

BAB II

TINJAUAN PUSTAKA DAN LANDASAN TEORI

2.1 Tinjauan Pustaka

Pembuatan sistem pakar untuk berbagai jenis bidang keilmuan telah banyak dibuat dan diteliti oleh orang-orang yang ahli ataupun mempunyai minat untuk mendalami bidang ini. Salah satu penelitian mengenai sistem pakar dalam bidang pertanian dilakukan oleh Pramudita Eka Hananto (2012) mengembangkan sistem diagnosa hama dan penyakit tanaman berdasarkan gejala-gejala yang ada pada tanaman cengkih dengan menggunakan metode inferensi forward chaining.

Berbeda dengan penelitian yang dilakukan Pramudita Eka Hananto (2012), Slamet Tridianto (2016) membuat sistem pakar diagnosa penyakit tanaman cabai merah dengan metode backward chaining berbasis web.

Kemudian penelitian serupa dilakukan oleh Wisnu Raharjo (2014) dalam skripsinya yang berjudul “Sistem Pakar Diagnosa Penyakit Tanaman Jambu Citra Menggunakan Metode Forward Chaining”, yang melakukan penelitian identifikasi penyakit dengan berdasarkan obyek pengamatan yang ada, misalnya daun, buah dan batang. Tujuan dari pengamatan ini adalah untuk memudahkan proses identifikasi tersebut, dengan menjawab pertanyaan dari sistem mengenai gejala-gejala yang terjadi pada tanaman jambu citra berdasarkan pengamatan maka jenis penyakit yang menyerang tanaman jambu citra dapat diketahui, kemudian setelah terdeteksi ditampilkan pula cara penanggulangan hama dan penyakitnya.

Selanjutnya, penerapan sistem pakar tidak hanya dalam bidang pertanian saja, bidang kesehatanpun telah mengimplementasikan sistem pakar. Penelitian yang dilakukan oleh Dedi Kurniawan (2009) adalah membangun sebuah sistem pakar pediagnosa penyakit berbasis web.

Dengan adanya berbagai macam penelitian yang dikembangkan oleh beberapa peneliti pada berbagai bidang, penulis mencoba untuk mengimplementasikan sistem pakar dalam bidang pertanian terutama dalam bidang hama dan penyakit tanaman pada tanaman tembakau, sistem ini dibuat untuk mendiagnosa hama atau penyakit yang diderita tanaman tembakau yang dilakukan layaknya seorang pakar terhadap pemilik

tanaman tembakau yang bermasalah. Sehingga dapat diketahui jenis hama atau penyakit dan bagaimana tata cara pengendaliannya, sebagai langkah pencegahan awal.

2.2 Sistem Pakar

Sistem pakar merupakan bagian dari kecerdasan buatan yang berisi kombinasi pemahaman teoritis tentang suatu persoalan dan sekumpulan aturan pemecahan heuristik yang dikembangkan oleh manusia untuk dapat memecahkan problema pada suatu domain yang spesifik.

Secara umum sistem pakar merupakan pengumpulan, gambaran dan penyimpanan pengetahuan sistem pakar ke dalam computer dan selanjutnya dapat diakses oleh pengguna. Dalam penyusunannya, sistem pakar mengkombinasikan kaidah-kaidah penarikan kesimpulan (*Inference Rule*) dengan basis pengetahuan tertentu yang diberikan oleh atau lebih pakar dalam bidang tertentu. Kombinasi dari dua hal tersebut disimpan dalam computer, yang selanjutnya digunakan dalam proses dalam pengambilan keputusan untuk penyelesaian masalah tersebut.

Kelebihan sistem pakar :

1. Menyimpan pengetahuan dan keahlian dari para pakar.
2. Memungkinkan orang awam dapat mengerjakan pekerjaan para ahli.
3. Bisa melakukan proses berulang secara otomatis.
4. Memiliki kemampuan untuk mengakses pengetahuan.
5. Menghemat waktu dalam pengambilan keputusan.
6. Dapat memecahkan masalah lebih cepat dari pada kemampuan manusia dengan catatan data yang sama.

Kekurangan sistem pakar :

1. Biaya yang diperlukan untuk membuat, memelihara dan mengembangkannya relative mahal.
2. Sulit dikembangkan, hal ini erat kaitannya ketersediaan pakar dibidangnya dan kepakaran sangat sulit diekstrak dari manusia.
3. Sistem pakar tidak 100% benar karena seseorang yang terlibat dalam pembuatan sistem pakar tidak selalu benar.

2.2.1 Konsep Dasar Sistem Pakar

Konsep dasar sistem pakar mengandung keahlian, pakar, pengalihan keahlian, inferensi, aturan dan kemampuan menjelaskan. Keahlian adalah suatu kelebihan penguasaan pengetahuan di bidang tertentu yang diperoleh dari pelatihan, membaca atau pengalaman. Pakar adalah seseorang yang mampu menjelaskan tanggapan, mempelajari hal-hal baru seputar topik permasalahan, menyusun kembali pengetahuan jika dipandang perlu, memecahkan aturan-aturan jika dibutuhkan dan menentukan relevan tidaknya keahlian mereka (Turban, 2005). Pengalihan keahlian dari para ahli ke komputer kemudian untuk dialihkan lagi ke orang lain yang bukan ahli, hal ini merupakan tujuan utama dari sistem pakar. Proses ini membutuhkan 4 aktivitas yaitu :

1. Tambahkan pengetahuan (dari para ahli atau sumber-sumber lainnya).
2. Representasi pengetahuan (ke komputer)
3. Inferensi pengetahuan
4. Pengalihan pengetahuan ke user

Pengetahuan yang disimpan dikomputer tersebut dengan nama basis pengetahuan. Ada 2 tipe pengetahuan, yaitu : fakta dan prosedur (biasanya berupa aturan).

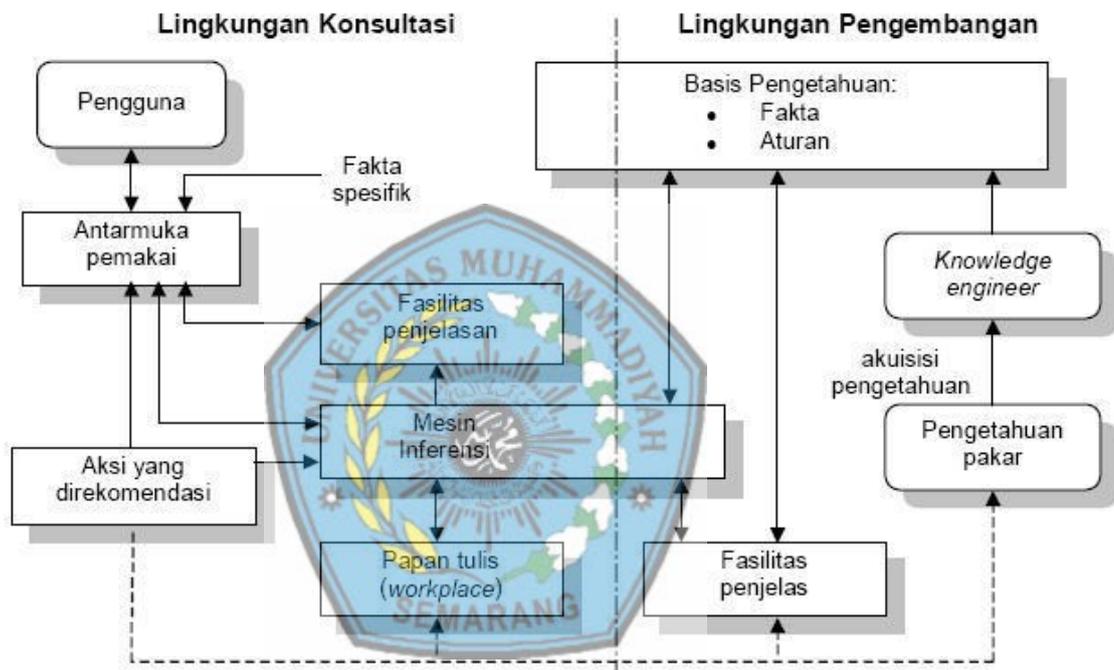
Salah satu fitur yang harus dimiliki oleh sistem pakar adalah kemampuan untuk menalar, jika keahlian-keahlian sudah tertimba sebagai basis pengetahuan dan sudah tersedia program untuk mengakses basis data, maka komputer harus dapat deprogram untuk membuat inferensi. Proses inferensi ini dikemas dalam bentuk motor inferensi (*inference engine*).

Sebagian sistem pakar komersial dibuat dalam bentuk rule-based systems, yang mana pengetahuannya disimpan dalam bentuk aturan-aturan. Aturan tersebut biasanya berbentuk IF-THEN.

Fitur lainnya dari sistem pakar adalah kemampuan untuk merekomendasikan, sistem inilah yang membedakan antara sistem pakar dengan sistem konvensional.

2.2.2 Struktur Sistem Pakar

Sistem pakar terdiri dari 2 bagian pokok, yaitu : lingkungan pengembang (development involvement) dan lingkungan konsultasi (consultation involvement). Lingkungan pengembang digunakan sebagai pembangunan sistem pakar baik dari segi pembangunan komponen maupun basis pengetahuan. Lingkungan konsultasi digunakan oleh orang yang bukan ahli berkonsultasi.



Gambar 2.1 Struktur Sistem Pakar

1. Subsistem penambahan pengetahuan

Bagian ini digunakan untuk memasukkan pengetahuan, mengkonstruksi atau memperluas pengetahuan dalam basis pengetahuan. Pengetahuan itu berasal dari : ahli, buku, basisdata, penelitian dan gambar.

2. Basis Pengetahuan

Berisi pengetahuan-pengetahuan yang dibutuhkan untuk memahami, memformulasikan dan menyelesaikan masalah. Basis pengetahuan tersusun atas fakta dan kaidah. Fakta adalah informasi tentang objek, peristiwa atau situasi.

Kaidah adalah cara untuk membangkitkan suatu fakta baru dari fakta yang sudah diketahui.

3. Motor Inferensi (*Inference engine*)

Program yang berisi metodologi yang digunakan untuk melakukan penalaran terhadap informasi-informasi dalam basis pengetahuan, serta digunakan untuk memformulasikan konklusi. Terdapat 3 elemen utama dalam motor inferensi, yaitu :

- a. *Interpreter* : mengeksekusi item-item agenda yang erpilih dengan menggunakan aturan-aturan dengan basis pengetahuan yang sesuai.
- b. *Scheduler* : mengontrol agenda
- c. *Consistency enforcer* : akan berusaha memelihara ke konsistenan dalam mempresentasikan solusi yang bersifat darurat.

4. Workspace

Merupakan arena dalam memori yang digunakan untuk merekam kejadian yang sedang berlangsung termasuk keputusan sementara. Ada 3 tipe keputusan yang dapat direkam yaitu :

- a. Rencana : bagaimana mengadai masalah.
- b. Agenda : aksi-aksi potensial yang sedang menunggu dieksekusi.
- c. Solusi : calon aksi yang akan dibangkitkan.

5. Antarmuka (Interface)

Merupakan batasan antara dua entitas, berfungsi sebagai penengah untuk media komunikasi antara user dan program. Berikut ini merupakan contoh antarmuka untuk pengguna :

- a. GUI : Graphical User Interfae
- b. WUI : Web besad User Interface

6. Subsistem Penjelasan

Digunakan untuk melacak respond an memberikan penjelasan tentang kelakuan sisem pakar seara interaktif melalui pertanyaan. Contohnya sebagai berikut :

- a. Mengapa suatu pertanyaan ditanyakan oleh sistem pakar?
- b. Bagaimana konklusi dicapai?

- c. Mengapa ada alternative yang dibatalkan?
 - d. Rencana apa yang digunakan untuk mendapatkan solusi?
7. Perbaikan

Sistem ini digunakan untuk mengevaluasi kinerja sistem pakar itu sendiri untuk melihat apakah pengetahuan-pengetahuan yang ada masih cocok untuk digunakan dimasa mendatang.

2.2.3 Klasifikasi Sistem Pakar

Pada penerapannya ada beberapa bidang aplikasi yang sesuai dengan teknologi sistem pakar. Bidang-bidang tersebut antara lain :

1. Kontrol

Sistem pakar digunakan untuk mengontrol kegiatan yang membutuhkan presisi waktu yang tinggi. Misalnya industri - industri berteknologi tinggi.

2. Prediksi

Keunggulan dari seorang pakar adalah kemampuannya memprediksi kedepan. Misal yang mudah ditemui, bagaimana seorang pakar meteorologi memprediksi cuaca besok berdasarkan data-data sebelumnya. Kemampuan ini juga dimiliki sistem pakar, penggunaan sistem pakar prediksi misalnya untuk peramal cuaca, penentuan masa tanam dan sebagainya.

3. Interpretasi

Sistem pakar ini digunakan untuk menganalisis data-data yang tidak lengkap, tidak teratur dan data kontradiktif. Misalnya untuk menginterpretasi citra.

4. Pengajaran

Sistem pakar ini digunakan untuk mengajar mulai dari SD sampai mahasiswa perguruan tinggi. Kelebihan dari sistem pakar untuk mengajar adalah membuat diagnose apa penyebab kekurangan dari seorang siswa, kemudian memberikan cara untuk memperbaikinya.

5. Perencanaan

Penggunaan sistem pakar untuk perencanaan sangat luas, mulai dari perencanaan mesin-mesin sampai manajemen bisnis. Penggunaan sistem pakar

ini akan menghemat biaya, waktu dan material, sebab pembuatan model sudah tidak dibutuhkan lagi. Misalnya penggunaan sistem konfigurasi komputer, tata letak sirkuit dan seandainya.

6. Diagnosis

Sistem pakar diagnosis biasanya digunakan untuk merekomendasikan obat untuk orang sakit, kerusakan mesin, kerusakan rangkaian elektronik, hama dan penyakit tanaman dan sebagainya. Prinsipnya adalah menemukan masalah atau kerusakan yang terjadi, sistem pakar diagnosis adalah jenis sistem pakar yang sangat populer saat ini. Biasanya sistem pakar diagnosis menggunakan pohon keputusan (decision tree) sebagai representasi pengetahuannya. Kebanyakan sistem pakar diagnosis menggunakan shell, sehingga sangat mudah untuk melakukan perubahan pada basis pengetahuannya.

2.2.4 Sifat Sistem Pakar

Sistem pakar merupakan program-program praktis yang menggunakan strategi heuristik yang dikembangkan oleh manusia untuk memecahkan permasalahan-permasalahan yang spesifik. Karena keheuristikannya itu dan sifatnya yang berdasarkan pada pengetahuan maka umumnya sistem pakar bersifat sebagai berikut :

1. Terbuka untuk diperiksa, baik dalam menampilkan langkah-langkah maupun dalam menjawab pertanyaan-pertanyaan tentang proses tertentu. Sangatlah penting bagi sebuah sistem pakar mempunyai sifat terbuka terhadap pemeriksaan, karena alasannya :
 - a. Untuk mempermudah penambahan sejumlah informasi atau aturan baru untuk memperbaharui basis pengetahuan dalam rangka mengembangkan kinerjanya.
 - b. Memuaskan user, akan kebenaran jawaban yang diberikan oleh sistem pakar.
 - c. Setiap aspek dan keputusan yang diambil selama proses untuk mendapatkan solusi dapat dievaluasi dengan baik.
2. Mudah dimodifikasi baik dengan menambah maupun menghapus suatu pengetahuan berbasis pengetahuan.

3. Heuristik dalam menggunakan pengetahuan (yang sering sekali tidak sempurna) untuk memperoleh solusi.
4. Fasilitas penalaran/penjelasan
Sistem pakar akan dapat memberikan informasi tentang kesimpulan yang diambil komputer dan memperlihatkan kaidah-kaidah yang dipergunakan serta urutan yang dilaksanakan.

2.2.5 Karakteristik Sistem Pakar

Karakteristik umum yang membedakan sistem pakar dengan perangkat lunak biasa adalah :

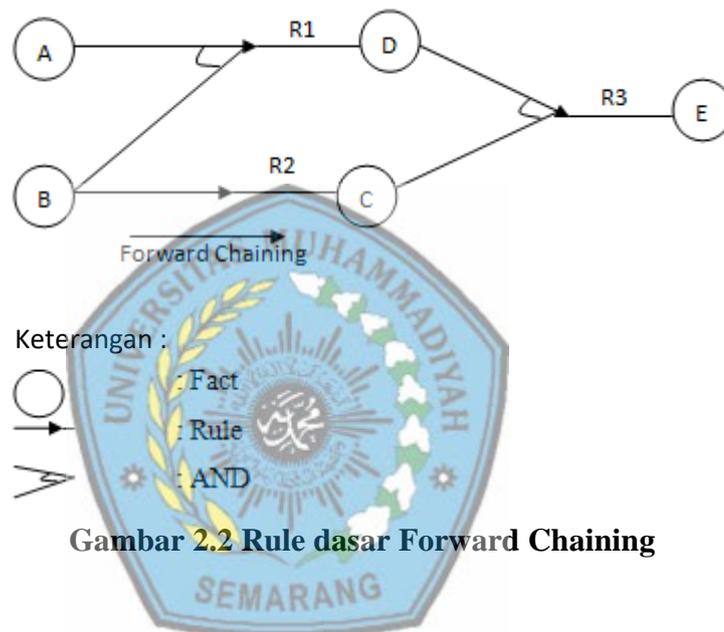
1. Terdapat banyak kemungkinan jawaban
Akan memakan banyak waktu untuk menguji dan mempelajari jawaban itu, karena ruang persoalan (problem space) berukuran besar dan tidak pasti.
2. Data kabur
Sistem pakar mencari konklusi yang tidak pasti karena informasi yang dipakainya sering berupa data yang kabur . Meskipun demikian sistem pakar diharapkan dapat member keputusan yang tergolong baik, dalam arti tingkat kesalahan tidak terlalu besar.
3. Heuristic
Bersifat heuristic dalam menggunakan pengetahuan untuk memperoleh suatu solusi.
4. Fasilitas informasi
Sistem pakar dapat memberikan kemudahan jawaban kepada user, sehingga user akan merasa puas dengan jawaban yang diberikan sistem pakar.

2.3 Metode Forward Chaining

Metode forward chaining adalah metode pencarian atau teknik pelacakan ke depan yang di mulai dari informasi yang ada dan penggabungan rule untuk menghasilkan suatu kesimpulan atau tujuan. (Russel S, Norvig P, 2003). Forward chaining merupakan proses peruntutan yang dimulai dengan menampilkan kumpulan data atau fakta yang meyakinkan menuju konklusi akhir. Jadi metode forward chaining

dimulai dari informasi masukan (if) dahulu kemudian menuju konklusi (then) atau dapat dimodelkan sebagai berikut :

Rule base	Workspace
R1 : IF A AND B THEN D	A,B
R2 : IF B THEN C	
R3 : IF C AND D THEN E	



2.3.1 Karakteristik Forward Chaining

Metode forward chaining memiliki beberapa karakteristik berikut ini :

- a. Perencanaan, monitoring dan control.
- b. Disajikan untuk masa depan.
- c. Antecedent ke konsekuen.
- d. Data memadu, penalaran dari bawah ke atas.
- e. Bekerja ke depan untuk mendapatkan solusi apa untuk mengikuti fakta.
- f. Breadth first dimudahkan.
- g. Antecedent menentukan pencarian.
- h. Penjelasan tidak di fasilitasi.

2.3.2 Kelebihan dan Kekurangan Metode Forward Chaining

Kelebihan Metode Forward Chaining :

1. Metode forward chaining akan bekerja dengan baik ketika problem bermula dari mengumpulkan/menyatukan informasi lalu kemudian mencari kesimpulan apa yang dapat diambil dari informasi tersebut.
2. Metode ini mampu menyediakan banyak sekali informasi dari hanya jumlah kecil data.

Kekurangan Metode Forward Chaining :

1. Kemungkinan tidak ada cara mengenali dimana beberapa fakta lebih penting dari fakta lainnya.
2. Sistem bisa saja menanyakan yang tidak berhubungan, walaupun jawaban dari pertanyaan tersebut penting. Namun hal ini akan membingungkan user untuk menjawab pada subjek yang yang tidak berhubungan.

2.4 Kaidah Produksi

Metode kaidah produksi biasanya dituliskan dalam bentuk jika-maka (IF-THEN). Kaidah produksi dapat dikatakan sebagai hubungan implikasi dua bagian yaitu bagian premis (jika) dan bagian konklusi (maka). Pada dasarnya penyajian pengetahuan secara kaidah produksi terdiri dari dua hal, yaitu :

1. Antecedent, bagian yang mengekspresikan situasi atau premis (pernyataan berawalan IF).
2. Consequent, bagian yang menyatakan suatu tindakan atau kesimpulan yang diperoleh jika premis bernilai benar (pernyataan berawalan THEN).

Konklusi atau kesimpulan yang dinyatakan pada bagian THEN dinyatakan benar, jika pernyataan IF pada bagian sistem tersebut juga benar atau sesuai dengan aturan tersebut.

Setiap aturan produksi dalam basis pengetahuan mengimplementasikan sebagai keahlian otonom yang dapat dikembangkan dan dimodifikasi secara terpisah dari aturan lain. Pada saat digabungkan dan dimasukkan ke dalam mesin inferensi, kumpulan

aturan berlaku sinergi dan memberikan hasil yang lebih dari pada jumlah aturan individu (Turban,2005).

Kelebihan dari Kaidah Produksi :

a. *Expressiveness dan Intuittiveness*

Kaidah-kaidah produksi secara esensial meyakinkan apa yang dilakukan dalam situasi tertentu.

b. *Simplicity*

Struktur seragam dari sintaks IF-THEN dalam sistem berbasis kaidah memberikan suatu kesederhanaan yang menarik untuk direpresentasi pengetahuan. Iri ini mengingatkan suatu keadaan yang dapat dibaca dari kaidah produksi dan komunikasi antara berbagai bagian dari program unggulan.

c. *Modularity dan Modifiability*

Kaidah produksi mengkodekan bentuk diskret informasi yang secara umum tidak berhubungan dengan kaidah produksi yang lain, kecuali jika ada kaidah produksi eksplisit yang menghubungkan mereka. Informasi dapat dihapus atau ditambah dari sistem asalkan tidak mengakibatkan efek samping yang mengganggu. Ciri modulasi sistem produksi menunjukkan kenaikan perbaikan dan setelan yang bagus dari sistem produksi dengan tidak mengalami pengurangan kinerja.

d. *Knowledge Intensive*

Basis pengetahuan disusun dari kaidah produksi yang pada gilirannya sangat utama adalah pengetahuan murni, karena membutuhkan isi bukan control maupun informasi pemrograman. Karena setiap kaidah produksi ekuivalen untuk suatu ringkasan dan kejelasan kalimatnya maka masalah sematic diselesaikan dengan struktur representasi.

Keurangan dari Kaidah Produksi :

- a. Pengetahuan yang kompleks membutuhkan kaidah yang banyak, yang memungkinkan sulit dalam membuatnya yang digunakan untuk sistem maupun perawatannya.

- b. Suatu sistem mempunyai banyak kaidah atau mempunyai batasan pencarian dalam mengontrol program. Ada beberapa program mempunyai kesulitan dalam mengevaluasi sistem berbasis kaidah dan membuat inferensi.

2.5 Hama dan penyakit

Peningkatan hasil pekebunan bisa dilakukan melalui beberapa upaya, diantaranya upaya perlindungan tanaman dari serangan hama dan penyakit. Tanaman yang tumbuh baik dan diperkirakan memberi hasil yang tinggi, terkadang tidak menjadi kenyataan, hanya karena serangan hama dan penyakit.

Istilah hama dan penyakit sering dikira sama, padahal sebenarnya keduanya berbeda. Hama merupakan organisme yang dianggap mengganggu atau merusak tanaman sehingga pertumbuhan dan perkembangannya terganggu. Sedangkan penyakit merupakan keadaan tanaman yang terganggu pertumbuhannya tetapi penyebabnya bukan binatang (hama). Penyebab penyakit dapat berupa bakteri, virus, jamur maupun karena kekurangan unsur hara.

Pengamatan terhadap hama dan penyakit tanaman sangat dibutuhkan untuk mengetahui sejauh mana tanaman terserang. Pengamatan tersebut berupa pengamatan kasar dan pengamatan halus. Pengamatan kasar untuk mengetahui secara kualitatif keadaan umum tanaman. Sedangkan pengamatan halus untuk memperoleh data kuantitatif yang meliputi populasi hama dan tingkat kerusakan tanaman dan berat serangan. Data tersebut dapat diperoleh dengan berdasarkan pengamatan beberapa tanaman sempe yang tersebar secara sistematis.

Berikut merupakan contoh hama dan penyakit yang menyerang tanaman tembakau dengan gejala serangannya :

Tabel 2.1 Hama dan Penyakit Tanaman

ID	PENYAKIT	GEJALA				SOLUSI
		AKAR	BATANG	DAUN	LAINNYA	
P01	Ulat Daun (<i>Spodoptera Litura</i> Dan <i>Prodenia Litura</i>).			1. Timbul lubang tidak beraturan. 2. Terdapat bekas gigitan berwarna putih.		1. Dapat dilakukan dengan menyempotkan isektisida kimia seperti regent, curacron atau prevaton.
P02	Kutu Tembakau (<i>Myzus Persicae</i>)			1. Daun lengket 2. Adanya cendawa berwarna putih		1. Mengurangi pemupukan 2. Melakukan penyemprotan insektisida
P03	Nematoda (<i>Meloydogyne sp</i>)	1. Terdapat bisul-bisul bulat pada akar	1. Batang kerdil	1. Daun layu 2. Daun berguguran		1. Dapat dilakukan pencegahan dengan pengolahan lahan yang baik, menjaga sanitasi lingkungan tanaman dan membersihkan gulma.
P04	Rebah Kecambah				1. Pangkal bibit berlekuk seperti terjepit 2. Bibit busuk 3. Bibit berwarna coklat	1. Pengaturan jarak tanam pembibitan. 2. Disinfeksi tanaman sebelum penaburan benih 3. Penyeprotan pembibitan 4. Pencelupan bibit

Tabel 2.1 Hama dan Penyakit Tanaman (Lanjutan)

ID	PENYAKIT	GEJALA				SOLUSI
		AKAR	BATANG	DAUN	LAINNYA	
P05	Penyakit Lanas		<ol style="list-style-type: none"> 1. Pangkal batang busuk berwarna coklat. 2. Batang menggantung 3. Empelur tanaman bersekat-sekat. 	<ol style="list-style-type: none"> 1. Daun mendadak layu 2. Terdapat bercak kelabu pada daun 		<ol style="list-style-type: none"> 1. Melakukan sanitasi pengolahan tanah yang matang memperbaiki drainase penggunaan pupuk kandang yang telah masak. 2. Penyemprotan fugsida pada pangkal batang dengan menggunakan fungsida makozeb, benomil atau propanokrab hidroklorida.
P06	Penyakit Kerupuk			<ol style="list-style-type: none"> 1. Daun berkerut 2. Tepi daun melengkung ke atas. 3. Tulang daun bengkok 4. Daun menebal 5. Daun berkerut 6. Daun kasar 		<ol style="list-style-type: none"> 1. Bersihkan tanaman inang 2. Segera cabut tanaman yang terserang dan musnahkan dengan cara dibakar.

Tabel 2.1 Hama dan Penyakit Tanaman (Lanjutan)

ID	PENYAKIT	GEJALA				SOLUSI
		AKAR	BATANG	DAUN	LAINNYA	
P07	Patik Daun			1. Terdapat bercak putih coklat pada daun 2. Daun rapuh		1. Lakukan isolasi pada daun tang bawah. 2. Daun yang sudah tua segera dipetik. 3. Semprot persemaian dengan bubuk Bordeaux atau bubuk California.
P08	Busuk Daun / Busuk Bakteri		1. Batang terkulai	1. Daun mendadak layu 2. Daun busuk		1. Rotasi tanaman, mencabut dan memusnahkan tanaman terinfeksi 2. Penyemprotan bakterisida seperti plantomicyn, bactoxyn atau agrept.
P09	Hangus Batang		1. Batang mengering 2. Batang berwarna coklat sampai hitam			1. Cabut tanaman yang terserang lalu dibakar 2. Untuk pencegahan awal menggunakan natural glio

Tabel 2.1 Hama dan Penyakit Tanaman (Lanjutan)

ID	PENYAKIT	GEJALA				SOLUSI
		AKAR	BATANG	DAUN	LAINNYA	
P10	Kutu kebul		1. Tanaman kerdil	1. Daun rusak		<ol style="list-style-type: none"> 1. Menjaga sanitasi 2. Membersihkan gulma 3. Penyemprotan akarisida bamex, demolish, agrimex, alfamex atau Pegasus.
P11	Kumbang Perusak Daun			<ol style="list-style-type: none"> 1. Daun terdapat bekas gigitan berkelok-kelok 2. Daun dan tunas habis 		<ol style="list-style-type: none"> 1. Menaburkan nematisida pada pangkal batang tanaman.
P12	Peyakit tepung			<ol style="list-style-type: none"> 1. Permukaan daun terdapat bercak putih 2. Daun sepenuhnya tertutup lapisan berwarna putih 		<ol style="list-style-type: none"> 1. Menjaga kebun jangan sampai lembab. 2. Segera petik daun yang sudah terserang

Tabel 2.1 Hama dan Penyakit Tanaman (Lanjutan)

ID	PENYAKIT	GEJALA				SOLUSI
		AKAR	BATANG	DAUN	LAINNYA	
P13	Busuk Tangkai		<ol style="list-style-type: none"> 1. Batang berlubang 2. Tangkai busuk 3. Empulunya berlubang dan busuk 			<ol style="list-style-type: none"> 1. Seleksi daun sebeum disimpan digudang 2. Jarum penyunduk sebelum digunakan 3. Disinfeksi dengan tepol
P14	Penyakit Karat			<ol style="list-style-type: none"> 1. Terdapat bintik-bintik yang meluas di permukaan daun 		<ol style="list-style-type: none"> 1. Menggunakan varietas resisten 2. Mendesinfeksi benih kedalam larutan formalin 0,2% selama 15 menit 3. Rotasi tanaman 4. Menanam tepat pada waktunya

Tabel 2.1 Hama dan Penyakit Tanaman (Lanjutan)

ID	PENYAKIT	GEJALA				SOLUSI
		AKAR	BATANG	DAUN	LAINNYA	
P15	Penyakit Rhizoctonia Sp				1. Benih tidak tumbuh 2. Bibit yang baru dicabut akan tampak akarnya busuk 3. Tanaman yang baru dipindahkan dari persemaian akan layu secara tiba-tiba	1. Benih didesinfeksi dengan larutan sublimat 1% selama 15 menit 2. Seterilkan tanah dengan larutan formalin 4% 3. Semprotkan larutan bubuk berdeaux K.O.C 4. Semprot dengan larutan fungisida dithane M 45 0,2%
P16	Penyakit virus		1. Pertumbuhan tanaman lambat	1. Daun terdapat bercak kuning		1. Menjaga sanitasi kebun 2. Cabut dan bakar tanaman yang terinfeksi

2.6 Rekayasa Perangkat Lunak

Rekayasa perangkat lunak adalah suatu disiplin ilmu yang membahas semua aspek produksi perangkat lunak, mulai tahap awal yaitu analisa kebutuhan pengguna, menentukan spesifikasi dari kebutuhan pengguna, desain, pengkodean, pengujian sampai pemeliharaan sistem setelah digunakan.

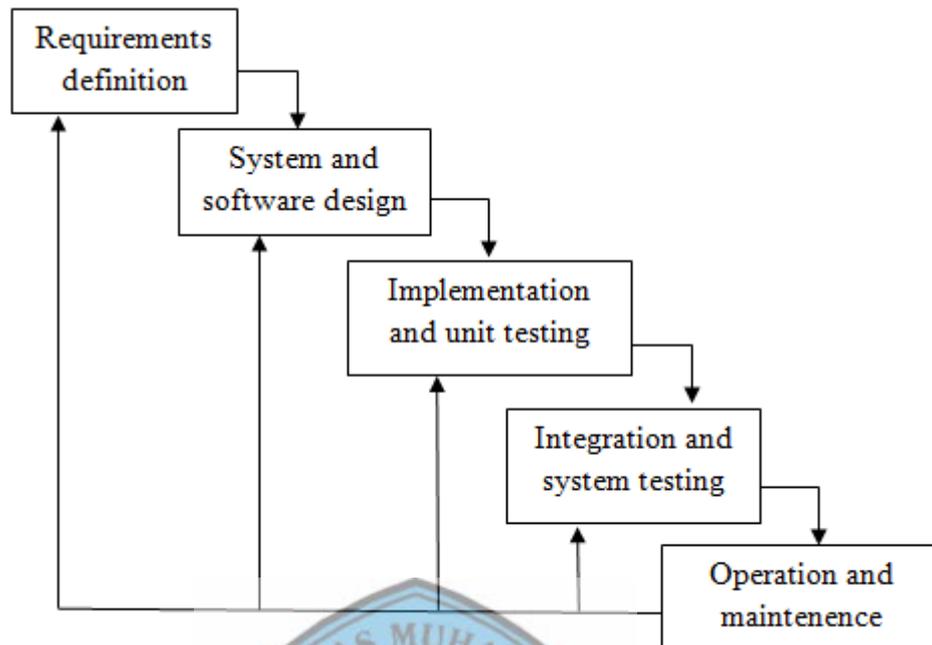
Ruang lingkup rekayasa perangkat lunak :

1. Software requirements berhubungan dengan spesifikasi kebutuhan dan persyaratan perangkat lunak
2. Software desain mencakup proses penampilan arsitektur, komponen, antar muka dan karakteristik lain dari perangkat lunak.
3. Software construction berhubungan dengan detail pengembangan perangkat lunak termasuk algoritma, pengkodean, pengujian dan pencarian kesalahan.
4. Software testing meliputi pada keseluruhan perilaku perangkat lunak.
5. Software maintenance mencakup upaya-upaya perawatan ketika perangkat lunak telah dioperasikan.
6. Software configuration management berhubungan dengan usaha perubahan konfigurasi perangkat lunak untuk memenuhi kebutuhan tertentu.
7. Software engineering management berkaitan dengan pengelolaan dan pengukuran rekayasa perangkat lunak, termasuk perencanaan proyek perangkat lunak.
8. Software engineering tools and methods mencakup kajian teoritis tentang alat bantu dan metode rekayasa perangkat lunak.

2.6.1 Model Waterfall

Model waterfall merupakan sebuah metode pengembangan software yang bersifat sekuensial. Inti dari metode waterfall adalah pengerjaan dari suatu sistem dilakukan secara berurutan atau secara linear. Jadi jika langkah satu belum dikerjakan maka tidak akan bisa melakukan pengerjaan langkah 2, 3 dan seterusnya. Secara otomatis tahapan ke-3 akan bisa dilakukan jika tahap ke-1 dan ke-2 sudah dilakukan.

Fase-fase dalam Waterfall Model :



Gambar 2.3 Model Waterfall (Sommerville)

Keterkaitan dan pengaruh antar tahap ini ada karena output sebuah tahap dalam Waterfall Model merupakan input bagi tahap berikutnya, dengan demikian ke tidak sempurnaan hasil pelaksanaan tahap sebelumnya adalah awal ketidaksempurnaan tahap berikutnya. Memperhatikan karakteristik ini, sangat penting bagi tim pengembang untuk secara bersama-sama melakukan analisa kebutuhan dan desain sistem sesempurna mungkin sebelum masuk ke dalam tahap penulisan kode program. Secara garis besar metode waterfall mempunyai langkah langkah sebagai berikut : Analisa, Design, Code dan Testing, Penerapan dan Pemeliharaan.

1. Analisa kebutuhan (*Requirement Analysis*)/(*Requirements analysis and definition*)

Langkah ini merupakan analisa terhadap kebutuhan sistem. Pengumpulan data dalam tahap ini bisa malakukan sebuah penelitian, wawancara atau study literatur. Seorang sistem analis akan menggali informasi sebanyak-banyaknya dari user sehingga akan tercipta sebuah sistem komputer yang bisa melakukan tugas-tugas yang diinginkan oleh user tersebut. Tahapan ini akan menghasilkan dokumen user requirement atau bisa dikatakan sebagai data yang berhubungan dengan keinginan user dalam pembuatan

sistem. Dokumen ini lah yang akan menjadi acuan sistem analis untuk menterjemahkan ke dalam bahasa pemrogram.

2. Design sistem (*System Design*)

Proses desain akan menerjemahkan syarat kebutuhan ke sebuah perancangan perangkat lunak yang dapat diperkirakan sebelum dibuat coding. Proses ini berfokus pada : struktur data, arsitektur perangkat lunak, *representasi interface*, dan detail (*algoritma*) prosedural. Tahapan ini akan menghasilkan dokumen yang disebut software requirment. Dokumen inilah yang akan digunakan programmer untuk melakukan aktivitas pembuatan sistemnya.

3. Coding & Testing/penulisan kode Program (*Implementation*)

Coding merupan penerjemahan design dalam bahasa yang bisa dikenali oleh komputer. Dilakukan oleh programmer yang akan meterjemahkan transaksi yang diminta oleh user. Tahapan ini lah yang merupakan tahapan secara nyata dalam mengerjakan suatu sistem. Dalam artian penggunaan komputer akan dimaksimalkan dalam tahapan ini. Setelah pengkodean selesai maka akan dilakukan testing terhadap sistem yang telah dibuat tadi. Tujuan testing adalah menemukan kesalahan-kesalahan terhadap sistem tersebut dan kemudian bisa diperbaiki.

4. Penerapan / pengujian program (*Integration & Testing*)

Tahapan ini bisa dikatakan final dalam pembuatan sebuah sistem. Setelah melakukan analisa, design dan pengkodean maka sistem yang sudah jadi akan digunakan oleh user.

5. Pemeliharaan (*Operation & Maintenance*)

Perangkat lunak yang sudah disampaikan kepada pelanggan pasti akan mengalami perubahan. Perubahan tersebut bisa karena mengalami kesalahan karena perangkat lunak harus menyesuaikan dengan lingkungan (periperal atau sistem operasi baru) baru, atau karena pelanggan membutuhkan perkembangan fungsional.

Keuntungan Metode Waterfall

- a. Kualitas dari sistem yang dihasilkan akan baik. Ini dikarenakan oleh pelaksanaannya secara bertahap. Sehingga tidak terfokus pada tahapan tertentu.
- b. Document pengembangan sistem sangat terorganisir, karena setiap fase harus terselesaikan dengan lengkap sebelum melangkah ke fase berikutnya. Jadi setiap fase atau tahapan akan mempunyai dokumen tertentu.
- c. Metode ini masih lebih baik digunakan walaupun sudah tergolong kuno, daripada menggunakan pendekatan asal-asalan. Selain itu, metode ini juga masih masuk akal jika kebutuhan sudah diketahui dengan baik.

Kelemahan waterfall

- a. Diperlukan manajemen yang baik, karena proses pengembangan tidak dapat dilakukan secara berulang sebelum terjadinya suatu produk.
- b. Kesalahan kecil akan menjadi masalah besar jika tidak diketahui sejak awal pengembangan yang berakibat pada tahapan selanjutnya.
- c. Pelanggan sulit menyatakan kebutuhan secara eksplisit sehingga tidak dapat mengakomodasi ketidakpastian pada saat awal pengembangan.
- d. Pelanggan harus sabar, karena pembuatan perangkat lunak akan dimulai ketika tahap desain sudah selesai. Sedangkan pada tahap sebelum desain bisa memakan waktu yang lama.
- e. Pada kenyataannya, jarang mengikuti urutan sekuensial seperti pada teori. Iterasi sering terjadi menyebabkan masalah baru.

2.7 Diagram Alir (*Flow Chart*)

Flow chart adalah gambaran dalam bentuk diagram alir dari algoritma-algoritma dalam suatu program, yang menyatakan arah alur program tersebut.

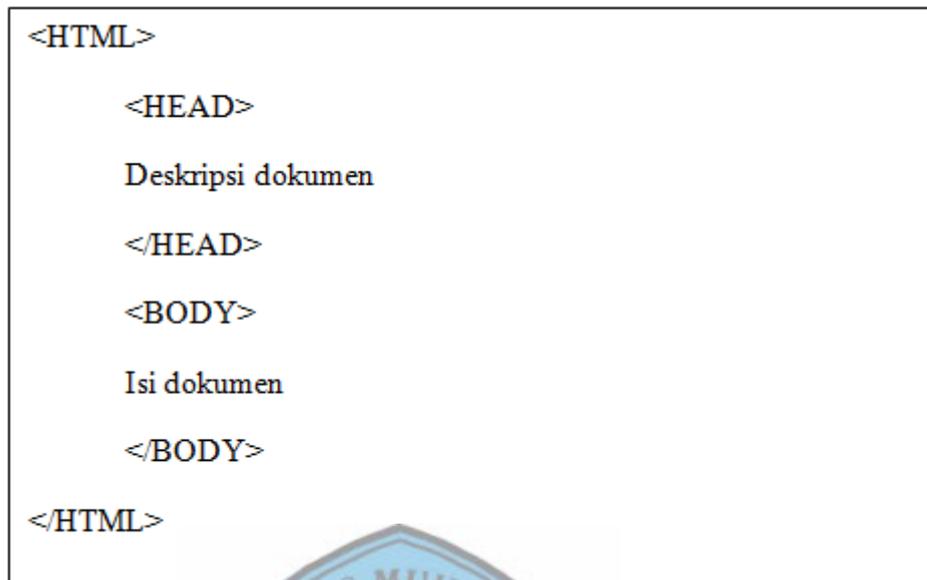
Tabel 2.2 Simbol Flow Chart

NO	Simbol	Keterangan
1.		Menunjukkan awal/akhir program
2.		Menunjukkan input/output
3.		Menunjukkan arus
4.		Menunjukkan proses
5.		Menunjukkan pengujian

2.8 *Hypertext Markup Language (HTML)*

HTML merupakan bahasa yang digunakan untuk menulis halaman web. Biasanya mempunyai ekstensi .htm, .html, .shtml (Suyanto,2007). Hypertext Markup Language (HTML) adalah sebuah bahasa markah 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. Dengan kata lain, berkas yang dibuat dalam perangkat lunak pengolah kata dan disimpan dalam format ASCII normal sehingga menjadi halaman web dengan perintah-perintah HTML.

Kerangka dasar HTML :



Gambar 2.4 Kerangka dasar HTML

Keterangan :

- HTML : Setiap dokumen HTML biasanya diawali dan ditutup dengan tag HTML, yang memberitahu bahwa dokumen tersebut adalah dokumen HTML.
- HEAD : Dibagian ini biasanya dimuat tag *TITLE* yang menampilkan judul halaman pada bagian title milik browser
- BODY : Dokumen *body* digunakan untuk menampilkan text, image link dan semua yang akan ditampilkan pada halaman web.

2.9 PHP (Hypertext Preprocessor)

PHP merupakan bahasa standart yang digunakan dalam dunia website. PHP adalah bahasa pemrograman yang berbentuk script yang diletakkan di dalam server web (Nugroho,2004). PHP banyak dipakai untuk memprogram situs web dinamis. PHP disebut bahasa pemrograman server side karena PHP diproses pada komputer server. Hal ini berbeda dibandingkan dengan bahasa pemrograman client-side seperti Java Script yang diproses pada web browser (*client*). PHP dapat digunakan dengan gratis (*free*) dan bersifat *Open Source*.

Kelebihan PHP dari bahasa pemrograman lain :

1. Bahasa pemrograman PHP adalah sebuah bahasa script yang tidak melakukan sebuah kompilasi dalam penggunaannya.
2. Web Server yang mendukung PHP dapat ditemukan dimana - mana dari mulai IIS sampai dengan apache, dengan konfigurasi yang relatif mudah.
3. Dalam sisi pengembangan lebih mudah, karena banyaknya milis - milis dan developer yang siap membantu dalam pengembangan.
4. Dalam sisi pemahaman, PHP adalah bahasa scripting yang paling mudah karena referensi yang banyak.
5. PHP adalah bahasa open source yang dapat digunakan di berbagai mesin (linux, unix, windows) dan dapat dijalankan secara runtime melalui console serta juga dapat menjalankan perintah-perintah system.

Kekurangan PHP dari bahasa pemrograman lain ;

1. PHP Tidak mengenal Package.
2. Jika tidak di encoding, maka kode PHP dapat dibaca semua orang & untuk meng encodingnya dibutuhkan tool dari Zend yang mahal sekali biayanya.
3. PHP memiliki kelemahan keamanan. Jadi Programmer harus jeli & berhati-hati dalam melakukan pemrograman & Konfigurasi PHP.

Cara penulisan PHP

PHP bersifat case sensitif yang artinya semua penulisannya harus sesuai dengan kamus data yang tersedia. Secara mendasar php memiliki struktur sebagai berikut :

```
<?php
//Ini contoh dalam php
Echo "<h2> contoh penulisan php</h2>";
$nama = "nama saya</p>";
Echo "<p>$nama</p>";
$nilai = 10;
$hasil = $nilai + 10;
Echo $hasil;
?>
```

Gambar 2.5 Kerangka dasar PHP

Kode diatas adalah contoh sederhana struktur PHP, berikut penjelasannya :

1. Awal kode harus diawali dengan "<?php" dan diakhiri dengan "?>", 2 perintah tersebut harus ada pada tiap bagian yang memanggil fungsi php.
2. Dalam barisan perintah php, kita bebas menyisipkan komentar dengan diawali tanda "//". Komentar digunakan untuk mempermudah mengingat kumpulan baris dengan proses yang sama
3. echo digunakan untuk mengirimkan hasil ke browser, sehingga yang dikirimkan adalah bagian yang berada setelah perintah echo
4. setiap variabel harus diawali dengan tanda dolar "\$" dan dapat dipanggil kapanpun dalam halaman yang sama.

2.10 XAMPP

XAMPP adalah satu paket web server yang mudah dioperasikan berbagai sistem operasi. Dalam paketnya sudah ada Apache (web server), MYSQL(data base), PHP (server side scripting), dan berbagai pustaka bantu. XAMPP tersedia untuk linux, Windows, MacOS maupun Solaris sehingga sangat memudahkan pembuatan webserver multiplatform.

2.11 Notepad ++

Notepad ++ adalah sebuah text editor yang sangat memudahkan dalam pemrograman dibanding notepad biasa. Notepad++ menggunakan komponen Scintilla untuk dapat menampilkan dan menyunting teks dan berkas kode sumber berbagai bahasa pemrograman yang berjalan diatas sistem operasi Microsoft Windows. Notepad++ ini dilisensikan *free/ gratis*, karena sourceforge.net memfasilitasi Notepad++ membebaskan untuk digunakan.

2.12 Macromedia dreamweaver 5

Macromedia Dreamweaver 5 adalah HTML editor profesional untuk mendesain, coding, dan pengembangan website , halaman web, dan aplikasi berbasis web. Dreamweaver memberikan kemudahan dengan *tools* yang dapat membantu dan memudahkan dalam berkreasi dan bereksperimen.

2.13 Pengujian *Black Box*

Pengujian *black box* berfokus pada persyaratan fungsional perangkat lunak. Dengan demikian, pengujian black box memungkinkan perancang perangkat lunak mendapatkan serangkaian kondisi input yang sepenuhnya menggunakan semua persyaratan fungsional untuk suatu program.