

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Sistem pakar merupakan salah satu cabang dari kecerdasan buatan yang telah berkembang pesat dalam beberapa dekade terakhir. Menurut Turban et al. (2021), sistem pakar adalah bagian dari kecerdasan buatan yang menggunakan basis pengetahuan dan inferensi untuk menyelesaikan masalah yang biasanya memerlukan keahlian manusia. Mereka menekankan bahwa sistem pakar dirancang untuk mensimulasikan kemampuan pengambilan keputusan seorang ahli dalam domain tertentu.

Russell dan Norvig (2021) mendefinisikan sistem pakar sebagai sistem komputer yang mengemulasi kemampuan seorang pakar manusia dalam memberikan nasihat atau membuat keputusan dalam bidang spesifik. Mereka menyoroti pentingnya akurasi pengetahuan yang dimasukkan ke dalam sistem untuk memastikan kehandalan dan validitas hasilnya. Selain itu juga, sistem pakar sebagai sistem yang memanfaatkan kecerdasan buatan untuk meniru kemampuan analisis dan pemecahan masalah dari seorang ahli dalam bidang tertentu (Gupta & Mittal, 2022). Sistem Pakar (expert system) adalah salah satu cabang dari kecerdasan buatan (AI) yang ada saat sekarang ini yang banyak dibahas pemakaiannya dalam dunia industri, pendidikan, dan bisnis. Salah satunya adalah terdapat pada bidang bisnis perkebunan tanaman jeruk. (Solecha et al., 2021).

Jeruk merupakan salah satu komoditas buah unggulan nasional yang memiliki nilai ekonomi tinggi serta keberadaannya menyebar hampir di seluruh

wilayah Indonesia. Jeruk merupakan salah satu jenis buah yang menjadi komoditas unggulan yang dikembangkan, karena mempunyai sebaran lokasi yang luas (banyak ditanam), mempunyai arti ekonomi serta rasa, aroma, warna dan mengandung vitamin C sehingga menarik konsumen untuk mengkonsumsi buah jeruk (Kristiandy et al, 2021). Jeruk (*Citrus sp*) adalah buah sub tropik yang telah berkembang luas di Indonesia dan menjadi komoditas unggulan nasional dalam program pengembangan usaha agribisnis buah. Sebagai komoditas buah unggulan, jeruk merupakan unggulan pertama dari 5 komoditas buah lainnya berturut-turut: mangga, manggis, durian, dan pisang (Rosita, et al 2021).

Tahun 2021, menurut data Food and Agriculture organization (FAO) bahwa Indonesia berhasil menempati urutan ke delapan sebagai penghasil jeruk terbesar di dunia sebanyak 2,722,952 ton dan di Provinsi Riau sebanyak 760,00 ton. Produksi sebesar ini belum bisa memenuhi kebutuhan nasional, maka harus ada penambaham luas areal 4.000 ha dan peningkatan hasil produksi. Lima Tahun Terakhir mengalami penurunan data terbaru yang terkait produksi, luas panen serta populasi sub sektor Kementerian Pertanian selama lima tahun . Salah satu penyebabnya karena kualitas dan produktifitas buah jeruk yang dihasilkan rata-rata masih rendah. Hal itu sesuai dengan hasil observasi yang dilakukan oleh penulis, yakni adanya kendala dalam hasil produktivitas dan kualitas panen buah jeruk yang tidak seimbang dengan luasnya lahan tanaman jeruk. Yang mana banyak terdapat hama dan penyakit yang menyerang tanaman jeruk tersebut. Hal ini disebabkan oleh beberapa faktor, diantaranya adalah teknik budidaya, kualitas bibit yang digunakan, serta penanganan hama maupun penyakit Tanaman Jeruk.

Selain itu, jeruk juga merupakan salah satu komoditi buah-buahan yang mempunyai peranan penting di pasaran dunia maupun dalam negeri, baik dalam bentuk segar maupun olahannya. Jeruk mempunyai nilai ekonomis tinggi, maka pemerintah mengarahkan pengelolaan pertanaman jeruk bagi petani kecil dan industri perkebunan jeruk. Oleh karena itu, penanaman jeruk nasional akan memiliki peranan penting karena disamping untuk meningkatkan pendapatan masyarakat, kesempatan kerja, konsumsi buah dan juga meningkatkan devisa ekspor nasional (Kristiandi et al, 2021).

OPT (Organisme Pengganggu Tanaman) merupakan kendala bagi petani dalam membudidayakan tanaman jeruk, CVPD (Citrus Vein Phloem Degeneration) sejenis 390 hama dan penyakit yang menyerang tanaman jeruk menyebabkan kerusakan pembuluh tapis yang ada pada tanaman jeruk, sehingga mempengaruhi kondisi maupun produktifitas tanaman jeruk (Foda et al., 2021). Salah satu contohnya yang ada di dalam penelitian ini adalah hama kutu loncat, hama penggerek buah jeruk (*citruspestis sagitifarella*), hama thrips (*scirtothrips citri*) yang mana semua hama dan penyakit tersebut mempengaruhi produktivitas dan kualitas panen tanaman jeruk. Adapun semua hama dan penyakit diatas juga terdapat/alami pada tanaman jeruk yang dijumpai oleh penulis pada perkebunan jeruk yang di Sei-emas kecamatan Kepenuhan Hulu.

Ketika semua hama dan penyakit tersebut dialami oleh tanaman jeruk, biasanya para petani maupun pembudidayaan butuh cara untuk mengatasi permasalahan tersebut. Seperti mencari informasi melalui internet, yang mungkin bisa membantu para petani dan pembudidaya tanaman jeruk, dan bisa juga

berkonsultasi dengan seorang ahli pakar dalam bidang tersebut untuk mengetahui apa yang harus dilakukan ketika tanaman jeruknya mengalami atau terserang hama dan penyakit seperti yang dijabarkan diatas. Namun terkait untuk berkonsultasi dengan seorang ahli pakar tidaklah sebuah hal yang mudah untuk dilakukan oleh para petani dan pembudidayaan, karena keterbatasan ketidaktahuan informasi mengenai seorang pakar dan bagaimana cara menjumpai pakar tersebut, serta terhalang biaya juga untuk berkonsultasi. Selain itu juga, para petani dan pembudidayaan yang ada disekitar kita sekarang ini maupun dari dulu hingga sekarang mayoritas adalah kaum amam yang tidak punya ide untuk mencari seorang ahli pakar tersebut atau sudah terbiasa menggunakan cara tersendiri yang mana belum tentu sesuai maupun tepat cara penanganan hama dan penyakitnya berdasarkan gejala-gejala yang ada pada tanaman jeruknya.

Dampak dari tingkat produktivitas dan kualitas panen para petani dan pembudidaya tanaman jeruk menjadi menurun menyebabkan kerugian yang besar bagi petani dan pembudidaya tanaman jeruk. Jadi oleh sebab itu, untuk mengatasi permasalahan yang ada dan untk membantu para petani dan pembudidayaan supaya lebih efektif dan efisien waktu dan materi, maka peneliti mengusulkan sebuah sistem yang bisa menyediakan informasi dengan tujuan untuk mengidentifikasi jenis penyakit berdasarkan gejala yang ditimbulkan. Melalui penelitian ini cara yang dapat dilakukan untuk mengatasi permasalahan tersebut adalah dengan membuat suatu sistem pakar diagnosis penyakit tanaman jeruk. Dalam sistem pakar ini metode yang digunakan ialah metode *Dempster Shafer*.

Metode Dempster-Shafer adalah kerangka kerja yang dirancang untuk menangani ketidakpastian dalam pengambilan keputusan. Pendekatan ini didasarkan pada teori himpunan bukti yang memungkinkan integrasi informasi dari berbagai sumber untuk mencapai kesimpulan yang lebih tepat. Metode ini memodelkan ketidakpastian melalui dua elemen utama: himpunan massa, yang mencerminkan tingkat kepercayaan terhadap suatu hipotesis, dan himpunan kesimpulan, yang menyajikan hasil dari pengolahan himpunan massa dengan mengesampingkan faktor ketidakpastian (Fadhilah & Triayudi, 2024).

Berdasarkan pemaparan permasalahan yang sudah dijelaskan, maka peneliti memberikan judul penelitian tugas akhir ini dengan judul **“Sistem Pakar Diagnosis Penyakit Tanaman Jeruk Menggunakan Metode *Dempster-Shafer* Berbasis Web”**.

1.2 Rumusan Masalah

Rumusan masalah ini didasarkan sesuai identifikasi pada latar belakang yang sudah dipaparkan, maka peneliti menemukan beberapa rumusan, yaitu :

1. Bagaimana merancang dan membuat sistem pakar yang mampu mendiagnosis penyakit pada tanaman jeruk sehingga mudah dan dapat memberikan keterangan beserta perawatannya?
2. Bagaimana mengimplementasikan metode *dempster-Shafer* ke dalam sistem pakar sehingga mendapatkan hasil tingkat keyakinan yang valid?

1.3 Ruang Lingkup Permasalahan

Untuk pembuatan Sistem Pakar Diagnosis Penyakit Tanaman Jeruk menggunakan metode *dempster-shafer* ini, peneliti memberikan beberapa batasan masalah sehingga lebih tertera dan spesifik, antara lain :

1. Penelitian ini dilakukan di Kantor Dinas Perkebunan dan Peternakan Rokan Hulu.
2. Sistem Pakar yang dibangun hanya untuk mendiagnosis penyakit pada tanaman jeruk menggunakan metode *dempster-Shafer*.
3. Variabel yang digunakan adalah hama kutu loncat, hama penggerak buah jeruk (*Citripestis sagittiferella*), hama Thrips (*Scirtithripsi citri*), busuk akar phytophthora, penyakit blendok, penyakit embun tepung.
4. Metode penelitian yang digunakan studi literatur, wawancara dan observasi.
5. Bahasa pemrograman yang digunakan dalam pembuatan sistem dengan teknologi Web PHP dan menggunakan MySQL.

1.4 Tujuan dan Manfaat Penelitian

1.4.1 Tujuan Penelitian

1. Untuk membantu para petani jeruk dalam melakukan diagnosis awal penyakit pada tanaman jeruk.
2. Merancang sistem pakar diagnosis penyakit tanaman jeruk
3. Menerapkan metode *dempster-Shafer* dalam mendiagnosis penyakit tanaman jeruk.
4. Dapat menghasilkan suatu sistem yang dapat digunakan untuk melakukan

diagnosis penyakit tanaman.

1.4.2 Manfaat Penelitian

1. Manfaat bagi Instansi

a. Untuk mempermudah dan mempercepat petani dalam melakukan diagnosis awal terhadap penyakit tanaman jeruk.

b. Sebagai informasi bagi masyarakat awam dan petani jeruk mengenai permasalahan penyakit yang menyerang tanaman jeruk.

2. Manfaat bagi Mahasiswa (Peneliti)

a. Untuk mengetahui bagaimana menganalisis dan membangun sistem pakar dengan menggunakan metode *dempster-Shafer* dalam memberikan hasil diagnosis awal penyakit yang diderita beserta keterangan dan solusi yang benar dan tepat.

b. Menambah penerapan metode *Dempster Shafer*

c. Dapat menambah wawasan dan pengalaman peneliti.

1.5 Metode Pengumpulan Data

Tahapan-tahapan yang akan dilakukan pada pelaksanaan penelitian adalah sebagai berikut:

1. Studi Literatur

Pada tahap ini penulis mengumpulkan bahan referensi berkaitan dengan Sistem Pakar Diagnosis Penyakit pada Tanaman Jeruk dengan Menggunakan Metode *Dempster Shafer*, dari berbagai jurnal, skripsi, artikel dan berbagai sumber referensi lainnya.

2. Analisis Masalah

Pada tahap ini dilakukan analisis untuk setiap informasi yang telah diperoleh dari tahap sebelumnya agar mendapatkan pemahaman akan menyelesaikan permasalahan.

3. Analisis dengan Metode *Dempster Shafer*

Setelah menganalisis masalah pada tahap sebelumnya, dilakukanlah analisis dengan menghitung secara manual menggunakan rumus dari metode *Dempster Shafer*

4. Observasi

Observasi adalah suatu metode mengumpulkan data dengan melakukan pengamatan langsung pada objek yang diteliti. Metode ini bertujuan untuk dapat mengetahui langsung bagaimana alur kerja yang terjadi pada objek yang diteliti.

5. Wawancara

Proses wawancara dilakukan langsung kepada Dinas Perkebunan tentang penyakit pada tanaman jeruk untuk mendapatkan informasi tentang jenis penyakit, gejala, dan nilai kemungkinan dari metode *dempster-shafer* pada setiap gejala untuk menentukan jenis penyakit.

1.6 Sistematika Penulisan

Sistematika penulisan dari skripsi ini terdiri dari lima bagian utama sebagai berikut :

BAB 1	PENDAHULUAN Bab ini akan menjelaskan mengenai latar belakang pemilihan judul tugas akhir “Sistem Pakar Diagnosis Penyakit Tanaman Jeruk
--------------	---

	Menggunakan Metode <i>Dempster-Shafer</i> Berbasis Web ”, rumusan masalah, batasan masalah, tujuan penelitian dan manfaat penelitian, metode pengumpulan data, dan sistematika penulisan.
BAB 2	LANDASAN TEORI Bab ini akan membahas teori–teori atau gambaran umum serta kebutuhan sistem yang berkaitan dengan sistem pakar(<i>expert system</i>) serta metode <i>dempster-shafer</i> .
BAB 3	METODOLOGI PENELITIAN Bab ini akan membahas mengenai identifikasi masalah, pengumpulan data, studi literature, analisis dan perhitungan metode <i>dempster-shafer</i> , perancangan sistem, implementasi serta pengujian sistem.
BAB 4	ANALISIS DAN PERANCANGAN SISTEM Bab ini berisi bagaimana menganalisis cara kerja sistem yang akan dibangun dan menjelaskan tahap perancangan sistem berdasarkan hasil analisis agar dimengerti oleh pengguna.
BAB 5	IMPLEMENTASI DAN PENGUJIAN Bab ini berisi tentang implementasi dan pengembangan perangkat lunak serta pengujian akhir terhadap sistem yang telah dibuat.
BAB 6	PENUTUP Bab ini berisi mengenai kesimpulan terhadap sistem yang dibuat dan saran untuk pengembangan terhadap sistem yang telah dibuat.

BAB 2

LANDASAN TEORI

2.1 Konsep Dasar Sistem

Sistem berasal dari bahasa Yunani yang mengandung arti “*Systema*” yang berarti kesatuan atau kumpulan. Ditinjau dari perkataan kata, sistem berarti sekumpulan objek yang bekerja bersama-sama untuk menghasilkan kesatuan metode, prosedur atau teknik yang digabungkan dan diatur sedemikian rupa sehingga menjadi satu kesatuan yang berfungsi untuk mencapai tujuan (Nitami et al., 2021).

Sistem adalah perangkat unsur yang secara teratur saling berkaitan sehingga membentuk suatu totalitas. Sistem juga diartikan sebagai susunan yang teratur dari pandangan, teori, asas, dan sebagainya. Sistem adalah sekumpulan objek-objek yang saling berelasi dan berinteraksi serta hubungan antara objek bisa dilihat sebagai satu kesatuan yang dirancang untuk mencapai suatu tujuan. Dengan demikian, secara sederhana, sistem dapat diartikan sebagai suatu kumpulan atau himpunan dari unsur variabel-variabel yang saling terorganisasi, saling berinteraksi, dan saling bergantung satu sama lain (Faradila, 2023).

Suatu sistem mempunyai karakteristik atau sifat-sifat tertentu, yaitu mempunyai komponen-komponen, batasan sistem, lingkungan luar sistem, penghubung, masukan, keluaran pengolah dan sasaran atau tujuan (Nitami et al., 2021). Berikut karakteristik pada sistem:

1. Komponen (*Components*)

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang artinya saling bekerja sama membentuk suatu kesatuan. Komponen- komponen sistem atau elemen-elemen sistem dapat berupa suatu subsistem atau bagian-bagian dari sistem.

2. Batasan sistem (*Boundary*)

Batasan sistem merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya. Batas sistem ini memungkinkan suatu sistem dipandang sebagai suatu kesatuan dan menunjukkan ruang lingkup dari sistem tersebut.

3. Lingkungan luar sistem (*Environment*)

Lingkungan luar sistem adalah apapun diluar batas dari sistem yang mempengaruhi operasi sistem. Lingkungan luar sistem dapat bersifat menguntungkan dan juga merugikan.

4. Penghubung sistem (*Interface*)

Penghubung sistem merupakan media yang menghubungkan antara suatu subsistem dengan yang lainnya. Melalui penghubung ini kemungkinan sumber-sumber daya mengalir dari suatu subsistem ke subsistem yang lainnya.

5. Masukan sistem (*Input*)

Masukan sistem adalah energi yang dimasukkan ke dalam sistem. Masukan dapat berupa masukan perawatan dan masukan sinyal *maintenance* input adalah energi yang dimasukkan supaya sistem tersebut dapat berjalan.

6. Keluaran sistem (*Output*)

Keluaran sistem adalah energi yang diolah dan diklasifikasikan menjadi

keluaran yang berguna. Keluaran dapat merupakan masukan untuk subsistem yang lain.

7. Pengolahan sistem (*Process*)

Pengolahan sistem dapat mempunyai suatu bagian pengolah atau sistem itu sendiri sebagai pengolahnya. Pengolah yang akan merubah masukan menjadi keluaran.

8. Sasaran sistem (*Objective*)

Suatu sasaran sistem mempunyai tujuan atau sasaran, kalau sistem tidak mempunyai sasaran maka sistem tidak akan ada. Suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuannya. Sasaran sangat berpengaruh pada masukan dan keluaran yang dihasilkan.

2.2 Konsep Dasar Pakar

Pakar adalah orang yang memiliki pengetahuan, penilaian, pengalaman, dan metode tertentu, serta kemampuan untuk menerapkan bakat ini dalam memberi nasehat dan memecahkan masalah. Pakar yang dimaksud ialah orang yang memiliki pengetahuan atau keahlian khusus yang tidak diketahui atau tidak ada pada kebanyakan orang (Yulianto et al., 2021).

Menurut Hendradi (2022) Seorang pakar atau ahli didefinisikan sebagai seseorang yang memiliki pengetahuan atau keahlian khusus yang tidak dimiliki oleh kebanyakan orang. Seorang pakar dapat memecahkan masalah yang orang lain tidak mampu memecahkan. Seorang pakar juga dapat mendapatkan pengetahuannya dari sistem pakar atau mendapatkan pengetahuannya sendiri dari

seorang pakar.

Dalam penelitian Solecha et al., (2021) Pakar merupakan seseorang ahli dalam suatu bidang ilmu pengetahuan tertentu, berarti expert memiliki suatu permasalahan yang tidak dapat dipecahkan oleh orang lain secara efisien. Seorang pakar adalah orang yang mempunyai keahlian dalam bidang tertentu, yaitu pakar yang mempunyai knowledge atau kemampuan khusus yang orang lain tidak mengetahui atau mampu dalam bidang yang dimilikinya (Octaviani et al., 2022).

Ahli atau pakar didefinisikan sebagai seseorang yang memiliki pengetahuan atau keahlian khusus yang tidak dimiliki oleh kebanyakan orang. Para ahli memiliki kemampuan untuk memecahkan masalah yang tidak dapat dipecahkan oleh orang lain dan dapat memperoleh pengetahuan dari sistem pakar (Prionggo Hendradi, 2022).

2.3 Sistem Pakar

Sistem Pakar (expert system) adalah salah satu bidang ilmu Komputer yang mendayagunakan komputer sehingga dapat berperilaku cerdas seperti manusia. Sistem pakar berusaha mengadopsi pengetahuan manusia ke dalam komputer, agar komputer dapat menyelesaikan masalah seperti yang biasa dilakukan oleh para ahli (Solecha et al., 2021)

Sistem pakar berasal dari istilah *knowledge base expert system*. Dengan sistem pakar ini orang awam pun dapat menyelesaikan masalah yang cukup rumit yang sebenarnya hanya dapat diselesaikan dengan bantuan para ahli. Bagi para ahli sistem pakar ini juga membantu aktivitasnya sebagai asisten yang sangat berpengalaman (Ramadhan & Nugroho, 2023).

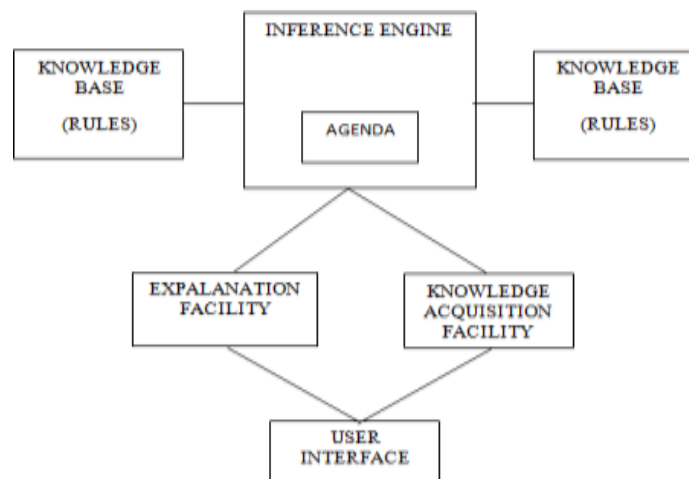
Sistem pakar adalah sebuah program komputer yang dirancang untuk mengambil keputusan seperti keputusan seorang pakar, dimana sistem pakar menggunakan pengetahuan (Knowledge), fakta dan teknik berfikir dalam menyelesaikan masalah yang biasanya hanya dapat diselesaikan oleh seorang pakar di bidang yang tertentu (Maharani et al., 2021).

Sistem pakar merupakan cabang dari AI (Artificial Intelligent) yang membuat ekstensi untuk spesialisasi pengetahuan guna memecahkan suatu permasalahan pada Human Expert (Solecha et al., 2021). Dalam jurnal Ramadhan & Nugroho (2023), dijelaskan bahwa Sistem pakar berasal dari istilah knowledge base expert. Dengan sistem ini, orang awam pun dapat menyelesaikan masalah yang cukup rumit yang sebenarnya hanya dapat diselesaikan dengan bantuan para ahli. Para ahli sistem pakar juga membantu aktivitas mereka sebagai asisten yang sangat berpengalaman.

Sistem pakar adalah suatu sistem yang dirancang untuk dapat menirukan keahlian seorang pakar dalam menjawab pertanyaan dan memecahkan suatu masalah. Sistem pakar akan memberikan pemecahan suatu masalah yang di dapat dari dialog dengan pengguna. Dengan bantuan sistem pakar seseorang yang bukan pakar atau ahli dapat menjawab pertanyaan menyelesaikan masalah serta mengambil keputusan yang biasanya dilakukan oleh seorang pakar. Sistem pakar dengan kata lain adalah sistem yang di desain dan diimplementasikan dengan bantuan bahasa pemrograman tertentu untuk dapat menyelesaikan masalah (Ramadhan & Nugroho, 2023).

2.4 Struktur Sistem Pakar

Sistem pakar disusun oleh dua bagian yaitu lingkungan pengembangan dan lingkungan konsultasi. Lingkungan pengembangan sistem pakar digunakan untuk memasukkan *knowledge* pakar ke dalam lingkungan sistem pakar, sedangkan lingkungan konsultasi digunakan oleh pengguna yang bukan pakar guna memperoleh *knowledge* pakar. Komponen-komponen yang terdapat dalam sistem pakar yaitu antar muka pengguna, basis pengetahuan (*knowledge-base*), akuisisi pengetahuan, mesin inferensi, *workplace*, fasilitas penjelasan, perbaikan pengetahuan. Berikut gambar strukturnya:



Gambar 2.1 Struktur Sistem Pakar
(Sumber: Panessai, I. Y. (2021))

Adapun komponen yang terdapat dalam struktur sistem pakar ini adalah :

a. Knowledge Base (Basis Pengetahuan)

Basis pengetahuan mengandung pengetahuan untuk pemahaman, formulasi dan penyelesaian masalah. Elemen dasarnya fakta dan ukuran.

b. Inference Engine (mesin Inferensi)

Merupakan otak dari sebuah sistem pakar dan dikenal juga dengan sebutan

control structure (struktur kontrol)atau rule interpreter (dalam sebuah sistem pakar berbasis kaidah) Forward Chaining.

c. *Working Memory*

Berguna untuk menyimpan fakta yang dihasilkan oleh inference engine dengan penambahan parameter berupa derajat kepercayaan atau dapat juga dikatakan sebagai global database dari fakta yang digunakan oleh rule-rule yang ada.

d. *Explanation Facility*

Menyediakan kebenaran dari solusi yang dihasilkan kepada user (reasoning chain).

e. *Knowledge Acquisition Facility*

Meliputi proses pengumpulan, pemindahan dan perubahan dari kemampuan pemecahan masalah seorang pakar atau sumber pengetahuan terdokumentasi keprogram sebuah komputer, dengan tujuan untuk memperbaiki atau mengembangkan basis pengetahuan.

f. *User Interface*

Mekanisme untuk memberi kesempatan kepada user dan sistem pakar untuk berkomunikasi. Antar muka menerima informasi dari pemakai dan mengubahnya ke dalam bentuk yang dapat diterima oleh sistem.

2.5 Diagnosis

Diagnosis adalah proses menemukan kelemahan atau penyakit apa yang dialami seseorang dengan melalui pengujian dan studi yang seksama mengenai gejala-gejalanya. Diagnosis memiliki proses yang tidak langsung ketika mengidentifikasi jenis penyakit dengan cara mengetahui jenisnya. Sehingga dapat

dikatakan bahwa itu penyakit yang dialami (Kusumah et al., 2022).

Menurut Kamus Besar Bahasa Indonesia (KBBI), diagnosis menentukan sifat penyakit dengan pemeriksaan gejalanya. Secara umum, diagnosis adalah upaya untuk mengetahui atau mengidentifikasi suatu penyakit atau masalah kesehatan yang diderita oleh pasien atau penderita. Diagnosis juga adalah penentu penyebab kerusakan dalam situasi kompleks berdasarkan gejala yang dialami, termasuk diagnostic medis, elektronik, mekanis, dan perangkat lunak (Yolla & Puspita, 2023).

2.6 Penyakit Tanaman Jeruk

Tanaman jeruk merupakan tanaman buah yang berasal dari Asia dan telah lama tumbuh di Indonesia, baik secara alami maupun melalui budidaya. Jeruk menjadi salah satu jenis tanaman hortikultura yang layak untuk dibudidayakan karena memiliki nilai ekonomis yang tinggi. Buah ini banyak diminati oleh masyarakat, baik dalam bentuk buah segar maupun berbagai hasil olahannya, seperti jus, manisan, atau produk lainnya. Selain itu, tanaman jeruk juga memiliki potensi besar untuk dikembangkan, terutama di wilayah dengan kondisi iklim dan tanah yang mendukung pertumbuhannya (Adlini & Umaroh, 2021).

1. Hama kutu loncat (penyakit CVPD) Gejalanya:

Kutu loncat ini menyerang kuncup, tunas, daun-daun muda dan tangkai daun. Serangannya mengakibatkan tunas-tunas muda keriting dan pertumbuhannya terhambat. Apabila serangan parah, bagian tanaman terserang biasanya kering secara perlahan kemudian mati. Kutu juga menghasilkan sekresi berwarna putih transparan berbentuk spiral, diletakkan berserak di atas permukaan daun atau tunas. Serangga ini selain sebagai hama juga sebagai vektor penyakit CVPD.



Gambar 2.2 Kutu loncat
(Sumber : hortikultura.pertanian.go.id)

Penanganannya:

Pengamatan dan pengendalian terhadap hama kutu loncat termasuk ke sanitasi lingkungannya harus bersih dari pada sampah-sampah maka dari itu biasanya para petani biasa melakukan pengasapan dibawah pohon jeruk itu.

2. Hama Penggerek Buah Jeruk (*Citrupestis sagittiferella*) Gejalanya:

Pada buah yang terserang hama penggerek, terlihat adanya lubang gerekan pada kulit buah dan juga mengeluarkan getah seperti blendok. Buah yang terserang mulai dari buah muda hingga buah menjelang panen, fase kritis serangan yaitu pada saat buah berumur 2-5 bulan. Buah yang terserang akan membusuk dan kemudian gugur sebelum matang.



Gambar 2.3 Kerusakan buah yang disebabkan oleh ulat frass atau gumming.

(Sumber : hortikultura.pertanian.go.id)



Gambar 2.4 Ngengat dewasa penggerek buah jeruk.

(Sumber : hortikultura.pertanian.go.id)

Penanganannya:

Untuk mencegah peletakan telur sebaiknya dilakukan pembungkusan pada buah (jeruk besar), memetik buah jeruk yang terserang kemudian dibenam dalam tanah atau dibakar. Pengendalian cara ini biasanya dilakukan sekaligus untuk mengendalikan lalat buah dan puru buah. Pengendalian dengan insektisida dilakukan sebelum telur menetas yaitu saat buah umur 2- 5 bulan sehingga larva yang baru keluar akan segera mati sebelum sempat menggerek. Di alam, populasi hama ini dikendalikan oleh parasit telur *Trichogramma nana* (16%). Pemanfaatan parasitoid ini dilakukan pada saat yang tepat dengan pelepasan dari hasil perbanyakan masal yang sudah banyak dilakukan.

2. Hama Thrips (*Scirtothrips citri*) Gejalanya:

Hama thrips menyerang dengan cara memarut mengisap permukaan kulit buah (Gazali & Ilhamiyah, 2022). Serangan pada buah terjadi mulai pada fase bunga dan ketika buah masih sangat muda. Akibat serangan thrips tersebut, pada buah

akan terdapat bekas luka berwarna coklat keabu-abuan yang disertai garis nekrotis di sekeliling luka. Bekas luka tampak di permukaan kulit buah di sekeliling tangkai atau melingkar pada sekeliling kulit buah. Thrips juga dapat menyerang bagian tangkai dan daun muda yang mengakibatkan helai daun menebal, kedua sisi daun agak menggulung ke atas dan pertumbuhannya tidak normal.



Gambar 2.5 Kutu Trips Jeruk
(Sumber : hortikultura.pertanian.go.id)



Gambar 2.6 Hewan hama Thrips
(Sumber : hortikultura.pertanian.go.id)

Penanganannya:

Untuk mencengah peletakan telur sebaiknya dilakukan pembungkusan pada buah (jeruk besar), memetik buah jeruk yang terserang kemudian dibanam dalam tanah atau dibakar. Pengendalian cara ini biasanya dilakukan sekaligus untuk mengendalikan lalat buah dan puru buah. Pengendalian dengan insektisida

dilakukan sebelum telur menetas yaitu saat buah umur 2- 5 bulan sehingga larva yang baru keluar akan segera mati sebelum sempat menggerak. Di alam, populasi hama ini dikendalikan oleh parasit telur *Trichogramma nana* (16%). Pemanfaatan parasitoid ini dilakukan pada saat yang tepat dengan pelepasan dari hasil perbanyakan masal yang sudah banyak dilakukan.

3. Busuk akar *Phytophthora*

Gejalanya:

Penyakit ini disebabkan oleh serangan cendawan *Phytophthora parasitica*. Umumnya penyakit ini berjangkit di tempat yang selalu teduh dan lembap. Tanaman jeruk manis yang diserang biasanya sudah berumur lebih dari satu tahun.

Batang dekat tanah (pangkal batang) dan akar yang besar merupakan sasaran serangan cendawan ini. Mula-mula kulit batang yang terserang berwarna hitam kebasah-basahan, kemudian mengeluarkan getah (blendok) yang berwarna coklat. Serangan ini bisa menjalar ke atas atau ke samping. Bagian tanaman jeruk yang terserang penyakit bila digosok akan memperlihatkan warna coklat, sedangkan bagian yang sehat berwarna hijau.

Selanjutnya penyakit ini dapat masuk ke dalam kambium, kemudian ke dalam kayu. Kulit yang terserang penyakit menjadi mengerut, retak-retak dan akhirnya mati. Bila kulit yang terserang penyakit ini terkelupas akan terlihat kambium berwarna coklat tua. Pada serangan yang sudah berat, semua akar membusuk dan kering. Daun-daun layu, mengering, dan bahkan seluruh batang bagian pohon pun mengering dan mati.



Gambar 2.7 Busuk akar Phytophthora
(Sumber : hortikultura.pertanian.go.id)

Penanganannya:

Busuk akar ini terjadi disebabkan kurangnya pengapuran. tanaman itu baiknya selalu dikasih kapur selalu maupun Dolomit akan terhindar dari penyakit busuk akar.

4. Penyakit Blendok

Gejalanya:

Penyakit Blendok/ Diplodia merupakan salah satu penyakit utama pada jeruk. Penyakit Blendok dapat terjadi apabila ada patogen menyerang cendawan *Botryodiplodia theobromae* Pat. yang patogenik menyerang tanaman yang rentan, yang tumbuh pada lingkungan yang sesuai untuk patogen, dan petani kurang intensif dalam pemeliharaan tanaman.

Tingkat serangan penyakit blendok dapat dipakai sebagai tolok ukur terhadap tingkat pemeliharaan yang sudah dilakukan, makin intensif pemeliharaan dapat menurunkan tingkat serangan penyakit. Kondisi lingkungan yang mempermudah serangan patogen diantaranya kondisi kekeringan, adanya pelukaan, perbedaan suhu siang dan malam yang tinggi dan pemeliharaan yang kurang

optimal. Penyakit ini juga dikenal sebagai penyakit blendok karena salah satu gejalanya adalah keluarnya blendok (gum) dari batang yang terinfeksi.

Penyakit blendok dapat diketahui dengan mudah apabila tanaman sudah bereaksi terhadap serangan patogen dengan mengeluarkan substansi pertahanan berupa blendok (gum/gumosis). Diketahui ada dua jenis Diplodia yaitu basah dan kering. Diplodia basah, batang, cabang, atau ranting yang terserang mengeluarkan blendok berwarna kuning keemasan dan pada stadia lanjut, kulit tanaman mengelupas. Diplodia kering, kulit batang atau cabang tanaman yang terserang akan mengering tanpa mengeluarkan blendok, sehingga gejalanya lebih sulit diamati. Pada bagian celah kulit terlihat adanya masa spora jamur berwarna putih atau hitam. Serangan pada batang utama akan lebih berbahaya dibandingkan pada cabang atau ranting. Serangan yang melingkar pada cabang mengakibatkan bagian tanaman di atas serangan akan kering dan mati.



Gambar 2.8 Penyakit Blendok
(Sumber: hortikultura.pertanian.go.id)

Penanganannya:

- Menjaga kebersihan kebun, memangkas bagian tanaman yang sakit
- Menjaga kebersihan alat pertanian dengan alkohol 70% atau Sodium hipoklorit 10%

- Pelaburan dengan bubur California Cara Pembuatan BuburCalifornia :

Bahan : 1 Vol. Belerang + 2 Vol. Kapur + 10 Vol. Air

1. Serbuk belerang direbus sampai larut
2. Kapur hidup dilarutkan di wadah terpisah
3. Larutan kapur dimasukkan ke dalam larutan belerang yang sedang mendidih dan campuran kedua bahan dibiarkan mendidih \pm 10 menit.

Cara Penggunaan:

Larutan sudah siap untuk melabur, batang tanaman terlebih dahulu dibersihkan dari blendok dan kulit kering yang mengelupas dengan cara disikat. Pelaburan batang dilakukan dua kali setahun yaitu awal dan akhir musim hujan.

5. Penyakit Embun Tepung Gejalanya:

Penyakit ini disebabkan oleh cendawan *Oidium tingtonium*, yang sering timbul di daerah yang beriklim basah (lembap) dan dataran tinggi. Biasanya penyakit ini menyerang bagian daun dan ranting tanaman jeruk yang masih muda. Pada ranting dan daun tersebut sering kali terdapat tepung putih. Adanya tepung ini mengakibatkan daun muda menjadi keriput, kemudian kering. Bila serangan penyakit ini cukup berat maka akan menyebabkan daun tua akan rontok. Serangan penyakit ini dapat pula menghambat pembungaan bila terjadi pada saat tanaman jeruk akan berbunga.

Cara terbaik pengendalian penyakit ini, adalah dengan menggunakan fungisida yang mengandung belerang. Misalnya, penghembusan tepung belerang pada bagian tanaman jeruk yang terserang pada pagi hari (masih ada embun pagi).

Penghembusan ini dilakukan terus-menerus seminggu sekali, sehingga tanaman jeruk sembuh. Selain itu dapat pula disemprot dengan Antracol 70 WP dengan selang waktu 5 – 7 hari sekali semprot.



Gambar 2.9 Penyakit Embun Tepung
(Sumber : hortikultura.pertanian.go.id)

Penanganannya:

Penanganan:

1. Pemangkasan tunas yang terserang.
2. selalu dilakukan pengasapan
3. Penyemprotan menjelang bertunas dan diulang saat daun muda, menggunakan pestisida dengan bahan aktif seperti Siprokonozal, Propineb, Copper Hidrocloride dan Benomyl.

2.7 Dempster Shafer

Dempster Shafer adalah suatu metode pengambilan keputusan yang membantu mengatasi ketidakpastian dan ketidakjelasan dalam proses pengambilan keputusan (Susilawati & Simanullang, 2023). Metode *Dempster Shafer* merupakan metode yang mengkombinasikan potongan informasi yang terpisah untuk mengkalkulasikan kemungkinan dari suatu kasus berdasarkan nilai bobot

atau kepercayaan yang diberikan (Istiadi et al., 2021). Metode *Dempster Shafer* juga suatu teori matematika untuk pembuktian berdasarkan *belief function and plausible reasoning* (fungsi kepercayaan dan pemikiran yang masuk akal) yang digunakan untuk mengkalkulasi kemungkinan dari suatu peristiwa (Sitohang et al., 2022).

Metode *Dempster Shafer* dikenalkan oleh Dempster yang melakukan percobaan model ketidakpastian dengan range probabilitas dari pada sebagai probabilitas tunggal tepatnya pada tahun (1967). Kemudian pada tahun itu juga dikembangkan oleh Shafer dan dipublikasikan pada sebuah buku yang berjudul *Mathematical Theory Of Evident* dimana pada teori tersebut dapat membedakan ketidakpastian dan ketidaktahuan, banyak metode atau model yang lengkap dan konsisten dalam menentukan ketidakpastian, namun kenyataannya banyak yang tidak bisa diselesaikan sepenuhnya dan konsisten. Ketidak konsisten ini disebabkan oelh penambahan fakta baru yang disebutkan dengan nonmonotik. Oleh karena itu, membuat *Dempster Shafer* memiliki kelebihan dalam mengatasi ketidak konsistenan (Belipati et al., 2021).

Jika diketahui X adalah sub-set dari q dengan m1 sebagai fungsi densitasnya, dan Y juga merupakan sub-set dari q dengan m2 sebagai fungsi densitasnya, maka dapat dibentuk suatu fungsi kombinasi m1 dan m2 sebagai m3, ditunjukkan pada persamaan berikut:

$$m_3(Z) = \frac{\sum_{x \cap y = z} m_1(x) \cdot m_2(y)}{m_1(X) \cdot m_2(Y)} \dots\dots\dots(1)$$

$$1 - \sum_{x \cap y = \emptyset} m_1(X) \cdot m_2(Y)$$

(X) adalah mass function dari evidence (X), yang diperoleh dari nilai keyakinan suatu evidence dikalikan dengan nilai disBelief dari evidence tersebut..1 m

(Y) adalah mass function dari evidence (Y), yang diperoleh dari nilai keyakinan suatu evidence dikalikan dengan nilai disBelief dari evidence tersebut. yang merupakan nilai kekuatan dari evidence Z yang diperoleh dari kombinasi nilai keyakinan sekumpulan evidence .

Secara umum teori *Dempster Shafer* ditulis dalam interval [Belief, Plausibility]. Belief (Bel) adalah ukuran kekuatan evidence dalam mendukung 20 suatu himpunan proposisi. Jika bernilai 0 maka mengindikasikan bahwa tidak ada evidence, dan jika bernilai 1 menunjukkan adanya kepastian. Plausibility (Pls) dinotasikan sebagai:

$$Pls(X) = 1 - Bel(X) \dots\dots\dots (2)$$

Jika yakin akan X, maka dapat dikatakan bahwa $Bel(X) = 1$, sedangkan nilai $Pls(X) = 0$.

Pada teori *Dempster Shafer* dikenal adanya frame of discernment yang dinotasikan dengan θ . Frame ini merupakan semesta pembicaraan dari sekumpulan hipotesis. Pada teori *Dempster Shafer* terdapat mass function yaitu tingkat kepercayaan dari suatu evidence measure yang dinotasikan dengan (m).

Fungsi Belief juga dapat diformulasikan sebagai berikut :

$$Bel(X) = \sum_{Y \in X} m(Y) \dots\dots\dots (3)$$

dan Plausibility dinotasikan sebagai berikut:

$$Pls(X) = 1 - Bel(X) = 1 - \sum_{Y \in X} m(Y) \dots\dots\dots (4)$$

Dimana:

$Bel(X) = \text{Belief}(X)$

$m(X) = \text{mass function dari } (X)$ $m(Y) = \text{mass function dari } (Y)$

Environment adalah elemen-elemen yang menggambarkan kemungkinan sebagai jawaban. Kemungkinan ini dalam teori *Dempster Shafer* disebut dengan power set $P(\theta)$, memiliki nilai interval antara 0 sampai 1 (Aldo and Putra 2020).

Fungsi Belief diformulasikan seperti pada Persamaan (1) dan fungsi Plausibility diformulasikan seperti pada Persamaan (2)

$$Bel(X) = \sum_{Y \subseteq X} m_1(Y) \dots \dots \dots (1)$$

$$Pls(X) = 1 - Bel(\bar{X}) \dots \dots \dots (2)$$

dimana:

X= Penyakit yang mengalami gejala 1

Y= Penyakit yang mengalami gejala 2

$Bel(X) = \text{Belief}(X)$, artinya nilai keyakinan atau kepastian penyakit X yang mengalami gejala 1

$Pls(X) = \text{Plausibility}(X)$, artinya nilai ketidakyakinan atau ketidakpastian

$m_1(X) = \text{Mass function atau tingkat kepercayaan dari evidence } (X)$ (MZ et al., 2020).

2.8 Alat Bantu Perancangan

2.9.1 Unified Modeling Language (UML)

Unified Modeling Language atau yang biasa dikenal dengan singkatan UML, adalah bahasa standar yang digunakan untuk menggambarkan, menjelaskan, dan membangun perangkat lunak. Ini merupakan metode dalam pengembangan sistem

berbasis objek yang juga berfungsi sebagai alat bantu dalam proses pengembangan sistem. Beberapa alat bantu yang digunakan dalam perancangan berbasis objek dengan menggunakan UML meliputi diagram *use case*, *activity diagram*, *sequence diagram*, dan *class diagram* (Ayu Binangkit et al., 2023).

Menurut Siska Narulita et al. (2024), Unified Modelling Language (UML) merupakan sebuah bahasa yang divisualisasikan dalam bentuk gambar atau grafik yang berfungsi untuk memberikan gambaran dan spesifikasi dalam pembangunan dan dokumentasi dari sebuah pengembangan sistem berorientasi objek (object oriented).










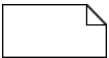
Unified Modeling Language (UML) berfungsi sebagai alat untuk mendeskripsikan dan merancang sistem perangkat lunak, terutama sistem yang dikembangkan menggunakan pendekatan pemrograman berorientasi objek. UML merupakan hasil penggabungan berbagai bahasa pemodelan grafis berorientasi objek yang berkembang pesat pada akhir 1980-an hingga awal 1990-an. Meskipun UML tidak terbatas pada metodologi tertentu, kenyataannya, UML paling sering digunakan dalam metodologi berbasis objek (Nistrina & Sahidah, 2022).

UML merupakan sebuah standar penulisan atau semacam *blue print* dimana didalamnya termasuk sebuah bisnis proses, penulisan kelas - kelas dalam sebuah bahasa yang spesifik. Terdapat beberapa diagram UML yang sering digunakan dalam pengembangan sebuah sistem menurut (Ramdany et al., 2024), yaitu:

a. *Use Case*: Merupakan gambaran dari fungsionalitas yang diharapkan dari sebuah sistem, dan mempresentasikan sebuah interaksi antara aktor dan sistem. Di Dalam *Use Case* terdapat aktor yang merupakan sebuah gambaran entitas dari

manusia atau sebuah sistem yang melakukan pekerjaan di sistem.

Tabel 2.1 Simbol Use Case

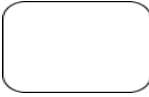




NO	GAMBAR	NAMA	KETERANGAN
1		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
2		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (<i>independent</i>).
3		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
4		<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara <i>eksplisit</i> .
5		<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
6		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
7		<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
8		<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor
9		<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (<i>sinergi</i>).
10		<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu

			sumber daya komputasi
--	--	--	-----------------------

Sumber : (Irfan et al., 2023)

b. *Activity Diagram*: Merupakan gambaran alir dari aktivitas - aktivitas di dalam sistem yang berjalan.

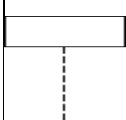
Tabel 2.2 Activity Diagram



NO	GAMBAR	NAMA	KETERANGAN
1		<i>Activity</i>	Memperlihatkan bagaimana masing- masing kelas antarmuka saling berinteraksi satu sama lain
2		<i>Action</i>	State dari sistem yang mencerminkan eksekusi dari suatu aksi
3		<i>Initial Node</i>	Bagaimana objek dibentuk atau diawali.
4		<i>Activity Final Node</i>	Bagaimana objek dibentuk dan dihancurkan
5		<i>Fork Node</i>	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran

Sumber : (Irfan et al., 2023)

c. *Sequence Diagram*: Menggambarkan interaksi antar objek di dalam dan di sekitar sistem yang berupa *message* yang digambarkan terhadap waktu.

Tabel 2.3 Sequence Diagram



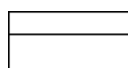


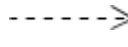
NO	GAMBAR	NAMA	KETERANGAN
1		<i>LifeLine</i>	Objek <i>entity</i> , antarmuka yang saling berinteraksi.

2		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi
3		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi

Sumber : (Irfan et al., 2023)

d. *Class Diagram*: Merupakan gambaran struktur dan deskripsi dari *class*, *package*, dan objek yang saling berhubungan seperti diantaranya pewarisan, asosiasi dan lainnya.

Tabel 2.4 Class Diagram

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
2		<i>Nary</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
3		<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
4		<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor
5		<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
6		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempegaruhi elemen yang bergantung padanya elemen yang tidak mandiri

7	————	<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya
---	------	--------------------	---

Sumber : (Irfan et al., 2023)

2.9 Bahasa Pemrograman

2.9.1 *Hyper Text Markup Language (HTML)*

HTML (*Hypertext Markup Language*) adalah sebuah bahasa *mark-up* yang digunakan untuk membuat sebuah halaman *web*, menampilkan berbagai informasi di dalam sebuah penjelajah web Internet dan pemformatan *hiperteks* sederhana yang ditulis dalam berkas format *ASCII* agar dapat menghasilkan tampilan wujud yang terintegrasi. Pada HTML dipergunakan *hypertext link* atau hubungan antara teks dan dokumen lain. Dengan demikian pembaca dokumen bisa melompat dari satu dokumen ke dokumen yang lain dengan mudah (Rio Ramdani et al., 2023).

HTML singkatan dari *Hypertext Mark-up Language*. HTML lebih merujuk ke Bahasa *markup* yang diperuntukkan untuk menentukan format atau *style* dari teks yang ditandai. Dokumen HTML merupakan dasar dari semua konten yang muncul di *World Wide Web (WWW)*, terdiri dari dua bagian penting: konten informasi dan satu set instruksi yang memberi tahu komputer Anda cara menampilkan konten tersebut (Mira Orisa et al., 2023).

Hypertext Mark-up Language (HTML) adalah bahasa *mark-up* yang digunakan untuk membuat halaman web. HTML berfungsi untuk memberikan struktur dan konten pada halaman web, seperti judul, teks, gambar, audio, video, dan tautan ke halaman web lainnya. HTML terdiri dari serangkaian elemen atau tag yang ditempatkan di dalam dokumen HTML. Setiap elemen HTML memiliki arti atau

fungsi tertentu, dan dapat diatur dengan menggunakan atribut, seperti warna, ukuran, atau style (Febriyani et al., 2023).

2.9.2 Cascading Style Sheets (CSS)

CSS kepanjangan dari *Cascading Style Sheet* adalah bahasa-bahasa yang merepresentasikan halaman web. Seperti warna, *layout*, dan *font*. Dengan menggunakan CSS, seorang *web developer* dapat membuat halaman web yang dapat beradaptasi dengan berbagai macam ukuran layar. Pembuatan CSS biasanya terpisah dengan halaman HTML. Meskipun CSS dapat disisipkan di dalam halaman HTML (Risti, 2023).

CSS adalah bahasa *Cascading Style Sheet* dan biasanya digunakan untuk mengatur tampilan elemen yang tertulis dalam Bahasa *mark-up*, seperti HTML. CSS berfungsi untuk memisahkan konten dari tampilan visualnya di situs. HTML dan CSS memiliki keterikatan yang erat, karena HTML adalah bahasa *mark-up* (fondasi situs) dan CSS memperbaiki *style* (untuk semua aspek yang terkait dengan tampilan *website*), maka kedua bahasa pemrograman ini harus berjalan beriringan (Rahmatuloh & Revanda, 2022).

Cascading Style Sheets (CSS) merupakan Bahasa pemrograman *mark-up* (HTML) yang biasa digunakan untuk membuat tampilan halaman *web* lebih menarik. *Cascading Style Sheets* (CSS) mengontrol format tampilan pada halaman HTML dan XHTML dan bisa juga diaplikasikan pada segala dokumen XML, SVG, dan XUL. CSS digunakan untuk memisahkan konteks utama dengan tampilan dokumen seperti *layout*, warna dan *font*. CSS merupakan rekomendasi dari W3C (*world wide web consortium*). *Style sheet* merupakan sebuah *text file*

sederhana yang berekstensi .css (Mira Orisa et al., 2023).

Cascading Style Sheet (CSS) merupakan file yang berisi rangkaian untuk mengatur komponen dalam sebuah halaman web sehingga akan lebih terstruktur dan rapi. Suatu *Style Sheet* merupakan tempat untuk mengontrol dan mengatur style-style yang ada. *Style Sheet* mendeskripsikan bagaimana tampilan dokumen HTML di layar. CSS dapat mengendalikan ukuran gambar, warna *text*, warna tabel, ukuran *border*, warna *hyperlink*, warna *mouse over*, spasi antar paragraf, spasi antar *text*, margin kiri, kanan, atas, bawah dan parameter lainnya. CSS terdiri dari selector, deklarasi, properti dan nilai (Sujarwo et al., 2023).

2.9.3 *Javascript*

JavaScript merupakan Bahasa pemrograman web, dimana sebagian besar situs *website* menggunakan *javascript*, dan semua *browser web modern* di desktop, tablet, dan ponsel menyertakan Bahasa *javascript*, menjadikan *javascript* yang merupakan bahasa pemrograman yang paling banyak atau umum digunakan. *JavaScript* merupakan bahasa pemrograman yang dinamis dan bahasa pemrograman dengan interpretasi yang sangat cocok untuk berorientasi objek dan *programming fungsional* (Fandopa & Santoso, 2022).

JavaScript merupakan bahasa pemrograman yang berupa kumpulan skrip. *JavaScript* berjalan pada suatu dokumen HTML. *JavaScript* sendiri adalah bahasa skrip pertama yang ada dalam sebuah website. Dengan adanya *javascript*, sebuah bahasa HTML mempunyai kemampuan tambahan dengan cara mengizinkan menjalankan perintah perintah di sisi user. *JavaScript* diselipkan kedalam dokumen HTML dan akan dijalankan saat halaman web diakses oleh pengguna

melalui *browser* ('Afifan, 2024).

Java Script adalah bahasa script yang biasa diletakkan bersama kode HTML untuk menentukan suatu aksi. JavaScript digunakan pada pemrograman web untuk meningkatkan atau menambah kemudahan pengguna ketika mengakses halaman web. Javascript adalah program dalam bentuk script yang dijalankan oleh interpreter yang telah ditanamkan kedalam browser web, sehingga browser web dapat mengeksekusi program javascript (Risti, 2023).

2.9.4 *Hypertext Preprocessor (PHP)*

PHP adalah singkatan dari “PHP: *Hypertext Preprocessor*”, yaitu bahasa pemrograman yang digunakan secara luas untuk penanganan pembuatan dan pengembangan sebuah situs web dan bisa digunakan bersamaan dengan HTML. PHP diciptakan oleh Rasmus Lerdorf pertama kali tahun 1994. PHP banyak dipakai untuk memrogram situs web dinamis (Rio Ramdani et al., 2023).

PHP (*Hypertext preprocessor*) merupakan *script* yang terintegrasi dengan HTML dan berada pada server. PHP juga dapat didefinisikan sebagai pembuat halaman *website* yang dinamis artinya halaman yang akan ditampilkan dibuat saat halaman itu diminta oleh klien. Ini dapat menyebabkan informasi yang diterima klien selalu yang terbaru. Semua *script* PHP dieksekusi pada server dimana *script* tersebut dijalankan. PHP juga dapat digunakan dalam berbagai sistem operasi seperti *Linux*, *Windows*, *Unix*. Fungsi PHP salah satunya untuk menangani kebutuhan standar pembuatan aplikasi *website*. PHP juga mempunyai banyak referensi bahasa *scripting* (Setiawan et al., 2022).

2.9.5 Structured Query Language (SQL)

Structured Query Language (SQL) merupakan konsep pengoprasian basisdata, untuk pemilihan, seleksi dan input data yang pengoprasian data dikerjakan dengan mudah yang pengoprasian data dikerjakan dengan mudah dan otomatis (Widiyanto et al., 2023). *Structured Query Language (SQL)* adalah bahasa pemrograman yang memberikan perintah untuk mengakses dan mengolah data pada sebuah sistem *database*. SQL banyak digunakan dalam manajemen basis data relasional sehingga dalam mengolah informasi perlu adanya data yang saling terhubung antara data satu dengan data lain nya. Dalam pengolahan data tersebut memiliki kemampuan untuk mengubah, menghapus mengatur data, memanipulasi dan menarik data yang tersimpan pada IBM *database management system* (Nandi Naufal et al., 2022).

2.10 Alat Bantu Pemrograman

2.10.1 My Structured Query Language (MySQL)

My Structured Query Language (MySQL) adalah sebuah perangkat lunak yang digolongkan sebagai RDBMS (*Relational Database Management System*). MySQL merupakan aplikasi yang berfungsi sebagai pengolahan database yang dikenal sangat cepat dalam mengirimkan dan menerima data selain itu juga multiuser dengan menggunakan perintah dasar SQL (Hanny & Ari Sulistiyawati, 2023).

My Structured Query Language (MySQL) merupakan suatu sistem manajemen database (*database management system*) atau DBMS, yaitu sistem yang berguna untuk melakukan proses pengaturan koleksi- koleksi struktur data (database) baik

yang meliputi proses pembuatan atau proses pengelolaan *database* (Suci et al., 2021).

Fungsi MySQL adalah sebuah database yang dapat digunakan sebagai *Client* maupun *Server*. Database MySQL merupakan suatu perangkat lunak database yang berbentuk database relasional atau disebut *Relational Database Management System* (RDBMS) yang menggunakan suatu bahasa permintaan yang bernama *Structured Query Language* (SQL) (Piu et al., 2021).

2.10.2 Laragon

Menurut Andarsyah et al. (2022) Laragon merupakan perangkat lunak open source yang berfungsi sebagai server virtual atau localhost, serta mendukung berbagai sistem operasi. Laragon memungkinkan penggunaan domain yang dapat disesuaikan sesuai kebutuhan, sering dikenal dengan istilah *pretty url's*. Aplikasi ini sangat cocok untuk mengelola aplikasi berbasis web secara efektif.

Laragon adalah perangkat lunak gratis yang dapat berjalan di berbagai sistem operasi dan berfungsi sebagai server mandiri atau localhost. Perangkat lunak ini menawarkan beragam layanan, alat, dan fitur yang mendukung pengembangan aplikasi berbasis web, seperti *Apache*, *MySQL*, *PHP Server*, *Memcached*, *Redis*, *Composer*, *Xdebug*, *PhpMyAdmin*, *Cmder*, hingga *Laravel*. Dengan berbagai fitur tersebut, Laragon menjadi solusi praktis dan efisien bagi pengembang untuk membangun dan mengelola aplikasi secara mudah (Handoyo & Anwar, 2023).

2.10.3 Website

Website merupakan sebuah platform yang berfungsi untuk menyampaikan informasi melalui internet dalam berbagai bentuk, seperti gambar, video, teks, suara, serta elemen interaktif yang memberikan pengalaman pengguna yang lebih dinamis. Salah satu fitur utama dari website adalah kemampuannya untuk menghubungkan satu dokumen dengan dokumen lainnya melalui hypertext, sehingga memudahkan pengguna dalam melakukan navigasi antar halaman atau sumber informasi terkait. Melalui perantara browser, website dapat diakses secara luas oleh pengguna di seluruh dunia, menjadikannya sebagai salah satu media yang paling efektif dalam penyampaian informasi, edukasi, maupun pengembangan bisnis di era digital (Alviano et al., 2023).

Website adalah suatu media yang terdiri dari beberapa halaman yang saling berkaitan satu sama lain, dan berfungsi sebagai media untuk menampilkan suatu informasi, baik berbentuk gambar, video, teks, suara, ataupun gabungan dari semuanya. *Website* bersifat *multi-platform* yang artinya dapat dibuka dari segala perangkat atau *device* yang terhubung dengan jaringan internet. Walaupun teknologi ini sudah cukup lama digunakan, namun saat ini masih banyak sekali perusahaan-perusahaan yang masih menggunakan *website* dalam menampilkan profil perusahaan (*company profile*), menjual produk, ataupun sebagai sistem yang dapat digunakan oleh pelanggan (Sonny, Sonny, 2021).

2.10.4 Visual Studio Code

Visual studio code merupakan code editor yang berfungsi sebagai software

pendukung, dalam pembuatan aplikasi. *Code editor* ini dapat dijalankan di berbagai perangkat desktop. *Visual Studio Code* sebagai *code editor* untuk menulis kode HTML, CSS dan PHP. Selain itu *code editor* ini dapat digunakan untuk melakukan koneksi dan operasi ke database MySQL sehingga dapat digunakan untuk menyiapkan database untuk aplikasi web yang sedang dibangun. *Visual studio code* juga merupakan *code editor yang open source* serta dapat digunakan untuk membuat dan mengedit *source code* dalam berbagai bahasa pemrograman (Budihartono, 2023).

2.10.5 Model View Controller (MVC)

Metode model view controller(MVC) adalah sebuah metode untuk membuat sebuah aplikasi dengan memisahkan data (model) dari tampilan (view) dan cara bagaimana memprosesnya (controller). MVC yang membangun aplikasi seperti memanipulasi data, antar muka pengguna dan bagian yang menjadi control dalam sebuah aplikasi web (A. Rohmad Basar, N. H. Adi, and S. Zebua, 2021).

Model View Controller (MVC) merupakan metode yang digunakan untuk merancang aplikasi web dengan memisahkan desain, data, dan proses. MVC merupakan model dalam perancangan aplikasi yang diurutkan berdasarkan komponen seperti basis data, interface, dan pengontrol. *Model View Controller* (MVC) digunakan untuk memisahkan logika bisnis dengan antarmuka pengguna agar para pengembang aplikasi lebih mudah dalam mengubah suatu bagian aplikasi tanpa mengubah bagian aplikasi lainnya (Oktasari & Kurniadi, 2020).

2.11 Framework Laravel

Laravel adalah framework berbasis PHP yang dirancang untuk mendukung optimasi dalam pengembangan website. Dengan menggunakan Laravel, website dapat berjalan lebih dinamis. Framework ini mudah dipahami dan dilengkapi dengan berbagai fitur, seperti otentikasi, routing, session manager, cache, dan komponen-komponen lain yang mempermudah pengembangan (Mangapul Siahaan & Wijaya, 2024).

Laravel menyediakan berbagai fitur yang memudahkan proses pengembangan aplikasi berbasis web, seperti routing, autentikasi, pengelolaan database, serta dukungan untuk migrasi database. Framework ini juga dikenal karena dokumentasinya yang lengkap dan komunitas pengembang yang aktif, yang menjadikannya mudah dipelajari dan diterapkan. Setelah melalui berbagai pembaruan dan penyempurnaan, Laravel mencapai kestabilan pada versi 5.2.36 yang dirilis pada 6 Juni 2016, dengan dukungan untuk sistem operasi lintas platform. Berkat keunggulan tersebut, Laravel telah menjadi salah satu framework PHP yang paling populer dan banyak digunakan oleh pengembang web di seluruh dunia (Rahmawati & Sumarsono, 2024).

2.12 Black Box Testing

Black box testing adalah metode pengujian perangkat lunak di mana tester fokus pada pengujian fungsionalitas sistem berdasarkan input yang diberikan, tanpa mengetahui struktur internal atau kode program. Metode ini bertujuan untuk memastikan bahwa seluruh persyaratan fungsional program tercakup dengan

memverifikasi apakah output yang dihasilkan sesuai dengan harapan dari berbagai kombinasi input yang diuji (Pratama et al., 2023).

Black box testing merupakan pengujian kualitas perangkat lunak yang berfokus pada fungsionalitas perangkat lunak. Pengujian *black box* bertujuan untuk menemukan fungsi yang tidak benar, kesalahan antarmuka, kesalahan pada struktur data, kesalahan performansi, kesalahan inisialisasi dan terminasi (Dwi Wijaya & Wardah Astuti, 2021).

2.13 Penelitian Terdahulu

Tabel 2.6 Penelitian Terdahulu

NO	JUDUL	PENULIS	TAHUN TERBIT	HASIL
1	Sistem Pakar Mendiagnosis Penyakit Spacelona Fawcetti Jenkins Pada Tanaman Jeruk Menggunakan Metode <i>Dempster Shafer</i>	Widiarti Rista Maya, Febry Andi Saragih, Elfitriani, Sri Murniyanti	2024	Diagnosis penyakit jeruk dengan akurasi 98%. Pencegahan dilakukan dengan pestisida Fungisida Dithiocarbamate.
2	Sistem Pakar Mendiagnosis Penyakit Tanaman Solanum Betaccum Menggunakan Metode <i>Dempster Shafer</i>	Depri Sabinus Lumbantu, Badrul Anwar, Muhammad Dahria	2022	Diagnosis penyakit Virus Kuning pada terong belanda dengan akurasi 50%. Membantu petani mengatasi masalah diagnosis.

NO	JUDUL	PENULIS	TAHUN TERBIT	HASIL
3	Sistem Pakar Diagnosis Hama dan Penyakit Bawang Merah Menggunakan Metode <i>Dempster Shafer</i>	Dasril Aldo	2021	Diagnosis hama dan penyakit bawang merah dengan akurasi 95%. Penanganan lebih terarah.
4	Sistem Pakar Diagnosis Penyakit Tanaman Kopi Menggunakan Metode <i>Dempster Shafer</i>	Bobby Riansyah, Didik Kurniawan, Made Same	2021	Diagnosis penyakit kopi berbasis web. Membantu meningkatkan produktivitas kopi.
5	Sistem Pakar Mendiagnosis Penyakit Pada Tanaman Mentimun Menggunakan Metode <i>Dempster Shafer</i> Berbasis Android	Syaiful Amri, R.Fanri Siahaan	2021	Diagnosis penyakit mentimun berbasis aplikasi Android. Memanfaatkan SQLite untuk pengelolaan data.
6	Sistem Pakar Mendiagnosis Penyakit Batu Karang Menggunakan Metode <i>Dempster Shafer</i>	Yolanda Wiguna, Faisal Tufik, Asyahri Hadi Nasyuha	2022	Diagnosis penyakit batu karang berbasis web. Mempermudah akses diagnosis untuk masyarakat umum.
7	Sistem Pakar Diagnosis Penyakit Tanaman Padi Menggunakan Forward Chaining dan <i>Dempster Shafer</i>	Dianmita Putri	2022	Diagnosis penyakit padi dengan akurasi 81,11%. Sistem mendapatkan kategori “Baik” pada evaluasi responden.

NO	JUDUL	PENULIS	TAHUN TERBIT	HASIL
8	Sistem Pakar Diagnosis Hama dan Penyakit Bawang Putih Menggunakan Metode Dempster Shafer	Risaldo Ratinamo	2021	Diagnosis penyakit bawang putih berbasis web. Membantu petani mendeteksi penyakit sejak dini.
9	Sistem Pakar Mendiagnosis Penyakit Pada Tanaman Kopi Robusta Menggunakan Metode <i>Dempster Shafer</i>	Tisah Endang Togatorop, Azlan, Ita Mariami	2022	Diagnosis penyakit kopi robusta. Membantu meningkatkan produktivitas kopi di Indonesia.
10	Sistem Pakar Diagnosis Penyakit Pada Tanaman Vanili Menggunakan Metode <i>Dempster Shafer</i> Berbasis WEB	Nanda, Bambang Hermanto, Albertus Sudirman	2021	Diagnosis penyakit vanili berbasis web dengan akurasi 99,50%. Responden menilai sistem berkategori “Sangat Baik”.

BAB 3

METODOLOGI PENELITIAN

3.1 Pendahuluan

Penelitian ini merupakan suatu kegiatan yang penting karena dengan penelitian ini akan didapatkan data yang berhubungan dengan tema dari penelitian tersebut. Selain itu, penelitian merupakan suatu kegiatan yang saling berhubungan antara kegiatan satu dengan kegiatan lainnya. Oleh karena itu, penting penulis merancang kerangka kerja dalam penelitian ini secara bertahap, sehingga setiap tahapan akan dapat dilihat pencapaian dan hasil akhir.

Pada sistem pakar diagnosis penyakit tanaman jeruk dengan metode *Dempster Shafer* ini dapat diakses siapapun yang sedang mengalami permasalahan pada tanaman jeruknya. Sistem ini dapat memberikan pemahaman kepada para petani jeruk dan masyarakat yang mempunyai tanaman jeruk juga tentang apa saja yang terjadi pada tanaman jeruknya dan bagaimana cara penanganannya.

Sistem ini dapat mendiagnosis awal dari gejala yang ditimbulkan. Pada sistem ini akan menyertakan informasi rekomendasi penanganan (produk) yang bisa digunakan dalam mengatasi permasalahan penyakit tanaman jeruk tersebut. Tentunya, sistem pakar ini juga memberikan informasi mengenai jenis penyakit tanaman jeruk berdasarkan gejala yang dialami, dan saran perawatan yang bisa dilakukan oleh para petani dan masyarakat.

Pada tahapan ini digunakan notasi-notasi yang berlaku dalam perancangan sistem flowchart untuk menggambarkan arus data sistem sehingga dapat membantu

dalam proses analisis sistem pakar. Flowchart digunakan untuk menggambarkan sistem baru yang akan dikembangkan secara logika terlebih dahulu mempertimbangkan lingkungan fisik dimana data tersebut diproses.

3.2 Kerangka Kerja Penelitian

Kerangka kerja penelitian merupakan langkah-langkah yang akan dilakukan dalam menyelesaikan permasalahan dalam penelitian ini. Dibawah ini telah digambarkan alur dari kerangka kerja penelitian sebagai berikut.



Gambar 3. 1 Kerangka Kerja Penelitian

1. Identifikasi Masalah

Melalui pengamatan secara langsung dan pengalaman pribadi yang dirasakan sendiri didapatkan identifikasi masalah terhadap objek ini dan dilakukan dengan

maksud agar dapat mengetahui secara jelas permasalahan yang terjadi. Setelah diidentifikasi masalah bahwa banyak masyarakat yang masih belum bisa menangani permasalahan penyakit pada tanaman buah-buahnya, khususnya tanaman jeruk. Sehingga, menyebabkan tanaman jeruk tersebut mengalami permasalahan penyakit yang serius dan berdampak buruk juga dengan tanamannya jeruknya.

Ini disebabkan masyarakat masih kesulitan untuk mendapatkan informasi dalam mengidentifikasi mengenai jenis penyakit tanaman jeruk yang dialami, saran atau tindakan perawatan yang harus dilakukan dan rekomendasi (produk) yang sesuai untuk mengatasi penyakit tanaman tersebut. Ketersediaan pakar pada bidangnya pun menjadi salah satu faktor, sehingga membutuhkan waktu yang lama untuk mengidentifikasi jenis penyakit apa saja dikarenakan keterbatasan itu. Tentunya juga untuk melakukan perawatan pada tanaman tersebut juga membutuhkan pengetahuan yang pas dan prosedur yang tentunya sesuai juga. Karena kalau tidak dilakukan seperti itu, berakibat fatal pada tanaman jeruk itu sendiri dan mengakibatkan kerugian bagi para petani maupun para pembudidayaan.

2. Analisis Masalah

Setelah menganalisis masalah yang terjadi diatas, diidentifikasilah bahwasanya adalah keterbatasan dalam mendapatkan informasi dan pengetahuan para petani dan pembudidayaan tanaman jeruk, sehingga lambat dalam melakukan tindakan perawatan yang menyebabkan gejala/penyakit yang dialami tanaman jeruk semakin memburuk. Pada dasarnya semua gejala-gejala yang dialami tanaman

jeruk tersebut bukanlah suatu hal yang berbahaya, karena bisa bisa atasi dengan cara penanganan yang tepat sesuai dengan gejala apa saja yang alami tanaman jeruk sehingga bisa dianggap menjadi penyakit tanaman tersebut.

Namun, dengan adanya gejala-gejala tersebutlah yang membuat menurunnya tingkat hasil panen buah jeruk itu sendiri dikarenakan adanya penghambat baik factor eksternal maupun factor internalnya dalam proses pertumbuhan tanaman tersebut baik pada akar, batang, daun, buahnya. Untuk melakukan perawatan dan penanganan terhadap gejala-gejala tersebut, tentunya akan memakan waktu yang lama dan menghabiskan banyak waktu untuk menjumapi seorang ahli pakar dan tentunya juga biaya yang dikeluarkan juga mahal. Dan salah satunya faktornya juga adalah keterbatasan seorang ahli pakar dibidang tersebut serta jauh dari jangkauan tempat tinggal para petani maupun pembudidayaan.

Oleh karena itu, perlu adanya media informasi dan pengetahuan yang memberi pemahaman terhadap para petani dan pembudidayaan, tentang bagaimana penanganan terhadap masalah yang dialami, saran atau tindakan perawatannya seperti apa saja dan rekomendasi (produk) ataupun bahan alami yang bisa digunakan dalam perawatan jenis gejala/penyakit tersebut. Pengembangan pada Sistem Pakar, yang mana sistem ini bekerja dengan cara memindahkan pengetahuan pakar ke dalam sistem komputer. Jadi, sistem ini dapat mengatasi permasalahan yang terjadi pada tanaman jeruk, meningkatkan efisiensi, dan memberikan informasi seperti pengetahuan pakar.

3. Studi Literatur

Pada tahap ini setelah melakukan identifikasi dan analisis masalah, peneliti mempelajari dan mendalami literatur-literatur yang berhubungan dengan permasalahan yang terjadi. Kemudian, satu persatu literatur dipelajari dan dipilihlah literatur yang berkaitan dengan penelitian ini supaya tidak keluar dari lingkup pembahasan masalah yang ditangani. Sumber literatur tersebut didapatkan dari jurnal dan artikel yang membahas tentang sistem informasi dan bahan bacaan lainnya yang mendukung penelitian.

4. Pengumpulan Data

Setelah tahap studi literature diatas, selanjutnya tahap pengumpulan data yang dapat dilakukan dengan beberapa cara yaitu :

a. Observasi

Dengan melakukan pengamatan secara langsung terhadap objek yang akan diteliti. Metode ini bertujuan untuk dapat mengetahui langsung bagaimana alur kerja yang terjadi pada objek yang akan diteliti tersebut.

b. Wawancara

Peneliti melakukan *interview* (wawancara) ini dilakukan secara langsung kepada para ahli pakar pada bidangnya untuk mendapatkan penjelasan dari masalah-masalah yang sebelumnya tidak diketahui maupun kurang jelas dan untuk menyakinkan bahwa data yang diperoleh atau dikumpulkan benar-benar akurat dan untuk mendapatkan nilai kepastian dari setiap gejala-gejala untuk menentukan jenis penyakit yang dialami tanaman jeruk.

5. Analisis dengan Metode Dempster-Shafer

Pada tahap ini setelah data-data telah terkumpul peneliti melakukan analisis dengan metode *dempster-Shafer*. Penerapan *dempster-Shafer* dalam sistem pakar ini diimplementasi dengan representasi, kombinasi dan propogasi ketidakpastian, dimana teori ini memiliki beberapa karakteristik yang secara instutif sesuai dengan cara berfikir seorang pakar, namun dasar matematika yang kuat. Karena *Dempster Shafer* ini merupakan teori pembuktian matematika berdasarkan nilai dan *belief* dan *plausibility* yang mana *Plausibility* adalah ukuran ketidakpercayaan terhadap *evidence*/gejala. Jika bernilai 1 maka mengindikasikan bahwa tidak ada kepastioan, dan jika bernilai 0 menunjukkan adanya kepastian.

6. Perancangan Sistem

Perancangan sistem merupakan tahap lanjutan dari tahap sebelumnya yaitu pengumpulan data. Perancangan sistem meliputi rencana bagaimana kegiatan-kegiatan dalam siklus pengembangan sistem dapat diterapkan secara efektif dan efisien sehingga mampu menghasilkan sebuah sistem yang sesuai dengan tujuan yang dimaksud. Tahapan desain terhadap kebutuhan sistem dari hasil analisis menggunakan *Data Flow Diagram (DFD)*, *Entity Relationship Diagram (ERD)*, dan *Flow Chart*.

7. Pembuatan Sistem

Setelah tahap perancangan sistem, selanjutnya adalah tahap pembuatan sistem. Pada tahap pembuatan sistem ini dilakukan untuk membuat program yang diperoleh perancangan program dari data yang ada. Tahap-tahap yang dilakukan

untuk penelitian guna perancangan dan pembuatan sistem tersebut secara terstruktur guna untuk menghasilkan sebuah sistem yang sesuai dengan kebutuhan.

8. Pengujian Sistem

Pengujian sistem dilakukan setelah tahap pembuatan sistem dilakukan, Pengujian ini dilakukan bertujuan agar aplikasi yang dibuat sesuai dengan kebutuhan pengguna. Selain itu, pengujian ini juga dilakukan untuk memastikan fungsionalitas dan logika dari sistem berjalan dengan baik tanpa terjadi error.

Pengujian yang dilakukan terdiri dari:

- a. Pengujian *blackbox*, digunakan untuk menguji tingkat kemampuan *user interface* terhadap sistem yang dibangun.
- b. Pengujian *User Acceptance Test (UAT)*.

9. Implementasi Sistem

Setelah pengujian sistem, selanjutnya tahap implementasi sistem. Pada tahap ini dilakukan Implementasi yang merupakan suatu konversi dari desain aplikasi yang telah dirancang ke dalam sebuah program komputer dengan aplikasi berbasis *web* dengan menggunakan bahasa pemrograman yang telah ditentukan, maka dari itu dapat diimplementasikan/diterapkan kedalam Sistem Pakar Diagnosis Penyakit Tanaman Jeruk Menggunakan Metode *Dempster Shafer*.