

**SISTEM PAKAR DIAGNOSA PENYAKIT TANAMAN
MENTIMUN DENGAN METODE *K-NEAREST
NEIGHBOR* (KNN) PADA DINAS
PERTANIAN KABUPATEN
POHUWATO**

Oleh

NUR INTAN ID SALEH

T3118282

SKRIPSI

**Untuk memenuhi salah satu syarat ujian
guna memperoleh gelar Sarjana**



**PROGRAM SARJANA
TEKNIK INFORMATIKA
UNIVERSITAS ICHSAN GORONTALO
GORONTALO
2022**

**SISTEM PAKAR DIAGNOSA PENYAKIT TANAMAN
MENTIMUN DENGAN METODE *K-NEAREST
NEIGHBOR* (KNN) PADA DINAS
PERTANIAN KABUPATEN
POHUWATO**

Oleh

NUR INTAN ID SALEH

T3118282

SKRIPSI

**Untuk memenuhi salah satu syarat ujian
guna memperoleh gelar Sarjana**



**PROGRAM SARJANA
TEKNIK INFORMATIKA
UNIVERSITAS ICHSAN GORONTALO
GORONTALO
2022**

PERSETUJUAN SKRIPSI

**SISTEM PAKAR DIAGNOSA PENYAKIT TANAMAN
MENTIMUN DENGAN METODE *K-NEAREST
NEIGHBOR* (KNN) PADA DINAS
PERTANIAN KABUPATEN
POHUWATO**

Oleh

NUR INTAN ID SALEH

T3118282

SKRIPSI

Untuk memenuhi salah satu syarat ujian

Guna memperoleh gelar Sarjana

Program Studi Teknik Informatika,

Ini telah disetujui oleh Tim Pembimbing

Gorontalo,

Pembimbing I


IWSDI, M.Kom

Pembimbing II


Sunarto Taliki, M.Kom

PENGESAHAN SKRIPSI

SISTEM PAKAR DIAGNOSA PENYAKIT TANAMAN MENTIMUN DENGAN METODE *K-NEAREST NEIGHBOR* (KNN) PADA DINAS PERTANIAN KABUPATEN POHUWATO

Oleh

NUR INTAN ID SALEH

T3118282

Diperiksa oleh Panitia Ujian Strata Satu (S1)
Universitas Ichsan Gorontalo

1. Ketua Penguji
Anas, M. Kom
2. Anggota
Ruhmi Sulachani, M. Kom
3. Anggota
Yulianty Lasena, M. Kom
4. Anggota
HUSDI, M. Kom
5. Anggota
Sunarto Taliki, M. Kom



Mengetahui :



Jorry Karim, M.Kom
NIDN: 0918077302

Ketua Program Studi

Sudirman S. Panna, M.Kom
NIDN: 09240

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 di perguruan 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 dipublikasikan orang lain, kecuali secara tertulis dicantumkan sebagai acuan/sitasi dalam naskah dan dicantumkan pula dalam daftar pustaka.
4. Pernyataan ini saya buat dengan sesungguhnya dan apabila dikemudian 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,

Yang Membuat Pernyataan,



Nur Intan Id Saleh

ABSTRACT

The lack of plant experts and knowledge of farmers in dealing with diseases in cucumber plants causes losses for farmers, especially cucumber farmers. Apart from that, this is also an obstacle for cucumber farmers to diagnose and carry out proper treatment of cucumber diseases that attack the plants they plant. Farmers do not understand the types of diseases in cucumber plants, so the process of handling cucumber plants is slow. An expert system is generally defined as a system designed to model problem-solving abilities like a human expert. Identification of the problem in this study is damage to cucumber plants due to the difficulty of identifying diseases in cucumber plants. The method used in diagnosing diseases in cucumber plants is the K-Nearest Neighbor method. Percentage off accuracy of the K-Nearest Neighbor method is able to classify and serve cucumber plant disease. The cucumber plant disease program was 70% with value of $k=17$. It is hoped that in making and expert system using the K-Nearest Neighbor Method in this study can help farmers to make an initial diagnosis of a cucumber plant disease and provide a solution to overcome the cucumber plant disease. The results of making this Expert System are able to prove the truth of a cucumber disease and provide a solution in dealing with the cucumber plant disease.

Keywords: *Cucumber Plant Disease, Expert System, K-Nearest Neighbor.*

ABSTRAK

Minimnya pakar tumbuhan dan pengetahuan petani dalam menangani penyakit pada tanaman mentimun menyebabkan kerugian bagi petani khususnya petani mentimun. Selain daripada itu hal ini juga menjadi hambatan bagi para petani mentimun untuk mendiagnosa dan melakukan penanganan yang tepat terhadap penyakit mentimun yang menyerang tanaman yang mereka tanam. Petani kurang paham terhadap jenis penyakit pada tanaman mentimun, sehingga lambatnya proses penanganan pada tanaman mentimun. Sistem pakar umumnya didefinisikan sebagai sebuah sistem yang dirancang untuk memodelkan kemampuan menyelesaikan masalah seperti layaknya seorang pakar (*human expert*). Identifikasi masalah dalam penelitian ini yaitu kerusakan pada tanaman mentimun akibat sulitnya mengidentifikasi penyakit pada tanaman mentimun. Metode yang digunakan dalam mendiagnosa penyakit pada tanaman mentimun ialah metode *K-Nearest Neighbor*. Persentase akurasi metode *K-Nearest Neighbor* mampu mengklasifikasikan dan melayani penyakit tanaman mentimun. Program penyakit tanaman mentimun adalah 70% dengan nilai $K=17$. Diharapkan dalam pembuatan sistem pakar dengan menggunakan metode *K-Nearest Neighbor* dalam penelitian ini dapat membantu petani dalam melakukan diagnosa awal terhadap suatu penyakit tanaman mentimun dan memberikan solusi untuk mengatasi penyakit tanaman mentimun. Hasil dari pembuatan Sistem Pakar ini mampu membuktikan kebenaran suatu penyakit mentimun dan memberi solusi dalam menangani penyakit tanaman mentimun

Kata kunci: Penyakit Tanaman Mentimun, Sistem Pakar, *K-Nearest Neighbor*.

KATA PENGANTAR

Segala Puji bagi Allah SWT karena dengan Taufiq dan Hidayah-Nya lah sehingga penulis dapat menyelesaikan usulan Skripsi ini dengan judul: **“Sistem Pakar Diagnosa Penyakit Pada Tanaman Mentimun Menggunakan Metode *K-Nearest Neighbor* Pada Dinas Pertanian Kabupaten Pohuwato”**, untuk memenuhi salah satu syarat Ujian Akhir guna memperoleh gelar Sarjana Komputer pada Program Studi 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 keikhlasan dan kerendahan hati, penulis mengucapkan banyak terima kasih dan penghargaan yang setinggi-tingginya kepada:

1. Ibu Dr. Juriko Abdussamad, M.Si, selaku Ketua Yayasan Pengembang Ilmu Pengetahuan dan Teknologi (YPIPT) Universitas Ichsan Gorontalo;
2. Bapak Dr. Abdul Gaffar Latjoke, M.Si, selaku Rektor Universitas Ichsan Gorontalo;
3. Bapak Jorry Karim, S.Kom., M.Kom, selaku Dekan Fakultas Ilmu Komputer Universitas Ichsan Gorontalo;
4. Bapak Sudirman Melangi, M.Kom, selaku Pembantu Dekan I Bidang Akademik Fakultas Ilmu Komputer Universitas Ichsan Gorontalo;
5. Ibu Irma Surya Kumala Idris, S.Kom., M.Kom, selaku Pembantu Dekan II Bidang Administrasi Umum dan Keuangan Fakultas Ilmu Komputer Universitas Ichsan Gorontalo;
6. Bapak Irvan Abraham Salihi, M.Kom, selaku Pembantu Dekan III Bidang Kemahasiswaan Fakultas Ilmu Komputer Universitas Ichsan Gorontalo;
7. Bapak Sudirman S. Panna, M.Kom, selaku Ketua Jurusan Teknik Informatika Fakultas Ilmu Komputer Universitas Ichsan Gorontalo;
8. Ibu Annahl Riadi, M.Kom, selaku Pembimbing I;

9. Bapak Abdul Yunus Labolo, M.Kom, selaku Pembimbing II;
10. Bapak dan Ibu Dosen Universitas Ichsan Gorontalo yang telah mendidik dan mengajarkan berbagai disiplin ilmu kepada penulis;
11. Kedua Orang Tua saya yang tercinta atas segala kasih sayang, jerih payah dan doa restunya dalam membesarkan dan mendidik penulis;
12. Suami saya yang selalu memberikan dukungan dan selalu mendoakan yang terbaik;
13. Rekan-rekan seperjuangan yang telah banyak memberikan bantuan dan dukungan moril yang sangat besar kepada penulis;
14. Kepada semua pihak yang ikut membantu dalam penyelesaian proposal ini yang tak sempat penulis sebutkan satu-persatu.

Semoga Allah, SWT Melimpahkan balasan atas jasa-jasa mereka kepada kami. Penulis menyadari sepenuhnya bahwa apa yang telah dicapai ini masih jauh dari kesempurnaan dan masih banyak terdapat kekurangan. Oleh karena itu, penulis sangat mengharapkan adanya kritik dan saran yang sangat konstruktif. Akhirnya penulis berharap semoga hasil yang telah dicapai ini dapat bermanfaat bagi kita semua, Aamiin.

Gorontalo, Maret 2022

Penulis

DAFTAR ISI

HALAMAN SAMPUL.....	i
HALAMAN JUDUL.....	ii
PERSETUJUAN SKRIPSI.....	ii
PENGESAHAN SKRIPSI.....	iii
PERNYATAAN SKRIPSI.....	iv
<i>ABSTRACT</i>	v
ABSTRAK.....	vi
KATA PENGANTAR.....	vii
DAFTAR ISI.....	ix
DAFTAR GAMBAR.....	xii
DAFTAR TABEL.....	xiv
BAB IPENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Identifikasi Masalah.....	2
1.3 Rumusan Masalah.....	3
1.4 Tujuan Penelitian.....	3
1.5 Manfaat Penelitian.....	3
BAB IILANDASAN TEORI.....	4
2.1 Tinjauan Studi.....	4
2.2 Tinjauan Pustaka.....	5
2.2.1 Sistem Pakar.....	5
2.2.2 Metode <i>K-Nearest Neighbor</i>	8
2.2.3 Mentimun.....	12
2.2.4 Pengembangan Sistem.....	16
2.2.5 Teknik Pengujian Sistem.....	21
2.2.6 Perangkat Lunak Pendukung.....	23
2.3 Kerangka Pemikiran.....	26
BAB IIIMETODE PENELITIAN.....	27
3.1 Jenis, Metode, Subjek, Objek, Waktu, dan Lokasi Penelitian.....	27

3.2	Pengumpulan Data.....	27
3.3	Pengembangan Sistem.....	28
3.3.1	Analisis Sistem.....	28
3.3.2	Desain Sistem.....	29
3.3.3	Pengujian Sistem.....	30
BAB IVHASIL PENELITIAN.....		33
4.1	Hasil Pengumpulan Data.....	33
4.1.1	Sejarah Dinas Pertanian Kabupaten Pohuwato.....	33
4.1.2	Struktur Organisasi Dinas Pertanian Kabupaten Pohuwato.....	34
4.1.3	Visi Dan Misi Dinas Pertanian Kabupaten Pohuwato.....	34
4.1.4	Tupoksi.....	35
4.1.5	Pengumpulan Data Mentimun.....	37
4.2	Hasil Pemodelan Data.....	43
4.3	Hasil Pengembangan Sistem.....	45
4.3.1	Analisa Sistem.....	45
4.3.2	Diagram Konteks.....	48
4.3.3	Diagram Berjenjang.....	49
4.3.4	Diagram Arus Data (DAD).....	50
4.3.5	Kamus Data.....	53
4.3.6	Arsitektur Sistem/Kebutuhan <i>Hardware</i> dan <i>Software</i>	57
4.3.7	Desain Secara Umum.....	58
4.3.8	Desain Secara Terinci.....	59
4.3.9	Pengujian Sistem.....	65
BAB VPEMBAHASAN PENELITIAN.....		71
5.1	Pembahasan Model.....	71
5.2	Pembahasan Sistem.....	75
5.2.1	Halaman Menu Utama.....	75
5.2.2	Halaman Menu Gejala.....	76
5.2.3	Halaman Menu Konsultasi.....	76
5.2.4	Halaman Menu Data Penyakit.....	77
5.2.5	Halaman Menu Data Solusi.....	78

5.2.6 Halaman Menu Data Kasus.....	79
5.2.7 Halaman Menu Informasi Daftar Penyakit.....	80
BAB VIKESIMPULAN DAN SARAN.....	81
6.1 Kesimpulan.....	81
6.2 Saran.....	81
DAFTAR PUSTAKA.....	82
LAMPIRAN.....	84

DAFTAR GAMBAR

Gambar 2. 1 : Konsep Dasar Sistem Pakar.....	6
Gambar 2. 2 : Penyakit Busuk Daun.....	13

Gambar 2. 3 : Penyakit Embun Tepung.....	14
Gambar 2. 4 : Penyakit Antraknosa.....	14
Gambar 2. 5 : Penyakit Bercak Daun.....	15
Gambar 2. 6 : Penyakit Virus Mosaik.....	15
Gambar 2. 7 : Penyakit Scab.....	16
Gambar 2. 8 : Model Waterfall.....	17
Gambar 2. 9 : Bagan Alir.....	22
Gambar 2. 10 : Grafik Alir.....	22
Gambar 2. 11 : PHP.....	23
Gambar 2. 12 : MySQL.....	24
Gambar 2. 13 : XAMPP.....	24
Gambar 2. 14 : Adobe Dreamweaver.....	25
Gambar 2. 15 : Adobe Photoshop CS3.....	25
Gambar 2. 16 : Microsoft Visio.....	25
Gambar 2. 17 : Kerangka Pikir.....	26

Gambar 4. 1 : Struktur Organisasi Dinas Pertanian 34

Gambar 4. 2 : Sistem Berjalan.....	47
Gambar 4. 3 : Sistem Yang Diusulkan.....	48
Gambar 4. 4 : Diagram Konteks.....	49
Gambar 4. 5 : Diagram Berjenjang.....	49
Gambar 4. 6 : Diagram Arus Data Level 0.....	50
Gambar 4. 7 : Diagram Arus Data Level 1 Proses 1.....	51
Gambar 4. 8 : Diagram Arus Data Level 1 Proses 2.....	52
Gambar 4. 9 : Diagram Arus Data Level 1 Proses 3.....	53
Gambar 4. 10 : Desain Daftar Hasil Konsultasi.....	60
Gambar 4. 11 : Desain Input Gejala.....	61
Gambar 4. 12 : Desain Input Penyakit.....	61
Gambar 4. 13 : Desain Input Solusi.....	62
Gambar 4. 14 : Desain Input Kasus.....	62
Gambar 4. 15 : Desain Input Konsultasi.....	63

Gambar 4. 16 : Desain Menu Utama.....	65
Gambar 4. 17 : Flowchart Gejala.....	66
Gambar 4. 18 : Flowgraph Gejala.....	67

DAFTAR TABEL

Tabel 2. 1 : Tinjauan Studi.....	4
Tabel 2. 2 : Sampel data kondisi fisik.....	10
Tabel 2. 3 : Data setelah ditambahkan jarak terhadap data pengujian.....	11

Tabel 2. 4 : Data setelah diurutkan berdasarkan jarak terkecil.....	12
Tabel 2. 5 : Simbol-simbol Diagram Alir (Flowchart).....	19
Tabel 4. 1 : Daftar Penyakit Tanaman Mentimun	38
Tabel 4. 2 : Daftar Gejala Penyakit Tanaman Mentimun.....	38
Tabel 4. 3 : Penyakit dan Gejala Tanaman Mentimun.....	39
Tabel 4. 4 : Daftar Solusi Tanaman Mentimun.....	40
Tabel 4. 5 : Penyakit dan Solusi Tanaman Mentimun.....	41
Tabel 4. 6 : Tabel Bobot Parameter.....	42
Tabel 4. 7 : Daftar Bobot Dari Setiap Gejala Penyakit.....	43
Tabel 4. 8 : Matriks Tanaman Penyakit Mentimun.....	44
Tabel 4. 9 : Gejala Yang Dipilih.....	44
Tabel 4. 10 : Konsultasi.....	45
Tabel 4. 11 : Kamus Data Pengguna.....	53
Tabel 4. 12 : Kamus Data Gejala.....	54
Tabel 4. 13 : Kamus Data Kasus.....	54
Tabel 4. 14 : Kamus Data Kasus Gejala.....	54
Tabel 4. 15 : Kamus Data Konsultasi.....	55
Tabel 4. 16 : Kamus Data Konsultasi Gejala.....	55
Tabel 4. 17 : Kamus Data Konsultasi Hasil.....	56
Tabel 4. 18 : Kamus Data Penyakit.....	56
Tabel 4. 19 : Kamus Data Penyakit Solusi.....	57
Tabel 4. 20 : Kamus Data Solusi.....	57
Tabel 4. 21 : Desain Output Secara Umum.....	58
Tabel 4. 22 : Desain Input Secara Umum.....	59
Tabel 4. 23 : Penyakit.....	63
Tabel 4. 24 : Gejala.....	63
Tabel 4. 25 : Kasus Gejala.....	64
Tabel 4. 26 : Konsultasi.....	64
Tabel 4. 27 : Konsultasi Hasil.....	64
Tabel 4. 28 : Pengujian Black Box.....	69

Tabel 5. 1 : Nilai Kondisi71

Tabel 5. 2 : Nilai P001 Dengan P01.....	71
Tabel 5. 3 : Nilai P002 Dengan P02.....	72
Tabel 5. 4 : Nilai P003 Dengan P03.....	72
Tabel 5. 5 : Nilai P004 Dengan P04.....	72
Tabel 5. 6 : Nilai P005 Dengan P05.....	73
Tabel 5. 7 : Nilai P006 Dengan P06.....	73

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan pertanian di Indonesia masih menjadi salah satu hal penting sebagai roda penggerak ekonomi negara. Hal ini dikarenakan pertanian dari segi produksi menjadi sektor kedua paling berpengaruh setelah industri pengolahan. Sedangkan bila dibandingkan sektor lainnya pertanian masih berada diposisi teratas selain sektor perdagangan dan sektor konstruksi. Dengan demikian sektor pertanian mampu mengangkat citra Indonesia di mata dunia, terutama sebagai negara agraris yang cukup produktif.

Mentimun merupakan salah satu jenis sayuran buah yang memiliki banyak kegunaan dalam kehidupan masyarakat sehari-hari. Timun (*Cucumis Sativus*) adalah tanaman sayuran yang banyak manfaat dan kegunaannya antara lain sebagai bahan makanan (diolah sebagai Acar Timun, Es Timun), bahan obat-obatan dan bahan kosmetika. Sebagai bahan makanan, timun mengandung zat gizi antara lain kalori, protein, lemak, karbohidrat, kalsium, fosfor, besi, vitamin A, vitamin B1, vitamin B2, vitamin C, serat, niacin dan air[1].

Peningkatan produksimentimun dapat dipacu dengan usaha intensifikasi, ekstensifikasi, dan diversifikasi melalui penerapan paket teknologi budidaya secara intensif. Namun terdapat kendala pada budidaya mentimun ialah adanya penyakit pada tanaman yang menyebabkan gagal panen[2]. Berdasarkan dari observasi, para petani di Pohuwato sulit mendiagnosa penyakit yang menyerang tanamannya karena sistem yang berjalan masih manual dimana petani yang ingin melakukan konsultasi harus konsultasi ke pakar langsung, cara tersebut kurang efisien karena membutuhkan waktu yang lama. Hal inilah yang mendorong pembuatan aplikasi sistem pakar pendiagnosa penyakit pada tanaman mentimun yang

dapat memudahkan petani dalam mendiagnosa penyakit yang menyerang tanamannya.

Sistem pakar adalah aplikasi berbasis komputer yang digunakan untuk menyelesaikan masalah sebagaimana yang dipikirkan oleh pakar. Pakar yang dimaksud disini adalah orang yang mempunyai keahlian khusus yang dapat menyelesaikan masalah yang tidak dapat diselesaikan oleh orang awam[3].

Penelitian yang dilakukan oleh Desy Pratiwi, 2020, “Sistem Pakar Pendiagnosa Penyakit Tanaman Mentimun Berbasis *Android* Menggunakan Metode *Forward Chaining*”. Hasil yang di dapat pada penelitian ini yaitu aplikasi sistem pakar ini dapat memberikan informasi mengenai penyakit tanaman cabai dan bagaimana cara penanganannya berdasarkan gejala yang ada[2].

Algoritma *K-Nearest Neighbor* (K-NN) adalah sebuah metode untuk melakukan klasifikasi terhadap objek berdasarkan data pembelajaran yang jaraknya paling dekat dengan objek tersebut. Alasan mengapa menggunakan metode ini, karenametode ini dapat mengklasifikasikan data secara akurat dengan memilih terlebih dahulu nilai k tetangga terdekat dengan tepat[4].

Berdasarkan penjelasan latar belakang di atas maka penulis mengangkat judul penelitian “**Sistem Pakar Diagnosa Penyakit Tanaman Mentimun Dengan Metode *K-Nearest Neighbor* Pada Dinas Pertanian Kabupaten Pohuwato**”. Pada penelitian ini akan buat suatu aplikasi sistem pakar yang memberikan informasi mengenai penyakit tanaman dan dapat mendiagnosa gejala–gejala penyakit tanaman Mentimun, sekaligus memberikan solusi penanggulangannya, yang nantinya dapat digunakan untuk mengurangi atau memperkecil resiko kerusakan tanaman mentimun.

1.2 Identifikasi Masalah

Berdasarkan latar belakang, maka diuraikan identifikasi masalah dalam penelitian ini adalah sebagai berikut :

1. Petani sulit mendiagnosa penyakit yang menyerang tanamannya karena sistem yang berjalan masih manual.
2. Pengembangan Mentimun mengalami penurunan jumlah produksi.

1.3 Rumusan Masalah

Permasalahan yang akan dipecahkan dalam penelitian ini sesuai dengan latar belakang diatas dijelaskan dan diuraikan sebagai berikut :

1. Bagaimana hasil uji coba Algoritma *K-Nearest Neighbor* untuk mendiagnosa penyakit pada tanaman Mentimun?
2. Bagaimana kinerja dan efektifitas Sistem Pakar Diagnosa Penyakit Tanaman Mentimun Menggunakan Metode *K-Nearest Neighbor* yang dapat diimplementasikan?

1.4 Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah :

1. Menguji coba Metode *K-Nearest Neighbor* untuk memperoleh akurasi yang terbaik pada Sistem Pakar Diagnosa Penyakit Mentimun.
2. Memperoleh Sistem Pakar Diagnosa Penyakit Tanaman Mentimun berbasis *web* yang handal dan efektif sehingga dapat diimplementasikan.

1.5 Manfaat Penelitian

Adapun manfaat dari penelitian sistem pakar ini, yaitu:

1. Manfaat Teoritis: Membantu masyarakat khususnya para pembudidaya mentimun dalam menganalisa gejala-gejala apa yang mungkin terjadi pada Mentimun merereka, sehingga mereka dapat tahu dan memutuskan, apakah Penyakit Pada Mentimun mereka harus segera ditindaki dengan serius atau menjalani perawatan.
2. Manfaat Praktis: Memberikan kemudahan kepada pihak Dinas Pertanian Kabupaten Pohuwato dalam menangani permasalahan-permasalahan Penyakit Pada Mentimun sehingga populasi di Kabupaten Pohuwato ini terus bertambah. Penelitian ini juga diharapkan dapat memberi masukan terhadap peneliti yang akan mengadakan penelitian selanjutnya untuk dapat menggunakannya dalam mengembangkan sistem yang bersangkutan.

BAB II

LANDASAN TEORI

2.1 Tinjauan Studi

Tinjauan studi terkait dengan penelitian ini mengenai tanaman Mentimun atau yang menggunakan Metode *K-Nearest Neighbor* sebagai berikut:

Tabel 2. 1 : Tinjauan Studi [5] [6]

No	Peneliti	Judul	Tahun	Metode	Hasil
1.	Ishak, Muhamma d Dahria, Rudi Gunawan	Sistem Pakar Diagnosa Penyakit Tanaman Mentimun menggunakan Metode <i>Dempster Shafer</i>	2019	<i>DEMPSTER SHAFTER</i>	Dapat memberikan solusi penanggulangannya, yang nantinya dapat digunakan untuk mengurangi atau memperkecil resiko pada penyakit Mentimun.
2.	Firman Dwi Wahyudi, Paulus Harsadi, Dwi Remawati	IMPLEMENTASI METODE K-NN UNTUK DETEKSI KERUSAKAN MESIN BUBUT	2018	<i>K-NEAREST NEIGHBOR</i>	Dengan aplikasi sistem pakar ini, deteksi kerusakan mesin bubut dapat ditangani secara tepat dan cepat dengan metode <i>K-Nearest Neighbour</i> .

2.2 Tinjauan Pustaka

2.2.1 Sistem Pakar

Sistem pakar adalah aplikasi berbasis komputer yang digunakan untuk menyelesaikan masalah sebagaimana yang dipikirkan oleh pakar. Pakar yang dimaksud disini adalah orang yang mempunyai keahlian khusus yang dapat menyelesaikan masalah yang tidak dapat diselesaikan oleh orang awam[3].

Sebagai contoh, dokter adalah seorang pakar yang mampu mendiagnosis penyakit yang diderita pasien serta dapat memberikan penata laksanaan terhadap penyakit tersebut. Sistem pakar adalah program *artificial intellegence* yang menggabungkan pangkalan pengetahuan (*Knowledge Base*) dengan sistem inferensi. Ini merupakan bagian *software* spesialisasi tingkat tinggi yang berusaha menduplikasi fungsi seorang pakar dalam satu bidang keahlian. Program ini bertindak sebagai seorang konsultan yang cerdas atau penasihat dalam suatu lingkungan keahlian tertentu, sebagai hasil himpunan pengetahuan yang telah dikumpulkan dari beberapa orang pakar[7].

2.2.1.1 Konsep Dasar Sistem Pakar

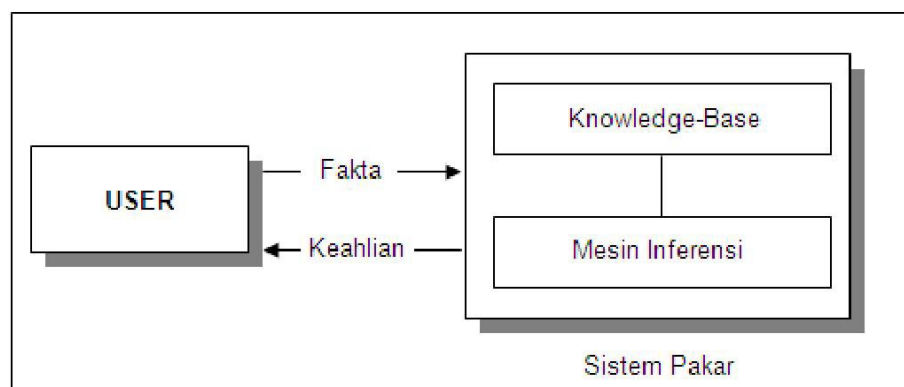
Sistem pakar terdiri dari beberapa konsep yang harus dimilikinya. Konsep dasar dari suatu sistem pakar[8]:

1. Keahlian merupakan pengetahuan khusus yang diperoleh dari latihan, belajar dan pengetahuan. Pengetahuan dapat berupa fakta, teori, aturan, strategi *global* untuk memecahkan masalah.
2. Ahli (*Expert*) Melibatkan kegiatan mengenali suatu masalah, menyelesaikan masalah secara cepat dan tepat,

menerangkan pemecahannya belajar dari pengalaman, *merestrukturisasi* pengetahuan, memecahkan aturan serta menentukan *relevansi*.

3. Mentransfer Keahlian (*Transferring Expertise*) Proses pentransferan keahlian seorang pakar kedalam komputer agar dapat digunakan oleh orang lain yang bukan seorang pakar. Pengetahuan tersebut ditempatkan ke dalam sebuah komponen yang dinamakan basis pengetahuan.
4. Menyimpulkan Aturan (*Infeferencing Rule*) Merupakan kemampuan komputer yang telah diprogram. Penyimpulan dilakukan mesin *Inferensi* yang telah memiliki prosedur tentang penyelesaian suatu permasalahan.
5. Aturan (*Rule*) dibutuhkan karena suatu sistem pakar bersifat *rule-basedsystems*, yang artinya pengetahuan disimpan dalam bentuk aturan.
6. Kemampuan menjelaskan (*Explanation Capability*) merupakan karakteristik dari sistem pakar yang memiliki kemampuan menjelaskan atau memberi saran mengapa tindakan tertentu dianjurkan atau tidak dianjurkan.

Gambar 2.1 menggambarkan konsep dasar suatu sistem pakar Knowledge-Base. Pengguna menyampaikan fakta atau informasi untuk sistem pakar dan kemudian menerima saran dari pakar atau jawaban ahlinya. Bagian dalam sistem pakar terdiri dari dua komponen utama, yaitu Knowledg-Base yang berisi knowledge dan mesin inferensi yang menggambarkan kesimpulan. Kesimpulan tersebut merupakan respons dari sistem pakar atas permintaan pengguna.



Gambar 2. 1 : Konsep Dasar Sistem Pakar

2.2.1.2 Ciri-Ciri Sistem Pakar

Ciri-ciri sistem pakar adalah sebagai berikut[8]:

1. Terbatas pada *domain* keahlian tertentu.
2. Dapat memberikan penalaran untuk data yang tidak pasti.
3. Dapat menjelaskan rangkaian alasan yang diberikan dengan cara yang dapat dipahami.
4. Berdasarkan pada kaidah atau *rule* tertentu.
5. Dirancang untuk dapat dikembangkan secara bertahap.
6. Pengetahuan dan mekanisme *inferensi* jelas terpisah.
7. Keluarannya bersifat anjuran.
8. Sistem dapat menggunakan kaidah secara searah dan sesuai yang dituntun oleh dialog dengan pemakai.

2.2.1.3 Manfaat dan Kekurangan Sistem Pakar

Sistem pakar sangat populer karena memiliki manfaat dan kemampuan yang diberikannya, sebagai berikut[8]:

1. Meningkatkan produktivitas, karena sistem pakar dapat bekerja lebih cepat daripada manusia.
2. Membantu seorang yang awam bekerja seperti seorang pakar.
3. Meningkatkan kualitas, dengan memberikan nasehat yang konsisten dan mengurangi kesalahan.
4. Mampu menangkap pengetahuan dan kepakaran seseorang.
5. Memudahkan akses pengetahuan seorang pakar.
6. Bisa digunakan sebagai media pelengkap dalam pelatihan. Pengguna pemula yang bekerja dengan sistem pakar akan menjadi lebih berpengalaman karena adanya fasilitas penjelas yang berfungsi sebagai guru.

7. Meningkatkan kemampuan dalam menyelesaikan suatu masalah karena sistem pakar mengambil sumber pengetahuan dari banyak pakar.

Selain manfaat, ada juga beberapa kekurangan yang ada pada sistem pakar, diantaranya:

1. Biaya yang sangat mahal untuk membuat dan memeliharanya.
2. Sulit dikembangkan karena keterbatasan keahlian dan ketersediaan pakar.
3. Sistem pakar tidak 100% bernilai benar.

2.2.2 Metode *K-Nearest Neighbor*

2.2.2.1 Definisi *K-Nearest Neighbor*

Pada umumnya *K-Nearest Neighbor* (K-NN) digunakan pada kasus yang memiliki pola kemiripan pada kasus sebelumnya, di dunia kesehatan penggunaan *K-Nearest Neighbor* (K-NN) juga digunakan dalam menentukan diagnosa terhadap pasien yang analisa gejala-gejalanya diperoleh pada kasus-kasus diagnose sebelumnya yang berguna untuk memastikan keakuratan terhadap pendiagnosaan yang telah dilakukan. Berikut fungsi persamaan dari *K-Nearest Neighbor*:

$$Similarity(T,S) = \sum_{i=1}^n f \frac{(T_i, S_i) * W_i}{W_1} \dots\dots\dots(1)$$

Algoritma *K-Nearest Neighbor* (K-NN) adalah sebuah metode untuk melakukan klasifikasi terhadap objek berdasarkan data pembelajaran yang jaraknya paling dekat dengan objek tersebut. *K-Nearest Neighbor* berdasarkan konsep '*learning by analogy*'. Data *learning* di deskripsikan dengan atribut

numerik n-dimensi. Tiap data *learning* merepresentasikan sebuah titik, yang di tandai dengan c , dalam ruang n-dimensi. Jika sebuah data *query* yang labelnya tidak di ketahui diinputkan, maka *K-Nearest Neighbor* akan mencari k buah data *learning* yang jaraknya paling dekat dengan data *query* dalam ruang n-dimensi. Jarak antara data *query* dengan data *learning* dihitung dengan cara mengukur jarak antara titik yang merepresentasikan data *query* dengan semua titik yang merepresentasikan data *learning* dengan rumus *Euclidean Distance* berikut[4].

$$\text{Euclidean Distance} = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \dots + (q_n - p_n)^2} \dots\dots\dots(2)$$

Pada fase *training*, algoritma ini hanya melakukan penyimpanan vektor-vektor fitur dan klasifikasi data *training sample*. Pada fase klasifikasi, fitur – fitur yang sama dihitung untuk testing data (klasifikasinya belum di ketahui). Jarak dari vektor yang baru ini terhadap seluruh vektor *training sample* di hitung, dan sejumlah k buah yang paling dekat diambil. Titik yang baru klasifikasinya di prediksi termasuk pada klasifikasi terbanyak dari titik–titik tersebut[4].

Nilai k yang terbaik untuk algoritma ini tergantung pada data; secara umumnya, nilai k yang tinggi akan mengurangi efek *noise* pada klasifikasi, tetapi membuat batasan antara setiap klasifikasi menjadi lebih kabur. Nilai k yang bagus dapat di pilih dengan optimasi parameter, misalnya dengan menggunakan *cross-validation*. Kasus khusus di mana klasifikasi di prediksi berdasarkan data pembelajaran yang paling dekat (dengan kata lain, $k = 1$) disebut algoritma *nearest neighbor*[4].

K buah data *learning* terdekat akan melakukan *voting* untuk menentukan label mayoritas. Label data *query* akan ditentukan berdasarkan label mayoritas dan jika ada lebih dari satu label mayoritas maka label data *query* dapat di pilih secara acak di antara label-label mayoritas yang ada[4].

2.2.2.2 Cara Kerja *K-Nearest Neighbor*

K-Nearest Neighbor (K-NN) memiliki prinsip kerja dengan mencari jarak terdekat antara K tetangga (*neighbour*) dengan data yang akan di evaluasi terdekat

dalam data latih. Algoritma K-NN memiliki cara kerja yang sederhana berdasarkan nilai jarak terdekat sehingga dapat ditentukan hasil K-NN-nya[4].

Langkah-langkah penyelesaian masalah menggunakan metode K-NN adalah sebagai berikut[9]:

1. Tentukan parameter K (K = Tetangga Terdekat)
2. Hitung jarak antara data uji dengan seluruh data latih.
3. Jarak tersebut diurutkan secara descending
4. Tentukan jarak terdekat sampai urutan K
5. Cari kelas yang sesuai dengan jarak yang telah diurutkan
6. Cari jumlah kelas terbanyak lalu klasifikasikan data tersebut.

2.2.2.3 Contoh Kasus

Jika ada 20 data sampel yang digunakan sebagai basis pengetahuan untuk mengetahui status gizi berdasarkan tinggi badan, berat badan, nilai persen lemak, tekanan darah sistolik, tekanan darah diastolic, detak jantung, lingkaran lengan atas, lingkaran lengan bawah, lingkaran perut dan lingkaran panggul seperti pada Tabel 2.2.

Tabel 2. 2 : Sampel data kondisi fisik

Tinggi Badan	Berat Badan	% Lemak	Sistolik	Diastolik	Detak Jantung	Lengan Atas	Lengan Bawah	Lingkar Perut	Lingkar Panggul	Status Gizi
163	59	25.4	102	68	69	23	14	74	94	Normal
170	125	42.9	113	73	79	40	19	112	135	Obesitas
172	75	31	114	78	86	26	16	79	100	Normal
166	58	19	128	69	69	26	16	72	92	Normal
167	50	16.5	111	80	89	21	13	71	88	Kurus
168	50	10.4	100	72	97	23	14	62	87	Kurus
173	56	18.4	114	73	62	24	15	66	93	Normal
168	73	22.7	107	73	81	30	18	77	96	Normal
177	60	17.4	107	71	104	24	15	71	90	Normal
168	52	13.9	122	82	101	22	15	68	84	Kurus
159	58	23.4	118	75	64	26	15	70	85	Normal
167	75	30.7	126	81	65	30	16	91	103	Obesitas
170	72	26.1	113	72	83	28	16	85	98	Normal
172	68	22.8	110	65	68	27	15	79	94	Normal
165	73	29.1	105	67	75	28	18	83	100	Obesitas
169.5	55	15.2	112	77	70	22	14	75	92	Kurus
160	54	15.7	138	104	78	27	15	73	86	Normal
173	56	17.9	120	76	97	25	14	72	88	Normal
162	54	18	108	70	76	24	15	71	88	Normal
169	79	22	123	76	70	29	17	84	101	Obesitas

Ingin diketahui status gizi seseorang dengan tinggi badan 175 cm, berat badan 67 kg, % lemak 23%, tekanan darah sistolik 97, tekanan darah diastolik 67, detak jantung 79/menit, lingkaran lengan atas 25 cm, lingkaran lengan bawah 15 cm, lingkaran perut 75 cm, lingkaran panggul 95 cm. Berdasarkan algoritma KNN, dapat dilakukan perhitungan sebagai berikut:

1. Tentukan nilai K, misal $K = 5$.
2. Hitung jarak setiap sampel data dengan data mahasiswa yang akan diuji berdasarkan persamaan (1).

Tabel 2.3 : Data setelah ditambahkan jarak terhadap data pengujian

Tinggi Badan	Berat Badan	% Lemak	Sistolik	Diastolik	Detak Jantung	Lengan Atas	Lengan Bawah	Lingkar Perut	Lingkar Panggul	Status Gizi	Jarak
163	59	25.4	102	68	69	23	14	74	94	Normal	18.6215
170	125	42.9	113	73	79	40	19	112	135	Obesitas	85.9477
172	75	31	114	78	86	26	16	79	100	Normal	26.2679
166	58	19	128	69	69	26	16	72	92	Normal	35.5387
167	50	16.5	111	80	89	21	13	71	88	Kurus	30.7449
168	50	10.4	100	72	97	23	14	62	87	Kurus	29.4747
173	56	18.4	114	73	62	24	15	66	93	Normal	25.6351
168	73	22.7	107	73	81	30	18	77	96	Normal	18.976
177	60	17.4	107	71	104	24	15	71	90	Normal	18.503
168	52	13.9	122	82	101	22	15	68	84	Kurus	38.5462
159	58	23.4	118	75	64	26	15	70	85	Normal	32.6827
167	75	30.7	126	81	65	30	16	91	103	Obesitas	40.8692
170	72	26.1	113	72	83	28	16	85	98	Normal	23.6561
172	68	22.8	110	65	68	27	15	79	94	Normal	17.4367
165	73	29.1	105	67	75	28	18	83	100	Obesitas	21.0763
169.5	55	15.2	112	77	70	22	14	75	92	Kurus	26.0594
160	54	15.7	138	104	78	27	15	73	86	Normal	60.7148
173	56	17.9	120	76	97	25	14	72	88	Normal	30.3317
162	54	18	108	70	76	24	15	71	88	Normal	25.671
169	79	22	123	76	70	29	17	84	101	Obesitas	34.2783

$$d_1 = \sqrt{(163-175)^2 + (59-67)^2 + (25.4-23)^2 + (102-97)^2 + \dots}$$

$$\sqrt{(68-67)^2+(69-79)^2+(23-25)^2+(14-15)^2}+\sqrt{(74-75)^2+(94-95)^2}=18.6215$$

$$d_2 = \sqrt{(170-175)^2+(125-67)^2+(42,9-23)^2+(113-97)^2}$$

$$\sqrt{(73-67)^2+(79-79)^2+(40-25)^2+(19-15)^2}+\sqrt{(112-75)^2+(135-95)^2}=85.957$$

dan seterusnya, sehingga diperoleh nilai jarak seperti yang terdapat pada Tabel 2.3.

3. Selanjutnya, data diurutkan berdasarkan jarak seperti yang terlihat pada Tabel 2.4.

Tabel 2. 4 : Data setelah diurutkan berdasarkan jarak terkecil

Tinggi Badan	Berat Badan	% Lemak	Sistolik	Diastolik	Detak Jantung	Lengan Atas	Lengan Bawah	Lingkar Perut	Lingkar Panggul	Status Gizi	Jarak
172	68	22.8	110	65	68	27	15	79	94	Normal	17.4367
177	60	17.4	107	71	104	24	15	71	90	Normal	18.503
163	59	25.4	102	68	69	23	14	74	94	Normal	18.6215
168	73	22.7	107	73	81	30	18	77	96	Normal	18.976
165	73	29.1	105	67	75	28	18	83	100	Obesitas	21.0763
170	72	26.1	113	72	83	28	16	85	98	Normal	23.6561
173	56	18.4	114	73	62	24	15	66	93	Normal	25.6351
162	54	18	108	70	76	24	15	71	88	Normal	25.671
169.5	55	15.2	112	77	70	22	14	75	92	Kurus	26.0594
172	75	31	114	78	86	26	16	79	100	Normal	26.2679
168	50	10.4	100	72	97	23	14	62	87	Kurus	29.4747
173	56	17.9	120	76	97	25	14	72	88	Normal	30.3317
167	50	16.5	111	80	89	21	13	71	88	Kurus	30.7449
159	58	23.4	118	75	64	26	15	70	85	Normal	32.6827
169	79	22	123	76	70	29	17	84	101	Obesitas	34.2783
166	58	19	128	69	69	26	16	72	92	Normal	35.5387
168	52	13.9	122	82	101	22	15	68	84	Kurus	38.5462
167	75	30.7	126	81	65	30	16	91	103	Obesitas	40.8692
160	54	15.7	138	104	78	27	15	73	86	Normal	60.7148
170	125	42.9	113	73	79	40	19	112	135	Obesitas	85.9477

4. Apabila ditetapkan nilai K=5, maka diambil 5 jarak terpendek.

5. Pada urutan pertama sampai kelima, ada 4 status gizi Normal dan 1 status gizi Kurus, sehingga untuk data evaluasi $r = (175,$

67, 23, 97, 67, 79, 25, 15, 75, 95) termasuk dalam status gizi Normal.

2.2.3 Mentimun

2.2.3.1 Pengertian Mentimun

Mentimun merupakan salah satu jenis sayuran buah yang memiliki banyak kegunaan dalam kehidupan masyarakat sehari-hari. Timun (*Cucumis Sativus L*) adalah tanaman sayuran yang banyak manfaat dan kegunaannya antara lain sebagai bahan makanan (diolah sebagai Acar Timun, Es Timun), bahan obat-obatan dan bahan kosmetika. Sebagai bahan makanan, timun mengandung zat gizi antara lain kalori, protein, lemak, karbohidrat, kalsium, fosfor, besi, vitamin A, vitamin B1, vitamin B2, vitamin C, serat, niacin dan air[1].

Timun adalah tanaman sayuran buah semusim (berumur pendek) seperti terong, labu dan tomat, tumbuh menjalar, berbentuk semak atau perdu bisa mencapai tinggi 2 meter. Akar berbentuk tunggang dan berair, batang lunak dan berair, daun berbentuk bulat dengan ujung daun runcing berganda, berbulu halus dan tulang daun menjari. Bunga terdiri dari bunga jantan dan betina dengan perbandingan jumlah yang relatif sama. Buah panjang silindris dengan bentuk beragam tergantung varietasnya[1].

2.2.3.2 Penyakit Mentimun

Penyakit adalah kerusakan yang terjadi pada tanaman yang disebabkan oleh jamur, bakteri atau virus yang menyerang atau merusak organ tanaman. Pada tanaman mentimun varietas roman, penyakit yang sering kali muncul antara lain sebagai berikut[10]:

1. Busuk Daun (*Downy mildew*)



Gambar 2. 2 : Penyakit Busuk Daun [11]

Penyakit yang disebabkan oleh jamur *Pseudoperonospora* yang merusak daun yang dipancarkan oleh angin sehingga menginfeksi kulit daun mentimun. Infeksi pada tanaman terjadi jika kelembaban udara tinggi. Gejala serangan yang diakibatkan daun berwarna kuning dan berjamur selain itu daun mudah membusuk. Untuk itu tindakan pengendalian jenis penyakit busuk daun bisa dilakukan dengan mengatur jarak tanam agar tidak terlalu dekat dengan kelembaban udara bisa berkurang. Tanaman yang terserang agar segera dikeluarkan / dibuang atau disemprot dengan fungisida Velimex 80 WP dengan dosis 2-2,5 g / L dan volume semprot 400-800 L / ha, penambahan varietas mentimun yang tahan terhadap penyakit, memperbaiki drainase tanah dan mengurangi kelembaban kebun dengan cara memperpanjang jarak tanam[10].

2. Penyakit Embun Tepung (*Powdery mildew*)



Gambar 2. 3 : Penyakit Embun Tepung [12]

Penyakit ini disebabkan oleh *Erysiphe cichoracearum*. Penyakit ini dipancarkan oleh angin, kemudian berkembang dengan cepat bila suhu dan keadaan tanahnya kering. Biasanya penyakit ini menyerang pada saat musim kemarau yang menyerang daun dan batang dengan gejala terdapat lapisan putih bertepung pada permukaan daun dan batang muda. Daun yang dilapisi lapisan putih bertepung akan berubah menjadi berwarna kuning dan akhirnya mengering. Untuk tindakan pengendaliannya dapat dilakukan dengan cara menanam mentimun varietas yang tahan (*resisten*) mencabut tanaman yang terserang cukup parah dan disemprot dengan fungisida yang mengandung bahan aktif *benomyl* atau *karbendazim*[10].

3. Antraknosa



Gambar 2. 4 : Penyakit Antraknosa [13]

Penyakit ini disebabkan oleh *Cendawan Colletotrichum* *Lagenarium* *Pass* dan mengakibatkan daun berwarna kuning atau kecoklatan, tangkai batang berwarna kuning dan buah berwarna kuning hingga kehitaman. Tindakan pengendalian yang dapat dilakukan dengan cara mengadakan pergiliran tanaman, menanam benih yang baik atau sebelumnya benih direndam dalam bantuan berbahan aktif tiram seperti Benlate T 20/20 WP. Selain itu, tanaman dapat disemprot dengan Velimex 80 WP dengan dosis 2-2,5 g / L udara dan volume semprot 400-800 L / ha atau Difolatan 4 F dengan dosis 2-3 cc / L udara[10].

4. Bercak Daun



Gambar 2. 5 : Penyakit Bercak Daun [14]

Penyakit ini disebabkan oleh *CendawanPseodomonas Laschrymans*, akibat yang ditimbulkan yaitu daun berwarna kuning, berlubang dan mengering. Tindakan pengendaliannya bias dilakukan yaitu melakukan pergiliran tanaman, menanam biji yang sehat dan varietas yang tahan (*resisten*) serta menyemprotkan bakterisida yang mengandung bahan aktif *streptomisin* atau *oksitetrasiklin*[10].

5. Virus Mosaik



Gambar 2. 6 : Penyakit Virus Mosaik [15]

Penyakit ini disebabkan oleh *Cucumber Mosaic Virus*. Akibat yang ditimbulkan virus ini adalah permukaan daun belang berwarna hijau tua dan hijau muda, daun layu dan mengering serta tanaman kerdil. Pengendaliannya adalah mengendalikan vektornya melalui penyemprotan insektisida sejak dini. Secara mekanis pengendalian dilakukan dengan mencabut dan merotasi dengan tanaman jenis lain yang berbeda *varietas*, menjaga kebersihan kebun dan kebersihan alat pertanian,

mengendalikan vektor serangga, dan mencabut serta memperbaiki tanaman yang terserang[10].

6. Kudis (*Scab*)



Gambar 2. 7 : Penyakit Scab [15]

Penyakit ini disebabkan oleh cendawan *Cladosporium Cucumerinum* dipancarkan oleh angin dan sisa-sisa tanaman. Faktor-faktor yang mempengaruhi perkembangan penyakit adalah cuaca lembab, berkabut, banyak embun, dan keadaan malam yang sejuk. Gejala yang ditimbulkan yaitu tiba-tiba buah mengeluarkan cairan bening sampai coklat, buah mudah busuk dan berjatuhan. Pengendalian serangan penyakit yang bisa dilakukan yaitu dengan cara melakukan pergiliran rotasi tanaman atau bisa juga dengan cara menanam benih atau bibit yang sehat serta *varietas* mentimun yang tahan dan disemprot fungisida ditiokarbamat[10].

2.2.4 Pengembangan Sistem

2.2.4.1 *System Development Life Cycle*

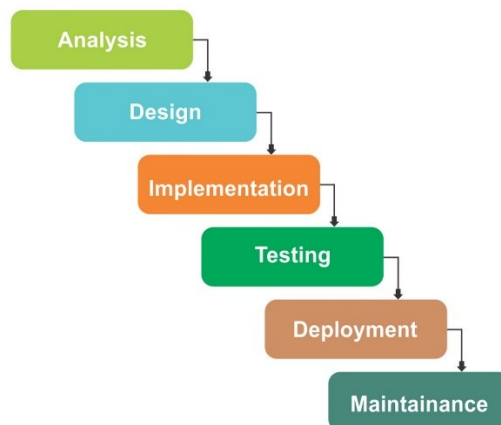
Siklus Hidup Pengembangan Sistem atau SDLC merupakan tahapan-tahapan pekerjaan terdefinisi, terstruktur dengan baik yang dilakukan oleh analisis sistem dan programmer dalam membangun suatu perangkat lunak yang berkualitas. SDLC yang dilakukan dengan benar dapat memungkinkan tingkat control dan dokumentasi manajemen tertinggi. Pengembang memahami

apa yang harus mereka bangun dan mengapa. Semua pihak menyetujui tujuan di depan dan melihat rencana yang jelas untuk mencapai tujuan tersebut. Ada beberapa macam model dalam SDLC, salah satunya yaitu model *Waterfall*[16].

2.2.4.2 Model *Waterfall*

Model *Waterfall* yang sering juga dikenal sebagai model air terjun adalah model proses pertama yang diperkenalkan. Ia sangat mudah dimengerti dan digunakan. Dalam model *Waterfall*, setiap fase harus diselesaikan sebelum fase berikutnya dapat dimulai dan tidak ada fase tumpang tindih. Model *Waterfall* adalah pendekatan SDLC paling awal yang digunakan untuk pengembangan perangkat lunak[16].

Pada pendekatan *Waterfall*, seluruh proses pengembangan perangkat lunak dibagi menjadi fase-fase terpisah. Hasil dari fase satu bertindak sebagai input untuk fase berikutnya secara berurutan. Ini berarti bahwa setiap fase dalam proses pengembangan dimulai hanya jika fase sebelumnya selesai[16].



Gambar 2. 8 : Model *Waterfall* [17]

Tahapan – tahapan yang ada pada model *waterfall* secara umum adalah[18]:

a. *Analysis* (Analisis Kebutuhan)

Analysis adalah proses pengumpulan kebutuhan yang dilakukan secara intensif untuk memesifikasi kebutuhan perangkat lunak agar dapat dipahami seperti apa yang dibutuhkan oleh user.

b. *Design* (Desain)

Design adalah proses multi langkah yang fokus pada desain pembuatan program perangkat lunak termasuk struktur data, *arsitektur* perangkat lunak, *representasi* antar muka dan prosedur pengodean.

c. *Implementation* (Pembuatan Kode Program)

Implementation adalah desain harus ditranslasikan ke dalam program perangkat lunak.

d. *Testing* (Pengujian)

Testing adalah fokus pada perangkat lunak secara dari segi logik dan fungsional dan memastikan bahwa semua bagian sudah diuji. Hal ini dilakukan untuk meminimalisir kesalahan dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan.

e. *Deployment* (Verifikasi)

Pada tahap ini pengguna menguji apakah sistem tersebut telah sesuai dengan yang disetujui.

f. *Maintenance*

Maintenance adalah instalasi dan proses perbaikan sistem sesuai yang disetujui.

2.2.4.3 Pengertian Sistem

Sistem adalah seperangkat komponen yang saling berhubungan dan bekerjasama antara satu sama lain untuk mencapai tujuan. Sistem adalah

kumpulan dari elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu[19].

2.2.4.4 Analisis Sistem

Analisis sistem adalah sebagai komponen yang utuh dari suatu sistem dengan tujuan dapat mengevaluasi permasalahan, kesempatan yang misal terjadi sehingga untuk penanganannya dapat diketahui. Jadi seandainya ditahap ini memiliki kesalahan, maka bisa juga akan menyebabkan kesalahan ditahap yang selanjutnya.

a. Studi kelayakan

Untuk mendapatkan keberhasilan dari solusi yang di usulkan, kita bisa menggunakan studi kelayakan.

b. Analisis kebutuhan

Untuk mendapatkan spesifikasi kebutuhan yang harus kita lakukan adalah mencari spesifikasi yang rinci tentang hal-hal yang dilakukn oleh sistem. Sekaligus spesifikasi digunakan supaya memuat kesepakatan antara pengembang.

2.2.4.5 Desain Sistem

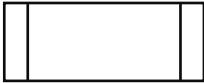
Desain sistem adalah berupa output yang dikehendaki untuk menganalisa persyaratan yang diperlukan untuk menyiapkan data yang perlu disimpan dalam file. Tahap perencanaannya sebagai berikut :

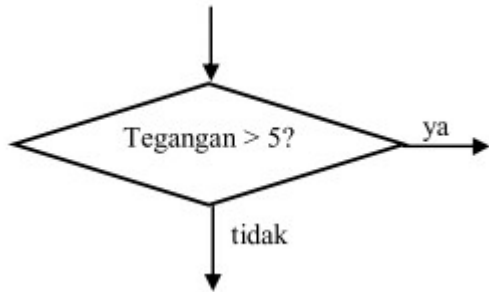
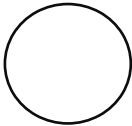
1. Merancang penerapan file- file yang dibutuhkan datanya.
2. Sistem dirancang secara global.
3. Pembuatan program untuk mendesain sistem.

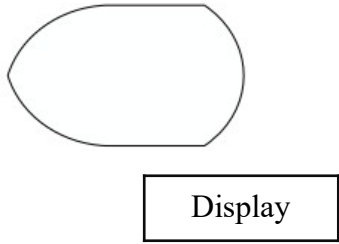
2.2.4.6 Flowchart

Diagram alir (*flowchart*) merupakan cara lain untuk menuangkan algoritma. Pendekatan yang dilakukan adalah dengan menggunakan gambar[20].

Tabel 2. 5 : Simbol-simbol Diagram Alir (*Flowchart*)

No	Simbol	Keterangan
1.	 (Terminator)	Simbol ini digunakan untuk menyatakan titik awal atau titik akhir diagram alir. Kemungkinan isinya seperti berikut: Mulai Selesai Dalam bahasa Inggris, kata <i>Start</i> dan <i>End</i> biasa dipakai.
2.	 (Proses)	Simbol ini digunakan untuk menyatakan sebarang proses misalnya untuk menyatakan suatu operasi <i>aritmetika</i>.
3.	 (Proses Terdefinisi)	Simbol ini menyatakan prosedur lain yang didiagramalirkan pada tempat lain.
4.	 (Input/Output atau kadang disebut Data)	Simbol ini menyatakan operasi pemasukan data atau penampilan data. Contoh: Masukkan Panjang ➤ Untuk memasukkan data panjang (membaca data panjang dari <i>keyboard</i>) Tampilkan Luas ➤ Untuk menampilkan data luas
5.	(Keputusan)	Simbol ini digunakan Untuk melakukan pengambilan keputusan. Dalam hal ini, yang ada dalam simbol ini berupa suatu pernyataan yang jawabannya berupa dua kemungkinan, yaitu “ya” atau “tidak”. Contoh

		penggunaannya seperti berikut:  Pada contoh tersebut, tanda panah yang menunjuk ke simbol keputusan menyatakan aliran menuju ke keputusan. Kondisi Tegangan > 5 ? menyatakan suatu pertanyaan yang berbunyi, “Apakah isi Tegangan lebih besar daripada 5?” kalau jawaban pertanyaan tersebut berupa ya, aliran menuju ke kanan. Sebaliknya, untuk jawaban tidak, maka aliran menuju ke bawah.
6.	 (Konektor)	Simbol ini digunakan untuk menghubungkan ke berbagai bagian dalam diagram alir. Contoh:  Bagian yang sebelah kiri menyatakan bahwa aliran dilanjutkan ke bagian yang ditandai dengan . Gambar sebelah kanan menyatakan bagian yang bertanda . Tanda panah menyatakan arah pemrosesan selanjutnya. Namun, simbol ini juga terkadang digunakan pada pertemuan dua aliran.

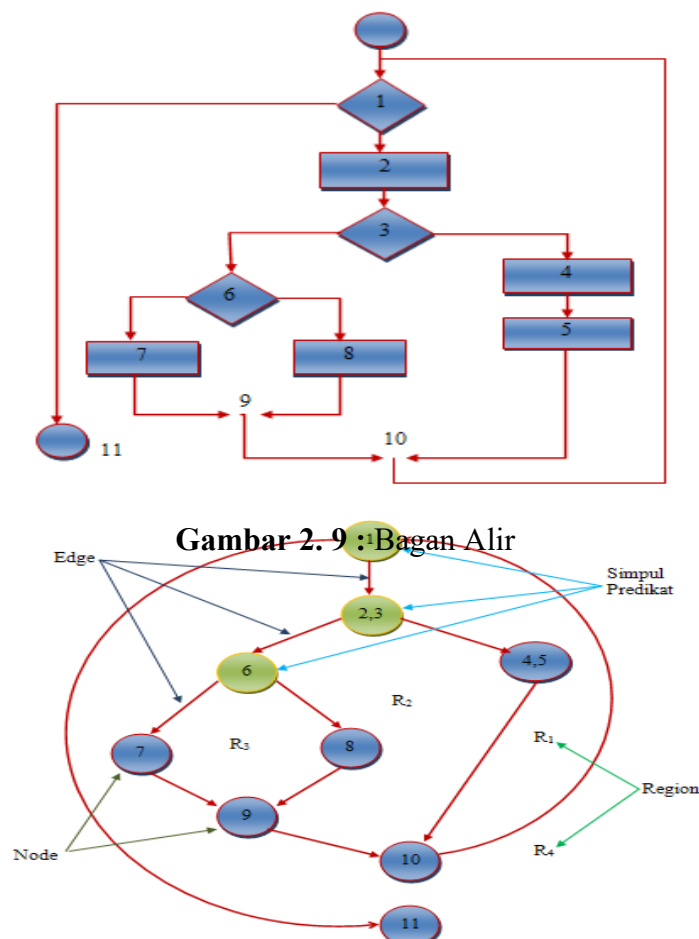
7.		Simbol ini digunakan untuk menyatakan peralatan output yang dipakai mulai dari <i>printer</i> , <i>plotter</i> dan layar.
----	---	---

2.2.5 Teknik Pengujian Sistem

2.2.5.1 White Box

Pengujian *white-box* (*glass box*), adalah metode desain *test case* yang menggunakan struktur kontrol desain prosedural untuk memperoleh *test case*. Dengan menggunakan metode pengujian *white-box*, perekayasa sistem dapat melakukan test case untuk memberikan jaminan bahwa[21]:

- Semua jalur independen pada suatu modul ditelusuri minimal 1 kali
- Semua jalur keputusan logis *True/False* dilalui
- Semua loop dieksekusi pada batas yang tercantum dan batas operasionalnya
- Struktur data internal digunakan agar validitas terjamin



Gambar 2. 10 : Grafik Alir

- a. *Node* adalah lingkaran yang merepresentasikan satu atau lebih statemen procedural.
- b. *Edge* adalah anak panah pada grafik alir.
- c. *Region* adalah area yang membatasi edge dan node
- d. Simpul predikat adalah simpul atau node yang berisi kondisi yang ditandai dengan dua atau lebih *edge* yang berasal darinya.

2.2.5.2 *Black Box*

Black Box adalah metode pengujian perangkat lunak yang menguji fungsionalitas aplikasi yang bertentangan dengan struktur internal atau kerja. Metode uji dapat diterapkan pada semua tingkat pengujian perangkat lunak: unit, integrasi, fungsional, sistem dan penerimaan[21].

Pengujian *Black-Box* berusaha menemukan kesalahan dalam kategori:

- a. Fungsi tidak benar atau hilang .
- b. Kesalahan antar muka.
- c. Kesalahan pada struktur data (pengaksesan basis data).
- d. Kesalahan inisialisasi dan akhir program.
- e. Kesalahan performasi.

2.2.6 Perangkat Lunak Pendukung

Penulis menggunakan perangkat lunak pendukung dengan menggunakan PHP, XAMPP, MySQL, Adobe Dreamweaver, dan Adobe Photoshop.

2.2.6.1 PHP (*Hypertext Preprocessor*)

PHP atau *Hypertext Preprocessor* merupakan bahasa berbentuk *script* yang ditempatkan dalam *server* dan dieksekusi di dalam *server* untuk selanjutnya dibaca oleh *client*. PHP juga bisa disisipkan dalam bahasa HTML[22].



Gambar 2. 11 : PHP

2.2.6.2 MySQL

MySQL diciptakan di negara Swedia oleh perusahaan MySQL AB. Perangkat lunak ini tersebar luas secara gratis karena memiliki lisensi GNU *General Public License*. MySQL adalah salah satu jenis *Database* yang sangat terkenal dan banyak digunakan untuk membangun aplikasi *web* yang menggunakan *database* sebagai sumber dan pengelolaan datanya[22].



Gambar 2. 12 : MySQL

2.2.6.3 XAMPP

Xampp adalah salah satu paket *installer* yang berisi *Apache* yang merupakan *web server* tempat menyimpan file-file yang diperlukan *website*, dan



PhpMyAdmin sebagai aplikasi yang digunakan untuk perancangan *databaseMySQL*[23].

Gambar 2. 13 : XAMPP

2.2.6.4 *Adobe Dreamweaver*

Adobe Dreamweaver merupakan aplikasi pengembang yang berfungsi untuk mendesain *web* yang dibuat, dikembangkan, dan diproduksi oleh *Adobe System*. Aplikasi pengembang *web* ini sangat digemari oleh *web* desainer dalam merancang *web*, sebab perangkat lunak ini memiliki kelebihan dan kemudahan dalam penggunaannya[24].



Gambar 2. 14 : *Adobe Dreamweaver*

2.2.6.5 *Adobe Photoshop*

Adobe Photoshop merupakan salah satu *software* pengolah gambar (*digital imaging*) yang banyak digunakan dan memberikan kontribusi besar bagi dunia *digital* dan cetak. *Adobe PhotoshopCS3* masih memuat fitur-fitur yang terdapat pada *Adobe Photoshop CS2* sebelumnya, namun memiliki beberapa kelebihan dan keunggulan dibanding *versi* sebelumnya[25].



Gambar 2. 15 : *Adobe Photoshop CS3*

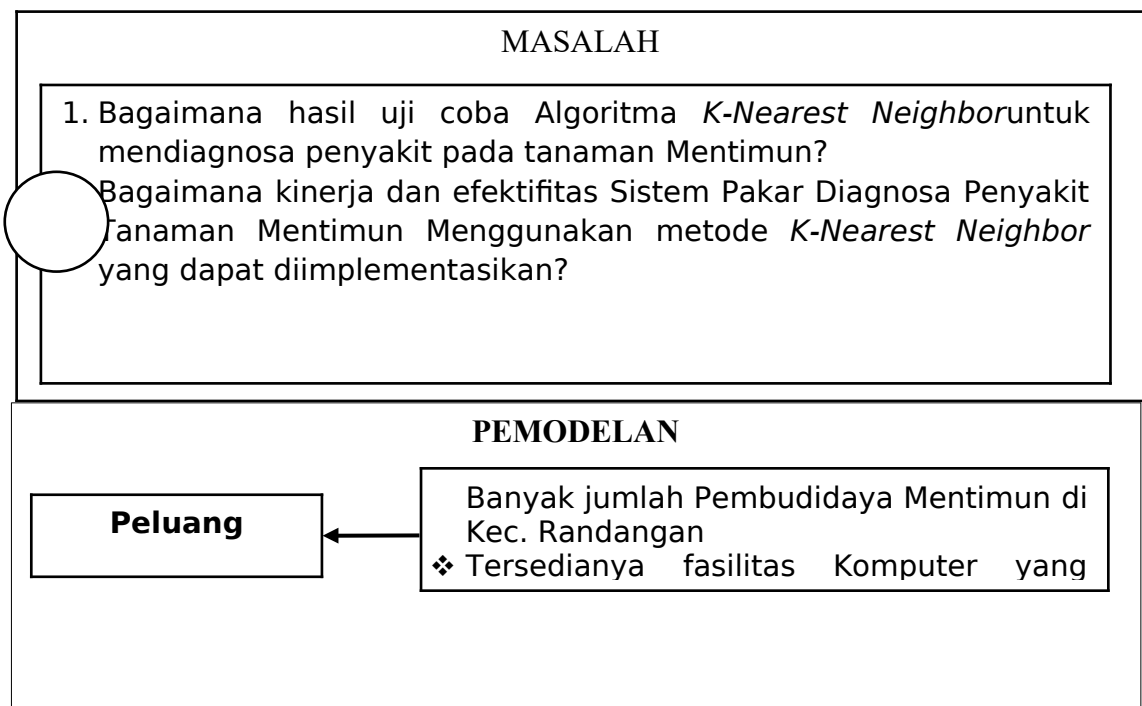
2.2.6.6 **Microsoft Visio**

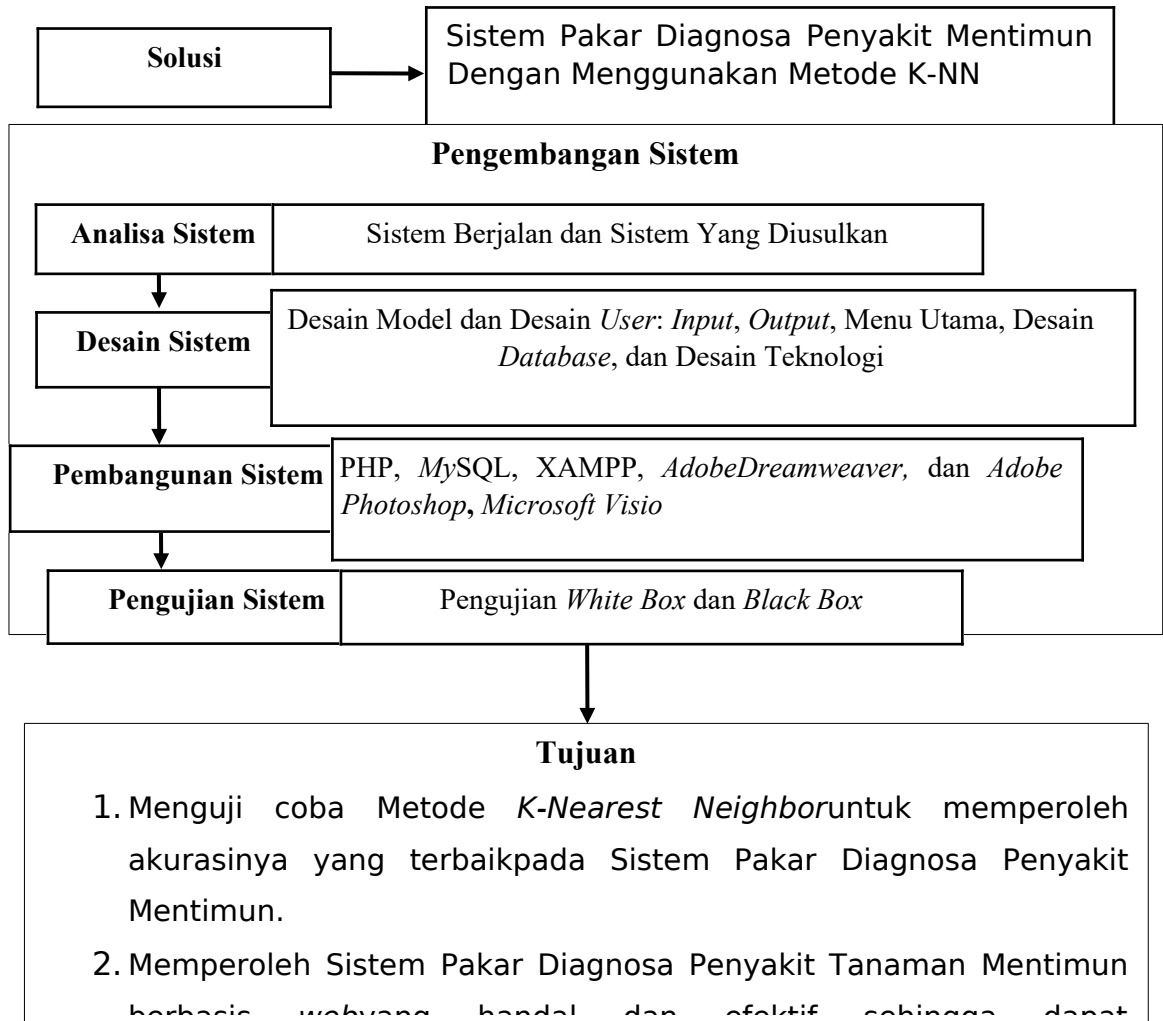
Microsoft Visio adalah sebuah aplikasi khusus untuk membuat semua diagram bisnis, mulai dari *flowchart*, *network diagram*, dan *organization charts*, untuk membuat denah dan *brainstorming* diagram.



Gambar 2. 16 : *Microsoft Visio*

2.3 Kerangka Pemikiran





Gambar 2. 17 : Kerangka Pikir

BAB III

METODE PENELITIAN

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

Penelitian

Dilihat dari informasi yang diolah, maka jenis penelitian ini merupakan penelitian *kuantitatif*. Penelitian *kuantitatif* merupakan penelitian yang lebih memfokuskan pada aspek pengukuran secara *objektif* terhadap fenomena sosial. Penelitian ini menggunakan metode penelitian studi kasus. Maka dengan itu jenis penelitian ini adalah penelitian *deskriptif*.

Subjek penelitian ini adalah *klasifikasi* pada objek “**Sistem Pakar Diagnosa Penyakit Tanaman Mentimun Dengan Metode K-Nearest Neighbor (KNN)**”. Penelitian ini dimulai dari bulan Oktober 2021 sampai dengan bulan Desember 2021. Objek yang akan dilakukan pada penelitian ini yaitu Penyakit pada tanaman Mentimun dan lokasi penelitiannya di Kantor Dinas Pertanian yang beralamatkan di Jl. Kusno Danupoyo Kompleks Blok Plan Perkantoran, Kecamatan Marisa, Kabupaten Pohuwato, Provinsi Gorontalo.

3.2 Pengumpulan Data

Untuk mendapatkan data yang benar-benar akurat, relevan, valid dan dapat dipercaya maka pengumpulan data dilakukan sebagai berikut:

- a. *Observasi*, Pengumpulan data melalui pengamatan dan pencatatan terhadap berbagai bentuk laporan-laporan yang akan dihasilkan dari sistem pakar. Pengamatan dilakukan di Kantor Dinas Pertanian Kabupaten Pohuwato.
- b. *Interview*, Pengumpulan data dengan cara tatap muka dan Tanya jawab langsung dengan sumber data, yaitu Pegawai

Dinas Pertanian Kabupaten Pohuwato sebagai pakar dari Sistem ini.

- c. Survey, Pengumpulan data dengan melakukan *survey* ke Dinas Pertanian Kabupaten Pohuwato.
- d. Studi pustaka, Pengumpulan data dengan cara membaca serta mempelajari dokumen-dokumen, *literature*, buku, jurnal, video/gambar yang berhubungan dengan obyek penelitian guna mendapatkan teori ataupun konsep yang dapat digunakan sebagai landasan teori dan kerangka pemikiran dalam penelitian dan untuk mencari metodologi yang sesuai serta membandingkan teori yang ada dengan fakta yang ada di lapangan.

3.3 Pengembangan Sistem

Dalam pembangunan aplikasi Sistem Pakar Diagnosa Penyakit Tanaman Mentimun Dengan Metode *K-Nearest Neighbor* (KNN) Pada Dinas Pertanian Kabupaten Pohuwato, menggunakan beberapa perangkat lunak antara lain:

- *PHP*
- *MySQL*
- *XAMPP*
- *Adobe Dreamweaver*
- *Adobe Photoshop*
- *Microsoft Visio*

3.3.1 Analisis Sistem

Untuk memulai pembuatan suatu sistem yang baru, harus dimulai dari awal dengan cara mengumpulkan informasi yang selengkap-lengkapnyanya. Khususnya jika ada kekurangan dari sistem lama, maka sistem baru yang dibuat harus dapat menutupi kekurangan sistem lama tersebut. Pada analisis sistem diuraikan suatu sistem informasi yang utuh kedalam bagian-bagian komponennya

dengan maksud untuk mengidentifikasi dan mengevaluasi suatu permasalahan, kesempatan, hambatan yang terjadi dan kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan.

3.3.1.1 Analisis Sistem Berjalan

Menganalisis sistem yang berjalan merupakan salah satu tahap untuk menganalisis suatu sistem dengan tujuan untuk mengetahui apakah ada masalah atau kesulitan yang ditemui sehingga dapat dilakukan suatu ide-ide baru dengan membangun sebuah sistem yang baru. Sistem yang berjalan belum menggunakan sebuah sistem. Petani biasanya berkonsultasi kepada penyuluh atau menggunakan pengetahuan terbatas yang dimiliki oleh Petani.

3.3.1.2 Analisis Sistem Yang Diusulkan

Sistem yang diusulkan yaitu membuat sebuah aplikasi sistem pakar yang dapat memudahkan petani atau pengguna mendiagnosa penyakit yang menyerang tanaman mentimun, sehingga penyakit tanaman tersebut dapat segera ditanggulangi. Aplikasi sistem pakar ini dibuat agar penyajian informasi dan dalam melakukan konsultasi tentang penyakit yang menyerang pada tanaman mentimun dapat memberikan solusi yang lebih cepat dan efektif. Secara umum dapat digambarkan bahwa sistem yang akan dibangun merupakan sebuah sistem pakar yang menggunakan metode *K-Nearest Neighbor*.

3.3.2 Desain Sistem

Desain sistem merupakan tahap selanjutnya setelah analisis sistem, mendapatkan gambaran tentang apa yang dikerjakan pada analisa sistem, maka dilanjutkan dengan memikirkan bagaimana membentuk sistem tersebut.

Pada tahap ini dilakukan desain sistem yakni desain *output*, desain *input*, desain *database*, desain teknologi dan desain model :

a. Desain *Output*

Keluaran (output) adalah produk dari Aplikasi yang dapat dilihat. Output dapat berupa hasil media keras seperti kertas, atau dapat pula hanya berupa tampilan informasi pada layar monitor.

b. *Desain Input*

Desain input adalah dokumen dasar yang digunakan untuk menangkap data, kode-kode input yang digunakan. Untuk tahap rancangan input secara umum yang perlu dilakukan analisis adalah mengidentifikasi terlebih dahulu input yang akan didesain secara rinci tersebut.

c. *Desain Database*

Merupakan kumpulan data yang saling berhubungan satu dengan yang lain. Database salah satu komponen yang penting di sistem informasi, karena berfungsi sebagai penyedia informasi bagi pemakainya.

d. *Desain Teknologi*

Desain teknologi terbagi dua yaitu desain teknologi secara umum dan terinci. Dipergunakan dalam menerima *input*, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan keluaran dan membantu pengendalian dari sistem secara keseluruhan.

e. *Desain Model*

Merupakan tahapan yang lebih berfokus pada spesifikasi detail berbasis sistem. Pada tahap ini digunakan *Data Flow Diagram* (DFD), dimana memodelkan persyaratan logis dari suatu sistem informasi.

3.3.3 Pengujian Sistem

Sebelum program diterapkan, maka program harus bebas terlebih dahulu dari kesalahan-kesalahan. Oleh sebab itu program harus diuji untuk menentukan kesalahan-kesalahan yang mungkin dapat terjadi. Program dites untuk tiap-tiap modul dan dilanjutkan dengan pengetesan untuk semua modul yang telah dirangkai. Kesalahan dari program yang mungkin

terjadi dapat diklasifikasikan dalam tiga bentuk kesalahan, yaitu sebagai berikut:

1. Kesalahan bahasa (*language errors*) atau disebut juga dengan kesalahan penulis (*syntax errors*) atau kesalahan tata bahasa (*grammatical errors*) adalah kesalahan didalam penulisan *source program* yang tidak sesuai dengan yang telah disyaratkan. Kesalahan ini relatif ditemukan dan diperbaiki, karena kompiler akan memberitahukan letak dan sebab kesalahannya sewaktu program dikompilasi.
2. Kesalahan sewaktu proses (*run time errors*), adalah kesalahan yang terjadi sewaktu *executable program* dijalankan. Kesalahan ini akan menyebabkan proses program berhenti sebelum selesai pada saatnya, karena compiler menemukan kondisi-kondisi yang belum terpenuhi yang tidak bisa dikerjakan. Kesalahan ini juga relatif mudah ditemukan, karena juga ditunjukkan letak serta sebab kesalahannya.
3. Kesalahan logika (*logical errors*), adalah kesalahan dari logika program yang dibuat. Kesalahan seperti ini sulit ditemukan, karena tidak ada pemberitahuan mengenai kesalahan dan tetap akan didapatkan hasil dari proses program, tetapi hasilnya salah. Kesalahan seperti ini merupakan kesalahan yang berbahaya karena bila tidak disadari dan tidak ditemukan, hasil yang salah dapat menyesatkan bagi yang menggunakannya. Cara mencari kesalahan logika dapat dilakukan dengan *test data*, yaitu dengan menjalankan program dengan menggunakan data tertentu dan membandingkan hasil pengolahannya dengan hasil yang sudah diketahui. Bila hasilnya berbeda, berarti mengalami kesalahan dan harus di lacak serta ditemukan sebab sebab kesalahannya. Proses melacak kesalahan ini dikenal dengan

istilah mencari kutu (*debugging*). Hasil pelacaknya adalah didapatkan kutu tersebut (*bug* yang berarti penyebab kesalahannya).

Program dapat diuji untuk tiap-tiap modulnya dan dilanjutkan dengan pengujian untuk semua modul yang telah dirangkai. Dengan demikian terdapat tiga tingkat pengujian yang dilakukan, yaitu:

1. Pengujian Modul

Pengujian untuk tiap-tiap modul program (dapat berupa program utama, sub routine, sub program) disebut dengan *stub testing*. Pengetesan suatu modul dapat saja dilakukan walaupun modul lainnya yang berhubungan dengannya belum ditulis. Hal ini mudah saja dilakukan, yaitu dengan cara mensimulasi modul yang dipanggil yang belum ditulis. Modul dipanggil yang disimulasi ini disebut dengan *stub*. Modul *stub* dapat berupa sub routine atau sub program yang tidak berisi dengan logika-logika program. Mungkin juga modul *stub* ini diisi dengan instruksi-instruksi yang akan mencetak parameter yang diterimanya untuk menunjukkan bahwa modul ini sudah dapat dipanggil dengan benar.

2. Pengujian Unit atau Pengujian Program

Setelah semua modul program selesai ditulis dan diuji secara independent sampai bebas dari kesalahan dan telah dirangkai menjadi satu unit program ini perlu diuji kembali. Pengetesan untuk ini disebut dengan *unit testing* atau *program testing* yang dimaksudkan untuk meyakinkan bahwa semua modul telah bekerja terintegrasi tanpa mengalami kesalahan.

3. Pengujian Sistem

Pengujian sistem biasanya dilakukan setelah pengujian program. Pengujian sistem dilakukan untuk memeriksa

kekompakan antar komponen sistem yang diimplementasikan. Tujuan utama dari pengujian sistem ini adalah untuk memastikan bahwa elemen-elemen atau komponen-komponen dari sistem telah berfungsi sesuai dengan yang diharapkan. Pengujian perlu dilakukan untuk mencari kesalahan-kesalahan atau kelemahan-kelemahan yang mungkin terjadi. Pengujian sistem termasuk juga pengujian program secara menyeluruh. Pada pengujian program, masing-masing program yang telah berjalan dengan benar dan baik bukan berarti tersebut juga akan dapat berjalan dengan program lainnya dalam sistem dengan baik. Kumpulan dari semua program yang telah diintegrasikan perlu diuji kembali untuk melihat apakah suatu program dapat menerima input data dengan baik, dapat memprosesnya dengan baik dan dapat memberikan output kepada program yang lainnya.

BAB IV

HASIL PENELITIAN

4.1 Hasil Pengumpulan Data

4.1.1 Sejarah Dinas Pertanian Kabupaten Pohuwato

Saat berdiri Dinas Pertanian Kabupaten Pohuwato yang berkantor sementara di Unit Pelayanan Perkebunan (UPP) di Desa Buntulia Tengah Kecamatan Marisa. Pada Tahun 2004 pindah dan berkantor di Rumah Ka Koha, Kemudian pada tahun 2005 Kembali berkantor di Balai Penyuluh Pertanian (BPP) di Desa Duhiadaa Kec. Marisa, Tahun 2006 pindah kantor lagi di Komplek Kantor Bupati di Jalan Pelabuhan Desa Marisa Selatan Kec. Marisa dan pada Tahun 2007 sudah berkantor di Bangunan Kantor Dinas Pertanian yang beralamatkan di Jl. Kusno Danupoyo Kompleks Blok Plan Perkantoran dan pada awal januari 2017 Dinas Pertanian Menggunakan dua Gedung kantor yakni Gedung A yang terletak di Jalan Kusno Danupoyo Komp. Blok Plan Marisa dan Gedung B beralamatkan di Jalan. Nani Wartabone Komp. Blok Plan Marisa sampai sekarang.

4.1.2 Struktur Organisasi Dinas Pertanian Kabupaten Pohuwato



Gambar 4. 1 : Struktur Organisasi Dinas Pertanian

4.1.3 Visi Dan Misi Dinas Pertanian Kabupaten Pohuwato

Visi:

“TERWUJUDNYA PERTANIAN YANG MAJU BERBASIS SUMBERDAYA LOKAL UNTUK KESEJAHTERAAN PETANI”

Misi:

1. Mewujudkan Sistem Pertanian Yang Maju
2. Mengembangkan Sarana Prasarana dan Infrastruktur Pertanian
3. Meningkatkan nilai tambah dan daya saing mutu produk pertanian berbasis ekonomi kerakyatan.
4. Meningkatkan Sumber Daya Manusia Pertanian

5. Mewujudkan Aparatur Pemerintah Bidang Pertanian Yang Amanah dan Profesional

4.1.4 Tupoksi

Dinas Pertanian merupakan Dinas Teknis yang berperan penting dalam mendukung program pemerintah untuk pengembangan ekonomi kerakyatan di bidang pertanian. Pelayanan SOPD dapat digambarkan secara rinci berdasarkan tata aturan atau regulasi yang bersifat fleksibel dan mengikat dengan penjelasan dan uraian sebagai berikut:

1. Kepala Dinas

Tugas: Menyelenggarakan program pertanian, perkebunan, peternakan dan penyuluhan di seluruh wilayah Kabupaten Pohuwato.

Fungsi:

- a. Penyiapan konsep kebijakan daerah dibidang Pertanian, perkebunan, peternakan dan penyuluhan;
- b. Penetapan standar pelayanan dan standar pelaksanaan tugas dibidang Pertanian, perkebunan, peternakan dan penyuluhan;
- c. Perencanaan, pelaksanaan dan pengendalian di bidang pertanian, perkebunan, peternakan dan penyuluhan;
- d. Pembinaan, bimbingan teknis dan pengawasan teknis dan pelaksanaan kegiatan dibidang Pertanian, perkebunan, peternakan dan penyuluhan;
- e. Memberikan masukan kepada Pemerintah Daerah dibidang Pertanian, perkebunan, peternakan dan penyuluhan;
- f. Melaksanakan tugas lain sesuai bidang tugas dan fungsinya

2. Sekretariat

Tugas: Melaksanakan pelayanan teknis administratif dan koordinasi pelaksanaan tugas dilingkungan Dinas Pertanian.

Fungsi:

- a. Pelaksanaan urusan dan tatalaksana.
- b. Rancangan peraturan dan fasilitas di bidang pertanian, perkebunan, peternakan dan penyuluhan;

- c. Kerja sama di bidang pertanian, perkebunan, peternakan dan penyuluhan;
- d. Pengelolaan Administrasi keuangan APBD, APBD Provinsi dan APBN;
- e. Pelayanan, dan pengolahan data, di bidang pertanian, perkebunan, peternakan dan penyuluhan;
- f. Koordinasi kebijakan, program, rencana dan kegiatan di bidang pertanian, perkebunan, peternakan dan penyuluhan;
- g. Pengelolaan Administrasi Kepegawaian;
- h. Koordinasi penyusunan bahan, dan pelaksanaan publikasi dan hubungan masyarakat di bidang pertanian, perkebunan, peternakan dan penyuluhan;
- i. Pengelolaan barang milik daerah di Lingkungan Dinas Pertanian; dan ,
- j. Koordinasi, monitoring, evaluasi dan pelaporan.

3. Bidang Tanaman Pangan dan Hortikultura

Tugas: Melaksanakan penyusunan, pemberian bimbingan, pemantauan dan evaluasi serta pelaksanaan kebijakan di bidang tanaman pangan dan hortikultura.

Fungsi :

1. Pengolahan dan pemasaran, produksi, dan penyusunan kebijakan perbenihan di bidang tanaman pangan dan hortikultura.
2. Penyusunan rencana kebutuhan dan penyediaan benih di bidang tanaman pangan dan hortikultura.
3. Pengawasan mutu dan peredaran benih.
4. pembinaan, pemberian, penerapan, dan peningkatan produksi, produktivitas dan mutu di bidang tanaman pangan dan hortikultura.

5. Penanggulangan dan pengendalian hama penyakit dibidang tanaman pangan dan hortikultura.
6. Evaluasi beserta pemantauan dibidang tanaman pangan dan hortikultura.
7. Pemberian bimbingan, pemasaran dan pengolahan hasil dibidang tanaman pangan dan hortikultura.
8. Melaksanakan tugas yang diberikan oleh kepala dinas sesuai arahan dan fungsinya.

4. Bidang Perkebunan

Tugas: Melaksanakan kebijakan, dan memberikan bimbingan, serta pemantauan dan evaluasi dibidang tanaman pangan dan hortikultura.

Fungsi :

- a. Penyusunan dibidang pembenihan, perlindungan, produksi, usaha, pemasaran dan pengolahan hasil perkebunan.
- b. Menyusun rencana kebutuhan dan penyediaan benih dibidang perkebunan.
- c. Pengawasan mutu dan peredaran benih.
- d. Penanggulangan gangguan usaha dan pencegahan pembakaran perkebunan.
- e. Pemberian bimbingan penerapan produksi dibidang perkebunan.
- f. Penanggulangan dan pengendalian hama penyakit.
- g. Pemberi izin usaha atau rekomendasi teknik dibidang perkebunan
- h. Pemberian bimbingan usaha, pasca panen, pengolahan dan pemasaran hasil dibidang perkebunan.
- i. Evaluasi dan pemantauan dibidang perkebunan, dan

- j. Melaksanakan tugas lain yang diberikan oleh kepala dinas sesuai tugas dan fungsinya.

Dari hasil pengumpulan data yang dilakukan dengan cara mewawancarai pakarnya yang berada di di Dinas Pertanian Kabupaten Pohuwato yang beralamat di Jl. Kusno Danupoyo Komp. Blok Plan Perkantoran Kecamatan Marisa Kabupaten Pohuwato.

4.1.5 Pengumpulan Data Mentimun

Pada Matriks penyakit Tanaman Mentimun terdapat 7 macam jenis Penyakit yang ditunjukkan oleh kode P01 sampai P07 yang terdiri dari Penyakit Busuk Daun (*Downy Mildew*), Penyakit Embun Tepung (*Powdery Mildew*), Penyakit Antraknosa, Penyakit Bercak Daun, Penyakit Virus Mozaik, Penyakit Kudis (*Scab*), dan penyakit Busuk Buah. Semua jenis-jenis penyakit ini terdapat 17 Gejala penyakit yang ditunjukkan oleh kode G01 sampai G17.

Penyakit yang terdiri dari 7 jenis penyakit dan 17 gejala yang dapat diolah, sehingga dapat menghasilkan kesimpulan serta solusi akhir dari permasalahan ini.

Tabel 4. 1: Daftar Penyakit Tanaman Mentimun

Kode Jenis Penyakit	Nama Macam Penyakit
P01	Penyakit Busuk Daun (<i>Downy Mildew</i>)
P02	Penyakit Embun Tepung (<i>Powdery Mildew</i>)
P03	Penyakit Antraknosa
P04	Penyakit Bercak Daun
P05	Penyakit Virus Mosaik
P06	Penyakit Kudis (<i>Scab</i>)

Tabel 4. 2 : Daftar Gejala Penyakit Tanaman Mentimun

Kode	Nama Gejala Penyakit
G01	Daun berwarna kuning
G02	Daun berjamur
G03	Bercak kuning cepat menyebar ke seluruh permukaan daun
G04	Daun mudah membusuk
G05	Terdapat lapisan putih bertepung pada daun dan batang muda
G06	Daun yang dilapisi lapisan putih bertepung, berubah menjadi warna kuning dan mengering
G07	Daun berwarna kuning atau kecoklatan
G08	Tangkai batang berwarna kuning
G09	Buah berwarna kuning hingga kehitaman
G10	Daun berlubang
G11	Daun mengering
G12	Daun berwarna hijau tua dan hijau muda
G13	Daun layu
G14	Tanaman Kerdil
G15	Buah mengeluarkan cairan bening sampai coklat
G16	Buah mudah busuk
G17	Buah berjatuhan

Tabel 4.3 : Penyakit dan Gejala Tanaman Mentimun

Kode Penyakit	Kode Penyakit	Kode Gejala	Nama Penyakit
P01	Busuk Daun (<i>Downy Mildew</i>)	G01	Daun berwarna kuning
		G02	Daun berjamur
		G03	Bercak kuning cepat menyebar ke seluruh permukaan daun
		G04	Daun mudah membusuk
P02	Embun Tepung (<i>Powdery Mildew</i>)	G05	Terdapat lapisan putih bertepung pada daun dan batang muda

		G06	Daun yang dilapisi lapisan putih bertepung, berubah menjadi warna kuning dan mengering
P03	Antraknosa	G07	Daun berwarna kuning atau kecoklatan
		G08	Tangkai batang berwarna kuning
		G09	Buah berwarna kuning hingga kehitaman
P04	Bercak Daun	G01	Daun berwarna kuning
		G10	Daun berlubang
		G11	Daun mengering
P05	Virus Mosaik	G12	Daun berwarna hijau tua dan hijau muda
		G13	Daun layu
		G11	Daun mengering
		G14	Tanaman Kerdil

P06	Kudis (<i>Scab</i>)	G15	Buah mengeluarkan cairan bening sampai coklat
		G16	Buah mudah busuk
		G17	Buah berjatuhan

Tabel 4. 4 : Daftar Solusi Tanaman Mentimun

Kode	Solusi
T01	Tanaman yang terserang agar segera di keluarkan / dibuang
T02	Penyemprotan fungisida Velimex 80 WP
T03	Memperbaiki drainase tanah
T04	Mengatur jarak tanam
T05	Penyemprotan fungisida yang mengandung bahan aktif benomy atau karbendazim
T06	Benih direndam dalam bantuan berbahan aktif tiram seperti Benlate T 20/20 WP
T07	Penyemprotan dengan Velimex 80 WP atau Difolatan 4 F
T08	Menanam biji yang sehat dan varietas yang tahan (resisten)
T09	Penyemprotan bakterisida yang mengandung bahan aktif streptomisin atau oksitetrasiklin
T10	Pengendalian vektornya melalui penyemprotan insektisida sejak dini
T11	Mencabut dan merotasi dengan tanaman jenis lain yang berbeda varietas
T12	Menjaga kebersihan kebun dan alat pertanian
T13	Penyemprotan fungisida ditiokarbamat

Tabel 4. 5 : Penyakit dan Solusi Tanaman Mentimun

Kode Penyakit	Nama Penyakit	Kode Solusi	Solusi
P01	Busuk Daun (<i>Downy Mildew</i>)	T01	Tanaman yang terserang agar segera dikeluarkan / dibuang
		T02	Penyemprotan fungisida Velimex 80 WP
		T03	Memperbaiki drainase tanah
		T04	Mengatur jarak tanam
P02	Embun Tepung (<i>Powdery Mildew</i>)	T01	Tanaman yang terserang agar segera dikeluarkan / dibuang
		T05	Penyemprotan fungisida yang mengandung bahan aktif benomy atau karbendazim
P03	Antraknosa	T06	Benih direndam dalam bantuan berbahan aktif tiram seperti Benlate T 20/20 WP
		T07	Penyemprotan dengan Velimex 80 WP atau Difolatan 4 F
P04	Bercak Daun	T08	Menanam biji yang sehat dan varietas yang tahan (resisten)
		T09	Penyemprotan bakterisida yang mengandung bahan aktif streptomisin atau oksitetrasiklin

P05	Virus Mosaik	T01	Tanaman yang terserang agar
-----	--------------	-----	-----------------------------

			segera dikeluarkan / dibuang
		T10	Pengendalian vektornya melalui penyemprotan insektisida sejak dini
		T11	Mencabut dan merotasi dengan tanaman jenis lain yang berbeda varietas
		T12	Menjaga kebersihan kebun dan alat pertanian
P06	Kudis (<i>Scab</i>)	T08	Menanam biji yang sehat dan varietas yang tahan (resisten)
		T13	Penyemprotan fungisida ditiokarbamat

Tabel 4. 6 : Tabel Bobot Parameter

Tingkat Gejala	Bobot/ Parameter
Gejala Penting	5
Gejala Sedang	3
Gejala Biasa	1

Tabel 4. 7 : Daftar Bobot Dari Setiap Gejala Penyakit

Kode	Nama Gejala Penyakit	Bobot
G01	Daun berwarna kuning	1
G02	Daun berjamur	3
G03	Bercak kuning cepat menyebar ke seluruh permukaan daun	5
G04	Daun mudah membusuk	3
G05	Terdapat lapisan putih bertepung pada daun dan batang muda	5
G06	Daun yang dilapisi lapisan putih bertepung, berubah menjadi warna kuning dan mengering	3
G07	Daun berwarna kuning atau kecoklatan	3
G08	Tangkai batang berwarna kuning	5
G09	Buah berwarna kuning hingga kehitaman	1
G10	Daun berlubang	3
G11	Daun mengering	1
G12	Daun berwarna hijau tua dan hijau muda	3
G13	Daun layu	3
G14	Tanaman kerdil	5
G15	Buah mengeluarkan cairan bening sampai coklat	5
G16	Buah mudah busuk	5
G17	Buah berjatuhan	3

1.2 Hasil Pemodelan Data

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

Tabel 4. 8 : Matriks Tanaman Penyakit Mentimun

P/G	P01	P02	P03	P04	P05	P06	P07
G01	*			*			
G02	*						
G03	*						
G04	*						
G05		*					
G06		*					
G07			*				
G08			*				
G09			*				
G10				*			
G11				*	*		
G12					*		
G13					*		
G14					*		
G15						*	
G16						*	*
G17						*	*

Tabel 4. 9 : Gejala Yang Dipilih

Gejala yang dipilih		
No	Kode	Nama Gejala
1	G03	Bercak kuning cepat menyebar ke seluruh permukaan daun
2	G05	Terdapat lapisan putih bertepung pada daun dan batang muda
3	G07	Daun berwarna kuning atau kecoklatan
4	G09	Buah berwarna kuning hingga kehitaman
5	G11	Daun mengering
6	G13	Daun layu
7	G16	Buah mudah busuk

Tabel 4. 10 : Konsultasi

Hasil Konsultasi			
No	Kasus	Penyakit	Kecocokan (%)
1	Kasus 3	Antraknosa	61,54
2	Kasus 6	Kudis (<i>Scab</i>)	60,00
3	Kasus 2	Embun Tepung (<i>Powdery Mildew</i>)	37,50
4	Kasus 1	Busuk Daun (<i>Downy Mildew</i>)	35,71
5	Kasus 5	Virus Mosaik	20,00
6	Kasus 4	Bercak Daun	20,00

Hasil Diagnosa:

Penyakit Antraknosa

Penyakit ini disebabkan oleh *Cendawan Colletotrichum Lagenarium Pass* dan mengakibatkan daun berwarna kuning atau kecoklatan, tangkai batang berwarna kuning dan buah berwarna kuning hingga kehitaman.

Solusi:

1. [T06] Benih direndam dalam bantuan berbahan aktif tiram seperti Benlate T 20/20 WP
2. [T07] Penyemprotan dengan Velimex 80 WP atau Difolatan 4 F

4.3 Hasil Pengembangan Sistem

4.3.1 Analisa Sistem

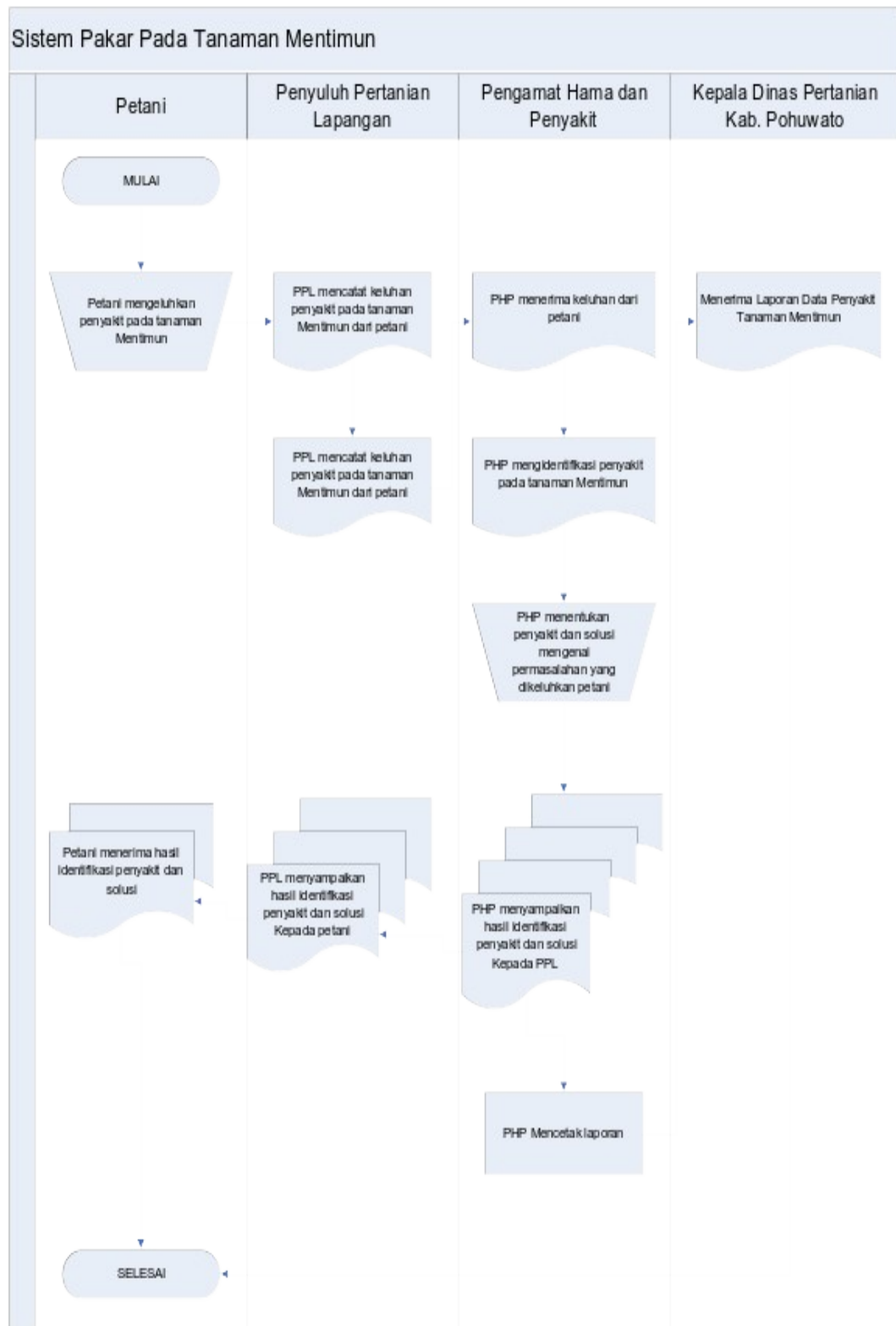
Analisa sistem adalah teknik pemecahan masalah dengan menguraikan sistem ke dalam komponen-komponen pembentuknya untuk bisa mengetahui bagaimana komponen-komponen itu bekerja dan berinteraksi satu sama lain untuk mencapai tujuan mereka.

4.3.1.1 Analisa Sistem Berjalan

Analisa sistem berjalan adalah menganalisa sistem yang sedang berjalan/ sistem lama dalam diagnosa penyakit tanaman mentimun. Sistem berjalan saat ini masih manual sehingga sangat membutuhkan waktu lama karena petani harus bertemu langsung dengan penyuluh kemudian penyuluh menyampaikan konsultasi dari petani ke Pengamat Hama dan Penyakit (PHP) setelah dilakukan pengamatan baru didiagnosa penyakitnya kemudian dilaporkan kepada pejabat yang berwenang, dengan sistem sekarang ini dapat mengakibatkan Tanaman Mentimun lebih mudah untuk mati karena lambatnya penanganan.

Beberapa penyakit tanaman mentimun jika tidak cepat ditangani, akan cepat menyebar melalui angin, sehingga lahan pertanian tanaman mentimun semua akan terjangkit yang menyebabkan gagal panen, akibatnya para petani merugi karena hasil panen yang rusak.

Oleh karena itu, cepatnya penanganan terhadap penyakit pada tanaman mentimun sangatlah penting, agar petani dengan cepat mengetahui masalah dan solusinya yang tepat, agar menjadikan lahan pertanian tanaman mentimun terhindar bahkan dapat diatasi dengan segera ketika gejala penyakit muncul.



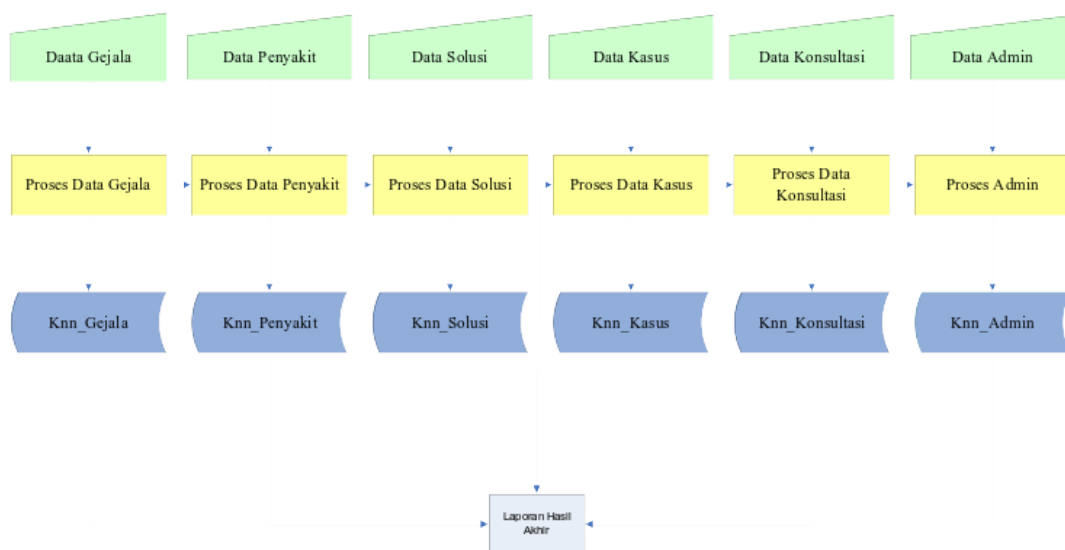
Gambar 4. 2 : Sistem Berjalan

4.3.1.2 Analisa Sistem Yang Diusulkan

Analisa sistem yang diusulkan adalah penguraian dari sistem utuh untuk setiap bagian komponen bertujuan untuk identifikasi dan evaluasi terhadap permasalahan setiap kesempatan dan hambatan yang ditimbulkan serta kebutuhan sehingga dapat di buat perbaikannya.

Setelah dilakukan analisa sistem berjalan atau sistem lama, selanjutnya yaitu menganalisa sistem yang baru. Data-data gejala penyakit yang dibutuhkan untuk pembuatan sistem ini dimasukkan kedalam analisa data sistem untuk mendiagnosa penyakit tanaman Mentimun.

Setelah dilakukan analisa, dapat diketahui sejauh mana sistem meneliti darimana asal data, bagaimana aliran data ke sistem, bagaimana operasi sistem dan hasil akhirnya.

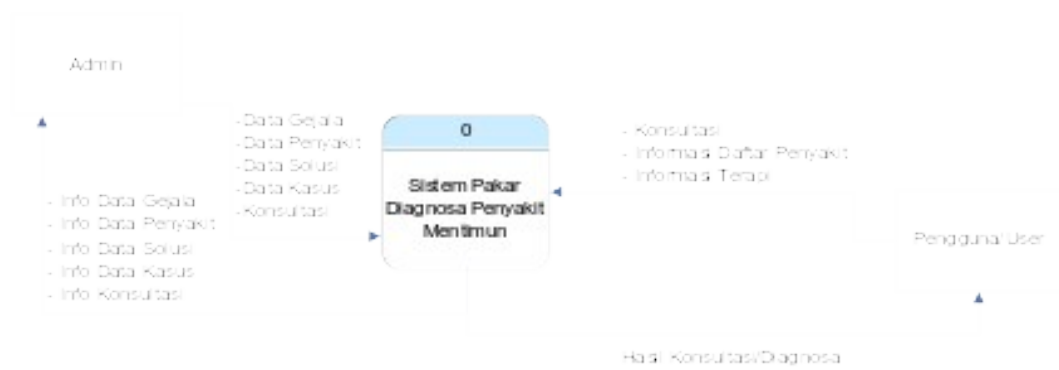


Gambar 4. 3 : Sistem Yang Diusulkan

4.3.2 Diagram Konteks

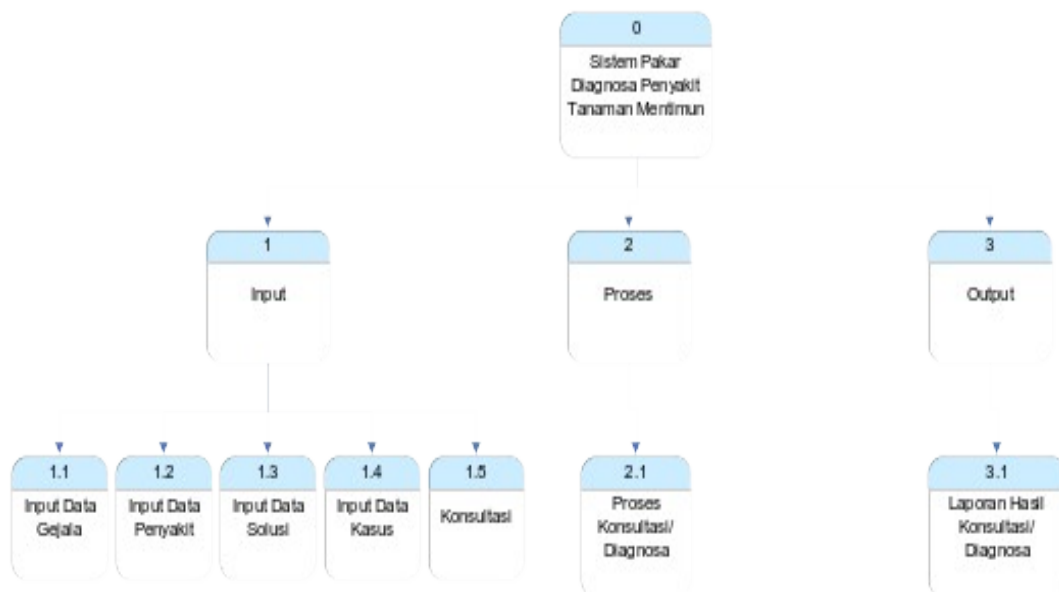
Diagram konteks merupakan level tertinggi dari DFD, yang terdiri dari suatu proses dan menggambarkan ruang lingkup suatu sistem. Diagram konteks terdiri dari dua entitas yaitu Admin dan pengguna. Pengguna bisa langsung konsultasi tanpa melalui admin, sedangkan Admin menginput data-data penyakit, gejala, solusi serta basis pengetahuannya atau kasusnya yang telah diperoleh dari

pakar. Sehingga nanti akan mengeluarkan output kepada pengguna berupa hasil diagnosa.



Gambar 4. 4 : Diagram Konteks

4.3.3 Diagram Berjenjang



Gambar 4. 5 : Diagram Berjenjang

4.3.4 Diagram Arus Data (DAD)

4.3.4.1 Diagram Arus Data (DAD) Level 0



Gambar 4. 6 : Diagram Arus Data Level 0

4.3.4.2 Diagram Arus Data (DAD) Level 1 Proses 1



Gambar 4. 7 : Diagram Arus Data Level 1 Proses 1

4.3.4.3 Diagram Arus Data (DAD) Level 1 Proses 2



Gambar 4. 8 : Diagram Arus Data Level 1 Proses 2

4.3.4.4 Diagram Arus Data (DAD) Level 1 Proses 3



Gambar 4. 9 : Diagram Arus Data Level 1 Proses 3

4.3.5 Kamus Data

Kamus data adalah suatu penjelasan mengenai seluruh data yang berada di dalam database. Kamus data untuk DAD Sistem Pakar Diagnosa Penyakit Tanaman Mentimun yaitu:

Tabel 4. 11 : Kamus Data Pengguna

Nama Database : Mentimun				
Nama Tabel : knn_admin				
No	Field	Type	Size	Keterangan
1	id_admin	Int (Primary)	11	Id Admin
2	nama	Varchar	50	Admin
3	username	Varchar	20	Username Admin
4	password	Varchar	50	Password Admin
5	tanggal	Date	-	Tanggal

Tabel 4. 12 : Kamus Data Gejala

Nama Database : Mentimun				
--------------------------	--	--	--	--

Nama Tabel : knn_gejala				
Fungsi : Tempat Untuk Menyimpan Data Gejala				
No	Field	Type	Size	Keterangan
1	id_gejala	Int (Primary)	11	Id gejala
2	kd_gejala	Varchar	5	Kode gejala
3	nm_gejala	Varchar	250	Nama gejala
4	bobot_parameter	Double	-	-

Tabel 4. 13 : Kamus Data Kasus

Nama Database : Mentimun				
Nama Tabel : knn_kasus				
Fungsi : Untuk Penyimpanan Kasus				
No	Field	Type	Size	Keterangan
1	id_kasus	Int (Primary)	11	Id Kasus
2	nama	Varchar	50	Nama Penyakit
3	id_penyakit	Int	11	Id Penyakit
4	tanggal	Date	-	Tanggal
5	status	Int	1	Status

Tabel 4. 14 : Kamus Data Kasus Gejala

Nama Database : Mentimun				
Nama tabel : knn_kasus_gejala				
Fungsi : Untuk Menyimpan Kasus Gejala				
No	Field	Type	Size	Keterangan
1	id_kasus_gejala	Int (Primary)	11	Id kasus gejala
2	id_kasus	Int	11	Id Kasus
3	id_gejala	Int	11	Id gejala

Tabel 4. 15 : Kamus Data Konsultasi

Nama Database : Mentimun				
Nama tabel : knn_konsultasi				
Fungsi : Tempat Untuk Menyimpan Konsultasi				
No	Field	Type	Size	Keterangan
1	id_konsultasi	Int (Primary)	11	Id konsultasi
2	nama	Varchar	100	Nama pasien
3	tanggal	Datetime	-	Tanggal konsultasi
4	status	Int	1	Status

Tabel 4. 16 : Kamus Data Konsultasi Gejala

Nama Database : Mentimun				
Nama tabel : knn_konsultasi_gejala				
Fungsi : Tempat Untuk Menyimpan Konsultasi Dari Gejala				
No	Field	Type	Size	Keterangan
1	id_konsultasi_gejala	Bigint (Primary)	20	Id konsultasi gejala
2	id_konsultasi	Int	11	Id konsultasi
3	Id_gejala	Int	11	Id gejala
4	Status	int	1	Status

Tabel 4. 17 : Kamus Data Konsultasi Hasil

Nama Database : Mentimun				
Nama tabel : knn_konsultasi_hasil				

Fungsi : Untuk Menyimpan Hasil Konsultasi				
No	Field	Type	Size	Keterangan
1	id_konsultasi_hasil	Bigint (Primary)	20	Id konsultasi hasil
2	id_konsultasi	Int	11	Id konsultasi
3	id_kasus	Int	11	Id kasus
4	id_penyakit	Int	11	Id penyakit
5	nilai	Double	-	Nilai
6	status	Int	1	Status

Tabel 4. 18 : Kamus Data Penyakit

Nama Database : Mentimun				
Nama tabel : knn_penyakit				
Fungsi : Tempat Penyimpanan Data Penyakit				
No	Field	Type	Size	Keterangan
1	id_penyakit	Int (Primary)	11	Id penyakit
2	kd_penyakit	Varchar	5	Kode gejala
3	nm_penyakit	Varchar	50	Nama penyakit
4	definisi	Text	-	Definisi

Tabel 4. 19 : Kamus Data Penyakit Solusi

Nama Database : Mentimun				
Nama tabel : knn_penyakit_solusi				

Fungsi : Tempat Penyimpanan Semua Id Solusi				
No	Field	Type	Size	Keterangan
1	id_penyakit_solusi	Int (Primary)	11	Id penyakit dan solusi
2	id_penyakit	Int	11	Id penyakit
3	id_solusi	Int	11	Id Solusi

Tabel 4. 20 : Kamus Data Solusi

Nama Database : Mentimun				
Nama tabel : knn_solusi				
Fungsi : Untuk Menyimpan Data Solusi				
No	Field	Type	Size	Keterangan
1	id_solusi	Int (Primary)	11	Id_solusi
2	kd_solusi	Varchar	5	Kode solusi
3	nm_solusi	Varchar	100	Nama solusi
4	keterangan	Text	-	Keterangan

4.3.6 Arsitektur Sistem/Kebutuhan *Hardware* dan *Software*

Dalam mengembangkan sistem ini penulis menggunakan pemrograman PHP dan *Database* MySQL. Untuk implementasi sistem membutuhkan perangkat lunak atau perangkat keras dasar yang digunakan untuk menjalankan program aplikasi, diantaranya yaitu:

1. Spesifikasi *Hardware* dan *Software*

- a. Processor Intel Core (TM) i3 - 5010U CPU @ 2.10GHz atau lebih
- b. RAM (*Memory*) 2 GB atau lebih
- c. HDD 500 atau lebih
- d. Monitor SVGA dengan Resolusi 1024 X 768
- e. Dan Peralatan I/O lainnya
- f. Windows 8, Windows 10, atau lebih

g. *Google Chrome* untuk membuka *Web*

h. *Dreamweaver CS4* atau lebih

4.3.7 Desain Secara Umum

4.3.7.1 Desain Output Secara Umum

Untuk : Sistem Pakar Penyakit Tanaman Mentimun

Sistem : Penerapan Metode *K-Nearest Neighbor* (KNN) untuk diagnosa penyakit tanaman Mentimun.

Tahap : Desain Output Secara Umum

Tabel 4. 21 : 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 Konsultasi	Internal/ Eksternal	Tabel	Layar	Monitor	Pengguna, Admin

4.3.7.2 Desain Input Secara Umum

Tujuan dari Desain Input secara umum yaitu untuk memberikan gambaran secara umum kepada user terkait sistem yang baru atau sistem yang diusulkan. Desain sistem secara umum mengidentifikasi komponen-komponen sistem informasi yang akan didesain secara rinci. Desain terinci digunakan untuk pemrograman komputer dan ahli yang mengimplementasikan sistem.

Untuk : Sistem Pakar Penyakit Tanaman Mentimun

Sistem : Penerapan Metode *K-Nearest Neighbor* (KNN) untuk Diagnosa Penyakit Mentimun

Tahap : Desain Input Secara Umum

Tabel 4. 22 : Desain Input Secara Umum

No	Nama	Tipe	Format	Media	Alat	Distribusi
1	Data Admin	Internal	Tabel	Layar	Monitor	Admin
2	Data Kategori Gejala	Internal	Tabel	Layar	Monitor	Admin
3	Data Gejala	Internal	Tabel	Layar	Monitor	Admin

Konsultasi – Hasil Diagnosa

Berikut data diagnosis dari gejala-gejala yang Anda/Pasien pilih.

Nama :

Gejala yang dipilih

No	Kode Gejala	Nama Gejala
1		
2		

Perhitungan

No	Penyakit	Gejala Kasus	Bobot	Gejala Dipilih (Cocok)	Bobot	Kedekatan
1						
2						

Hasil Konsultasi

No	Kasus	Penyakit	Kedekatan
1			



Gambar 4. 10 : Desain Daftar Hasil Konsultasi

4.3.8.2 Desain Input Secara Terinci

1)Input Data Gejala

Berikut adalah gambar desain input gejala yang dibuat untuk menginput data gejala penyakit Tanaman Mentimun ke table knn_gejala dalam *database*.

Tambah Gejala	
Kode Gejala	ex. G01
Nama Gejala	
Bobot Parameter	
Simpan Batal	

Gambar 4. 11 : Desain Input Gejala

2)Input Data Penyakit

Tambah Penyakit	
Kode Penyakit	ex. P01
Nama Penyakit	
Definisi (Informasi)	
Simpan Batal	

Tambah Solusi	
Kode Solusi	
Nama Solusi	
Keterangan	

3)I

ex. T01

Simpan
Batal

Gambar 4. 13 : Desain Input Solusi

4)Input Data Kasus

Tambah Data Kasus	
Nama Kasus	
Pilih Penyakit	-Pilih- v
Status	Aktif

Simpan
Batal

Gambar 4. 14 : Desain Input Kasus

5)Input Konsultasi

Konsultasi	
Nama Anda/Pasien	

Gambar 4. 15 : Desain Input Konsultasi

4.3.8.3 Desain Database Secara Terinci

Tabel 4. 23 : Penyakit

No	Tabel	Tipe Data	Ukuran	Kunci
1	id_penyakit	Int	11	Primary Key
2	kd_penyakit	Varchar	5	
3	nm_penyakit	Varchar	50	
4	definisi	Text	-	

Tabel 4. 24 : Gejala

No	Tabel	Tipe Data	Ukuran	Kunci
1	id_gejala	Int	11	Primary Key
2	kd_gejala	Varchar	5	
3	nm_gejala	Varchar	250	
4	bobot_parameter	Double		

Tabel 4. 25 : Kasus Gejala

No	Tabel	Tipe Data	Ukuran	Kunci
1	id_kasus_gejala	Int	11	Primary Key
2	id_kasus	Int	11	
3	id_gejala	Int	11	

Tabel 4. 26 : Konsultasi

No	Tabel	Tipe Data	Ukuran	Kunci
1	id_konsultasi	Int	11	Primary Key
2	nama	Varchar	100	
3	tanggal	Datetime		
4	status	Int	1	

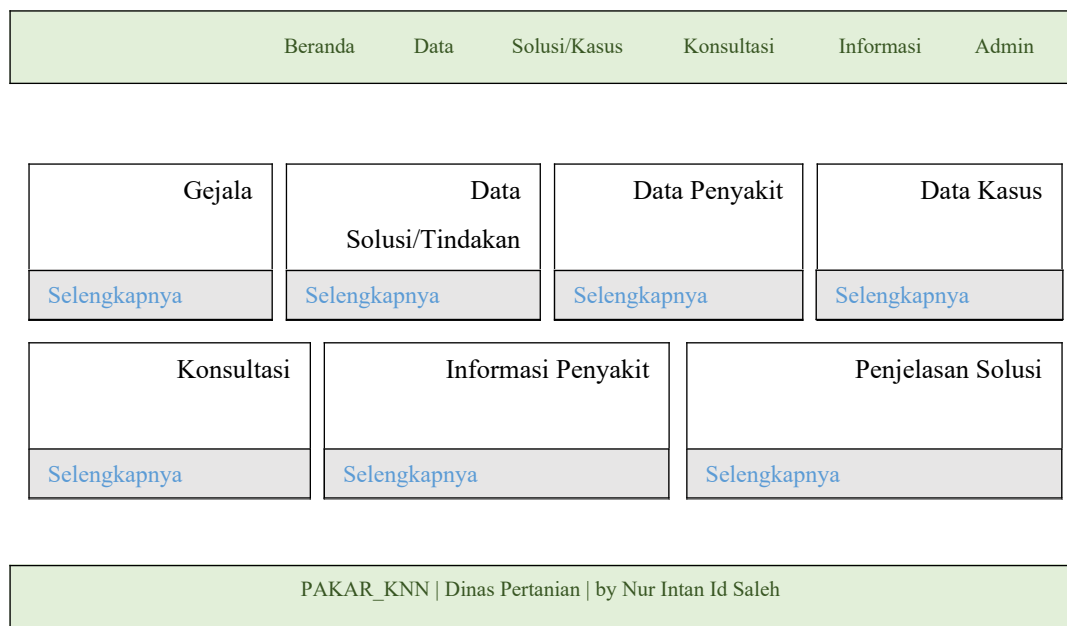
Tabel 4. 27 : Konsultasi Hasil

No	Tabel	Tipe Data	Ukuran	Kunci
1	id_konsultasi_hasil	Bigint	20	Primary Key

SISTEM PAKAR MENTIMUN

Metode K-Nearest Neighbor





Gambar 4. 16 : Desain Menu Utama

4.3.9 Pengujian Sistem

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 *independent* 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.

- Menggunakan struktur data internal untuk menjamin validitas jalur keputusan.

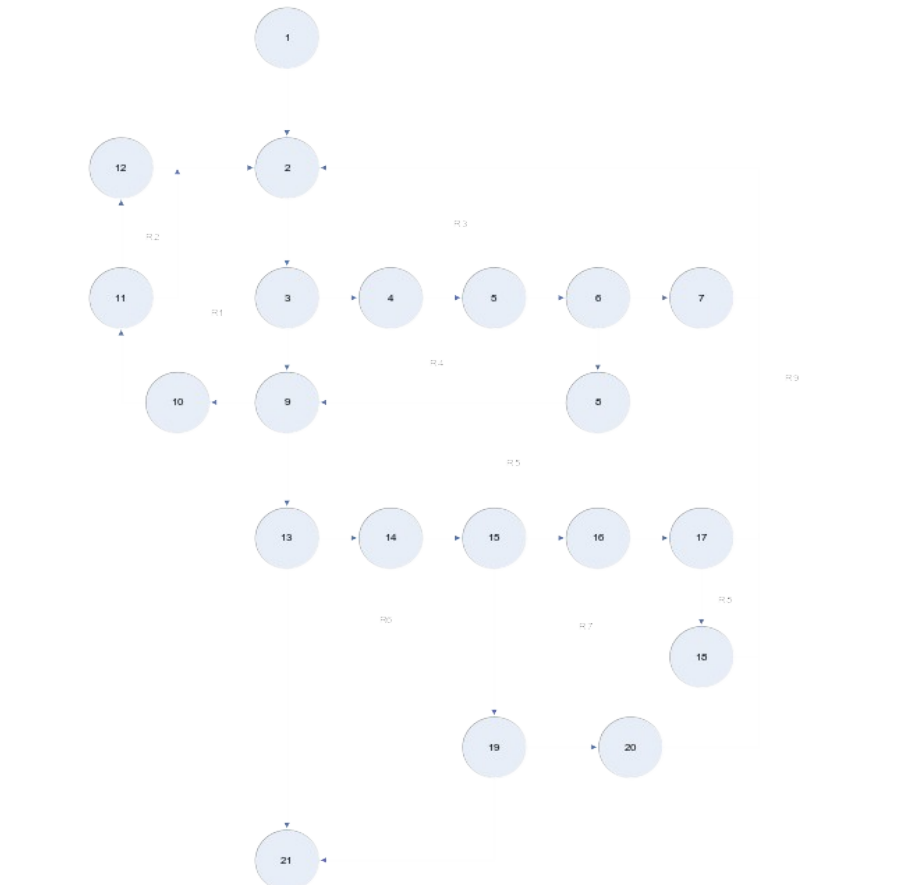
Berikut adalah pengujian *White Box* menggunakan *Flowchart* dan *Flowgraph*. Peneliti menggunakan *Flowchart* Gejala.

a. *Flowchart* Gejala



Gambar 4. 17 : *Flowchart* Gejala

b. *Flowgraph* Gejala



Gambar 4. 18 : *Flowgraph* Gejala

Dari *Flowgraph* di dapatkan

- Region (R) = 9
- Node (N) = 21
- Edge (E) = 28
- Predicate Node (P) = 8

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:

$$\begin{aligned} V(G) &= \text{Edge} - \text{Node} + 2 \\ &= 28 - 21 + 2 \\ &= 9 \end{aligned}$$

Atau

$$\begin{aligned} V(G) &= 8 \text{ Predicate} + 1 \\ &= 9 \end{aligned}$$

Angka 7 dari hasil perhitungan *cyclomatic complexity* menunjukkan jumlah *independent path* dari *basis path testing*, atau dengan kata lain menunjukkan jumlah pengujian yang harus dijalankan untuk memastikan semua *statement* pada program dijalankan minimal sekali (semua *statement* telah diuji).

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

R1 : 1 - 2 - 3 - 9 - 13 - 21

R2 : 1 - 2 - 3 - 9 - 13 - 14 - 15 - 19 - 21

R3 : 1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 9 - 13 - 21

R4 : 1 - 2 - 3 - 9 - 10 - 11 - 2 - 3 - 9 - 13 - 21

R5 : 1 - 2 - 3 - 4 - 5 - 6 - 7 - 2 - 3 - 9 - 13 - 21

R6 : 1 - 2 - 3 - 9 - 10 - 11 - 12 - 2 - 3 - 9 - 13 - 21

R7 : 1 - 2 - 3 - 9 - 13 - 14 - 15 - 16 - 17 - 2 - 3 - 9 - 21

R8 : 1 - 2 - 3 - 9 - 13 - 14 - 15 - 19 - 20 - 2 - 3 - 9 - 13 - 21

R9 : 1 - 2 - 3 - 9 - 13 - 14 - 15 - 16 - 17 - 18 - 2 - 3 - 9 - 13 - 21

Catatan:

- *Independent path* adalah sebuah *path* yang dilalui program yang menunjukkan set baru dari pemrosesan *statement* atau dari sebuah kondisi baru.
- *Independent path* pada *flowgraph* harus melewati sedikitnya satu *edge* yang belum pernah dilewati oleh *path* sebelumnya.
- *Independent path* selalu dimulai dari *node* awal hingga ke *node* akhir.
- *Independent path* yang dibuat pertama kali adalah *independent path* terpendek.

4.3.9.2 Pengujian *Black Box*

Pengujian *Black Box* merupakan pendekatan komplementer dari teknik *White Box*, metode ini digunakan untuk mengetahui apakah perangkat lunak berfungsi dengan benar. dibawah ini adalah pengujian dengan menggunakan metode *Black Box* dari perangkat lunak yang telah dibuat.

Tabel 4. 28 : Pengujian *Black Box*

No	Input/Event	Fungsi	Hasil	Hasil
1	Halaman Utama	Menampilkan Halaman Utama	Halaman Utama Tampil	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 Salah	Sesuai
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 Tambahan 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 Data Solusi/Tindakan	Menampilkan Data Solusi	Data Solusi Tampil	Sesuai
13	Klik Tambah Data Solusi	Menampilkan Tambahan Data Solusi	Tambahan Data Solusi Di Tampilkan	Sesuai
14	Klik Data Kasus	Menampilkan Data Kasus	Data Kasus Tampil	Sesuai
15	Klik Tambah Data Kasus	Menampilkan Tambahan Data Kasus	Tambahan Data Kasus Di Tampilkan	Sesuai
16	Klik Menu Konsultasi	Menampilkan Form Konsultasi	Form Konsultasi Tampil	Sesuai
17	Klik Sub Daftar Konsultasi	Menampilkan Daftar Konsultasi	Daftar Konsultasi Tampil	Sesuai
18	Klik Detail Pada Daftar Konsultasi	Menampilkan Hasil Konsultasi	Hasil Konsultasi Tampil	Sesuai
19	Klik Menu Informasi Penyakit	Menampilkan Informasi Penyakit	Informasi Penyakit Tampil	Sesuai
20	Klik Sub Data	Menampilkan Data Admin	Data Admin	Sesuai

	Admin		Tampil	
21	Klik Sub Ubah Password Pada Menu Administrator	Menampilkan Form Data Ubah Password	Form Data Ubah Password Tampil	Sesuai
22	Klik Sub Logout Pada Menu Administrator	Menampilkan Halaman Utama	Halaman Utama Tampil	Sesuai

BAB V

PEMBAHASAN PENELITIAN

5.1 Pembahasan Model

Tahap pembahasan model, pada tahapan ini akan dilakukan penerapan analisa *K-Nearest Neighbor* dalam melakukan proses pencarian *similarity* terhadap kondisi sebelumnya dengan kondisi yang baru akan didiagnosa.

Rumus :

$$Similarity = \sum_{i=1}^n f \frac{(T_i, S_i) * W_i}{W_i}$$

Penyelesaian:

1. Menentukan Nilai Bobot Kondisi

Tabel 5. 1 : Nilai Kondisi

No	Kondisi	Nilai
1	Kondisi Lama sama dengan Kondisi Baru	1
2	Kondisi Lama tidak sama dengan Kondisi Baru	0

2. Mencari nilai kondisi dengan menghubungkan kasus yang lama dengan kasus yang baru
 - a. Nilai Kondisi dari kasus lama P001 dengan kasus baru P01

Tabel 5. 2 : Nilai P001 Dengan P01

Kode Penyakit	G01	G02	G03	G04
P001	Ya	Ya	Ya	Ya
P01	Tidak	Tidak	Ya	Tidak

Nilai Kondisi	0	0	1	0
Bobot	3	3	5	3

b. Nilai Kondisi dari kasus lama P002 dengan kasus baru P02

Tabel 5. 3 : Nilai P002 Dengan P02

Kode Penyakit	G05	G06
P002	Ya	Ya
P02	Ya	Tidak
Nilai Kondisi	1	0
Bobot	3	5

c. Nilai Kondisi dari kasus lama P003 dengan kasus baru P03

Tabel 5. 4 : Nilai P003 Dengan P03

Kode Penyakit	G07	G08	G09
P003	Ya	Ya	Ya
P03	Ya	Tidak	Ya
Nilai Kondisi	1	0	1
Bobot	3	5	5

d. Nilai Kondisi dari kasus lama P004 dengan kasus baru P04

Tabel 5. 5 : Nilai P004 Dengan P04

Kode	G01	G10	G11
-------------	------------	------------	------------

Penyakit			
P004	Ya	Ya	Ya
P04	Tidak	Tidak	Ya
Nilai Kondisi	0	0	1
Bobot	3	1	1

e. Nilai Kondisi dari kasus lama P005 dengan kasus baru P05

Tabel 5. 6 : Nilai P005 Dengan P05

Kode Penyakit	G11	G12	G13	G14
P005	Ya	Ya	Ya	Ya
P05	Ya	Tidak	Ya	Tidak
Nilai Kondisi	1	0	1	0
Bobot	1	3	1	5

f. Nilai Kondisi dari kasus lama P006 dengan kasus baru P06

Tabel 5. 7 : Nilai P006 Dengan P06

Kode Penyakit	G15	G16	G17
P006	Ya	Ya	Ya
P06	Tidak	Ya	Tidak

Nilai Kondisi	0	1	0
Bobot	1	3	1

3. Menghitung nilai kedekatan kasus baru terhadap kasus lama:

$$ND = \sum \frac{(\text{Nilai}_\text{Kondisi} * \text{Nilai}_\text{Bobot})}{\text{Total}_\text{Nilai}_\text{Bobot}}$$

a. Kasus P001 dengan Kasus P01

$$K1 = \frac{(0*3)+(0*3)+(1*5)+(0*3)}{3+3+5+3}$$

$$= \frac{5}{14} = 0,36 \text{Dijadikan } \% = 0,3571 * 100 = 35,71\%$$

b. Kasus P002 dengan Kasus P02

$$K2 = \frac{(1*3)+(0*5)}{3+5}$$

$$= \frac{3}{8} = 0,38 \text{Dijadikan } \% = 0,375 * 100 = 37,50\%$$

c. Kasus P003 dengan Kasus P03

$$K3 = \frac{(1*3)+(0*5)+(1*5)}{3+5+5}$$

$$= \frac{8}{13} = 0,62 \text{Dijadikan } \% = 0,6153 * 100 = 61,53\%$$

d. Kasus P004 dengan Kasus P04

$$K4 = \frac{(0*3)+(0*1)+(1*1)}{3+1+1}$$

$$= \frac{1}{5} = 0,20 \text{Dijadikan } \% = 0,20 * 100 = 20,00\%$$

e. Kasus P005 dengan Kasus P05

$$K4 = \frac{(1*1)+(0*3)+(1*1)+(0*5)}{1+3+1+5}$$

$$= \frac{2}{10} = 0,20 \text{Dijadikan } \% = 0,20 * 100 = 20,00\%$$

f. Kasus P006 dengan Kasus P06

$$K4 = \frac{(0*1)+(1*3)+(0*1)}{1+3+1}$$

$$= \frac{3}{5} = 0,60 \text{Dijadikan } \% = 0,60 * 100 = 60,00\%$$

4. Mencari nilai kedekatan yang paling tinggi

$$Max(k1...k6) = (0,36), (0,38), (0,62), (0,20), (0,20), (0,60)$$

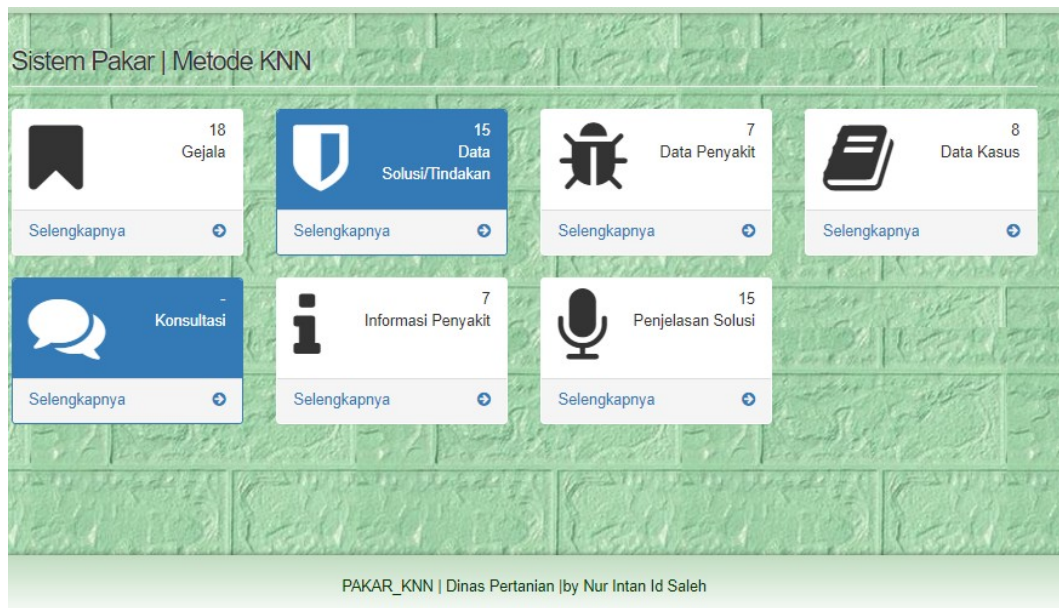
$$= (0,62)$$

5.2 Pembahasan Sistem

Pada tahap ini dilakukan penerapan hasil perancangan antarmuka kedalam sistem yang dibangun dengan menggunakan perangkat lunak yang telah di paparkan pada sub bab implementasi perangkat lunak.

5.2.1 Halaman Menu Utama





Gambar 5. 1 : Halaman Menu Utama

Pada tampilan menu utama terdapat 6 menu yang digunakan pada sistem, yaitu menu Beranda, Data, Solusi/Kasus, Konsultasi, Informasi, dan Admin. Sedangkan pada tampilan dibawahnya terdapat link untuk mempermudah cara cepat untuk menampilkan gejala, penyakit, solusi, dan sebagainya.

5.2.2 Halaman Menu Gejala

Daftar Gejala

No	Kode	Nama Gejala	Bobot	+ Tambah
	G01	Daun berwarna kuning	3	Ubah Hapus
	G02	Daun berjamur	3	Ubah Hapus
	G03	Bercak kuning cepat menyebar ke seluruh permukaan daun	5	Ubah Hapus
	G04	Daun mudah membusuk	3	Ubah Hapus
	G05	Terdapat lapisan putih bertepung pada daun dan batang muda	3	Ubah Hapus
	G06	Daun yang dilapisi lapisan putih bertepung, berubah menjadi warna kuning dan mengering	5	Ubah Hapus
	G07	Daun berwarna kuning atau kecoklatan	3	Ubah Hapus
	G08	Tangkai batang berwarna kuning	5	Ubah Hapus
	G09	Buah berwarna kuning hingga kehitaman	5	Ubah Hapus
	G10	Daun berlubang	1	Ubah Hapus
	G11	Daun mengering	1	Ubah Hapus
	G12	Daun berwarna hijau tua dan hijau muda	3	Ubah Hapus
	G13	Daun layu	1	Ubah Hapus
	G14	Tanaman kerdil	5	Ubah Hapus
	G15	Buah mengeluarkan cairan bening sampai coklat	1	Ubah Hapus
	G16	Buah mudah busuk	3	Ubah Hapus
	G17	Buah berjatuhan	1	Ubah Hapus

Gambar 5. 2 : Halaman Menu Gejala

Pada tampilan menu gejala, digunakan untuk melihat data gejala-gejala pada tanaman Mentimun. Pada pada form ini juga digunakan untuk menginput/menambah, mengubah dan menghapus gejala.

5.2.3 Halaman Menu Konsultasi

Daftar Konsultasi

No	Tanggal	Nama	Penyakit	Kedekatan	
	10 Feb 2022	Konsultasi 1	[P03] Antraknosa	0.61538461538462	Detail Hapus

Gambar 5. 3 : Halaman Menu Daftar Konsultasi

Pada tampilan menu konsultasi, tampilan di atas merupakan form untuk daftar konsultasi, maksudnya setiap user yang melakukan konsultasi datanya yang disimpan bisa dilihat di halaman ini. Dalam halaman ini admin juga dapat melihat detail gejala-gejala dan penyakit yang dialami tanaman Mentimun, d

halaman ini admin juga dapat menghapus data user yang telah melakukan konsultasi tanaman Mentimun.

5.2.4 Halaman Menu Data Penyakit

Daftar Penyakit

No	Kode	Nama Penyakit	Solusi	+ Tambah
	P01	Busuk Daun (Downy Mildew)	[T01] Tanaman yang terserang agar segera dikeluarkan / dibuang ✖ [T02] Penyemprotan fungisida Velimex 80 WP ✖ [T03] Memperbaiki drainase tanah ✖ [T04] Mengatur jarak tanam ✖	☒ Ubah ✖ Hapus
	P02	Embus Tepung (Powdery Mildew)	[T01] Tanaman yang terserang agar segera dikeluarkan / dibuang ✖ [T05] Penyemprotan fungisida yang mengandung bahan aktif benomy atau karbendazim ✖	☒ Ubah ✖ Hapus
	P03	Antraknosa	[T06] Benih direndam dalam bantuan berbahan aktif tiram seperti Benlate T 20/20 WP ✖ [T07] Penyemprotan dengan Velimex 80 WP atau Difolatan 4 F ✖	☒ Ubah ✖ Hapus
	P04	Bercak Daun	[T08] Menanam biji yang sehat dan varietas yang tahan (resisten) ✖ [T09] Penyemprotan bakterisida yang mengandung bahan aktif streptomisin atau oksitetrasiklin ✖	☒ Ubah ✖ Hapus
	P05	Virus Mosaik	[T01] Tanaman yang terserang agar segera dikeluarkan / dibuang ✖ [T10] Pengendalian vektornya melalui penyemprotan insektisida sejak dini ✖ [T11] Mencabut dan merotasi dengan tanaman jenis lain yang berbeda varietas ✖ [T12] Menjaga kebersihan kebun dan alat pertanian ✖	☒ Ubah ✖ Hapus
	P06	Kudis (Scab)	[T08] Menanam biji yang sehat dan varietas yang tahan (resisten) ✖ [T13] Penyemprotan fungisida difitokarbamat ✖	☒ Ubah ✖ Hapus

Gambar 5. 4 : Halaman Menu Data Penyakit

Pada halaman ini admin bisa juga melihat data-data penyakit yang sering menimpa tanaman Mentimun, Admin juga dapat menambah, mengubah dan menghapus data penyakit.

5.2.5 Halaman Menu Data Solusi

Daftar Solusi/Tindakan

No	Kode	Nama Solusi	+ Tambah	
T01		Tanaman yang terserang agar segera dikeluarkan / dibuang	Ubah	Hapus
T02		Penyemprotan fungisida Velimex 80 WP	Ubah	Hapus
T03		Memperbaiki drainase tanah	Ubah	Hapus
T04		Mengatur jarak tanam	Ubah	Hapus
T05		Penyemprotan fungisida yang mengandung bahan aktif benomy atau karbendazim	Ubah	Hapus
T06		Benih direndam dalam bantuan berbahan aktif tiram seperti Benlate T 20/20 WP	Ubah	Hapus
T07		Penyemprotan dengan Velimex 80 WP atau Difolatan 4 F	Ubah	Hapus
T08		Menanam biji yang sehat dan varietas yang tahan (resisten)	Ubah	Hapus
T09		Penyemprotan bakterisida yang mengandung bahan aktif streptomisin atau oksitetrasiklin	Ubah	Hapus
T10		Pengendalian vektornya melalui penyemprotan insektisida sejak dini	Ubah	Hapus
T11		Mencabut dan merotasi dengan tanaman jenis lain yang berbeda varietas	Ubah	Hapus
T12		Menjaga kebersihan kebun dan alat pertanian	Ubah	Hapus
T13		Penyemprotan fungisida difitokarbamat	Ubah	Hapus

Gambar 5. 5 : Halaman Menu Data Solusi

Pada halaman ini, admin bisa menambah dan menghapus solusi untuk setiap penyakit. Jadi untuk solusi yang sudah ditambahkan nanti akan muncul pada saat konsultasi, tergantung dari penyakit yang dialami tanaman Mentimun.

5.2.6 Halaman Menu Data Kasus

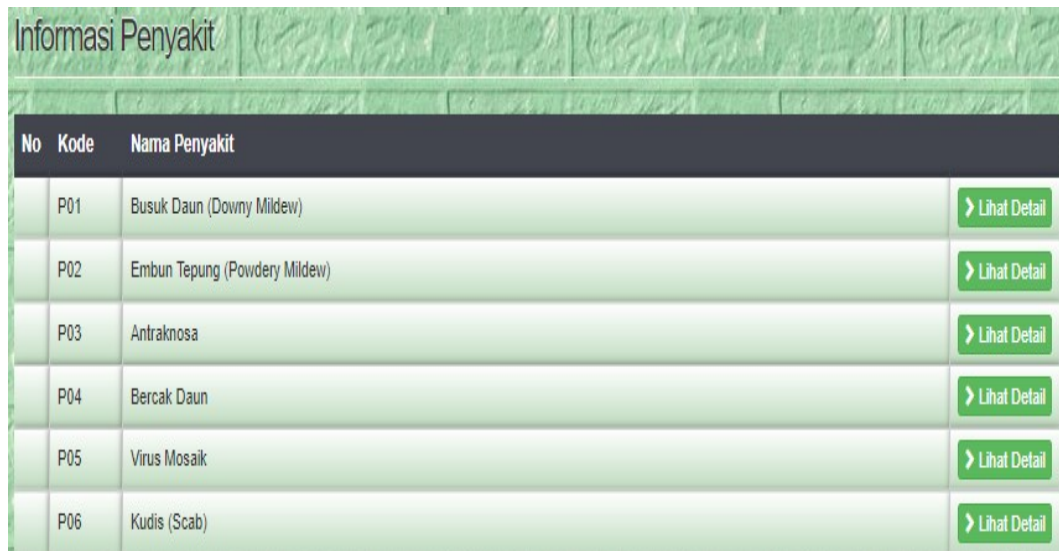
Daftar Kasus

No	Nama	Penyakit	Gejala	Tanggal	Status	+ Tambah
Kasus 1		[P01] Busuk Daun (Downy Mildew)	[G01] Daun berwarna kuning ✖ [G02] Daun berjamur ✖ [G03] Bercak kuning cepat menyebar ke seluruh permukaan daun ✖ [G04] Daun mudah membusuk ✖	28/01/2022	Aktif	Ubah ✖ Hapus
Kasus 2		[P02] Embun Tepung (Powdery Mildew)	[G05] Terdapat lapisan putih bertepung pada daun dan batang muda ✖ [G06] Daun yang dilapisi lapisan putih bertepung, berubah menjadi warna kuning dan mengering ✖	28/01/2022	Aktif	Ubah ✖ Hapus
Kasus 3		[P03] Antraknosa	[G07] Daun berwarna kuning atau kecoklatan ✖ [G08] Tangkai batang berwarna kuning ✖ [G09] Buah berwarna kuning hingga kehitaman ✖	28/01/2022	Aktif	Ubah ✖ Hapus
Kasus 4		[P04] Bercak Daun	[G01] Daun berwarna kuning ✖ [G10] Daun berlubang ✖ [G11] Daun mengering ✖	28/01/2022	Aktif	Ubah ✖ Hapus
Kasus 5		[P05] Virus Mosaik	[G11] Daun mengering ✖ [G12] Daun berwarna hijau tua dan hijau muda ✖ [G13] Daun layu ✖ [G14] Tanaman kerdil ✖	28/01/2022	Aktif	Ubah ✖ Hapus
Kasus 6		[P06] Kudis (Scab)	[G15] Buah mengeluarkan cairan bening sampai coklat ✖ [G16] Buah mudah busuk ✖ [G17] Buah berjatuhan ✖	28/01/2022	Aktif	Ubah ✖ Hapus

Gambar 5. 6 : Halaman Menu Data Kasus

Pada halaman ini, admin bisa menambah kasus untuk setiap penyakit. Maka dari itu kasus yang ditambahkan nantinya bisa muncul pada hasil konsultasi, kembali lagi tergantung dari penyakit yang dialami tanaman mentimun. Dan pada halaman ini juga Admin bisa menambah dan menghapus data.

5.2.7 Halaman Menu Informasi Daftar Penyakit



No	Kode	Nama Penyakit	
	P01	Busuk Daun (Downy Mildew)	Lihat Detail
	P02	Embun Tepung (Powdery Mildew)	Lihat Detail
	P03	Antraknosa	Lihat Detail
	P04	Bercak Daun	Lihat Detail
	P05	Virus Mosaik	Lihat Detail
	P06	Kudis (Scab)	Lihat Detail

Gambar 5. 7 : Halaman Menu Informasi Daftar Penyakit

Pada halaman ini, Admin bisa melihat secara detail setiap penyakit serta solusi yang ditampilkan.

BAB VI

KESIMPULAN DAN SARAN

6.1 Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan dan pembahasan yang telah diuraikan sebelumnya, maka dapat ditarik kesimpulan bahwa:

1. Dapat diketahui cara merekayasa Sistem Pakar Diagnosa Penyakit Tanaman Mentimun Dengan Metode *K-Nearest Neighbor*.
2. Sistem pakar ini dapat diimplementasikan untuk Mendiagnosa Penyakit Tanaman Mentimun. Hal ini dapat dibuktikan dengan hasil pengujian sistem menggunakan pengujian *White Box Testing* dan *Basis Path Testing* yang menghasilkan nilai *Cyclomatic Complexity* = 9.

6.2 Saran

Penulis akan menyampaikan beberapa saran atau masukan bagi siapa saja yang ingin memperbaiki atau menyempurnakan dari Sistem Pakar ini yaitu:

1. Sistem Pakar ini masih belum memberikan keakuratan data sebab metode perhitungan ketidak pastian yang digunakan hanya satu.
2. Gunakan metode lain untuk menjadi bahan perbandingan
3. Pada sistem pakar ini penulis hanya mengambil data penyakit pada tanaman mentimun, untuk penulis lain dapat menambahkan data hama.

DAFTAR PUSTAKA

- [1] L. Pasaribu, "Sistem Pakar Mendiagnosa Hama Dan Penyakit Tanaman Mentimun Menggunakan Metode Naïve Bayes," *Pelita Inform. Inf. dan ...*, vol. 7, no. April, 2019.
- [2] D. PRATIWI, *Sistem Pakar Pendiagnosa Penyakit Pada Tanaman Mentimun Berbasis Android Forward Chaining*. repository.uncp.ac.id, 2020.
- [3] suhendro Suhendro and N. Sevani, "Penerapan Metode Dempster-Shafer untuk Deteksi Penyakit Demam Berdarah dan Tipus," *Tek. dan Ilmu Komput.*, vol. 7, no. 28, pp. 459–471, 2018.
- [4] M. Kafil, "PENERAPAN METODE K-NEAREST NEIGHBORS UNTUK PREDIKSI PENJUALAN BERBASIS WEB PADA BOUTIQ DEALOVE BONDOWOSO," *JATI (Jurnal Mhs. Tek. Inform.*, vol. 3, no. 2, 2019, doi: 10.36040/jati.v3i2.860.
- [5] M. Dahria and R. Gunawan, "Penerapan Metode Dempster Shafer Mendiagnosa Penyakit Mentimun," *J. Teknol. Inform.*, vol. 2, no. 1, 2019.
- [6] F. D. Wahyudi, P. Harsadi, and D. Remawati, "IMPLEMENTASI METODE K-NN UNTUK DETEKSI KERUSAKAN MESIN BUBUT," *Jurnal TIKomSiN*, vol. 6, no. 2, 2018.
- [7] A. Arifsyah and A. Sindar, "Sistem Pakar Diagnosa Penyakit Pohon Karet Dengan Metode Certainty Factor," *J. Nas. Komputasi dan Teknol. Inf.*, vol. 2, no. 2, 2019, doi: 10.32672/jnkti.v2i2.1568.
- [8] B. H. Hayadi, "Sistem Pakar: Penyelesaian Kasus Menentukan Minat Baca, Kecenderungan, dan Karakter Siswa Dengan Metode Forward Chaining," Yogyakarta: Deepublish, 2018.
- [9] D. Kurnianingtyas, B. A. Rahardian, D. P. Mahardika, A. K. A., and D. A. K., "Sistem Pendukung Keputusan Diagnosis Penyakit Sapi Potong Menggunakan K-Nearest Neighbour (K-NN)," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 4, no. 2, 2017, doi: 10.25126/jtiik.201742308.
- [10] T. Rismawan, A. W. Irawan, W. Prabowo, and S. Kusumadewi, "SISTEM PENDUKUNG KEPUTUSAN BERBASIS POCKET PC SEBAGAI PENENTU STATUS GIZI MENGGUNAKAN METODE KNN (K-NEAREST NEIGHBOR)," *J. Teknoin*, vol. 13, no. 2, 2008.
- [11] A. N. Rahmi, I. Vewawati, and M. Kurniasih, "Sistem pakar diagnosa penyakit dan hama pada tanaman mentimun menggunakan metode forward chaining," 2019.

- [12] W. Nurwijayo, 2020, *Cara Ampuh Mengendalikan Penyakit Embun Bulu Pada Tanaman*. Diambil kembali dari gdmorganic: <http://gdmorganic.com/embun-bulu/>
- [13] F. S. Lamadi, 2020, *MENGENAL PENYAKIT EMBUN PADA TANAMAN* Diambil kembali dari dppk.fakfakkab: <http://dppk.fakfakkab.go.id/?p=406>
- [14] T. Sanne, 2018, *Penyakit Antraknosa Ketimun - Penanganan Antraknosa Pada Tanaman Ketimun*. Diambil kembali dari id.les-jardins-de-sanne: http://id.les-jardins-de-sanne.com/penyakit-antraknosa-ketimun-penanganan-antraknosa/#google_vignette
- [15] C. Jayanto, 2020, *Ini Jenis Hama dan Penyakit yang Sering Merusak Tanaman Mentimun*. Diambil kembali dari meditani.co: <http://meditani.co/ini-jenis-hama-dan-penyakit-yang-sering-merusak-tanaman-mentimun/amp/>
- [16] F.G Countryside, 2021, *Bagaimana menangani penyakit timun*. Diambil kembali dari ms.farmforage: <http://ms.farmforage.com/1060-how-to-deal-with-diseases-of-cucumbers.html>
- [17] E. Ali, “REKAYASA PERANGKAT LUNAK,” Yogyakarta: CV MFA, 2019.
- [18] E. Billah, 2019, *Pengertian dan Tahap Metode SDLC Waterfall*. Diambil kembali dari medium.com: <http://medium.com/@ersandibillah03/sdlc-waterfall-3a3c893be77b>
- [19] R. A. Sukanto and M. Salahuddin, “Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek,” Bandung: Informatika, 2016.
- [20] P. Yunita, “Aplikasi perhitungan payroll dosen pada stmik dumi,” *J. Inf. Manajemen dan Komput.*, vol. 10, no. 1, 2018.
- [21] A. Kadir, “Dasar Logika Pemrograman Komputer,” Jakarta: PT. Elex Media Komputindo, 2017.
- [22] R. Sulaehani, “Sistem Pakar Diagnosa Penyakit Tanaman Kakao Menggunakan Metode Case Base Reasoning (Cbr) Pada Kelompok Tani Gapoktan Desa Makarti Jaya,” *Simtek J. Sist. Inf. dan Tek. Komput.*, vol. 4, no. 1, pp. 74–83, 2019, doi: 10.51876/simtek.v4i1.51.
- [23] A. Andelheid and K. Nst, “Buku Pintar Menguasai PHP MySQL,” Jakarta: Mediakita, 2012.
- [24] R. Abdulloh, “Easy & Simple Web Programming,” Jakarta: PT Elex Media Komputindo, 2016.
- [25] R. Mandar, “Solusi Tepat Menjadi Pakar Adobe Dreamweaver CS6,” Jakarta: PT Elex Media Komputindo, 2017.

- [26] W. Komputer, "Buku Latihan Teknik Mengolah Foto Digital dengan Adobe Photoshop CS3," Jakarta: PT Elex Media Komputindo, 2007.

LAMPIRAN

1. Listing Program

1.1 Listing Program Index

```
<?php
include "config/library.php";
include "config/koneksi.php";
include "config/fungsi_indotgl.php";
$mod=antiinjec($koneksi, @$_GET['mod']);
$sid_admin=antiinjec($koneksi,
@$_SESSION['ses_admwsadevuid']);
$stipe=0;
if($sid_admin!="") {
$sql = "SELECT id_admin, nama, username, password
FROM knn_admin WHERE id_admin='$sid_admin'";
$result = $koneksi->query($sql);
$d_admin=$result->fetch_assoc();
$stipe=1;
}
?>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html
xmlns="http://www.w3.org/1999/xhtml"
xml:lang="en">

<head>
<title>SISTEM PAKAR</title>
<!-- jQuery -->
```

```

<script
src="bower_components/jquery/dist/jquery.min.js"></scri
pt>
<!-- Bootstrap Core JavaScript -->
<script
src="bower_components/bootstrap/dist/js/bootstrap.min.js
"></script>
<!-- Metis Menu Plugin JavaScript -->
<script
src="bower_components/metisMenu/dist/metisMenu.min.js"
"></script>
<!-- DataTables JavaScript -->
<script
src="bower_components/datatables/media/js/jquery.dataT
ables.min.js"></script>
<script
src="bower_components/datatables-plugins/integration/b
ootstrap/3/dataTables.bootstrap.min.js"></script>

<!-- Custom Theme JavaScript -->
<script src="dist/js/sb-admin-2.js"></script>
<meta name="description" content="free website
template" />
<meta name="keywords" content="enter your keywords
here" />
<meta http-equiv="content-type" content="text/html;
charset=utf-8" />
<meta http-equiv="X-UA-Compatible" content="IE=9" />
<link rel="stylesheet" type="text/css"
href="css/style.css" />
<script type="text/javascript"
src="js/jquery.min.js"></script>
<script type="text/javascript"
src="js/image_slide.js"></script>
<script type="text/javascript" src="js/jquery.easing-
sooper.js"></script>
<script type="text/javascript"
src="js/jquery.sooperfish.js"></script>

```

```
<script type="text/javascript">
```

```
$(document).ready(function() {  
    $('ul.sf-menu').sooperfish();  
});
```

```
</script>
```

```
<link          rel="icon"          type="image/x-icon"  
href="dist/img/admin_page.png" />
```

```
<!-- Bootstrap Core CSS -->
```

```
<link  
href="bower_components/bootstrap/dist/css/bootstrap.min.css" rel="stylesheet">
```

```
<!-- MetisMenu CSS -->
```

```
<link  
href="bower_components/metisMenu/dist/metisMenu.min.css" rel="stylesheet">
```

```
<!-- DataTables CSS -->
```

```
<link  
href="bower_components/datatables-plugins/integration/bootstrap/3/dataTables.bootstrap.css" rel="stylesheet">
```

```
<!-- DataTables Responsive CSS -->
```

```
<link  
href="bower_components/datatables-responsive/css/dataTables.responsive.css" rel="stylesheet">
```

```
<!-- Custom CSS -->
```

```
<link href="dist/css/tabel.css" rel="stylesheet">
```

```
<!-- Morris Charts CSS -->
```

```
<!--<link href="../bower_components/morrisjs/morris.css" rel="stylesheet">-->
```

```
<!-- Custom Fonts -->
```

```
<link href="bower_components/font-awesome/css/font-awesome.min.css" rel="stylesheet" type="text/css">
```

```
<script          type="text/javascript"  
src="js/jquery.form.min.js"></script>
```

```

<!-- HTML5 Shim and Respond.js IE8 support of HTML5
elements and media queries -->
<!-- WARNING: Respond.js doesn't work if you view the
page via file:// -->
<!--[if lt IE 9]>
<script
src="https://oss.maxcdn.com/libs/html5shiv/3.7.0/html5sh
iv.js"></script>
<script
src="https://oss.maxcdn.com/libs/respond.js/1.4.2/respon
d.min.js"></script>
<![endif]-->
<script                                type="text/javascript"
src="tinymce/tinymce.min.js"></script>
<!-- place in header of your html document -->
<script>
tinymce.init({
selector: "textarea.info",
theme: "modern",
    relative_urls: false,
height: 280,
plugins: [
    "advlist autolink link image lists charmap print
preview hr anchor pagebreak spellchecker",
    "searchreplace wordcount visualblocks
visualchars code fullscreen insertdatetime media
nonbreaking",
    "save table contextmenu directionality emoticons
template paste textcolor responsivefilemanager"
],
    content_css: "css/content.css",
    toolbar1: "insertfile undo redo | styleselect |
fontselect | fontsizeselect | bold italic underline | alignleft
aligncenter alignright alignjustify | bullist numlist outdent
indent | link unlink image | print preview media fullpage |
forecolor backcolor emoticons | responsivefilemanager ",
    image_advtab: true ,

```

```

        style_formats: [
            {title: 'Bold text', inline: 'b'},
            {title: 'Red text', inline: 'span', styles: {color:
'ff0000'}},
            {title: 'Red header', block: 'h1', styles: {color:
'ff0000'}},
            {title: 'Example 1', inline: 'span', classes:
'example1'},
            {title: 'Example 2', inline: 'span', classes:
'example2'},
            {title: 'Table styles'},
            {title: 'Table row 1', selector: 'tr', classes:
'tablerow1'}
        ]
    });
</script>
</head>

```

```

<body>
<div id="main">
<div id="dark_container">
<div id="banner">
<div id="welcome">
<h1>SISTEM PAKAR MENTIMUN</h1>
</div><!--close welcome-->
<div id="welcome_slogan">
<h1>Metode K-Nearest Neighbor</h1>
</div><!--close welcome_slogan-->
<div id="slideshow">
    <ul class="slideshow">
<li class="show"></li>
<li></li>

```



```

<li></li>
</ul>
<ul class="sf-menu" id="nav">
<?php if($sid_admin!="") { ?>
<li class="dropdown">
<a class="dropdown-toggle" data-toggle="dropdown"
href="#">
<i class="fa fa-user fa-fw"></i>&nbsp;<?php echo
@$d_admin['nama']; ?>&nbsp;&nbsp;&nbsp;<i class="fa fa-
caret-down"></i>
</a>
<ul class="dropdown-menu dropdown-user">
<li><a href="?mod=ubah-password"><i class="fa fa-key
fa-fw"></i> Ubah Password</a>
</li>
<li class="divider"></li>
<li><a href="logout.php"><i class="fa fa-sign-out fa-
fw"></i> Logout</a>
</li>
</ul>
<!-- /.dropdown-user -->
</li>
<?php } else { ?>
<li class="dropdown">
<a class="dropdown-toggle" href="login.php">
<i class="fa fa-user fa-fw"></i>&nbsp;&nbsp;&nbsp;Login Admin</i>
</a>
</li>
<?php } ?>
</div><!--close slideshow-->
</div><!--close banner-->

</div><!--close dark_container-->

```

```
<div id="menu_container">
<div id="menubar">
<ul class="sf-menu" id="nav">
<li>
<a href="?mod=beranda"> Beranda</a>
</li>
<?php if($stipe==1) { ?>
<li><a href="#">Data</a>
<ul>
<li>
<a href="?mod=gejala"> Data Gejala</a>
</li>
<li>
<a href="?mod=penyakit"> Data Penyakit</a>
</li></ul></li>
<li><a href="#">Solusi/Kasus</a>
<ul>
<li>
<a href="?mod=solusi"> Data Solusi</a>
</li>

<li>
<a href="?mod=kasus"> Data Kasus</a>
</li></ul></li>
<li><a href="#">Konsultasi</a>
<ul>
<li>
<a href="?mod=daftar-konsultasi">          Daftar
Konsultasi</a>
</li>
<li>
<a href="?mod=konsultasi"> Konsultasi</a>
</li></ul></li>
```

```

<?php } ?>
<li>
<a href="#"> Informasi</a>
<ul>
<li><a href="?mod=info-penyakit">Daftar
Penyakit</a></li>
<li><a href="?mod=info-terapi">Terapi
Pengobatan</a></li>
</ul>
</li>

<li>
<a href="#"> Admin</a>
<ul>
<li><a href="?mod=admin">Data Admin</a></li>
</ul>
</li>
</ul>
</div><!--close menubar-->
</div><!--close menu_container-->

```

```

<div id="site_content">
<?php
    if($mod==" " || $mod=="beranda")
{ include "pages/modul/home.php"; }
    elseif($mod=="gejala" && $stipe==1)
{ include "pages/modul/mod_pakar/lst_gejala.php"; }
    elseif($mod=="gejala-input" &&
$stipe==1) { include
"pages/modul/mod_pakar/inp_gejala.php"; }
    elseif($mod=="gejala-kategori" &&
$stipe==1) { include
"pages/modul/mod_pakar/lst_gejala_kategori.php"; }

```

```

elseif($mod=="gejala-kategori-input" &&
$stipe==1) { include
"pages/modul/mod_pakar/inp_gejala_kategori.php"; }
elseif($mod=="solusi" && $stipe==1)
{ include "pages/modul/mod_pakar/lst_solusi.php"; }
elseif($mod=="solusi-input" &&
$stipe==1) { include
"pages/modul/mod_pakar/inp_solusi.php"; }
elseif($mod=="penyakit" && $stipe==1)
{ include "pages/modul/mod_pakar/lst_penyakit.php"; }
elseif($mod=="penyakit-input" &&
$stipe==1) { include
"pages/modul/mod_pakar/inp_penyakit.php"; }
elseif($mod=="penyakit-solusi-input" &&
$stipe==1) { include
"pages/modul/mod_pakar/inp_penyakit_solusi.php"; }
elseif($mod=="kasus" && $stipe==1)
{ include "pages/modul/mod_pakar/lst_kasus.php"; }
elseif($mod=="kasus-input" &&
$stipe==1) { include
"pages/modul/mod_pakar/inp_kasus.php"; }
elseif($mod=="kasus-gejala-input" &&
$stipe==1) { include
"pages/modul/mod_pakar/inp_kasus_gejala.php"; }

elseif($mod=="konsultasi") { include
"pages/modul/mod_konsultasi/konsultasi_awal.php"; }
elseif($mod=="konsultasi-gejala")
{ include
"pages/modul/mod_konsultasi/konsultasi_gejala.php"; }
elseif($mod=="konsultasi-hasil")
{ include
"pages/modul/mod_konsultasi/konsultasi_hasil.php"; }
elseif($mod=="konsultasi-selesai")
{ include
"pages/modul/mod_konsultasi/konsultasi_selesai.php"; }
elseif($mod=="daftar-konsultasi")
{ include
"pages/modul/mod_konsultasi/konsultasi_list.php"; }

```

```

elseif($mod=="daftar-konsultasi-view") {
include
"pages/modul/mod_konsultasi/konsultasi_list_view.php"; }

```

```

elseif($mod=="info-penyakit") { include
"pages/modul/mod_informasi/info_penyakit.php"; }
elseif($mod=="info-penyakit-view")
{
include
"pages/modul/mod_informasi/info_penyakit_view.php"; }
elseif($mod=="info-terapi") { include
"pages/modul/mod_informasi/info_solusi.php"; }
elseif($mod=="info-terapi-view")
{
include
"pages/modul/mod_informasi/info_solusi_view.php"; }

```

```

//Pengaturan
elseif($mod=="admin" && $stipe==1)
{
include
"pages/modul/mod_pengguna/lst_pengguna.php"; }
elseif($mod=="admin-input" &&
$stipe==1) {
include
"pages/modul/mod_pengguna/inp_pengguna.php"; }

```

```

//User
elseif($mod=="ubah-password")
{
include
"pages/modul/mod_pengguna/z_user_password.php"; }

```

```

?>

```

```

<div class="sidebar_container">
    <div class="sidebar"><!--close sidebar_item-->
</div><!--close sidebar-->
    <div class="sidebar"><!--close sidebar_item-->
</div><!--close sidebar-->
    <div class="sidebar"><!--close sidebar_item-->
</div><!--close sidebar-->

```

```

        <div class="sidebar"><!--close sidebar_item-->
</div><!--close sidebar-->
<div class="sidebar"><!--close sidebar_item-->
</div><!--close sidebar-->
</div><!--close sidebar_container-->

<div id="content">
<div class="content_item">

        <div                class="content_container"><!--close
button_small-->
        </div><!--close content_container-->
<div class="content_container"><!--close button_small--
>
        </div><!--close content_container-->
        </div><!--close content_item-->
</div><!--close content-->
</div><!--close site_content-->

</div><!--close main-->

<div id="footer_container">
<div id="footer">
<a                href="http://validator.w3.org/check?
uri=referer">PAKAR_KNN</a>                |                <a
href="http://fotogrph.com/">Dinas Pertanian</a> |by <a
href="http://www.araynordesign.co.uk">Nur    Intan    Id
Saleh</a>
</div><!--close footer-->
</div><!--close footer_container-->

<?php
        if($mod=="daftar-konsultasi")        {        $col_sort=1;
$sort_tipe="desc"; }
        else { $col_sort=1; $sort_tipe="asc"; }

```

```

?>
<script>
    $(document).ready(function() {
        var t = $('#tabel_data').DataTable( {
            "oLanguage": {
                "sSearch": "Pencarian:",
                "sLengthMenu": "Tampilkan _MENU_
baris",
                "sZeroRecords": "Data tidak ditemukan
(kosong).",
                "sInfo": "Baris (_START_ sampai _END_)
dari total _TOTAL_ baris",
                "sInfoEmpty": "Nol (0) Baris",
                "sInfoFiltered": " - dari _MAX_ baris"
            },

            "columnDefs": [ {
                "searchable": false,
                "orderable": false,
                "targets": 'no_sort'
            } ],

            "order": [[ <?php echo $col_sort; ?>, '<?
php echo $sort_tipe; ?>' ]]
        } );

        t.on( 'order.dt search.dt', function () {
            t.column(0, {search:'applied',
order:'applied'}).nodes().each( function (cell, i) {
                cell.innerHTML = i+1;
            } );
        } ).draw();

    } );

```

```
</script>
</body>
</html>
```

1.2 Listing Program Koneksi

```
<?php
//koneksi.php
$dbhost="localhost";
$dbname="Mentimun";
$dbuser="root";
$dbpassword="";

// create connection
$koneksi = new mysqli($dbhost, $dbuser, $dbpassword,
$dbname);

// check connection
if($koneksi->connect_error) {
die("Koneksi Gagal : " . $koneksi->connect_error);
}

if(mysqli_connect_error()) {
die('Connect Error (' . mysqli_connect_errno() . ') '
. mysqli_connect_error());
}
?>
```

1.3 Listing Program Library

```
<?php
session_start();
date_default_timezone_set('Asia/Jakarta');
$root_url="http://localhost/knn_test"; //Ubah ini sesuai url
localhost di komputer Anda
```



```
$seminggu =  
array("Minggu","Senin","Selasa","Rabu","Kamis","Jumat","  
Sabtu");
```

```
$hari = date("w");  
$hari_ini = $seminggu[$hari];
```

```
$tgl_sekarang = date("Ymd");  
$tgl_skrng = date("d");  
$bln_sekarang = date("m");  
$thn_sekarang = date("Y");  
$jam_sekarang = date("H:i:s");
```

```
$nama_bln=array(1=> "Januari", "Februari", "Maret",  
"April", "Mei",  
"Juni", "Juli", "Agustus", "September",  
"Oktober", "November", "Desember");
```

```
function antiinjec($connect, $string = '') {  
if(!empty($string)) {  
    $stripsql = $connect->real_escape_string($string);  
return $stripsql;  
}  
}
```

```
function antiinjec_f($connect, $string = '') {  
if(!empty($string)) {  
    $stripsql = $connect->real_escape_string($string);  
return $stripsql;  
}  
}
```

```
$tgl_full=date("Y-m-d H:i:s");
```

```
$sesinf_adminid=1;
```

```
function tgl_waktu($data){
$ttl_waktu=date("d-m-Y H:i:s", strtotime($data));
return $ttl_waktu;
}
```

```
function buat_text($text) {
$TMPBAGIAN = array();
$TMP=explode("-",
strip_tags(html_entity_decode($text)));
$jml=count($TMP);
for($i=0;$i<=$jml;$i++)
{
$TMPBAGIAN[$i] = $TMP[$i];
}
$VIEW = implode(" ",$TMPBAGIAN);
return ucfirst($VIEW);
}
```

```
function ukuran_file($size)
{
if ($size >= 1073741824) {
$fileSize = round($size / 1024 / 1024 / 1024,1) . 'GB';
} elseif ($size >= 1048576) {
$fileSize = round($size / 1024 / 1024,1) . 'MB';
} elseif($size >= 1024) {
$fileSize = round($size / 1024,1) . 'KB';
} else {
$fileSize = $size . ' bytes';
}
return $fileSize;
}
?>
```

1.4 Listing Program Konsultasi Hasil

```
<?php
$id_konsul=antiinjec($koneksi,
@$_SESSION['knn_konsultasi']);

if(isset($_SESSION['knn_konsultasi'])) {
$h_cek_gejala=$koneksi->query("SELECT
count(a.id_gejala)
FROM knn_gejala as
a, knn_konsultasi_gejala as c
WHERE
a.id_gejala=c.id_gejala AND c.status=1 AND
c.id_konsultasi='$id_konsul'
ORDER BY
a.kd_gejala ASC");
$d_cek_gejala=$h_cek_gejala->fetch_row();
if($d_cek_gejala[0]==0) {
?>
<script language="JavaScript">alert('Belum ada gejala
yang dipilih,\nSilahkan pilih dulu gejala yang sesuai.');
```

```
<div class="col-lg-12">
<h3 class="page-header" style="margin-top:29px;">Konsultasi - Hasil Diagnosis</h3>
</div>
</div>
```

```
<div class="row">
<div class="col-lg-12">
Berikut data diagnosis dari gejala-gejala yang Anda/Pasien
pilih.<br />
Nama : <span style="color:#06C; font-weight:bold;"><?php echo @$data['nama'];
?></span><br /><br />
</div>
</div>
```

```
<div class="row">
<div class="col-lg-12">
<h4>Gejala yang dipilih <a href="?mod=konsultasi-gejala" class='btn btn-success btn-xs'><i class="fa fa-edit fa-fw"></i> Ubah Gejala</a></h4>
<table class="csstable responsive" width="100%" border="0" cellpadding="4" cellspacing="0">
<thead>
<tr>
<th width="26" class="no_sort">No</th>
<th width="42">Kode</th>
<th width="892">Nama Gejala</th>
</tr>
</thead>
<tbody>
<?php
$no=0;
$hquery=$koneksi->query("SELECT a.id_gejala, a.kd_gejala, a.nm_gejala, a.bobot_parameter
FROM knn_gejala as a, knn_konsultasi_gejala as c
```

```
WHERE a.id_gejala=c.id_gejala AND c.status=1 AND  
c.id_konsultasi='$id_konsul'
```

```
ORDER BY a.kd_gejala ASC");
```

```
while($data2=$hquery->fetch_assoc()){
```

```
    $no++;
```

```
    ?>
```

```
<tr>
```

```
<td><?php echo $no; ?></td>
```

```
<td><?php echo $data2['kd_gejala']; ?></td>
```

```
<td><?php echo $data2['nm_gejala']; ?></td>
```

```
</tr>
```

```
<?php } ?>
```

```
</tbody>
```

```
</table>
```

```
</div>
```

```
</div>
```

```
<br />
```

```
<div class="row">
```

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

```
<h4>Perhitungan</h4>
```

```
<table class="csstable responsive" width="100%"  
border="0" cellpadding="4" cellspacing="0">
```

```
<thead>
```

```
<tr>
```

```
<th width="22" class="no_sort">No</th>
```

```
<th width="237">Penyakit</th>
```

```
<th width="200">Gejala Kasus</th>
```

```
<th width="202">Bobot</th>
```

```
<th width="216">Gejala Dipilih (Cocok)</th>
```

```
<th width="200">Bobot</th>
```

```
<th width="196">Kedekatan</th>
```

```

</tr>
</thead>
<tbody>
<?php
//Hitung
//Data kasus yang akan dicocokkan
$no3=0;

$update_konsul=$koneksi->query("UPDATE
knn_konsultasi_hasil      SET      status=0      WHERE
id_konsultasi='$id_konsul'");

$h_kasus=$koneksi->query("SELECT a.id_kasus, a.nama,
a.id_penyakit,    a.tanggal,    a.status,    b.kd_penyakit,
b.nm_penyakit
                                FROM knn_kasus as a,
knn_penyakit as b WHERE a.id_penyakit=b.id_penyakit
AND a.status=1");
while($d_kasus=$h_kasus->fetch_assoc()){

//Hitung nilai total kesamaan gejala pasien dengan kasus
$h_pasien=$koneksi->query("SELECT
SUM(c.bobot_parameter) as jml_1
                                FROM knn_kasus as
a, knn_kasus_gejala as b, knn_gejala as c,
knn_konsultasi_gejala as d
                                WHERE
a.id_kasus=b.id_kasus AND b.id_gejala=c.id_gejala AND
b.id_gejala=d.id_gejala
                                AND
d.id_konsultasi='$id_konsul'
                                AND
a.id_kasus='$d_kasus[id_kasus]' AND d.status=1");
$d_pasien=$h_pasien->fetch_assoc();

//Pembagi #2 (Jumlah Bobot Per Kasus)
$h_kasus2
                                =$koneksi->query("SELECT
SUM(c.bobot_parameter) as jml_2

```

```

FROM knn_kasus as a,
knn_kasus_gejala as b, knn_gejala as c
WHERE a.id_kasus=b.id_kasus
AND b.id_gejala=c.id_gejala AND
a.id_kasus='$d_kasus[id_kasus]');
$d_kasus2 =$h_kasus2->fetch_assoc();
//echo "pasien=$d_pasien[jml_1],
kasus=$d_kasus2[jml_2]<br>";

```

```

$arr_kode1=array();
$arr_kode2=array();
$arr_bobot1=array();
$arr_bobot2=array();

```

```

$h_kode=$koneksi->query("SELECT c.kd_gejala,
c.bobot_parameter
FROM knn_kasus as a,
knn_kasus_gejala as b, knn_gejala as c,
knn_konsultasi_gejala as d
WHERE
a.id_kasus=b.id_kasus AND b.id_gejala=c.id_gejala AND
b.id_gejala=d.id_gejala
AND
d.id_konsultasi='$id_konsul' AND
a.id_kasus='$d_kasus[id_kasus]' AND d.status=1");
while($d_kode=$h_kode->fetch_assoc()){
    $arr_kode1[]=$d_kode['kd_gejala'];
    $arr_bobot1[]=$d_kode['bobot_parameter'];
}

```

```

$h_kode2=$koneksi->query("SELECT c.kd_gejala,
c.bobot_parameter
FROM knn_kasus as a,
knn_kasus_gejala as b, knn_gejala as c
WHERE
a.id_kasus=b.id_kasus AND b.id_gejala=c.id_gejala AND
a.id_kasus='$d_kasus[id_kasus]');

```

```

while($d_kode2=$h_kode2->fetch_assoc()){
    $arr_kode2[]=$d_kode2['kd_gejala'];
    $arr_bobot2[]=$d_kode2['bobot_parameter'];
}

$hasil=0;
if($d_kasus2['jml_2']>0) {
    $hasil=$d_pasien['jml_1']/$d_kasus2['jml_2'];
}
if($hasil>0) {
    $no3++;
    ?>
<tr>
<td><?php echo $no3; ?></td>
<td><?php echo $d_kasus['nm_penyakit']; ?></td>
<td><?php echo implode(" ", $arr_kode2); ?></td>
<td><?php echo implode(" ", $arr_bobot2); ?></td>
<td><?php echo implode(" ", $arr_kode1); ?></td>
<td><?php echo implode(" ", $arr_bobot1); ?></td>
<td><?php echo "($d_pasien['jml_1']/" .
$d_kasus2['jml_2'])." =
".number_format($hasil,2,',','')."."; ?></td>
</tr>
<?php

//Cek dahulu pernah disimpan belum
$d_cek=$koneksi->query("SELECT id_konsultasi,
id_kasus, nilai FROM knn_konsultasi_hasil WHERE
id_konsultasi='$id_konsul' AND
id_penyakit='$d_kasus[id_penyakit]')->fetch_assoc();

//Simpan data diagnosa
if($d_cek['id_konsultasi']== "") {

```



```

                                $koneksi->query("INSERT INTO
knn_konsultasi_hasil (id_konsultasi, id_kasus, id_penyakit,
nilai, status)
                                VALUES ('$id_konsul',
'$d_kasus[id_kasus]', '$d_kasus[id_penyakit]', '$hasil',
1)");
        } else {
            if($d_cek['id_kasus']==$d_kasus['id_kasus']) {
                $koneksi->query("UPDATE
knn_konsultasi_hasil SET nilai='$hasil', status=1
                                WHERE
id_konsultasi='$id_konsul' AND
id_kasus=$d_kasus[id_kasus] AND
id_penyakit='$d_kasus[id_penyakit]'");
            } else {
                if($d_cek['nilai']<$hasil) {
                    $koneksi->query("UPDATE
knn_konsultasi_hasil SET nilai='$hasil',
id_kasus=$d_kasus[id_kasus], status=1
                                WHERE
id_konsultasi='$id_konsul' AND
id_kasus=$d_cek[id_kasus] AND
id_penyakit='$d_kasus[id_penyakit]'");
                }
            }
        }
    }
}
?>
</tbody>
</table>
</div>
</div>
<br />

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

```

<h4>Hasil Konsultasi</h4>

<table class="csstable responsive" width="100%"
border="0" cellpadding="4" cellspacing="0">

<thead>

<tr>

<th width="25" class="no_sort">No</th>

<th width="200">Kasus</th>

<th width="450">Penyakit</th>

<th width="488">Kedekatan</th>

</tr>

</thead>

<tbody>

<?php

\$no2=0;

\$hquery=\$koneksi->query("SELECT
b.id_konsultasi_hasil, a.nama, c.nm_penyakit, b.nilai

FROM knn_kasus as a, knn_konsultasi_hasil as b,
knn_penyakit as c

WHERE a.id_kasus=b.id_kasus AND
b.id_penyakit=c.id_penyakit AND
b.id_konsultasi='\$id_konsul'

AND b.status=1

ORDER BY b.nilai DESC");

while(\$data2=\$hquery->fetch_assoc()){

\$no2++;

?>

<tr>

<td><?php echo \$no2; ?></td>

<td><?php echo \$data2['nama']; ?></td>

<td><?php echo \$data2['nm_penyakit']; ?></td>

<td><?php echo number_format(\$data2['nilai'],2,',','.'); ?
></td>

```

</tr>
<?php } ?>
</tbody>
</table>
</div>
</div>
<div class="row">
<div class="col-lg-12">
    <?php
        $h_hasil=$koneksi->query("SELECT
        b.id_konsultasi_hasil,      a.nama,      c.nm_penyakit,
        c.id_penyakit, c.definisi, b.nilai
        FROM knn_kasus
        as a, knn_konsultasi_hasil as b, knn_penyakit as c
        WHERE
        a.id_kasus=b.id_kasus AND b.id_penyakit=c.id_penyakit
        AND b.id_konsultasi='$id_konsul'
        AND
        b.status=1
        ORDER BY b.nilai
        DESC LIMIT 0, 1");
        $d_hasil=$h_hasil->fetch_assoc();
        ?>
    <br />
    Hasil diagnosa Anda adalah <span style="color:#63C;
    font-weight:bold;"><?php echo $d_hasil['nm_penyakit']; ?
    ></span><br />
    <span style="color:#666;"><?php echo
    html_entity_decode($d_hasil['definisi']); ?></span><br />
    Solusi:<br />
    <?php
        $lin=$koneksi->query("SELECT a.id_penyakit_solusi,
        b.kd_solusi, b.nm_solusi
        FROM
        knn_penyakit_solusi as a, knn_solusi as b, knn_penyakit as
        c

```

```

WHERE
a.id_solusi=b.id_solusi AND a.id_penyakit=c.id_penyakit
AND c.id_penyakit='$d_hasil[id_penyakit]");
while($din=$lin->fetch_assoc()){
echo          "<b>[". $din['kd_solusi']."]</b>"          ".
$din['nm_solusi']. "<br>";
    }
    ?>
<br />
<a href="/pdf/hasil.php" class='btn btn-danger'
target="_blank"><i class="fa fa-file-pdf-o fa-fw"></i>
Export PDF</a>
<a href="?mod=konsultasi-selesai" class='btn btn-
warning'><i class="fa fa-close fa-fw"></i> Selesai
Konsultasi</a>

<br /><br />
</div>
</div>
<?php } else { ?>
<div class="row">
<div class="col-lg-12">
<h3 class="page-header" style="margin-
top:29px;">Konsultasi</h3>
</div>
</div>
<div class="row">
<div class="col-lg-12">
Anda diperbolehkan memulai konsultasi dari halaman <a
href="?mod=konsultasi">konsultasi</a>.<br />
</div>
</div>
<?php } ?>

```

DAFTAR RIWAYAT HIDUP



Nama : Nur Intan Id Saleh

TTL : Lomuli, 22 Juli 1999

Alamat : Desa Taluduyunu

Jenis Kelamin : Perempuan

Status Kawin : Kawin

Agama : Islam

Email : nurintanidsaleh22@gmail.com

ORANG TUA

Ayah : Idris Saleh

Ibu : Hasna Hunowu

PENDIDIKAN

1. Tahun 2011, Menyelesaikan Pendidikan di SDN 01 Buntulia.
2. Tahun 2014, Menyelesaikan Pendidikan di Mts Alkhairaat Buntulia.
3. Tahun 2017, Menyelesaikan Pendidikan di MA Alkhairaat Buntulia.
4. Tahun 2017, Mendaftar dan diterima menjadi Mahasiswa di Kampus IAIN Sultan Amai Gorontalo, Fakultas Ushuluddin dan Dakwah, Jurusan Ilmu Al-Qur'an dan Tafsir.
5. Tahun 2018, Mendaftar dan diterima menjadi Mahasiswa di Fakultas Ilmu Komputer, Jurusan Teknik Informatika, Universitas Ichsan Gorontalo.