon, "select *from tb_gejala where  
No = '15'");
```

```
while ($q= mysqli_fetch_array($query)) {
```

```
echo $q ['Gejala'];
```

```
}
```

```
?>
```

```
</td><td><input type='radio' name='g15' value='0.5'> Ya  
<input type='radio' name='g15' value='0' checked> Tidak</td>
```

```
</tr>
```

```
<tr><td>18. </td><td>
```

```
<?php
```

```
include_once "koneksi.php";
```

```
$query = mysqli_query($kon, "select *from tb_gejala where  
No = '16'");
```

```
while ($q= mysqli_fetch_array($query)) {
```

```
echo $q ['Gejala'];
```

```
}
```

```
?>
```

```

        </td><td><input type='radio' name='g16' value='0.5'> Ya
        <input type='radio' name='g16' value='0' checked> Tidak</td>

```

```
</tr>
```

```
<tr><td>19. </td><td>
```

```
<?php
```

```
include_once "koneksi.php";
```

```

        $query = mysqli_query($kon, "select *from tb_gejala where
        No = '17'");

```

```
while ($q= mysqli_fetch_array($query)) {
```

```
    echo $q ['Gejala'];
```

```
}
```

```
?>
```

```

        </td><td><input type='radio' name='g17' value='0.5'> Ya
        <input type='radio' name='g17' value='0' checked> Tidak</td>

```

```
</tr>
```

```
<tr><td>20. </td><td>
```

```
<?php
```

```
include_once "koneksi.php";
```

```
$query = mysqli_query($kon, "select *from tb_gejala where  
No = '18'");
```

```
while ($q= mysqli_fetch_array($query)) {
```

```
echo $q ['Gejala'];
```

```
}
```

```
?>
```

```
</td><td><input type='radio' name='g18' value='0.5'> Ya  
<input type='radio' name='g18' value='0' checked> Tidak</td>
```

```
</tr>
```

```
<tr><td>21. </td><td>
```

```
<?php
```

```
include_once "koneksi.php";
```

```
$query = mysqli_query($kon, "select *from tb_gejala where  
No = '19'");
```

```
while ($q= mysqli_fetch_array($query)) {
```

```
echo $q ['Gejala'];
```

```
}?>
```

```

        </td><td><input type='radio' name='g19' value='0.5'> Ya
        <input type='radio' name='g19' value='0' checked> Tidak</td>

```

```

    </tr>

```

```

    <tr><td>22. </td><td>

```

```

<?php

```

```

    include_once "koneksi.php";

```

```

    $query = mysqli_query($kon, "select *from tb_gejala where
    No = '20'");

```

```

    while ($q= mysqli_fetch_array($query)) {

```

```

        echo $q ['Gejala'];

```

```

    }

```

```

?>

```

```

        </td><td><input type='radio' name='g20' value='0.5'> Ya
        <input type='radio' name='g20' value='0' checked> Tidak</td>

```

```

    </tr>

```

```

    <tr><td>23. </td><td>

```

```

<?php

```

```

    include_once "koneksi.php";

```

```

    $query = mysqli_query($kon, "select *from tb_gejala where
    No = '21'");

```

```

while ($q= mysqli_fetch_array($query)) {

echo $q ['Gejala'];

}

?>

</td><td><input type='radio' name='g21' value='0.5'> Ya
<input type='radio' name='g21' value='0' checked> Tidak</td>

</tr>

<tr><td>24. </td><td>

<?php

include_once "koneksi.php";

$query = mysqli_query($kon, "select *from tb_gejala where
No = '22'");

while ($q= mysqli_fetch_array($query)) {

echo $q ['Gejala'];

}

?>

</td><td><input type='radio' name='g22' value='0.5'> Ya
<input type='radio' name='g22' value='0' checked> Tidak</td>

</tr>

<tr><td>25. </td><td>

```



```
<?php
```

```
    include_once "koneksi.php";
```

```
    $query = mysqli_query($kon, "select *from tb_gejala where  
No = '23'");
```

```
    while ($q= mysqli_fetch_array($query)) {
```

```
        echo $q ['Gejala'];
```

```
    }
```

```
?>
```

```
    </td><td><input type='radio' name='g23' value='0.5'> Ya  
<input type='radio' name='g23' value='0' checked> Tidak</td>
```

```
    </tr>
```

```
    <tr><td>26. </td><td>
```

```
<?php
```

```
    include_once "koneksi.php";
```

```
    $query = mysqli_query($kon, "select *from tb_gejala where  
No = '24'");
```

```
    while ($q= mysqli_fetch_array($query)) {
```

```
        echo $q ['Gejala'];
```

```
    }
```

```
?>
```

```

        </td><td><input type='radio' name='g24' value='0.5'> Ya
        <input type='radio' name='g24' value='0' checked> Tidak</td>

    </tr>

    <tr><td>27. </td><td>

```

```

<?php

```

```

    include_once "koneksi.php";

```

```

    $query = mysqli_query($kon, "select *from tb_gejala where
    No = '25'");

```

```

    while ($q= mysqli_fetch_array($query)) {

```

```

        echo $q ['Gejala'];

```

```

    }

```

```

?>

```

```

        </td><td><input type='radio' name='g25' value='0.5'> Ya
        <input type='radio' name='g25' value='0' checked> Tidak</td>

```

```

    </tr>

```

```

    <tr><td>28. </td><td>

```

```

<?php

```

```

    include_once "koneksi.php";

```

```

    $query = mysqli_query($kon, "select *from tb_gejala where
    No = '26'");

```

```

while ($q= mysqli_fetch_array($query)) {

    echo $q ['Gejala'];

}

?>

</td><td><input type='radio' name='g26' value='0.5'> Ya
<input type='radio' name='g26' value='0' checked> Tidak</td>

</tr>

</td></tr>

<tr ><td colspan='3'><hr></td ></tr>

<tr><td colspan='2'><input type='reset' value='Hapus form'>

    <input          type='submit'          value='simpan'
name='simpan_testing'></td></tr>

</table>

</form>

</div>

<div class="col-lg-12">

```

LAMPIRAN 2 : RIWAYAT HIDUP



DIDANG WIKATAMA

DIDANG WIKATAMA, Lahir di TINUKARI KECEMATAN WAWO KABUPATEN KOLAKA UTARA SULAWESI TENGGARA

pada tanggal 22 FEBRUARI 2002, anak pertama dari 3 bersaudara dari pasangan bapak firman sada dan ibu rusmiani djaya

Riwayat pendidikan

1. Tahun 2011, menyelesaikan Sekolah Dasar di SD N 1 Tinukari.
2. Tahun 20017, menyelesaikan Sekolah Menengah Pertama di SMP Negeri 3 Ranteangin.
3. Tahun 2020, menyelesaikan Sekolah Madrasa Aliyah di MA AL-HIKMAH RANTEANGIN.
4. Tahun 2020, mendaftar dan di terima menjadi mahasiswa di Universitas Icshan Gorontalo.

LAMPIRAN 3 : Surat Penelitian



KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI
UNIVERSITAS ICHSAN GORONTALO
LEMBAGA PENELITIAN
 Kampus Unisan Gorontalo Lt.3 - Jln. Achmad Nadjamuddin No. 17 Kota Gorontalo
 Telp: (0435) 8724466, 829975 E-Mail: lembagapenelitian@unisan.ac.id

Nomor : 5089/PIP/LEMLIT-UNISAN/GTO/III/2024
 Lampiran : -
 Hal : Permohonan Izin Penelitian

Kepada Yth,
 Kepala Dinas Perkebunan Kabupaten Kolaka Utara
 di,-
 Tempat

Yang bertanda tangan di bawah ini :

Nama	: Dr. Rahmisyari, ST.,SE.,MM
NIDN	: 0929117202
Jabatan	: Ketua Lembaga Penelitian

Meminta kesediannya untuk memberikan izin pengambilan data dalam rangka penyusunan **Proposal / Skripsi**, kepada :

Nama Mahasiswa	: Didang Wikatama
NIM	: T3120076
Fakultas	: Fakultas Ilmu Komputer
Program Studi	: Teknik Informatika
Lokasi Penelitian	: DINAS PERKEBUNAN KABUPATEN KOLAKA UTARA
Judul Penelitian	: IMPLEMENTASI ALGORITMA MODIFIED NEAREST NEIGHBOR (MKNN) UNTUK DIAGNOSIS PENYAKIT TANAMAN CENGKEH

Atas kebijakan dan kerja samanya diucapkan banyak terima kasih.

Gorontalo, 18 Maret 2024



Dr. Rahmisyari, ST.,SE.,MM
NIDN 0929117202

LAMPIRAN 4 : Surat Rekomendasi Penelitian



Nomor : 500.6.16 / 91 / 2024
 Lampiran :
 Psthal : Persetujuan Izin Penelitian

Kepada
 Yth. Universitas Ichsan Gorontalo
 (Lembaga Penelitian)
 di -
 Tempat

Schubungan dengan Surat dari Kementrian Pendidikan, Kebudayaan, Riset dan Teknologi Universitas Ichsan Gorontalo (Lembaga Penelitian) Nomor : 5089/PIP/LEMLIT-UNISAN/GTO/III/2024 tanggal 18 Maret 2024 tentang Permohonan Izin Penelitian :

Nama : Didang Wikatama
 NIM : T3120076
 Fakultas : Fakultas Ilmu Komputer
 Program Studi : Teknik Informatika
 Lokasi Penelitian : Dinas Perkebunan dan Peternakan
 Judul Penelitian : Implementasi Algoritma Modified Nesrest Neighbor (MKNN) untuk diagnosis Penyakit Tanaman Cengkeh

Dengan ini menerima untuk malkukan penelitian di wilayah kabupaten kolaka utara melalui Dinas Perkebunan dan Peternakan Kabupaten Kolaka Utara dengan lokus wilayah Desa Lanipa-nipa dan Desa Katoi Kecamatan Katoi

Demikian Surat Persetujuan Izin Penelitian ini kami sampaikan, atas perhatian dan kerjasamanya, diucapkan terima kasih.

Lasusua 12 Agustus 2024

Pt. Kepala Dinas Perkebunan dan Peternakan Kab. Kolaka Utara

 Muhammad Shadik, SP
 Pengantar Tk. I, Gol. IV/b
 730812 200312 1 005

Tembusan:
 1. Pj. Bupati Kolaka Utara sebagai Laporan
 2. Arsip,-

LAMPIRAN 5 : Surat Keterangan Bebas Pustaka

**KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI**
UNIVERSITAS ICHSAN GORONTALO
FAKULTAS ILMU KOMPUTER
UPT. PERPUSTAKAAN FAKULTAS
SK. MENDIKNAS RI NO. 84/D/0/2001
Jl. Achmad Nadjamuddin No.17 Telp(0435) 829975 Fax. (0435) 829976 Gorontalo

SURAT KETERANGAN BEBAS PUSTAKA
No : 044/Perpustakaan-Fikom/VI/2024

Perpustakaan Fakultas Ilmu Komputer (FIKOM) Universitas Ichsan Gorontalo dengan ini menerangkan bahwa :

Nama Anggota : Didang Wikatama
No. Induk : T3120076
No. Anggota : M202464

Terhitung mulai hari, tanggal : Senin, 24 Juni 2024, dinyatakan telah bebas pinjam buku dan koleksi perpustakaan lainnya.


Demikian keterangan ini di buat untuk di pergunakan sebagaimana mestinya.



Gorontalo, 24 Juni 2024
Mengetahui,
Kepala Perpustakaan

Apriyanto Alhamad, M.Kom
NIDN : 0924048601

LAMPIRAN 6 : HASIL TURNITIN


Similarity Report ID: old:25211:61780504

PAPER NAME	AUTHOR
SKRIPSI_T3120076_DIDANG_WIKATAMA.pdf	DIDANG WIKATAMA wdidang@gmail.com

WORD COUNT	CHARACTER COUNT
9760 Words	57931 Characters
PAGE COUNT	FILE SIZE
65 Pages	1.7MB
SUBMISSION DATE	REPORT DATE
Jun 20, 2024 9:26 PM GMT+8	Jun 20, 2024 9:27 PM GMT+8

● **24% Overall Similarity**

The combined total of all matches, including overlapping sources, for each database.

- 24% Internet database
- 2% Publications database
- Crossref database
- Crossref Posted Content database
- 5% Submitted Works database

● **Excluded from Similarity Report**

- Bibliographic material
- Quoted material
- Cited material
- Small Matches (Less than 30 words)



Similarity Report ID: oia:25211:61780504

24% Overall Similarity

Top sources found in the following databases:

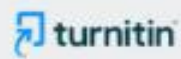
- 24% Internet database
- 2% Publications database
- Crossref database
- Crossref Posted Content database
- 5% Submitted Works database

TOP SOURCES

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	repository.ub.ac.id	8%
	Internet	
2	123dok.com	4%
	Internet	
3	titonkadir.blogspot.com	2%
	Internet	
4	LL Dikti IX Turnitin Consortium on 2019-07-16	1%
	Submitted works	
5	property145.com	1%
	Internet	
6	andi.ddns.net	1%
	Internet	
7	ejurnal.methodist.ac.id	1%
	Internet	
8	scribd.com	1%
	Internet	

Sources overview



Similarity Report ID: oid:25211:61780504

9	ejournal.catursakti.ac.id Internet	<1%
10	pt.scribd.com Internet	<1%
11	ojs.serambimekkah.ac.id Internet	<1%
12	ejurnal.dipaneegara.ac.id Internet	<1%
13	ejurnal.unisan.ac.id Internet	<1%
14	eprints.utdi.ac.id Internet	<1%
15	LL Dikti IX Turnitin Consortium on 2019-07-16 Submitted works	<1%

[Sources overview](#)