

**PERANGKAT LUNAK DIAGNOSA HAMA DAN PENYAKIT PADA
TANAMAN PADI BERBASIS SISTEM PAKAR**

Skripsi

Oleh

DIAN NINDARIANSARI



**FAKULTAS TEKNIK
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2017**

ABSTRAK

PERANGKAT LUNAK DIAGNOSA HAMA DAN PENYAKIT TANAMAN PADI BERBASIS SISTEM PAKAR

Oleh

DIAN NINDARIANSARI

Peran penting pertanian bagi kehidupan manusia sangat dibutuhkan, hal ini dikarenakan ada hasil pertanian yang digunakan sebagai makanan pokok yaitu beras. Kendala yang dialami oleh petani yaitu kerusakan yang disebabkan oleh serangan hama dan penyakit yang terlambat dikendalikan sehingga mencapai tahap yang meluas. Ahli pertanian dalam hal ini mempunyai pengetahuan khusus mengenai pengendalian serangan tersebut, akan tetapi tempat dan waktu menjadi kendala utama petani untuk dapat berkonsultasi dengan seorang ahli. Untuk itu perlu dibuat perangkat lunak diagnosa hama dan penyakit tanaman padi yang dapat membantu pengambilan keputusan dalam menentukan jenis kerusakan dari gejala-gejala yang ditimbulkan.

Penelitian ini membahas tentang cara kerja seorang ahli pertanian yang diadopsi pada sistem pakar. Pembuatan sistem ini menggunakan metode rekayasa perangkat lunak metode *waterfall* dengan bahasa pemrograman amzi prolog. Metode *waterfall* terdiri dari 5 tahapan, yaitu analisa kebutuhan, perancangan (*design*), implementasi, pengujian dan pemeliharaan (*maintenance*) perangkat lunak (*software*).

Sistem pakar yang dikembangkan memiliki kemampuan untuk mengelola data berupa gejala kerusakan, jenis penyebab kerusakan dan cara pengendalian kerusakan pada tanaman padi. Hasil yang didapat dari pengujian fungsionalitas sistem pakar dari 30 jenis kerusakan yang diuji dan terbagi dalam 6 kategori pengujian adalah total pengujian yang sukses sebesar 90% dan kegagalan pengujian sebesar 10%.

Kata kunci: Prolog (*Program Logic*), Sistem Pakar, Padi

ABSTRACT

PEST AND DISEASE OF RICE PLANTS DIAGNOSTIC SOFTWARE BASED ON EXPERT SYSTEMS

By

DIAN NINDARIANSARI

Agriculture have an important role in human existence, this is due to the important agricultural product which is used for primary food. The problem faced by the farmer is the damage caused by pest and diseases which is too late to be controlled thefore, it is widespread. Agricultural expert have knowledge to mitigate the pest, however, there are problems that faced by farmer to consult with the expert concerning availability of time and place. Therefore, a software to diagnose pest and disease of rice plant has to be made in order to help making decision requirement base on the symptoms that is caused.

This development of this expert system use amzi prolog tool to implement the model into software. Waterfall model consist of five steps: design requirement, design analysis, implementation, testing and maintenance.

Functional test of expert systems from 30 items divided into 5 categories and resulted in 90% running successfully and 10% failed. The expert system been developed has the ability to process data of symptoms of damage, types of damage and dagame control of rice plants.

Keywords: Prolog, Expert System, Rice

**PERANGKAT LUNAK DIAGNOSA HAMA DAN PENYAKIT PADA
TANAMAN PADI BERBASIS SISTEM PAKAR**

Oleh

DIAN NINDARIANSARI

Skripsi

**Sebagai Salah Satu Syarat untuk Mencapai Gelar
SARJANA TEKNIK**

Pada

**Jurusan Teknik Elektro
Fakultas Teknik Universitas Lampung**



**FAKULTAS TEKNIK
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2017**

Judul Skripsi : **PERANGKAT LUNAK DIAGNOSA HAMA
DAN PENYAKIT TANAMAN PADI
BERBASIS SISTEM PAKAR**

Nama Mahasiswa : **Dian Nindariansari**


Nomor Pokok Mahasiswa : 1015031033

Jurusan : Teknik Elektro

Fakultas : Teknik

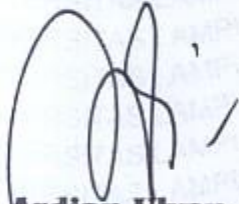
MENYETUJUI

1. Komisi Pembimbing


Ing. Hery Dian Septama, S.T.
NIP 19850915 200812 1 001


Meizano Ardhi M, S.T., M.T.
NIP 19810528 201212 1 001

2. Ketua Jurusan Teknik Elektro


Dr. Ing. Ardian Ulvan, M.Sc.
NIP 19731128 199903 1 001

MENGESAHKAN

1. Tim Penguji

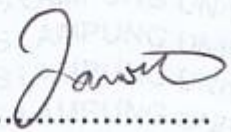
Ketua

: **Ing. Hery Dian Septama, S.T.**

.....

Sekretaris

: **Meizano Ardhi M, S.T., M.T.**

.....

Penguji

Bukan Pembimbing : **Yessi Mulyani, S.T., M.T.**

.....
.....

2. Dekan Fakultas Teknik



Prof. Drs. Suharno, M.Sc., Ph.D.

NIP 19620717 198703 1 002

Tanggal Lulus Ujian Skripsi : **27 Juli 2017**

RIWAYAT HIDUP



Penulis dilahirkan di Palembang Provinsi Sumatera Selatan pada tanggal 18 April 1991, sebagai anak ke tiga dari empat bersaudara dari Bapak Djunaidi dan Ibu Darmilah Maad, Spd,Sd..

Riwayat pendidikan penulis dimulai dari TK YWKA Palembang diselesaikan pada tahun 1997, Sekolah Dasar (SD) Negeri 612 Palembang diselesaikan pada tahun 2003, Sekolah Menengah Pertama (SMP) Negeri 40 Palembang diselesaikan pada tahun 2006 dan Sekolah Menengah Atas (SMA) Negeri 13 Palembang diselesaikan pada tahun 2009.

Pada tahun 2010, penulis terdaftar sebagai mahasiswi Jurusan Teknik Elektro Fakultas Teknik Universitas Negeri Lampung melalui jalur Ujian SNMPTN 2010. Pada tahun 2015, penulis melaksanakan kerja praktik di PT Semen Baturaja (PERSERO) Tbk Unit Panjang Jl. Yos Sudarso Km. 7, Panjang, Way Lunik Teluk Betung, Bandar Lampung pada 23 Februari- 23 Maret 2015.

SURAT PERNYATAAN

Dengan ini menyatakan bahwa dalam penelitian ini tidak terdapat karya yang pernah dilakukan oleh orang lain dan sepanjang sepengetahuan saya juga tidak terdapat karya atau pendapat yang ditulis atau diterbitkan oleh orang lain, kecuali tertulis diacu dalam naskah ini sebagaimana yang disebutkan di dalam daftar pustaka. Selain itu saya menyatakan pula bahwa penelitian ini dibuat oleh saya sendiri.

Apabila pernyataan saya tidak benar maka saya bersedia dikenakan sanksi sesuai dengan hukum yang berlaku.

Bandar Lampung, 25 Oktober 2017



Dian Nindariansari
NPM.1015031033

Motto:

*Berjuanglah tanpa pernah takut
gagal, karena bagi Allah SWT
tidak ada yang mustahil selagi
ikhtiar dan doa sesuai di
panjatkan.*

*Bealive to **ALLAH SWT***

SANWACANA

Alhamdulillahirobbil'alamin, segala puji bagi Allah SWT atas limpahan nikmat hidup, kesehatan,serta hidayah-Nya sehingga penulis dapat menyelesaikan skripsi ini. Sholawat serta salam senantiasa tercurah kepada Nabi besar Muhammad SAW sang penutup para Nabi dan Rasul hingga akhir zaman, berserta keluarga, sahabat, dan pengikutnya.

Penelitian dengan judul **“Perangkat Lunak Diagnosa Hama dan Penyakit Tanaman Padi Berbasis Sistem Pakar”** sebagai salah satu syarat untuk memperoleh gelar Sarjana Teknik pada Jurusan Teknik Elektro Fakultas Teknik Universitas Lampung.

Dalam kesempatan ini penulis mengucapkan terima kasih yang tulus kepada :

1. Bapak Ing. Hery Dian Septama, S.T.,M.Sc., sebagai Dosen Pembimbing Utama atas kesediaanya membimbing, membantu, meluangkan waktu serta memberikan saran dan kritik kepada penulis dalam menyelesaikan penulisan skripsi ini.
2. Bapak Meizano A.M. Djausal, S.T., M.T., sebagai Dosen pembimbing Pendamping atas kesediaannya membimbing, membantu, meluangkan waktu serta memberi saran dan kritik kepada penulis dalam penyelesaian penulisan skripsi ini.

3. Ibu Yessi Mulyani, S.T., M.T., Sebagai Dosen Penguji yang telah memberikan kritikan dan saran dalam penyelesaian tugas akhir ini.
4. Bapak Agus Trisanto, S.T., M.T., selaku Pembimbing Akademik.
5. Bapak Dr. Ardian Ulvan, S.T., M.Sc., selaku Ketua Jurusan Teknik Elektro Universitas Lampung.
6. Bapak Dr. Herman H Sinaga, Selaku Sekertaris Jurusan Teknik Elektro Universitas Lampung.
7. Segenap Dosen dan pegawai di Jurusan Teknik Elektro yang telah memberikan ilmu dan wawasan yang tak terlupakan oleh penulis.
8. Orang tua tercinta Bapak. Djunaidi dan Ibu Darmilah, M., Spd.Sd., yang telah membesarkan dengan segenap kasih sayang yang tulus kepada penulis.
9. Abang dan Adik saya dr. Daddy Cahyadinata, Wahyudi Darma Rizky, dan Gina Aretha Safalia yang selalu memberikan semangat, dukungan, doa, dan kasih sayang tak terhingga.
10. Kakak ipar dr. Pusvita Anggraini dan Novita Sari yang telah memberikan doa dan dukungannya.
11. Keluarga besarku (Alm.) Yosoef pt, dan para saudara sepupuan yang tidak pernah bosan mengingatkan untuk menyelesaikan tugas akhir ini : dr. Fieka Soraya, M. Fachrialdi, S.T., M. Irfani, S.T., M.T., dr. M. Fachrendi, dr. Agung Evasha Putra Pusvita Wulandari, S.P. yang telah memberikan kasih sayang, doa, dan semangatnya.

12. Orang terdekat yang selalu menemani dan membantu dari awal hingga akhir perjuangan penyelesaian tugas akhir ini Robby Triady Maulana, Amd
13. Teman seperjuangan dan sepersaudaraan keluarga besar Teknik Elektro 2010 atas segala bantuan tenaga, doa, materi, semangat, dan dukungan dalam setiap kesulitan yang dihadapi dari awal hingga akhir perjuangan menyelesaikan tugas akhir ini.
14. Wanita terhebat TE 2010 Ayu sintianingrum, S.T., Devy Andini, S.T., Kiki Aprilya, S.T., Maria Ulfa Muthmainah S.T., dan Novia Malinda, S.T., tanpa kalian penulis tidak bisa setegar dan sekuat ini berjuang menyelesaikan segala kesulitan selama ini.
15. Rekan TE 2010 Anwar Solihin, S.T., Derry Ferdiansyah, S.T., Ahmad Surya Arifin, S.T., Budi Wahyu Nugroho, S.T., Sandi Alfino, S.T., Irvika Rohman, S.T., dan lain-lain yang tidak bisa disebutkan satu persatu yang telah banyak membantu dalam kesulitan yang dihadapi oleh penulis, terima kasih.
16. Para senior teknik elektro yang telah memberikan bantuan, doa, dan semangat abang Adam Hussein, S.T., Mbak Ranny Dwidayanti, S.T., , Mbak Anisa Rahcman, S.T. teknik mesin, terima kasih buat marah-marahnya.
17. Para junior Teknik Elektro Universitas Lampung yang telah memberikan dukungan dan semangatnya.

18. Sahabat terbaikku Lely Oktaria, Am.Kep, Riska Yulianti, S.Kel, Elva Wulandari, Spd. dan Okta Noviana, Spd. yang telah memberikan doa dan semangatnya.
19. Sahabat kecil Apridayani, Spd dan Ro'aini yang telah memberikan doa dan semangatnya.
20. Sahabat tak terganti dan terlupakan Alm. Gita Erlangga, S.T., yang telah memberikan doa, dukungan, dan semangatnya.
21. Teman seperjuangan di kota rantauan Ika Dwi Wahyuni, S.E., dan Sri Eka Yati, S.Kom. yang telah memberikan doa dan semangatnya.
22. Semua pihak yang tidak dapat penulis sebutkan satu persatu atas bantuan dan dukungan serta doa dalam menyelesaikan penyusunan tugas akhir ini.

Penulis meminta maaf atas segala kesalahan dan ketidaksempurnaan dalam penyusunan penelitian ini. Saran dan kritik membangun sangat diharapkan penulis demi kebaikan dimasa yang akan datang. Sekali lagi penulis ucapkan terima kasih dan semoga Allah SWT membalas kebaikan anda semua dan diberi kemudahan dalam segala urusan. Amin Allahuma Amin.

Bandar Lampung, Oktober 2017

Dian Nindariansari
NPM. 1015031033

DAFTAR ISI

	Halaman
DAFTAR ISI	i
DAFTAR GAMBAR	iii
DAFTAR TABEL.....	v
I. PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Tujuan Penelitian	4
1.3 Manfaat Penelitian	4
1.4 Rumusan Masalah	6
1.5 Batasan Masalah.....	6
1.6 Sistematika Penulisan	7
II. TINJAUAN PUSTAKA	8
2.1 Sistem Diagnosa Hama dan Penyakit Tanaman Padi Berbasis Perangkat Lunak.....	8
2.2 Cara Diagnosa Seorang Ahli Pertanian.....	9
2.2.1 Pengecekan Fisik.....	10
2.2.2 Pengecekan Penunjang.....	10
2.3. Ilmu Kecerdasan Tiruan/Buatan (<i>Artificial Intelligence</i>)	11
a. Pengertian Kecerdasan Tiruan/Buatan (<i>Artificial Intelligence</i>)	11
b. Sejarah singkat Kecerdasan Tiruan/Buatan (<i>Artificial Intelligence</i>).....	11
c. Aplikasi-Aplikasi dari Kecerdasan Tiruan/Buatan (<i>Artificial Intelligence</i>)	13
2.3.3 Sistem Pakar (<i>Expert System</i>)	13
a. Pengertian Sistem Pakar (<i>Expert System</i>)	13
b. Tipe-Tipe Sistem Pakar.....	15
c. Komponen Sistem Pakar	16
2.3.4 Rekayasa Perangkat Lunak	24
2.5 Aplikasi-Aplikasi Pendukung	29
2.5.1 Sekilas Mengenai Prolog	30
2.5.2 Sejarah Singkat Prolog.....	31
2.5.3 Pemrograman Logika	32
2.5.4 Ciri Bahasa Prolog	34
2.5.5 Bahasa Deklaratif	38

III. METODE PENELITIAN	40
3.1 Diagram Alir Penelitian	40
3.2 Waktu dan Tempat	41
3.3 Alat dan Bahan	41
3.4 Tahap Perancangan Penelitian	41
3.5 Analisis kebutuhan dan definisi	44
a. Investigasi	44
b. Analisis	46
3.6 Desain sistem dan software	57
a. Pembuatan Algoritma	59
b. <i>Knowledge Base</i> (Basis Pengetahuan)	59
c. Desain <i>Knowledge base</i>	63
3.7. Implementasi	64
3.8. Pengujian (<i>Testing</i>)	68
3.9. Pemeliharaan (<i>Maintenance</i>)	68
3.10 Gambaran Umum Penelitian	70
IV. HASIL DAN PEMBAHASAN	71
4.1. Hasil dan Pembahasan	71
4.2 Impementasi Perangkat Lunak Diagnosa Hama dan Penyakit Tanaman Padi berbasis Sistem Pakar.....	71
4.3 Menu Diagnosa	75
4.4 Tampilan Halaman AntarMuka (<i>User Interface</i>) Berbasis Sistem Pakar	81
4.5 Hasil Pengujian Perangkat Lunak Diagnosa Hama dan Penyakit Tanaman Padi	85
4.6 <i>Functional Test</i>	90
4.7 <i>User Acceptant Test</i>	92
V. KESIMPULAN DAN SARAN	97
5.1 KESIMPULAN	97
5.2 SARAN	98

DAFTAR PUSTAKA

LAMPIRAN

DAFTAR TABEL

Tabel	Halaman
3.1 Hama dan Penyakit pada Tanaman Padi	48
3.2 Daftar <i>Attribute</i> Beserta Kuncinya.....	63
4.1 Gejala Kerusakan Tanaman Padi	72
4.2 <i>Functional Test</i>	91
4.3 <i>User Acceptance Test</i>	93

DAFTAR GAMBAR

Gambar	Halaman
2.1 Sistem Pakar Program Mandiri.....	15
2.2 Sistem Pakar Program terkait	15
2.3 Sistem Pakar Program Terhubungkan	16
2.4 Contoh Sistem Operasi Berbasis CLI.....	21
2.5 Contoh Sistem Operasi Berbasis GUI	22
2.6 <i>Development Engine</i>	23
2.7 Model <i>Waterfall</i>	26
3.1 Diagram Alir Penelitian.....	40
3.2 Metode <i>Waterfall</i>	42
3.3 Blok Diagram Diagnosa Kerusakan dengan Sistem Lama.....	44
3.4 <i>Usecase</i> Diagram	58
3.5 Kode Program pada <i>Knowledge Base</i>	64
3.6 Gambaran Umum Sistem.....	70
4.1 Bahasa Program Basis Data (<i>Knowledge Base</i>) Penyebab Kerusakan ...	75
4.2 Kode Program Menu Pembuka (<i>User Interface</i>)	76
4.3 Kode Program Pengolah Pertanyaan dari Basis Data (<i>Inference Engine</i>)	78
4.4 Tampilan Halaman Antar Muka Sistem Pakar Amzi Prolog	81
4.5 Halaman Pembuka Sistem Pakar	82
4.6 Sistem Pakar Memulai Diagnosa Penyebab Kerusakan	83

4.7	Sistem Pakar Berhasil Menemukan Penyebab Kerusakan	84
4.8	Sistem Pakar Tidak Menemukan Penyebab Kerusakan	84
4.9	Pengujian Perangkat Lunak pada Penggerek Batang Padi Kuning	85
4.10	Pengujian Perangkat Lunak pada Penggerek Batang Padi Putih.....	86
4.11	Pengujian Perangkat Lunak pada Walang Sangit.....	87
4.12	Pengujian Perangkat Lunak pada Ganjur	88
4.13	Pengujian Perangkat Lunak pada Lalat Bibit	89
4.14	Pengujian Perangkat Lunak pada Blas	89

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Kehidupan pertanian mempunyai arti yang penting bagi kehidupan manusia, selama manusia ada selama itu juga pertanian akan tetap dibutuhkan. Hal ini dikarenakan makanan merupakan kebutuhan manusia paling pokok selain air dan udara. Dari tahun ke tahun kebutuhan akan makanan terus meningkat seiring meningkatnya jumlah populasi manusia. Secara khusus beras merupakan hasil pertanian dari tanaman padi yang digunakan sebagai salah satu makanan pokok manusia.

Hal yang sering terjadi, banyak kerugian akibat kerusakan yang disebabkan oleh hama dan penyakit yang terlambat ditanggulangi dan bahkan sudah mencapai tahap yang parah dan mengakibatkan terjadinya gagal panen. Kerusakan tanaman tersebut sebelum mencapai tahap yang lebih parah dan meluas, sebenarnya masih menunjukkan gejala-gejala yang ringan pada tanaman dan masih dapat ditanggulangi. Akan tetapi kurangnya pengetahuan petani yang hanya mengandalkan pengetahuan turun temurun menganggap bahwa gejala kerusakan adalah hal yang biasa terjadi pada masa tanam dan gejala tersebut terabaikan, sehingga sudah terlambat untuk dikendalikan.

Ahli pertanian khususnya di bidang hama dan penyakit tanaman dalam hal ini mempunyai kemampuan untuk menganalisa gejala-gejala kerusakan tanaman yang disebabkan oleh serangan hama dan penyakit, namun jarak dan waktu menjadi kendala petani untuk berkonsultasi langsung dengan para ahli. Dampaknya pengetahuan petani tetap saja minim di zaman yang sudah semakin berkembang seperti sekarang ini.

Seiring dengan berkembangnya teknologi informasi serta komunikasi pada saat ini, maka kebutuhan akan informasi yang cepat dan akurat sudah menjadi kebutuhan yang utama bagi setiap insan. Hal ini mendorong para ahli untuk mengembangkan teknologi informasi bagaimana mengadopsi cara kerja seorang ahli ke dalam.

Hal ini dapat diwujudkan dengan cara menerapkan ilmu kecerdasan buatan (*Artificial Intelligence*) dengan membuat sistem pakar (*Expert System*) yang didalamnya memuat informasi tentang penyakit pada tanaman padi.

Semakin majunya peradaban maka kesibukan setiap manusia akan semakin bertambah. Untuk itu diperlukan suatu sistem yang dapat digunakan praktis dalam menyelesaikan suatu kendala yang dihadapi sehingga manusia dengan mudah melakukannya. Salah satu perangkat lunak yaitu sistem pakar dapat mengatasi kendala ini karena kemudahan dan kepraktisan yang dapat membantu masalah yang dihadapi oleh penggunanya.

Salah satu pelayanan yang mungkin dapat ditawarkan kepada masyarakat adalah sebuah sistem penanggulangan penyakit tanaman padi berbasis sistem pakar. Hal ini tentu akan mempermudah masyarakat untuk melakukan

tindakan jika sewaktu-waktu hama dan penyakit merusak tanaman padi mereka.

Penanggulangan kerusakan tanaman padi berbasis sistem pakar ini akan mengadopsi cara kerja seorang pakar/ahli pertanian dalam mengatasi masalah gejala kerusakan tanaman padi akibat serangan hama dan penyakit. Perangkat ini akan memberikan penjelasan mengenai kerusakan yang menyerang pada tanaman tersebut serta memberikan cara penanggulangan/pengobatan maupun pencegahannya. Sistem ini akan mencocokkan gejala-gejala kerusakan yang menyerang tanaman dengan pusat data berupa basis data tentang hama dan penyakit tanaman padi. Selanjutnya hasil dari pencocokan data tersebut akan diambil yang memiliki tingkat ketepatan yang paling tinggi. Sehingga diketahui diagnosa tentang penyakit tanaman tersebut, maka kemudian dapat diketahui pula tindakan yang cocok untuk memulihkan kerusakan tanaman tersebut.

Namun demikian, tentunya sistem ini tidak akan serta merta menggantikan peran serta kerja seorang ahli pertanian secara keseluruhan. Sistem ini dapat berfungsi dengan optimal untuk kasus-kasus gangguan kerusakan ringan, menangani beberapa kasus kerusakan hama dan penyakit padi yang biasa menyerang.

1.2 Tujuan Penelitian

Tujuan dilakukan penelitian ini, yaitu :

1. Merancang aplikasi sistem pakar untuk mendiagnosa gejala-gejala kerusakan tanaman padi akibat serangan hama dan penyakit dengan menggunakan metode *Interview* dan studi literatur.
2. Membangun aplikasi yang dapat membantu pengambilan keputusan dalam menentukan jenis kerusakan oleh hama dan penyakit tanaman padi dari beberapa gejala yang dimasukkan.
3. Dapat membuktikan bahwa sistem pakar yang dibangun merupakan bagian dari ilmu kecerdasan buatan (*Artificial Intelligence*).

1.3 Manfaat Penelitian

Hasil dari penelitian ini diharapkan dapat memberikan manfaat sebagai berikut:

1. Dapat digunakan oleh petani sebagai alat bantu untuk mengetahui penyebab kerusakan pada tanaman padi dari gejala kerusakan yang ditimbulkan tanpa harus mendatangkan seorang pakar/ ahli.
2. Dapat membantu petani (*user*) dalam hal menanggulangi kerusakan tanaman padi yang disebabkan oleh serangan hama dan penyakit dan memberikan informasi mengenai penyebab kerusakan dan cara penanggulangannya, dengan cara memasukkan gejala-gejala kerusakan yang terjadi pada tanaman tersebut ke dalam sistem pakar yang dibangun.

1.4 Rumusan Masalah

Pada penelitian ini akan dilakukan pengaplikasian sistem pakar pendiagnosa gejala kerusakan hama dan penyakit tanaman padi yang dirancang dengan menggunakan bahasa pemrograman *Prolog* yang diintegrasikan dalam bentuk *knowledge base*. Pada sistem pakar yang dibangun terdapat 30 jenis penyebab kerusakan, yaitu : penggerek batang padi kuning, penggerek batang padi putih, penggerek batang padi bergaris, wereng cokelat, wereng hijau, kepinding tanah, walang sangit, tikus, ganjur, hama putih palsu, hama putih, ulat tentara, ulat tanduk hijau, ulat jengkal palsu hijau, orong-orong, lalat bibit, keong mas, burung, hawar daun bakteri, bakteri daun bergaris, blas, hawar pelepah, busuk batang, busuk pelepah, bercak cokelat, bercak cerospora, hawar daun jingga, tungro, kerdil rumput dan kerdil hampa.

Sistem ini mengadopsi cara kerja seorang ahli pertanian dalam mendiagnosa gejala-gejala kerusakan yang terjadi pada tanaman padi. Sistem ini memungkinkan pengguna untuk mengetahui penyebab kerusakan yang menyerang dengan memasukan gejala-gejala yang terjadi pada tanaman padi kedalam sistem pakar. Kemudian gejala-gejala tersebut akan diolah dan dicocokkan ke dalam pusat data (*knowledge base*). Dan kemudian akan ditarik sebuah jawaban yang berupa kesimpulan mengenai penyebab kerusakan dan cara penanggulangan kerusakan tersebut.

1.5 Batasan Masalah

Adapun batasan masalah dalam penelitian ini adalah sebagai berikut:

1. Sistem pakar yang dibangun terbatas hanya membahas kerusakan tanaman padi yang disebabkan oleh hama dan penyakit.
2. Metode penelitian yang akan digunakan pada sistem ini menggunakan metode wawancara (*interview*) dan studi literatur dengan metode pengembangan menggunakan metode *waterfall*.
3. Penyebab kerusakan yang terdapat pada basis data pada sistem pakar terbatas yaitu hanya 30 jenis penyebab kerusakan, hal ini dikarenakan untuk mencoba apakah sistem yang dibangun dapat berjalan dengan baik atau tidak.

1.6 Sistematika Penulisan

Dalam rangka penulisan tugas akhir ini, disusun suatu sistematika penulisan dengan membaginya menjadi beberapa bab. Susunan sistematika tersebut adalah:

BAB I PENDAHULUAN

Dalam bab ini membahas mengenai latar belakang, tujuan penulisan, manfaat penelitian, rumusan masalah, batasan masalah, dan sistematika penulisan.

BAB II TINJAUAN PUSTAKA

Bab ini membahas mengenai teori-teori dasar mengenai cara diagnosis seorang pakar/ahli pertanian yang diadopsi ke dalam sistem pakar, Ilmu Kecerdasan Tiruan, Rekayasa Perangkat Lunak dan *Prolog*.

BAB III METODE PENELITIAN

Bab ini membahas mengenai tahapan-tahapan yang dilakukan dalam merancang sistem pakar diagnosis hama dan penyakit tanaman padi berbasis sistem informasi ini, seperti pemodelan sistem pakar, *knowledge base*, *inference engine* dan keterhubungan tabel-tabel pada basis data dan jenis-jenis hama dan penyakit tanaman padi.

BAB IV HASIL DAN PEMBAHASAN

Bab ini membahas mengenai analisis simulasi, uji coba, analisa untuk kinerja perangkat lunak saat menjalankan sistem ini serta tanggapan pengguna terhadap sistem pakar pendiagnosa penyakit berbasis sistem informasi.

BAB V KESIMPULAN DAN SARAN

Bab ini membahas mengenai kesimpulan dan saran yang diperoleh berdasarkan hasil pembahasan.

BAB II

TINJAUAN PUSTAKA

2.1 Sistem Diagnosa Hama Dan Penyakit Tanaman Padi Berbasis Perangkat Lunak

Tanaman padi merupakan salah satu tanaman budidaya terpenting untuk kelangsungan hidup manusia. Padi menghasilkan beras yang merupakan salah satu makanan pokok masyarakat Indonesia. Akan tetapi hal yang sering terjadi, banyak kerugian akibat kerusakan tanaman padi yang disebabkan oleh serangan hama dan penyakit pada tanaman padi. Sebenarnya setiap penyakit tanaman sebelum mencapai tahapan yang parah masih menunjukkan gejala-gejala ringan yang masih bisa diatasi sebelum terjadinya gagal panen. Dalam hal ini para petani masih sulit untuk menangani penyakit tanaman secara cepat dan tepat, padahal para ahli pertanian khususnya bidang ilmu hama dan penyakit tanaman mempunyai kemampuan untuk menganalisa gejala-gejala kerusakan akibat hama dan penyakit tanaman tersebut.

Teknologi informasi semakin berkembang pesat, hal ini mendorong manusia untuk terus memperbarui sistem pelayanan kepada seorang pengguna jasa. Sistem pelayanan yang dimaksud akan membahas tentang sistem pelayanan

mengenai penyakit tanaman padi. Sistem ini bertujuan untuk mengatasi beberapa kendala petani Indonesia dalam hal pengetahuan mengenai kerusakan tanaman padi yang disebabkan oleh hama dan penyakit. Hal yang perlu menjadi pertimbangan adalah bahwa tanaman padi merupakan makanan pokok masyarakat Indonesia dan merupakan salah-satu mata pencaharian sebagian besar petani Indonesia. Akan tetapi sering terjadinya gagal panen tidak membuat para petani berkeinginan untuk berkonsultasi langsung dengan para ahli yang lebih memahami permasalahan kerusakan tanaman padi mereka. Padahal jika mereka berkonsultasi dengan para ahli kemungkinan terjadinya gagal panen di wilayah Indonesia bisa teratasi dan kebutuhan produksi makanan pokok terpenuhi tanpa harus membebani biaya yang tinggi untuk harga suatu makanan pokok.

Alur sistem pendiagnosaan seperti ini sebenarnya dapat diganti dengan sistem yang lebih praktis yaitu berbasis perangkat lunak yang cara kerjanya mengadopsi keahlian seorang ahli dalam mendiagnosa suatu penyakit. Sehingga sistem ini akan sangat mudah dijangkau oleh setiap orang dalam pengaksesannya.

2.2 Cara diagnosa seorang ahli pertanian

Diagnosa kerusakan pada tanaman tidak semudah seorang dokter mendiagnosa penyakit pada manusia yang dapat secara langsung berinteraksi dengan pasien untuk mengetahui gejala dan keluhan sakit yang diderita.

Seorang ahli pertanian dalam mendiagnosa kerusakan suatu tanaman melalui dua tahapan yang akan dilakukannya, sebagai berikut :

a. **Pengecekan Fisik**

Tahapan pertama ini merupakan tahap dimana seorang ahli pertanian melihat tanda-tanda fisik, gejala dan pola yang ditimbulkan dari tanaman, mulai dari akar, batang, daun hingga buah. Pengamatan gejala penyakit dapat dibedakan menjadi dua yaitu gejala primer dan gejala sekunder. Gejala primer adalah gejala yang timbul segera dan secara langsung, umumnya penyebab penyakit sangat dekat dengan jaringan tanaman yang rusak. Sedangkan gejala sekunder adalah gejala yang timbul jauh dari jaringan tanaman yang terserang. Gejala-gejala tersebut seperti adanya struktur, galls, cairan atau misellium. Contoh : apabila suatu bagian tanaman terdapat misellium misalnya pada bagian daun maka dapat diprediksikan daun tersebut terserang patogen berupa jamur atau bakteri.

a. **Pengecekan Penunjang**

Pengecekan laboratorium merupakan tahap dimana seorang ahli pertanian memerlukan penelitian yang lebih mendalam agar dapat diketahui penyakit tersebut disebabkan oleh hama atau penyakit oleh patogen (jamur, virus dan bakteri). Pada tahap ini selain mengamati kerusakan yang terjadi pada tanaman, hal yang perlu dilakukan yaitu mengilustrasi keadaan tanaman tersebut mulai dari faktor iklim, lingkungan, unsur yang

terkadang dalam tanah, melakukan perbandingan tanaman yang sakit dengan tanaman yang tumbuh sehat, menentukan distribusi penyakit di dalam lapangan atau kebun. Pengecekan laboratorium biasanya diperlukan jika penyakit yang diderita merupakan penyakit yang tidak kasat mata dan belum pernah terjadi sebelumnya sehingga perlu adanya pendalaman penyakit untuk mendapatkan hasil yang lebih akurat.

2.3 Ilmu Kecerdasan Tiruan/Buatan (*Artificial Intelligence*)

a. Pengertian Kecerdasan Tiruan/Buatan (*Artificial Intelligence*)

Kecerdasan buatan (*Artificial Intelligence*) (AI) adalah bagian dari ilmu komputer yang mempelajari cara membuat agar mesin komputer dapat melakukan pekerjaan seperti dan sebaik yang dilakukan oleh manusia [1].

b. Sejarah singkat Kecerdasan Tiruan/ Buatan (*Artificial Intelligence*)

Kecerdasan buatan adalah bagian dari pengembangan di bidang teknologi informasi. Pada tahun 1950-an para ilmuan dan peneliti memikirkan bagaimana caranya agar mesin dapat melakukan pekerjaannya seperti yang bisa dikerjakan oleh manusia. Ahli matematikawan yang berasal dari negara Inggris bernama Alan Turing adalah pencetus ide mengenai kepintaran sebuah mesin dalam menyelesaikan suatu masalah seperti manusia. Hasil uji coba tersebut dikenal dengan sebutan *Turing Test*. Turing berpendapat jika mesin dapat membuat seseorang percaya bahwa dirinya mampu berkomunikasi dengan orang lain, maka dapat dikatakan bahwa mesin tersebut cerdas (seperti layaknya manusia) [1].

Seorang professor berasal *Massachusetts Institute Of Technology* yang bernama John McCarthy adalah yang pertama kali mempopulerkan ilmu kecerdasan buatan pada tahun 1956 dalam acara *Dartmouth Conference* yang dihadiri oleh para peneliti AI. Pada konferensi tersebut juga didefinisikan tujuan utama dari kecerdasan buatan, yaitu : mengetahui dan memodelkan proses-proses berpikir manusia dan mendesain mesin agar dapat menirukan kelakuan atau keahlian manusia.

Berikut beberapa contoh pengembangan kecerdasan buatan yang dilakukan pada tahun 1956-1966 :

1. *Dartmouth Conference* adalah ilmuwan yang membuat *Logic Theorist*, dimana pada program ini dapat membuktikan teorema-teorema matematika.
2. Robert K. Lindsay (1960) adalah ilmuwan yang membuat proyek Sad Sam, dimana program ini dapat mengetahui kalimat-kalimat sederhana yang ditulis dalam Bahasa Inggris dan mampu memberikan jawaban dari fakta-fakta yang didengar dalam sebuah percakapan.
3. Joseph Weizenbaum (1967) adalah ilmuwan yang membuat proyek ELIZA, dimana Program ini mampu melakukan terapi terhadap pasien dengan memberikan beberapa pertanyaan [6].

c. Aplikasi-Aplikasi dari Kecerdasan Tiruan/Buatan (*Artificial Intelligence*)

Bagian dari kecerdasan buatan terdiri dari tiga bagian, yaitu :

1. Sistem berbasis pengetahuan atau Sistem Pakar (*expert system*)/*knowledge base system*), yaitu sistem yang ada pada program komputer yang mempunyai fungsi untuk menyelesaikan masalah seperti layaknya manusia dengan domain pengetahuan yang dimilikinya.
2. Sistem bahasa alami (*natural language system*), yaitu pemrograman yang mengerti bahasa manusia.
3. Sistem dengan kemampuan memahami (*perception system*), yaitu sistem untuk penglihatan, pembicaraan atau sentuhan [5].

Perangkat-perangkat lunak ini dapat dijalankan dengan komputer pribadi, sehingga pengembangan untuk aplikasi kecerdasan tiruan dapat dilakukan dengan mudah dan dengan biaya yang murah [5].

2.3.3 Sistem Pakar (*Expert System*)

a. Pengertian Sistem Pakar (*Expert System*)

Sistem pakar adalah program dari AI dengan basis pengetahuan (*knowledge base*) yang diperoleh dari pengalaman/pengetahuan pakar/ahli dalam memecahkan persoalan pada bidang tertentu dan didukung mesin inferensi (*inference engine*) yang melakukan penalaran/pelacakan terhadap fakta-fakta yang diberikan oleh pemakai (*user*), dicocokkan (*matching*)

dengan fakta-fakta dan atauran/kaidah yang ada di basis pengetahuan setelah dilakukan pencarian, sehingga dicapai kesimpulan. Dapat disimpulkan sistem pakar adalah sistem yang berfungsi untuk mengadopsi keahlian dan pengetahuan manusia dalam menyelesaikan masalah ke dalam komputer. Sistem pakar atau sistem berbasis pengetahuan adalah yang paling banyak aplikasinya dalam membantu menyelesaikan masalah-masalah dalam dunia nyata. Contoh aplikasi dari program ini antara lain yaitu :

- Delta dari general elektrik untuk konsultasi kerusakan lokomotif.
- Prospector dari *Stanford Research Institut* untuk penafsiran prospek mineral.
- Dan lain-lain [2].

Program kecerdasan tiruan ini dapat dilakukan dengan menggunakan suatu program paket, yaitu alat pengembang sistem aplikasi pengetahuan (*knowledge system application development tool*) seperti:

- Pengetahuan pada sistem pakar mudah disimpan dan di *copy*
- Pengetahuan yang tidak mudah hilang
- Selalu membentuk opini terbaik dalam batas pengetahuannya.

Kerugian *expert system* :

- Kurang personalitinya
- Tidak dapat menyelesaikan masalah yang dibutuhkan intuisi [7].

b. Tipe-Tipe Sistem Pakar

Tipe-tipe sistem pakar berdasarkan struktur program, ada tiga tipe :

a. Program mandiri

Sistem pakar murni dan berdiri sendiri, artinya program utamanya tanpa mengandung *subroutine* yang memakai algoritma konvensional/*function* yang lain.



Gambar 2.1 Sistem Pakar Program Mandiri

b. Program terkait

Sistem pakar yang dikelilingi program lainnya, artinya sebuah *subroutine* yang akan dipanggil oleh program utama.

Misalnya memiliki subroutine untuk :

- Perhitungan matematik (MathLab)
 - Pembuatan grafik (SPSS/ PC+)
 - Persamaan linier (lindo)
- } Algoritma Konvensional

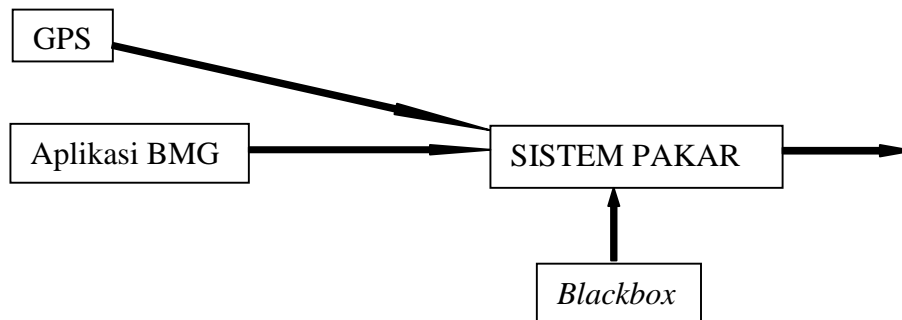


Gambar 2.2 Sistem Pakar Program terkait

c. Program terhubungkan

Sistem pakar merupakan program yang dapat berhubungan dengan paket program atau aplikasi lainnya, misalkan :

- Aplikasi *global position system* (GPS) yang terhubungkan pada satelit, radar dan sonar sebagai sensor masukan yang di *remote* oleh GPS.
- Aplikasi BMG (badan meteorologi dan geofisika) dengan sensor seperti *barometer, seismograf, tsunami*, dan sebagainya
- Kotak hitam (*blackbox*)



Gambar 2.3 Sistem Pakar Program Terhubungkan

c. **Komponen Sistem Pakar**

Sebuah program sistem pakar terdiri atas komponen-komponen sebagai berikut:

a. Basis Pengetahuan (*knowledge base*)

Basis Pengetahuan (*knowledge base*) adalah inti program sistem pakar. Merupakan representasi pengetahuan (*knowledge representation*) dari seorang pakar. Tersusun atas fakta yang berupa objek dan kaidah/aturan (*rule*) yang merupakan informasi tentang cara bagaimana membangkitkan fakta baru dari

fakta yang sudah diketahui. Daftar fakta-fakta (*Facts list*) berisikan hasil observasi dan sesuatu kenyataan yang dibutuhkan selama pengolahan. Bagian yang mengandung semua fakta-fakta, baik fakta awal pada sistem mulai beroperasi maupun fakta-fakta yang didapatkan pada suatu pengambilan kesimpulan. *Knowledge* ada dua jenis:

- Tacid Knowledge* (berupa ide, inovasi, pengalaman (*experience*) seorang pakar/ahli) yang belum dalam bentuk tulisan, masih berupa hasil pemikiran
- Explicit knowledge* adalah *knowledge* yang sudah dalam bentuk tulisan seperti yang ada pada buku pengetahuan, jurnal/artikel ilmiah [5].

b. Mesin inferensi (*Inference Engine*)

Bagian yang mengandung mekanisme fungsi berpikir dan pola-pola penalaran sistem yang digunakan seorang pakar. Mekanisme ini akan menganalisa sesuatu masalah tertentu dan selanjutnya mencari jawaban/kesimpulan yang terbaik. Memilih pengetahuan yang relevan dalam rangka mencapai kesimpulan. Memulai pelacakannya dengan mencocokkan kaidah (*rule*) dalam basis pengetahuan dengan fakta-fakta yang ada dalam *facts list* disimpan dalam basis pengetahuan di *hard-disk* [5].

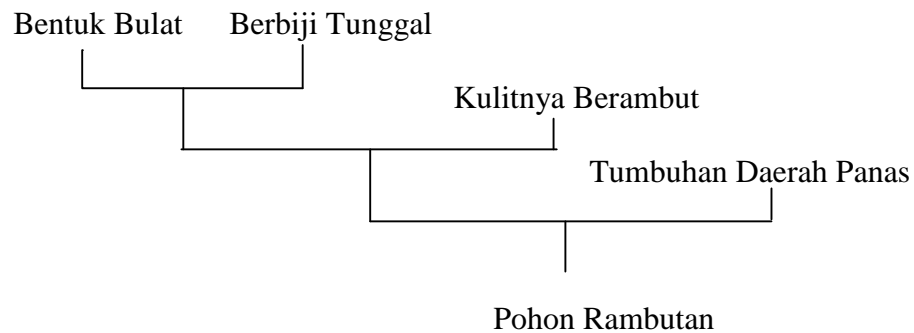
Dasar untuk membentuk *inference engine*

1. Metode *forward-chaining*

Inference engine sering disebut juga *data-driven* karena *inference engine* menggunakan informasi yang ditentukan oleh *user* untuk memindahkan keseluruhan jaringan dari logika ‘AND’ dan ‘OR’ sampai sebuah terminal

ditentukan sebagai objek. Jika *inference engine* tidak dapat menentukan objek maka akan meminta informasi lain. Aturan (*rule*) dimana menentukan objek, membentuk lintasan (*path*) yang mengarah ke objek. Oleh karena itu, hanya satu cara untuk mencapai objek adalah memenuhi semua aturan. Dengan kata lain pelacakan kedepan (*forward channing*) memulai dari sekumpulan fakta-fakta (data) dengan mencari kaidah yang cocok dengan dugaan/hipotesa yang ada menuju kesimpulan [4].

Contoh 1 :



Contoh 2 : F.C. menggunakan implikasi

Implikasi 1 : p1: *a college professor teaches in the summer*

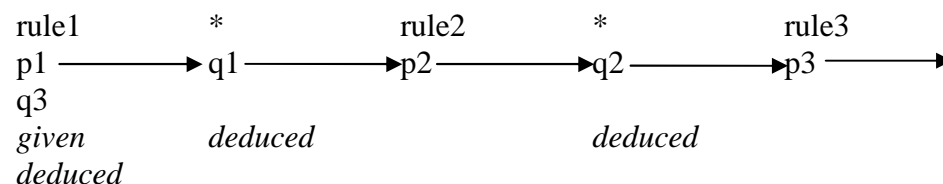
q1: *the professor can't do anything but teach*

implikasi 2 : p2: *the professor can't do anything but teach (=q1)*

q2: *the professor does not have time to do re-search*

implikasi 3: p3: *the professor does not have time to do research (=q2)*

q3: *the professor is unhappy*



hasil : $p1 \xrightarrow{\text{deduced}} q3$
given *deduced*

basis pengetahuan (*knowledge base*) dapat dituliskan dengan menggunakan variabel substitusi:

$p1 \xrightarrow[\text{given}]{\text{rule1}} p2 \xrightarrow[\text{deduced}]{\text{rule2}} p3 \xrightarrow[\text{deduced}]{\text{rule3}} q3$

Tanda * adalah *symbolic matching approach*. atau dapat digambarkan sebagai berikut:

$p1 \xrightarrow{\text{deduced}} q3$
given *deduced*

gambar diatas memperlihatkan bentuk dari *forward channing* yang akan menghasilkan:

basis aturannya (*rule base*):

Step1: $((p1 \xrightarrow{\text{deduced}} p2) \quad p1 = \text{True}) \xrightarrow{\text{deduced}}$
 Step2: $((p2 \xrightarrow{\text{deduced}} p3) \quad p2 = \text{True}) \xrightarrow{\text{deduced}}$
 Step3: $((p3 \xrightarrow{\text{deduced}} q3) \quad p3 = \text{True}) \xrightarrow{\text{deduced}}$

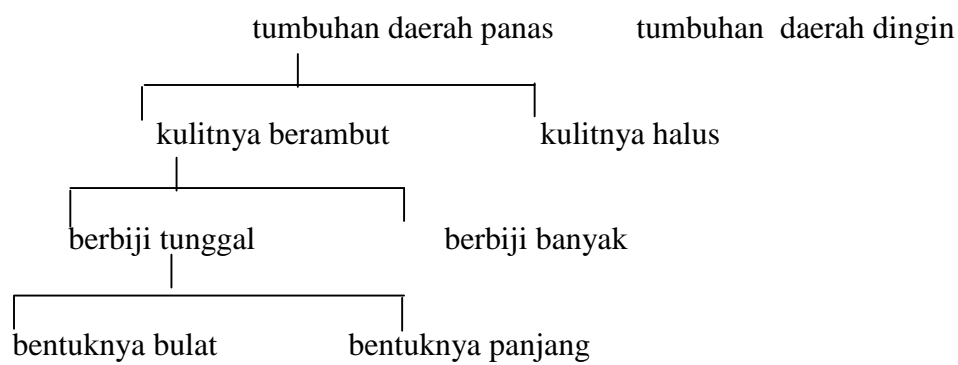
2. Metode *Backward Channing*

Merupakan kebalikan dari *forward channing* dimana mulai dengan sebuah hipotesa/sebuah objek dan meminta informasi untuk meyakinkan atau mengabaikan *backward chaining inference engine* sering disebut : '*objek-driven/goal driven*'. Karena sistem mulai dengan objek dan mencoba untuk

menverifikasi objek. Atau dapat diartikan pelacakan ke belakang (*backward channing*) yang memulai penalarannya dari kesimpulan (*goal*), dengan mencari sekumpulan hipotesa-hipotesa yang mendukung menuju fakta-fakta yang mendukung sekumpulan hipotesa-hipotesa tersebut [4].

Contoh :

start goal: pohon rambutan



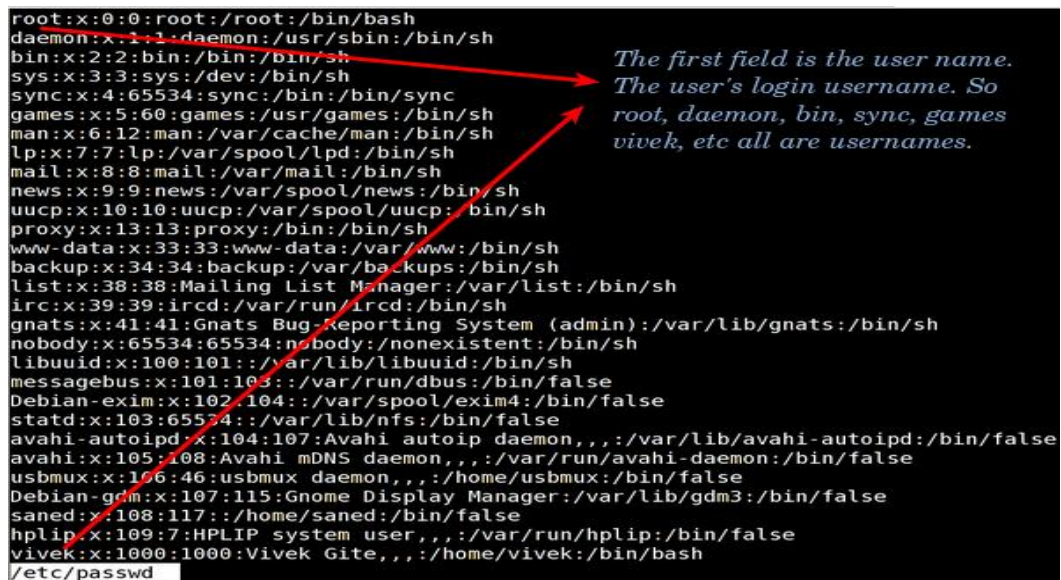
c. Antarmuka Pemakai (*User Interface*)

Antarmuka pemakai (*user interface*) adalah penghubung antara pengguna komputer dengan sistem pada perangkat keras komputer. Pengertian sistem operasi secara umum adalah pengelola seluruh sumber daya yang terdapat pada sistem komputer dan menyediakan sekumpulan layanan (*system calls*) ke pengguna untuk memudahkan dan menyamankan penggunaan serta pemanfaatan sumber daya sistem komputer [4].

Terdapat 2 jenis sistem operasi berdasarkan tampilan antarmuka kepada penggunanya, yaitu :

1. Sistem operasi berbasis CLI (*Command Line Interface*)

Pengertian CLI adalah antarmuka pada sistem operasi yang berbasis perintah atau teks. Dalam berinteraksi dengan sistem operasi pengguna hanya dapat menggunakan *keyboard* dengan cara mengetikkan perintah (*command*) tertentu. Berikut contoh sistem operasi berbasis CLI :



```

root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
mail:x:8:8:mail:/var/mail:/bin/sh
news:x:9:9:news:/var/spool/news:/bin/sh
uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh
proxy:x:13:13:proxy:/bin:/bin/sh
www-data:x:33:33:www-data:/var/www:/bin/sh
backup:x:34:34:backup:/var/backups:/bin/sh
list:x:38:38:Mailing List Manager:/var/list:/bin/sh
irc:x:39:39:ircd:/var/run/ircd:/bin/sh
gnats:x:41:41:Gnats Bug Reporting System (admin):/var/lib/gnats:/bin/sh
nobody:x:65534:65534:nobody:/nonexistent:/bin/sh
libuuid:x:100:101::/var/lib/libuuid:/bin/sh
messagebus:x:101:103::/var/run/dbus:/bin/false
Debian-exim:x:102:104::/var/spool/exim4:/bin/false
statd:x:103:65534::/var/lib/nfs:/bin/false
avahi-autoipd:x:104:107:Avahi autoip daemon,,,:/var/lib/avahi-autoipd:/bin/false
avahi:x:105:108:Avahi mDNS daemon,,,:/var/run/avahi-daemon:/bin/false
usbmux:x:106:46:usbmux daemon,,,:/home/usbmux:/bin/false
Debian-gdm:x:107:115:Gnome Display Manager:/var/lib/gdm3:/bin/false
saned:x:108:117::/home/saned:/bin/false
hplip:x:109:7:HPLIP system user,,,:/var/run/hplip:/bin/false
vivek:x:1000:1000:Vivek Gite,,,:/home/vivek:/bin/bash
/etc/passwd

```

The first field is the user name.
The user's login username. So root, daemon, bin, sync, games vivek, etc all are usernames.

Gambar 2.4 contoh sistem operasi berbasis CLI

2. Sistem Operasi berbaris GUI (*Graphical User Interface*)

Graphical User Interface (GUI) adalah antarmuka (*user interface*) yang menggunakan menu grafis agar bisa mempermudah *user* untuk berinteraksi dengan komputer dan lebih difokuskan untuk membuat sistem operasi yang *user-friendly* agar para pengguna lebih nyaman menggunakan komputer. Tampilan dari menu grafis itu sendiri seperti ada gambar-gambar dan tampilan yang tujuannya untuk memudahkan para pengguna menggunakan

sistem operasi ini. Contoh dari sistem operasi berbasis *Graphical User Interface* (GUI) yaitu tampilan menu pada *Microsoft word* sebagai berikut :



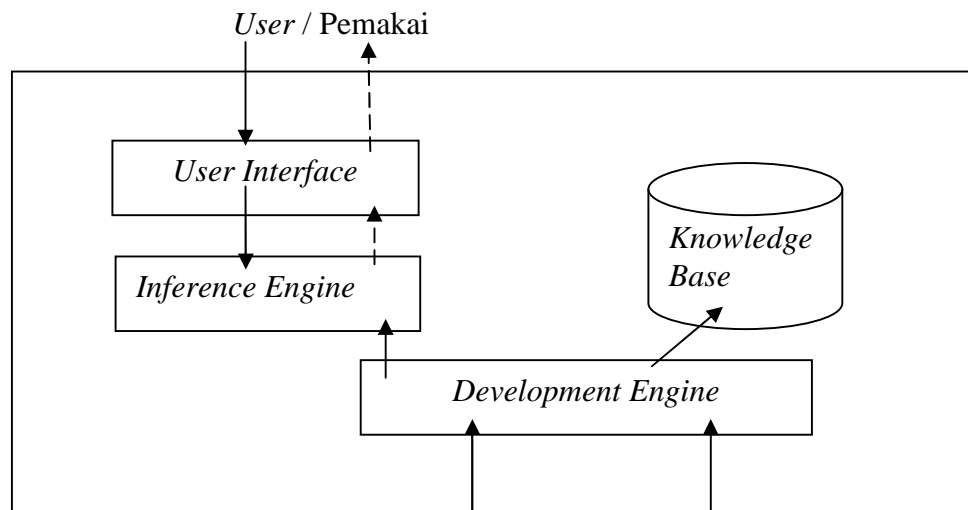
Gambar 2.5 Contoh Sistem Operasi berbasis GUI (*Graphical User Interface*)

Program sistem pakar yang dibangun menggunakan *user interface* berbasis CUI (*Character User Interface*) dimana pada sistem ini pengguna harus mengetikkan perintah-perintah pada *prompt*. Sistem pakar akan mengajukan pertanyaan-pertanyaan yang merupakan fakta mengenai gejala kerusakan yang ditimbulkan pada padi dan jawaban dari pertanyaan tersebut hanya memerlukan jawaban “ya”/”tidak”, berbentuk panduan menu (*menu driven*), pernyataan-pernyataan bahasa alami (*natural language*), dan *graphics interface style*. program sistem pakar akan mengambil kesimpulan berdasarkan jawaban-jawaban dari pengguna (*user*) [6].

d. *Development Engine*

Bagian dari sistem pakar sebagai fasilitas untuk mengembangkan mesin inferensi dan penambahan basis pengetahuan yang akan dilakukan oleh *knowledge engineer* yang harus punya keahlian dalam mengerti bagaimana pakar menerapkan pengetahuan dalam memecahkan masalah, mampu mengekstraksikan penjelasan (*knowledge equisition*) mengenai pengetahuan dari pakar, bila si pakar menemukan pengetahuan dan aturan yang baru dari

pengalaman ia bekerja. Dengan kata lain *development engine* adalah fasilitas yang digunakan oleh sistem pakar untuk memodifikasi *update knowledge base* dan *inference engine* [10].



Gambar 2.6 *Development engine*

Klasifikasi sistem pakar berdasarkan kegunaannya :

a. Diagnosa :

-Digunakan untuk merekomendasikan:

Pengobatan untuk orang sakit, kerusakan mesin, kerusakan rangkaian elektronik.

-Menemukan apa masalah atau kerusakan yang terjadi.

-Menggunakan pohon keputusan (*decision tree*) sebagai presentasi pengetahuannya.

b. Pengajaran :

-Digunakan untuk pengajaran, mulai dari SD s/d PT

- Membuat diagnosa apa penyebab kekurangan dari siswa
kemudian memberikan cara untuk memperbaikinya.

c. *Interpretasi*

- Untuk menganalisa data yang tidak lengkap, tidak teratur, dan data yang kontradiktif
misal : untuk interpretasi citra

d. *Prediksi*

- Contoh : bagaimana seorang pakar meteorologi memprediksi cuaca besok berdasarkan data-data sebelumnya.
- Untuk peramalan cuaca
- Penentuan masa tanam

e. *Perencanaan :*

- Perencanaan bisnis dengan tujuan untk menghemat biaya, waktu & material, sebab pembuatan model
-contoh : sistem konfigurasi komputer

f. *Kontrol*

- Untuk mengontrol kegiatan yang membutuhkan presisi waktu tinggi
- Misal : pengontrolan pada industri-industri berteknologi tinggi [5].

2.3.4 Rekayasa Perangkat Lunak

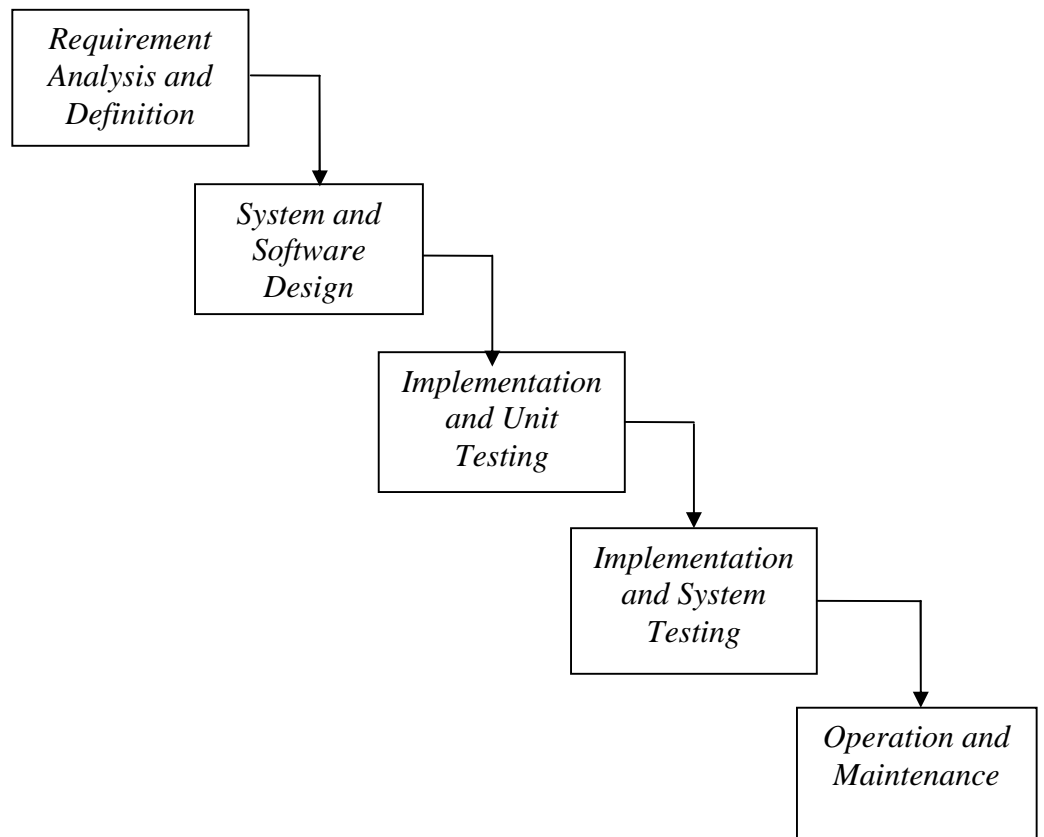
Rekayasa perangkat lunak adalah pengubahan perangkat lunak itu sendiri guna mengembangkan, memelihara dan membangun kembali dengan

menggunakan prinsip rekayasa untuk menghasilkan perangkat lunak yang dapat berkerja lebih efisien dan efektif untuk pengguna [6].

➤ **Metode *Waterfall* dalam Rekayasa Perangkat Lunak**

a. *Model Sekuensial Linear/Model Waterfall*

Model *waterfall* adalah pengembangan perangkat lunak yang sistematis, berurutan/sekuensial tahapan yang dilakukan dimulai pada pemenuhan seluruh persyaratan dalam analisis, perancangan desain, pengkodean, pengujian (*testing*), dan tahap pemeliharaan dalam membangun perangkat lunak *software (software)*. Gambar model *waterfall* dapat dilihat pada gambar 2.7 di bawah ini :



Gambar 2.7 Model *Waterfall*

Berikut tahapan-tahapan yang dilakukan pada model *waterfall* :

1. Analisis

Tahap ini adalah proses pengumpulan kebutuhan secara lengkap untuk didefinisikan analisis kebutuhan yang harus dipenuhi dan oleh program yang akan dibuat, yaitu memahami domain masalah, tingkah laku, unjuk kerja dan antar muka (*interface*).

2. Desain

Tahap ini melibatkan empat proses atribut sebuah program yaitu struktur data, arsitektur, perangkat lunak, representasi *interface*, dan detail algoritma prosedural.

3. Pengkodean

Pada tahap ini adalah proses penterjemahan desain ke dalam bentuk bahasa mesin yang dapat dilakukan secara mekanis.

4. Pengujian

Tahap ini dilakukan setelah kode dirancang dan difokuskan pada fungsi dan kesalahan untuk diperbaiki.

5. Pemeliharaan

Tahap ini meliputi penyesuaian atau perubahan mengikuti perkembangan seiring dengan adaptasi perangkat lunak dengan kondisi atau situasi yang sebenarnya setelah disampaikan pada konsumen atau pelanggan [2].

Pada penelitian ini dilakukan perancangan sistem menggunakan metode *waterfall* dikarenakan dari keunggulan metode *waterfall* itu sendiri yaitu mudah diaplikasikan karena urutan-urutan pengerjaan sudah sering dipakai. Selain itu juga cocok untuk *software* berskala besar dan yang bersifat umum. Disamping itu langkah-langkahnya sangat sekuensial, pengerjaan proyek akan mudah dikontrol dan terjadwal dengan baik. Pada rekayasa perangkat lunak terdapat beberapa model yang dapat digunakan selain metode *waterfall* diantaranya, sebagai berikut:

a. Model *Prototyping*

Pada model ini menyajikan gambaran yang lengkap dari sistem, terdiri dari model kertas, model kerja dan program. Programmer perlu melakukan identifikasi kebutuhan pemakai, menganalisa sistem dan melakukan studi kelayakan serta studi terhadap kebutuhan pemakai, meliputi model *interface*, teknik prosedural dan teknologi yang akan dimanfaatkan.

b. Model *Rapid Application Development (RAD)*

Model *Rapid Application Development* adalah model *waterfall* dan model *component based construction*. Dimana proses pembangunan perangkat lunak yang menekankan pada siklus pengembangan yang pendek dan singkat

c. Model *Evolutionary software process* model, terbagi dua:

-Model *Incremental*

Metode ini merupakan gabungan dari elemen-elemen dari model *waterfall* yang diaplikasikan secara berulang. Dimana dihasilkan produk dengan spesifikasi tertentu kemudian proses dimulai dari awal kembali hingga muncul hasil yang spesifikasinya lebih lengkap dari sebelumnya dan tentunya memenuhi kebutuhan pemakai.

-Model *Spiral Model/Spiral Boehm*

Metode ini kombinasi antara model *prototyping* dengan pengulangannya dan model *waterfall* dengan pengendalian dan sistematikanya. Pembuatan model ini memadupadankan beberapa model dari yang suatu masalah yang umum menjadi khusus.

e. Model *Component Assembly* (CAM)

Metode ini adalah gabungan dari berbagai sifat dan karakter model spiral Boehm dan model RAD (*Rapid Application Development*). Dimana pembuatan aplikasinya dibuat dari serangkaian komponen yang telah ada sebelumnya. Akan tetapi, waktu yang dibutuhkan lebih efektif dari pengerjaan program awal.

f. Model *The Concurrent Development*

Metode ini adalah menggambarkan secara skematik sebagai serial dari kegiatan teknis utama, tugas-tugas, dan hubungan antar bagian-bagian yang saling terkait di mana aktifitas analisa seperti desain/rancangan atau komunikasi pelanggan dapat diskemakan dengan cara yang sama.

g. Model *Formal Method*

Metode ini digunakan notasi matematika yang terperinci dan penuh ketelitian dalam mengidentifikasi desain dan menguji sistem yang berbasis komputer.

h. Model *Fourth Generation Techniques*

Model ini adalah penggabungan antara model prototype dengan model konvensional [14].

2.5 Aplikasi Pendukung.

Untuk membangun sebuah aplikasi sistem pakar diagnosa hama dan penyakit tanaman padi berbasis perangkat lunak dibutuhkan beberapa aplikasi pendukung seperti *software Amzi Prolog*. Bahasa pemrograman yang dipakai adalah bahasa pemrograman prolog (*Program in Logic*) [8].

2.5.1 Sekilas mengenai Prolog (*Program in Logic*)

Bahasa prolog merupakan bahasa generasi kelima yang mendorong pemrograman kedalam dimensi baru. Bahasa prolog dibangun atas dasar pemrograman alamiah dan logika. Untuk itulah lahir nama prolog, yaitu singkatan dari *Programming in Logic*.

Prolog juga merupakan bahasa *deklaratif*. Artinya, jika diberikan fakta dan aturan, prolog akan menyelesaikan problem secara deduktif; atau dari banyak fakta dan aturan kemudian diturunkan kesimpulan sebagai jawaban. Hal ini berbeda sekali dengan bahasa prosedural seperti *Pascal* atau *Fortran*. Dalam bahasa prosedural, pemrogram harus memberi perintah untuk memecahkan persoalan, langkah demi langkah; dengan kata lain, pemrogram harus tahu terlebih dahulu bagaimana memecahkan masalah itu. Bila dibandingkan dengan bahasa program prolog yang lebih praktis dimana hanya perlu memberikan penjelasan masalah (fakta) dan aturan dasar untuk memecahkannya. Pada penyelesaiannya prolog dibiarkan untuk menentukan sendiri bagaimana mencari jawaban. Dengan demikian program aplikasi penunjang kecerdasan buatan seperti basis pengetahuan, sistem pakar, antarmuka ke bahasa alami, pemrograman simbolik serta sistem manajemen informasi yang cerdas, dan pengenalan citra, akan lebih mudah diwujudkan [8].

2.5.2 Sejarah Singkat Prolog

Kelahiran prolog diawali ketika Alain Colmeaurer dengan sekelompok peneliti menghadapi masalah penerjemahan bahasa dengan komputer di Montreal. Hal ini terutama akibat hasil karya Chomsky dalam ilmu bahasa. Dia dan koleganya menelusuri cara mempertemukan bahasa alami dengan bahasa komputer. Dan disusul dengan dilakukannya pengembangan bahasa percobaan di Montreal dan Marseilles, yaitu Tarzan yang merupakan bahasa untuk memanipulasi *tree*, Col Meraurer, Roussel yang melakukan pengembangan disegi deduksi pada sistem percobaannya dan menentukan kemungkinan penggunaan logika [8].

Dari segi teori, pembangunan gagasan pemrograman dengan logika (*programming in logic*) yang pertama adalah Kowalski di Edinburgh. Kemudian didemonstrasikan secara eksperimental oleh Maarten Van Emden juga di Edinburgh, dan diwujudkan oleh Alain Colmeaurer di Universitas Marseilles, dengan berhasil membuat interpreter prolog yang pertama bersama kelompoknya pada tahun 1972 [8].

Yang paling berpengaruh dalam implementasi prolog ini adalah dihasilkannya kompiler prolog pertama untuk komputer DEC-10 karya Warren dan rekan-rekannya pada tahun 1979. Sistem ini menggunakan sintaks prolog baku (*standard*) dan efisiensinya membuktikan bahwa prolog merupakan bahasa tingkat tinggi. Selanjutnya karya David Warren juga melakukan persaingan dalam pengembangan bahasa kecerdasan buatan (*Artificial Intelligence*) yaitu Lisp dalam hal kecepatan. Hasil yang

didapat pada konferensi tentang komputer generasi kelima di London yang diselenggarakan oleh PL International, ditampilkan berbagai keberhasilan penggunaan prolog yang memberi kesan bahwa prolog adalah bahasa kecerdasan buatan yang lebih baik dari Lisp [8].

2.5.3 Pemrograman Logika

Untuk mengenal pemrograman logika, terlebih dahulu perlu mengetahui sekilas tentang logika predikat orde satu (*first order predicat logic*), yang kemudian dilanjutkan dengan logika predikat/kalkulus predikat. Logika predikat adalah logika yang juga mencakup *aljabar Boole*. Pada logika predikat, fakta dan aturan dinyatakan dalam predikat, seperti:

lelaki(Beck)	(fakta)
menikah(Beck, Ewon)	(fakta)
$\forall x \forall y$ [menikah(x,y) \wedge lelaki(x) \rightarrow \sim lelaki(y)]	(aturan)
$\forall x \exists y$ [orang(x) \rightarrow punyaibu(x,y)]	(aturan)

Kalimat pertama menyatakan Beck seorang lelaki. Yang kedua menyatakan bahwa Beck menikah dengan Ewon. Dan yang ketiga mengatakan untuk setiap x dan untuk semua y, jika x menikah dengan y dan x lelaki, maka y bukan lelaki. Dan yang terakhir menyatakan bahwa untuk setiap x, ada y, jika x orang, maka y adalah ibu dari x.

Jika dibuat dalam bentuk analogi contoh diatas symbol pada predikat dinyatakan: lelaki, menikah, orang punya ibu. Simbol konstanta dalam contoh tadi adalah Beck dan Ewon, dengan operator \wedge (dan), \rightarrow

(implikasi), \sim (negasi), \forall (untuk setiap), \exists (ada). Perhatikan bahwa ada dua simbol terakhir merupakan besaran logika (*kwantor*).

Fakta dan aturan dalam prolog tidak tertulis tepat seperti cara diatas tapi dalam bentuk yang terkadang disebut bentuk klausa *Kowalski*. Bentuk klausa ini sebenarnya diturunkan dari bentuk klausa *Horn*. Yang dimaksud klausa disini adalah suatu hubungan disjungsi beberapa literal fakta dan aturan, dan klausa Horn adalah klausa yang hanya terdiri dari (paling banyak) satu literal positif. Contoh klausa *Horn* dan penurunan menjadi klausa prolog adalah (tanda \vee berarti atau disjungsi):

$$\sim \text{lelaki}(X) \vee \sim \text{menikah}(X,Y) \vee \text{perempuan}(Y) \quad (\text{klausa Horn})$$

(perempuan merupakan satu-satunya literal positif) dengan menggunakan hukum De Morgan dan rumus:

$$\sim A \vee B = A \rightarrow B$$

diperoleh:

$$\begin{aligned} \sim(\text{lelaki}(X) \wedge \text{menikah}(X,Y)) \vee \text{perempuan}(Y) = \\ \sim \text{lelaki}(X) \wedge \text{menikah}(X,Y) \rightarrow \text{perempuan}(Y) \end{aligned}$$

Yang dalam prolog ditulis sebagai:

$$\text{Perempuan}(Y) \text{ if } \text{lelaki}(X) \text{ and } \text{menikah}(X,Y)$$

Gagasan yang melatarbelakangi pemrograman logika adalah menggantikan pemrograman dan komputasi dengan deskripsi logika suatu masalah dan mekanisme pembuktian untuk mendeduksi jawabannya. Pemikiran dasar dalam pemrograman logika membawa satu langkah

kedepan, yakni komputasi adalah deduksi. Sebagai contoh, fungsi faktorial biasa didefinisikan sebagai berikut:

```
faktorial(n) : = if n = 0 then 1
              else n * faktorial(n-1);
```

jika sebagai pengganti program diatas ditulis predikat dalam Prolog yang ditulis:

```
faktorial (n,m)
```

yang secara logika benar jika m adalah faktorial n, dan dengan klausa:

```
faktorial(0,1).
faktorial(n,m) if n1 = n-1 and faktorial(n1,m1) and
                m = n*m1.
```

Maka dengan klausa logika diatas dapat dihitung nilai faktorial. Inilah yang merupakan pokok pemrograman logika, yakni dengan klausa logika dapat dilakukan perhitungan [9].

2.5.4 Ciri Bahasa Prolog

Ciri-ciri bahasa prolog yang membedakan dengan bahasa program lain yaitu selain mencari kesimpulan atau jawaban secara logika terhadap pernyataan yang diajukan dapat pula memberi jawaban lain atau memberi beberapa kemungkinan jawaban. Berikut ciri lain yang membedakan bahasa prolog dengan bahasa program lain:

a. Predikat

Pada Prolog dibutuhkan fakta-fakta yang terungkap dalam relasi dan sifat untuk mencari jawaban. Contoh:

```
Irham suka permen
```

Permen manis

Digambarkan Irham sebagai objek dan permen sebagai relasi suka, dan sifat manis pada permen. Relasi bernilai benar jika memenuhi kondisi tertentu. Hubungan ini disebut aturan (*rule*).

Contoh aturan adalah:

Irham suka permen jika rasa permen asam

Pada prolog, aturan dan fakta dinyatakan dalam kalimat yang disebut klausa (*clause*) seperti;

suka (Irham,permen).

Sintaks untuk menyatakan fakta seperti diatas disebut logika predikat (*predicate logic*). Aturan di atas dapat dituliskan dalam prolog sebagai berikut:

Suka(Irham, permen) if rasa (permen, asam)

atau

suka (Irham, permen) if asam (permen)

Logika predikat ini disebut sebagai predikat yang merupakan nama simbolik suatu relasi atau sifat, sedangkan objek yang suka dikaitkan disebut *argument*. Contoh : predikat diatas adalah suka dengan argumen Irham dan permen.

b. Basis Data

Pada prolog untuk menunjang orientasi pada fakta yang ada dalam basis pengetahuan, prolog menyediakan fasilitas basis data.

c. Deduksi

Pada prolog dapat dilakukan deduksi (penarikan kesimpulan).

d. Pemadanan dan Unifikasi

Untuk menarik sesuatu kesimpulan (*reasoning*) untuk mencari jawaban, prolog melakukan pemadanan atau pencocokan. Argumen pertama dicocokkan dengan argumen pertama dengan predikat lain yang sama, dan seterusnya. Tujuan dilakukan pemadanan adalah untuk melakukan unifikasi, yakni penggabungan argument dalam suatu predikat dengan predikat lain yang lebih indentik. Contoh : sebuah bujur sangkar adalah bangun segi empat, dengan ukuran sama dan bersudut 90 derajat. Jika diminta untuk mendeteksi bagaimana suatu bangun berbentuk bujur sangkar. Langkah pertama yang dilakukan adalah menduga ciri-ciri dari bentuk bangun bujur sangkar. Jika ketiga ciri diatas terpenuhi, maka dugaan tersebut benar bahwa bangun yang di periksa adalah bujur sangkar. Pada prolog, dugaan suatu bangun berbentuk bujur sangkar adalah unifikasi variabel, Misal nilai bujur sangkar dilambangkan dengan X. Hal ini terjadi karena ada pencocokan fakta dan aturan antara bangun (X) dengan bangun (bujur sangkar) yang sudah identik, selanjutnya dilakukan unifikasi antara X dengan bujur sangkar.

Pada bahasa program lain, pemberian nilai terhadap suatu variabel dilakukan dengan statemen pemberian (*assignment statement*) seperti $y = 2$. Ini bersifat mutlak selama tidak ada statemennya tidak diubah. Unifikasi bersifat sementara selama belum ada lacak balik dan hanya berlaku satu klausa, sebagaimana hanya berlaku dalam satu fungsi seperti dalam C.

e. Pembuktian Mundur (*Backward Chaining*)

Penyelesaian suatu masalah dengan metode pembuktian mundur terhadap suatu bangun berbentuk bujur sangkar, yaitu dengan cara menganggap suatu bangun berbentuk bujur sangkar terlebih dahulu, kemudian baru dilakukan pembuktian. Metode ini disebut pembuktian mundur (*backward chaining*) atau metoda *top-down* karena penyelesaian dimulai dari kenyataan pokok (bujur sangkar) ke masalah rinci (sudut dan sisi).

f. Lacak Balik (*backtracking*)

Penyelesaian suatu masalah dengan metode lacak balik (*backtracking*) pada prolog dilakukan dengan cara melacak suatu fakta atau aturan dengan melakukan pemadanan (*matching*) berturut-turut. Jika berhasil maka akan terjadi *unifikasi* dan dilanjutkan dengan pembuktian yang terakhir dilakukan pemadanan serupa.

Jika dilogikakan pembuktian lacak balik (*backtracking*) dalam kasus permasalahan seperti diatas dan terjadi kegagalan, katakanlah karena sudut-sudutnya bukan 90 derajat yang harus dilakukan adalah mengubah anggapan bahwa suatu bangun berbentuk bujur sangkar. Selanjutnya, mencari anggapan baru dengan definisi baru, misalnya bentuk trapesium. Kejadian seperti ini pada prolog disebut sebagai lacak balik (*backtracking*). Jadi, ketika ditemui kegagalan, dimana hipotesa pertama tidak dapat digunakan harus dibatalkan. Kemudian Prolog melakukan pelacakan mundur (lacak balik) untuk mencari hipotesa baru. Proses

penyelesaian lacak balik ini akan berhenti jika ditemui jawaban atau kehabisan fakta untuk menyimpulkan jawaban [9].

2.5.5 Bahasa Deklaratif

Prolog disebut sebagai bahasa deklaratif karena prolog hanya membutuhkan deklarasi atau uraian masalah sedangkan yang lain memerlukan perintah. Untuk memecahkan suatu masalah, prolog melakukan dengan cara deduksi yaitu dari sekumpulan fakta ditarik kesimpulan atau dapat dikatakan kumpulan fakat yang bersifat umum diturunkan ke dalam bentuk yang lebih khusus. Dalam pembuatan prolog progamer hanya berkewajiban memberikan perincian masalah secara benar, bukannya membuat sekumpulan perintah untuk dikerjakan komputer [9].

Perbedaan bahasa deklaratif dengan bahasa *procedural* dapat digambarkan dengan perumpamaan seperti sedang naik taksi. Secara deklaratif, penumpang hanya meminta supir taksi untuk dibawa ke suatu tempat tujuan, tidak peduli jalan mana saja yang harus ditempuh. Penentuan jalan yang ditempuh diserahkan kepada supir. Dalam pengertian *procedural*, penumpang yang menentukan jalan mana saja yang harus ditempuh dan boleh saja supir taksi tidak tahu kemana tujuan penumpang sebenarnya [9].

Sebagai bukti bahwa prolog merupakan bahasa deklaratif adalah:

1. Misal : bahwa A adalah orang tua B, dalam prolog cukup ditulis:

Orangtua(A,B).

2. A adalah leluhur Z, maka harus didefinisikan terlebih dahulu logikanya yang dalam bahasa Indonesia berbunyi sebagai berikut:

A adalah leluhur Z jika A orang tua Z atau
A adalah leluhur Z jika orang tua B dan B leluhur Z.

Aturan seperti ini dalam prolog ditulis sebagai aturan rekursif seperti berikut:

leluhur(A,Z) if orangtua(A,Z).
leluhur(A,Z) if orangtua(A,B) and leluhur(B,Z).

Dengan cara menuliskan aturan seperti ini, dapat menentukan siapa leluhur Z, siapa keturunan A atau membuktikan kebenaran bahwa A leluhur Z dengan memberikan data hubungan (fakta) orang tua secukupnya seperti diatas tanpa harus merinci cara mencari jawaban atau cara membuktikannya. [9]

Aturan seperti diatas diturunkan berdasarkan logika, sehingga pemrograman prolog disebut pemrograman logika (*logic programming*).

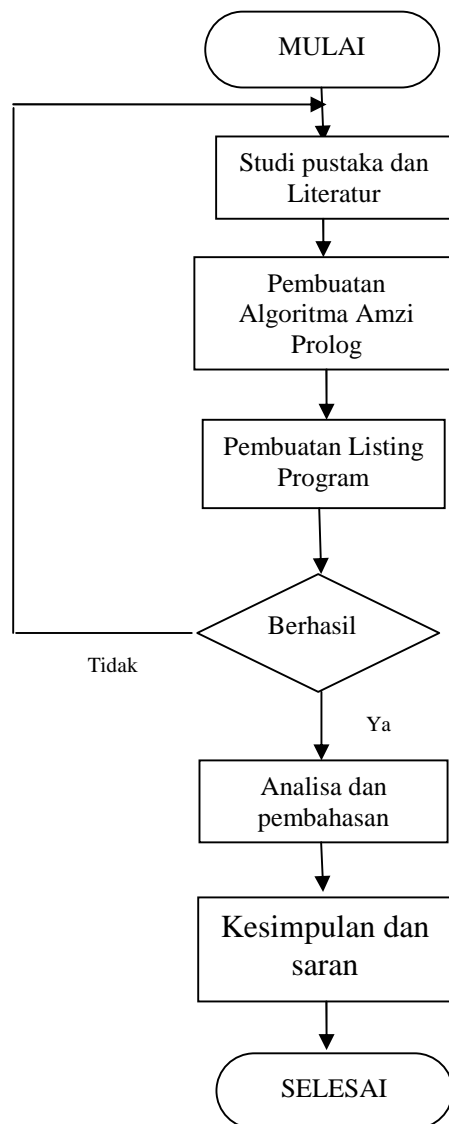
Dari contoh diatas dapat dilihat bahwa prolog adalah bahasa berjenis *deklaratif*, sedangkan bahasa program lainnya berjenis *procedural*.

Dengan adanya Prolog, diharapkan komputer dapat berfungsi sebagai pemecah masalah (*Intelligence*) atau komputer yang harus memahami masalah yang harus dipecahkan [9].

BAB III

METODE PENELITIAN

3.1 Diagram Alir Penelitian



Gambar 3.1 Diagram alir penelitian

3.2 Waktu dan Tempat

Penelitian tugas akhir ini dilaksanakan pada :

Waktu : Desember 2016 – Maret 2017

Tempat : Lab.Terpadu Teknik Sistem Komunikasi dan Informatika
Jurusan Teknik Elektro, Universitas Lampung

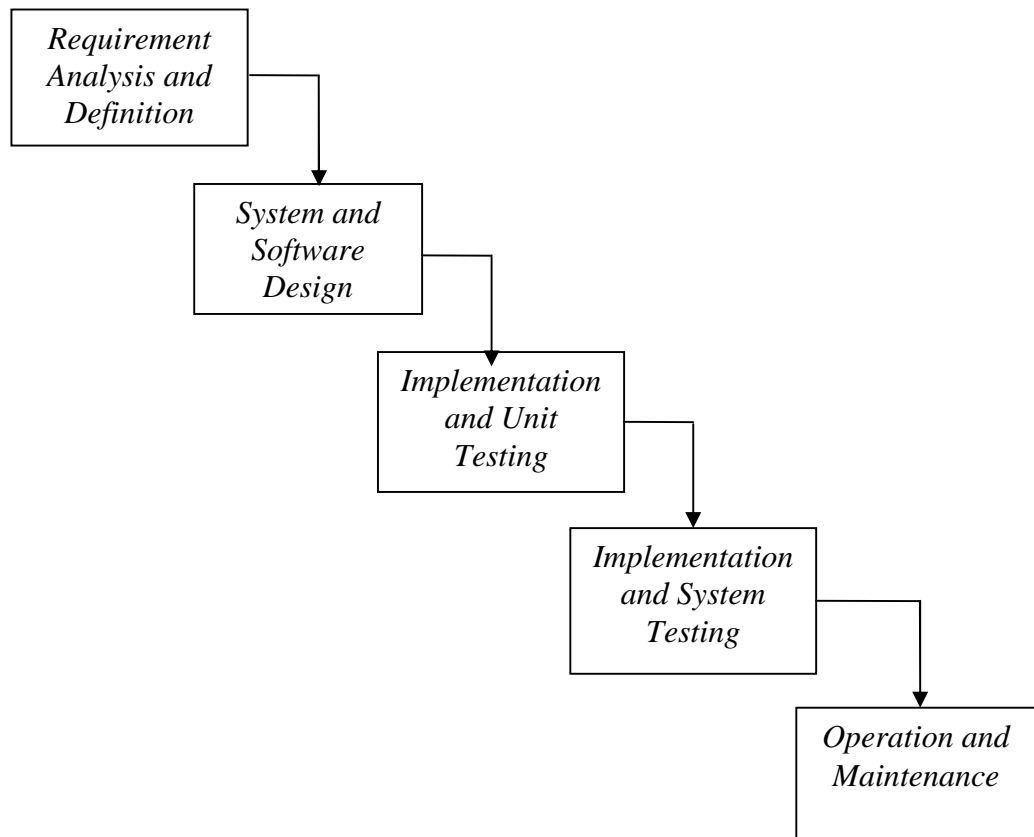
3.3 Alat dan Bahan

Peralatan yang digunakan pada penelitian ini adalah sebagai berikut:

1. *Notebook merk TOSHIBA NB 520 Intel atom N2800 1.86Ghz, RAM 2GB (32-bit)*
2. *Software Amzi Prolog (ter-install pada notebook) sebagai program untuk mensimulasikan penelitian yang dilaksanakan.*

3.4 Tahap Perancangan Penelitian

Dalam perancangan pembuatan sistem pakar diagnosa hama dan penyakit tanaman padi ini dilakukan beberapa tahap-tahap perancangan menggunakan metode *waterfall* dengan 5 tahapan yang harus dilewati oleh sebuah sistem dalam pengembangannya apabila menggunakan implementasi dari metode pengembangan *waterfall*. Berikut ini adalah kelima tahapan yang harus dilewati oleh pengembangan sistem tersebut:



Gambar 3.2 Metode *Waterfall*

Tahapan pada metode *waterfall* :

1. Tahapan analisis

Tahapan analisis mengacu pada fenomena dan juga permasalahan yang terjadi, dan mengapa sebuah aplikasi sangat penting untuk dibuat dalam mengatasi masalah atau fenomena tersebut.

2. Tahapan desain

Tahapan berikutnya adalah pembuatan desain dari sebuah sistem. Dalam tahapan ini, tidak hanya desain *user interface* sistemnya saja yang dikembangkan, namun juga dikembangkan desain dari alur sistem tersebut,

hingga bagaimana satu sistem tersebut bisa bekerja, mulai dari tampilan awal, fungsi-fungsi tombol, hingga output yang akan dihasilkan nantinya.

3. Tahapan Pengkodean

Pengkodean merupakan tahapan yang wajib dilakukan oleh mereka yang mengerti bahasa pemrograman, Untuk menjalankan desain sistem yang sudah dibuat, maka kemudian kode dan juga script akan dimasukkan ke dalam desain sistem tersebut, sehingga nantinya desain dari sistem tersebut bisa berjalan dengan lancar dan juga baik.

4. Tahapan Pengujian

Setelah sistem selesai dilakukan pengkodean, maka sistem tersebut akan diuji sebelum digunakan oleh *user*. Dalam pengujian dilihat apakah sistem dapat bekerja dengan baik, tampilan *interface* sesuai harapan, dan semua fungsinya bisa digunakan dengan baik dan lancar.

5. Tahapan Pemeliharaan (*Maintenance*)

Tahapan pemeliharaan (*maintenance*) mengacu pada *update-update* dari sebuah sistem yang mungkin mengalami kerusakan, perbaikan terhadap sistem yang mengalami *corrupt* dan kerusakan, serta penambahan fitur-fitur baru pada sistem tersebut. Tahap ini ditentukan oleh kebutuhan dari *user*, dan apabila sebuah sistem memiliki *support* yang baik, maka sistem tersebut akan berkembang dengan sangat baik [7].

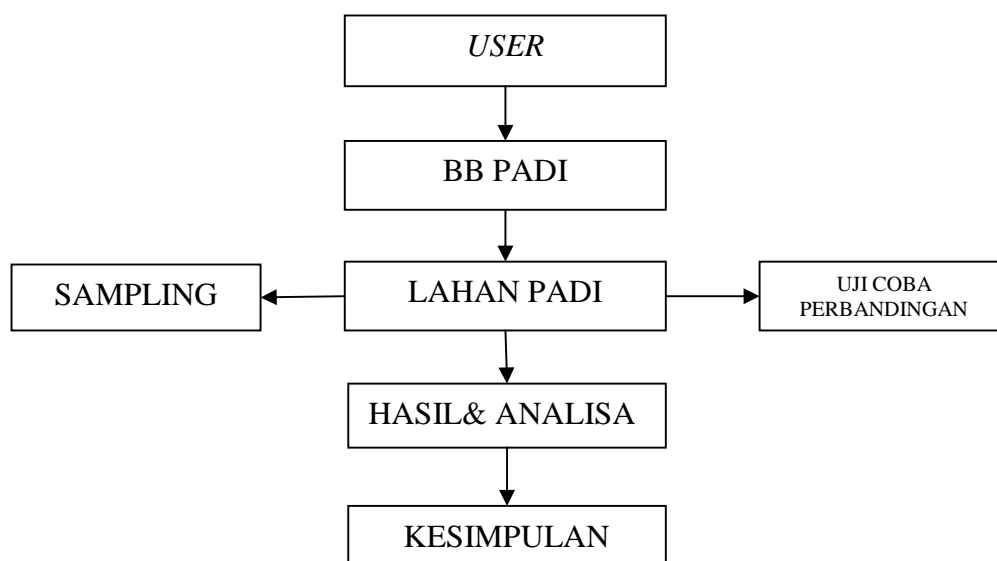
Pada penelitian yang dilakukan dirancang dengan metode *waterfall* karena metode ini memiliki proses yang urut, mulai dari analisa hingga *maintenance*. Setiap proses memiliki spesifikasinya sendiri, sehingga

sebuah sistem dapat dikembangkan sesuai dengan apa yang diinginkan *user*. Setiap proses tidak dapat saling tumpang tindih.

3.5 Analisis kebutuhan dan definisi

a) Investigasi

Untuk membuat suatu sistem yang baru, harus dimulai dari awal dengan cara mengumpulkan informasi selengkap-lengkapnyanya. Khususnya jika ada kelemahan-kelemahan dari sistem lama, maka sistem yang baru harus dapat menutupi kelemahan sistem lama. Pencarian data yang berkaitan dengan kelemahan sistem lama dilakukan dengan investigasi pada diagnosa sistem lama. Berikut adalah blok diagram diagnosa kerusakan tanaman padi dengan sistem lama :



Gambar 3.3 Blok diagram diagnosa kerusakan tanaman dengan sistem lama

Dari gambar 3.3 diatas dapat dilihat tahap diagnosa kerusakan tanaman padi yang disebabkan oleh hama dan penyakit, yaitu : petani perlu mendatangi BB PADI (Balai Besar Penelitian Tanaman Padi) untuk konsultasi mengenai gejala kerusakan yang ditimbulkan pada tanaman padi. Pada tahap konsultasi tidak sama seperti konsultasi seorang dokter dengan pasiennya mengingat objek yang sakit adalah benda mati. Untuk itu tidak hanya sekedar berkonsultasi, terkadang diperlukan peninjauan langsung pada wilayah yang terserang penyakit karena faktor lingkungan (suhu, kelembaban dan vegetasi alam) dan unsur kandungan tanah juga menjadi salah satu faktor. Selain itu membandingkan tanaman sehat dengan tanaman yang sakit juga merupakan salah satu tahapan diagnosa.

Dapat dilihat bahwa pada sistem yang lama terdapat kelemahan, diantaranya ialah:

- Petani harus direpotkan datang ke klinik penyakit dan hama tanaman (PHT) atau laboratorium pertanian atau BB PADI (Balai Besar Penelitian Tanaman Padi) untuk melakukan konsultasi mengenai kerusakan yang diserang pada tanaman.
- Diperlukan waktu dan biaya yang cukup banyak untuk berkonsultasi langsung dengan seorang ahli hama dan penyakit tanaman.

Pada sistem yang baru, petani bisa melakukan pencarian data mengenai jenis penyakit dan penanggulangan tanpa harus datang menemui seorang ahli pertanian atau klinik PHT atau BB PADI (Balai Besar Penelitian Tanaman Padi). Petani hanya perlu menginstall *software amzi prolog* yang

didalamnya terdapat *database* sistem pakar hama dan penyakit tanaman padi yang telah dibangun, sehingga tidak perlu mengeluarkan biaya dan waktu yang banyak untuk menemui langsung seorang pakar.

b). Analisa

Metode pengambilan data yang dilakukan pada penelitian ini menggunakan dua metode, yaitu studi literatur dan wawancara (*interview*). Metode ini digunakan karena praktis dan cukup menghemat biaya dan waktu. Metode wawancara (*interview*) dilakukan dengan tiga narasumber yang berbeda. Narasumber pertama adalah dosen pertanian di bidang hama dan penyakit tanaman di Politeknik Negeri Lampung (POLINELA) yaitu Bapak Ir. Iwan Gunawan, M.P.. Kedua dosen pertanian di bidang hama dan penyakit tanaman di Politeknik Negeri Lampung (POLINELA) yaitu Ibu Ir. Raida Kartina, M.P. dan ketiga Bapak Deroni yaitu salah satu petani padi yang ada di wilayah Palembang tepatnya di desa talang putri yang digunakan sebagai referensi nyata yang ada di lapangan.

Metode *interview* ini dilakukan bertujuan untuk mendapatkan data informasi yang dibutuhkan untuk menunjang ketepatan informasi pada sistem pakar seperti apa yang dibutuhkan *user*. Informasi tersebut kemudian dimasukkan ke dalam bentuk *database* yang ada pada *knowledge base* yang merupakan pengetahuan yang dibutuhkan untuk memahami, menformulasikan, dan memecahkan masalah pada sistem pakar yang dibangun.

Pada tahap wawancara yang dilakukan dengan seorang dosen pertanian didapat informasi data hama dan penyakit tanaman secara langsung dan tidak langsung. Informasi secara langsung yang dimaksud yaitu sebagai berikut :

➤ **Pertumbuhan Tanaman Padi**

- a. Stadia vegeratif – dari perkecambahan sampai terbentuknya bulir.
Pada varietas padi yang berumur pendek (120 hari) stadia ini lamanya sekitar 55hari, sedangkan pada varietas padi berumur panjang (150hari) lamanya sekitar 85 hari.
- b. Stadia reproduktif – dari terbentuknya bulir sampai pembungaan.
Pada varietas berumur pendek lamanya sekitar 25 hari, sedangkan pada varietas berumur panjang sekitar 35 hari juga.
- c. Stadi pembentukan gabah atau biji dari pembungaan sampai pemasakan biji. Lamanya stadia ini sekitar 30 hari, baik untuk varietas padi berumur pendek maupun berumur panjang [13].

➤ **Pengertian Hama Dan Penyakit Tanaman Padi**

Hama adalah sejenis hewan atau makhluk hidup yang mengganggu tanaman dan berpotensi merusak tanaman serta merugikan petani dari segi ekonomi. Sedangkan penyakit adalah kerusakan tanaman yang disebabkan oleh gangguan yang dibawah oleh vektor atau serangga atau makhluk hidup dan makhluk non-hidup.

➤ Penyebab Kerusakan Padi Yang Biasa Menyerang Tanaman Padi

Berikut adalah data informasi hama dan penyakit tanaman padi yang diperoleh dari wawancara dan studi literatur yang telah dilakukan dan telah dimasukkan ke dalam tabel [3]:

Tabel 3.1 Hama dan penyakit pada tanaman padi

Jenis Hama	Gejala serangan	Pengendalian /Penanggulangan
Penggerek batang (<i>stem borer</i>) -penggerek batang padi kuning (<i>Scirpophaga incertulas</i>)	-adanya ngengat penggerek batang padi atau kupu-kupu -ngengat penggerek batang padi berwarna kuning kecokelatan -adanya titik hitam dibagian belakang sayap depan -adanya ulat -pangkal batang tanaman terpotong (sundep) -matinya tunas-tunas padi -matinya malai (daun padi)	- <i>insektisida</i> , berbahan aktif : <i>Karbonfuran</i> , <i>bensultap</i> , <i>karbosulfan</i> , <i>dimenhipo</i> , <i>amitraz</i> & <i>fipronil</i> . Kecuali, kupu-kupu, jangan memakai pestisida semprot untuk sundep dan beluk.
-penggerek batang padi putih (<i>S. innotata</i>)	-adanya ngengat penggerek batang padi atau kupu-kupu -ngengat penggerek batang padi berwarna putih -adanya ulat -pangkal batang tanaman terpotong (sundep) -matinya tunas-tunas padi -matinya malai (daun padi)	- <i>insektisida</i> , berbahan aktif : <i>Karbonfuran</i> , <i>bensultap</i> , <i>karbosulfan</i> , <i>dimenhipo</i> , <i>amitraz</i> & <i>fipronil</i> . Kecuali, kupu-kupu, jangan memakai pestisida semprot untuk sundep dan beluk.
-penggerek batang padi bergaris (<i>Chilo suppressalis</i>)	-adanya ngengat penggerek batang padi atau kupu-kupu -ngengat penggerek batang padi berwarna kecokelatan -adanya ulat -pangkal batang tanaman terpotong (sundep) -matinya tunas-tunas padi -matinya malai (daun padi) -pada ulat (larva) terdapat 2 garis memanjang	- <i>insektisida</i> , berbahan aktif : <i>Karbonfuran</i> , <i>bensultap</i> , <i>karbosulfan</i> , <i>dimenhipo</i> , <i>amitraz</i> & <i>fipronil</i> . Kecuali, kupu-kupu, jangan memakai pestisida semprot untuk sundep dan beluk.
Wereng cokelat (<i>brown planthopper-BPH</i>)	- daun-daun menguning -daun yang terhisap hanya bagian pinggir	-Dikendalikan dengan varietas tahan -Penanaman padi dengan jarak

<p><i>Nilaparvata lugens</i></p>	<p>-daun-daun mengering - daun tampak seperti terbakar -Kerusakan yang disebabkan dimulai dari satu titik, kemudian menyebar ke segala arah dalam bentuk lingkaran -perkembangan akar terganggu -bagian bawah tanaman menjadi terlapisi oleh jamur</p>	<p>tanam (tidak terlalu rapat) -Pergiliran varietas -insektisida dengan bahan aktif : <i>amitraz, bupofresin, beauveria, bassiana</i> 6.20x10¹⁰cfu/ml, <i>BPMC, fipronil, amidaklopid, karbofuran, karbosulfan, metolcarb, MIPCI, propoksur, atau tiametoksan.</i></p>
<p>Wereng hijau (<i>green leafhopper</i>) -<i>Nephotettix virescens</i> -<i>N. nigropictus</i> -<i>N. cincticeps</i> -<i>N. malayanus</i></p>	<p>-daun-daun menguning -daun yang terhisap hanya bagian pinggir -daun menguning hingga jingga - jumlah anakan menurun -pertumbuhan terhambat (batang padi memendek). -pertumbuhan daun menjadi kerdil</p>	<p>-Mengurangi pemupukkan unsur <i>nitrogen</i> (memperlambat perkembangan biakkan WH) -dianjurkan menanam varietas tahan tungro seperti tukad petani, kalimas dan bondoyudo -<i>Insektisida</i> berbahan aktif :<i>BPMC, bufrezin, imidklopid, karbonfuran, MIPC</i> atau <i>tiametoksan</i></p>
<p>Kepinding tanah (<i>black bug</i>) <i>Scotinophara coarctata</i></p>	<p>-daun-daun menguning -daun menguning hingga kemerahan - jumlah anakan menurun -pertumbuhan terhambat (batang padi memendek) - pertumbuhan daun menjadi kerdil -gabah hampa (kosong)</p>	<p>-membersihkan lahan dari berbagai gulma agar sinar matahari dapat mencapai dasar kanopi tanaman padi -menanam varietas padi berumur genjah, untuk menghambat peningkatan populasi kepinding tanah</p>
<p>Walang sangit (<i>rice bug</i>) <i>Leptocorisa oratorius</i></p>	<p>-fase terserang pada saat gabah sedang mengisi -gabah menjadi hampa -beras berubah warna -beras mengapur</p>	<p>-mengendalikan gulma, baik yang ada di sawah maupun yang ada disekitar pertanaman -meratakan lahan dengan baik dan memupuk tanaman secara merata agar tanaman seragam -menangkap walang sangit dengan menggunakan jaring sebelum stadia pembungaan -mengumpan walang sangit dengan bangkai ikan / bangkai daging/ kotoran ayam -insektisida di pagi atau sore hari, ketika walang sangit berada di kanopi Insektisida berbahan aktif : <i>BPMC, fipronil, metolcarb, MIPC</i> atau <i>propoksur.</i></p>

<p>Tikus (<i>Rat</i>) <i>Rattus argentiventer</i></p>	<ul style="list-style-type: none"> -daun-daun menguning -kerusakan yang diserang mulai dari tengah petak meluas ke arah pinggir dan menyisakan 1-2 petakan di pinggir petakan -jumlah anakan menurun -adanya jejak-jejak kaki -adanya jalur jalan -adanya kotoran -adanya lubang aktif 	<p>-Secara fisik : memburu tikus secara masal dengan cara mengepung tikus kemudian dipukul menggunakan kayu</p> <p>-pemasangan TBS (<i>Trap Barrier System</i>), yaitu pemasangan pagar plastik setinggi 60cm, dan ditegakkan dengan bambu yang berjarak 1m, memiliki bubu perangkap pada setiap sisi pagar plastik dengan lubang menghadap keluar dan dilengkapi dengan tanggul sempit sebagai jalan masuk tikus. Tbs dikelilingi parit dengan lebar 50cm yang selalu tergenang air untuk mencegah tikus menggali / melubangi pagar plastik. Prinsip kerja TBS menarik tikus dari lingkungan sawah disekitarnya karena tikus tertarik padi yang ditanam lebih awal dan bunting lebih dahulu, sehingga dapat mengurangi populasi tikus sepanjang pertanaman</p> <p>- Membuat LTBS, hamper sama dengan TBS yaitu membuat pagar plastik dan dilengkapi bubu perangkap pada dua sisi berselang-seling agar mampu menangkap tikus dari dua arah</p> <p>-fumigasi, yaitu membunuh tikus beserta anaknya didalam lubangnya (lubang reproduksi)</p>
<p>Ganjur (Gall midge) <i>Orseolia oryzae</i></p>	<ul style="list-style-type: none"> -tunas padi yang terserang menjadi tergulung menyerupai daun bawang - ganjur berbentuk seperti nyamuk kecil - ganjur aktif pada malam hari dan sangat tertarik pada lampu 	<ul style="list-style-type: none"> -atur waktu tanam agar puncak curah hujan tidak bersamaan dengan stadia vegetatif -bajak ratun/tunggul yang berasal dari tanaman sebelumnya dan buang/bersihkan semua tanaman inang alternatif seperti padi liar (<i>oryzae rufipogon</i>) selama masa bera -tanam varietas tahan seperti ciliwung dan asahan

		<p>-gunakan lampu perangkap untuk menarik perhatian ganjur dewasa</p> <p>-<i>insektisida</i> berbahan aktif : <i>karbonfuran</i></p>
<p>Hama putih palsu (<i>leaffolder</i>) <i>Cnaphalocrocis medinakis</i></p>	<p>-adanya ngengat atau kupu-kupu</p> <p>-ngengat berwarna kuning coklat yang memiliki 3 buah pita hitam dengan garis lengkap / terputus pada bagian sayap depan</p> <p>-ngengat bertentuk segitiga</p> <p>-adanya bercak pada daun</p> <p>-bercak berwarna putih</p> <p>-daun berubah warna menjadi putih dan terlipat</p>	<p>-upayakan pemeliharaan tanaman sebaik mungkin agar pertanaman tumbuh secara baik, sehat dan seragam</p> <p>-pergunakan <i>insektisida</i> (bila diperlukan) berbahan aktif : <i>fipronil</i> atau <i>karbonfuran</i></p> <p>-jangan menggunakan <i>insektisida</i> sampai tanaman berumur 30hari setelah tanam pindah atau 40 hari sesudah sebar benih</p> <p>-tanaman padi yang terserang pada fase ini dapat pulih apabila air dan pupuk dikelola dengan baik</p>
<p>Hama putih (<i>caseworm</i>) <i>Nymphula depunctalis</i></p>	<p>-adanya ngengat atau kupu-kupu</p> <p>-ngengat berwarna putih</p> <p>-daun terpotong seperti digunting</p> <p>-daun yang terpotong menyerupai tabung untuk membungkus larva dengan benang-benang sutranya</p>	<p>-tindakan pengendalian diperlukan kalau tingkat serangan mencapai >25% daun rusak atau 10daun rusak per-rumpun</p> <p>-<i>insektisida</i> (bila diperlukan) berbahan aktif: <i>fipronil</i> atau <i>karbonfuran</i></p>
<p>Ulat tentara /grayak (<i>armyworm</i>) <i>-spodoptera mauritia acronyctoides -mythimna separata -spodoptora exempta -spodoptera litura</i></p>	<p>-adanya atau kupu-kupu</p> <p>-ngengat cenderung lebih aktif pada malam hari</p> <p>Ngengat menyukai cahaya</p> <p>-daun-daun yang dimakan mulai dari tepi daun dan hanya meninggalkan tulang daun dan batang</p>	<p>-<i>insektisida</i> (bila diperlukan) berbahan aktif : <i>BPMC</i> atau <i>karbofuran</i></p>
<p>Ulat tanduk hijau (<i>green horned caterpillar</i>) <i>Melanitis leda ismene cramer</i></p>	<p>-adanya ngengat atau kupu-kupu</p> <p>-ngengat cenderung lebih aktif pada malam hari</p> <p>-ngengat tidak menyukai cahaya</p> <p>-ngengat berukuran besar dan</p>	<p>-hama ini sebaiknya dikendalikan dengan cara memanfaatkan musuh alami, seperti parasit telur <i>Trichogrammatidae</i>. Oleh karena itu pengendalian secara kimiawi dengan <i>insektisida</i></p>

	<p>pada sayap terdapat bercak berbentuk seperti mata</p> <ul style="list-style-type: none"> -larva memiliki 2 pasang tanduk -daun-daun yang dimakan mulai dari pinggiran hingga ujung daun 	<p>tidak dianjurkan pada saat tanaman berumur 30hari setelah tanaman pindah atau 40hari setelah sebar benih.</p>
<p>Ulat jengkal palsu hijau (<i>green semilooper</i>) <i>Naranga aenescens</i></p>	<ul style="list-style-type: none"> -adanya ngengat atau kupu-kupu -adanya bercak putih pada daun -daun yang di makan dari pinggiran daun -tanaman padi yang diberi pupuk berlebihan pemicu meningkatnya populasi ulat jengkal - selama musim hujan populasi meningkat 	<ul style="list-style-type: none"> -Untuk mengendalikan hama ini sebaiknya dengan memanfaatkan musuh alami seperti parasit telur <i>Trichogrammatidae</i>, parasit larva dan pupa seperti <i>Ichneumonidae</i>, <i>Braconidae</i>, <i>Eulophidae</i>, <i>Chalcidae</i>, dan laba-laba pemangsa ngengat
<p>Orong-orong (<i>mole cricket</i>) <i>Gryllotalpa orientalis brumeister</i></p>	<ul style="list-style-type: none"> -pada pangkal tanaman terpotong -rusaknya akar-akar muda dan bagian pangkal tanaman yang berada di bawah tanah -padi muda yang diserang mati sehingga terlihat adanya spot-spot kosong di sawah 	<ul style="list-style-type: none"> -perataan tanah agar tergenang merata -penggenangan sawah 3-4hari dapat membantu membunuh telur orong-orong tanah -Penggunaan umpan (sekam di campur insektisida) -<i>insektisida</i> (bila diperlukan) berbahan aktif <i>karbofuran</i> atau <i>fipronil</i>
<p>Lalat bibit (<i>rice whorl maggot</i>) <i>Hydrellia philippina ferino</i></p>	<ul style="list-style-type: none"> -adanya bercak kuning pada tepian daun -daun yang terserang menjadi berubah bentuk dan menggulung -telur-telur diletakkan dipermukaan atas daun -pada daun yang tergulung terdapat larva berwarna kuning kehijauan yang tembus cahaya 	<ul style="list-style-type: none"> -hama ini dapat dikendalikan dengan cara mengeringkan sawah -<i>insektisida</i> (bila diperlukan) berbahan aktif : <i>bensultap</i>, <i>BMPC</i> atau <i>karbofuran</i>
<p>Keong mas (<i>golden apple snail</i>) <i>Pomacea canaliculata</i></p>	<ul style="list-style-type: none"> -jaringan tanaman rusak -daun dan batang terpotong seperti digunting -bekas potongan daun dan batang mengambang -adanya bibit yang hilang (per-tanaman) -adanya telur (keong) pada 	<ul style="list-style-type: none"> -secara fisik, membuang keong dengan secara langsung menggunakan tangan -secara mekanis, pungut keong dan kemudian hancurkan. -bila di suatu lokasi sudah diketahui bahwa keong mas adalah hama utama, sebaiknya

	tanaman berwarna merah muda	tanam bibit umur 21hari dan tanam lebih dari satu bibit per-rumpun, kemudian buat caren di dalam dan di sekeliling petakan sawah - <i>pestisida</i> (bila diperlukan) berbahan aktif : <i>niclos amida</i> , <i>pestisida botani</i> seperti, lerak, deris dan saponin (aplikasi pestisida di caren, cekungan-cekungan yang airnya terdapat keong berkumpul)
Burung (<i>bird</i>) - <i>Lonchura spp</i> - <i>Ploceus sp.</i>	-biji hampa -adanya gejala seperti beluk -biji banyak yang hilang	-penjagaan burung mulai dari jam 6-10pagi dan jam 2-6sore karena waktu-waktu tersebut merupakan waktu kritis tanaman padi terserang oleh burung -gunakan jaring untuk mengisolasi sawah dari serangan burung, luas sawah yang diisolasi kurang lebih 0,25hektar -bila tanam tabela : -Benih yang sudah disebar disawah,ditutup dengan tanah -Benih yang digunakan harus lebih banyak -Gunakan orang-orangan sawah yang diberi plastik untuk menakut-nakuti burung -pekerjakan penjaga burung Tanam serentak dengan sekitarnya (warga) -kendalikan habitat/sarang burung
Hawar daun bakteri (<i>bacterial leaf blight-BLB</i>) <i>Xanthomonas campestris pv.oryzae</i>	-adanya bercak pada daun -bercak berwarna abu-abu (kekuningan) -daun-daun menguning -daun menguning hingga mengering dan layu (seperti terbakar/ tersiram air panas) -daun melipat dan menggulung	-penyakit HDB efektif dikendalikan dengan varietas tahan : pemupukan lengkap, dan pengaturan air -untuk daerah endemis, tanam varietas tahan seperti code dan angke dan gunakan pupuk NPK dalam dosis tepat -bila memungkinkan hindari penggenangan yang terus menerus, mis: 1hari digenangi 3hari di keringkan

<p>Bakteri daun bergaris (<i>bacterial leaf streak</i>) <i>Xanthomonas campestris</i> pv. <i>oryzicola</i></p>	<p>-adanya bercak pada daun -bercak hijau gelap hingga kuning tembus cahaya diantara pembuluh daun -bercak lama kelamaan membesar berwarna cokelat -fase penyebaran meluas, seluruh tanaman berwarna oranye kekuning-kuningan</p>	<p>-buang atau hancurkan tunggul-tunggul dan jerami-jerami dari tanaman sakit yang sudah terdekomposisi sempurna sebelum tanam pindah -gunakan benih atau bibit -atur jarak tanam tidak terlalu rapat -berakan tanah sesudah panen</p>
<p>Blas (<i>blast</i>) Jamur ; <i>Pyricularia grisea</i></p>	<p>-adanya bercak pada daun -bercak berwarna cokelat kehitaman berbentuk belah ketupat dengan pusat bercak berwarna putih -adanya bercak cokelat kehitaman pada pangkal leher yang dapat mengakibatkan leher malai(daun) tidak mampu menopang malai dan patah</p>	<p>-lakukan penanaman varietas tahan secara bergantian untuk mengantisipasi perubahan ras blas yang sangat cepat dan penumpukkan NPK yang tepat -penanaman dalam waktu yang tepat serta perlakukan benih dapat diupayakan -<i>fungisida</i> (bila diperlukan) berbahan aktif : <i>metil tiofanat</i>, <i>fosdifen</i>, atau <i>kasugamisin</i></p>
<p>Hawar pelepah (<i>sheath blight</i>) <i>Rizoctonia solanikuhn</i></p>	<p>-adanya bercak pada daun -bercak berbentuk bulat atau oval memanjang -bercak berwarna keabu-abuan pucat pada pelepah -tanaman menjadi mudah rebah (tumbang) -gabah kurang terisi bahkan berakibat hampa</p>	<p>-penggunaan pupuk tanaman dengan dosis 250kg <i>urea</i>, 100kg <i>SP36</i> dan 100kg <i>kcl per ha</i> dapat menekan perkembangan penyakit ini -atur jarak tanaman agar jangan terlalu rapat -keringkan sawah beberapa hari pada saat anakan maksimum -bajak yang dalam untuk mengubur sisa-sisa tanaman yang terinfeksi -rotasi tanaman dengan kacang untuk menurunkan serangan penyakit -buang gulma dan tanaman yang sakit dari sawah -<i>fungisida</i> (bila diperlukan) berbahan aktif : <i>heksa konadol</i>, <i>kerbendazim</i>, <i>tebukonazol</i>, <i>belerang</i>, <i>flutalonil</i>, <i>difonokonazol</i>, <i>pripokonazol</i>, atau <i>validamisin A</i></p>
<p>Busuk batang (<i>stem rot</i>)</p>	<p>-adanya bercak pada daun -bercak berwarna kehitam-</p>	<p>- tunggul-tunggul padi selesai dipanen dibakar atau</p>

<p><i>-Magnaporthe sakninii</i> (<i>Cattaneo</i>) R.A. <i>Krause & R.K. Webster</i> (<i>telemorph</i>) <i>Helminthosporium sigmoideum var.irregulare</i></p>	<p>hitaman pada pelepah daun -bercak berbentuk tidak teratur secara beratahap dan membesar -anakan tanaman padi mati -terjadi kerebahan pada tanaman padi -pada batang dalam kanopi tanaman padi membusuk -terjadi kerebahan pada tanaman, tanpa terjadi hujan atau hujan dengan angin yang kencang</p>	<p>didekomposisi -keringkan petakan dan biarkan tanah sampai retak sebelum diairi lagi -gunakan pemupukan berimbang ; pupuk <i>nitrogen</i> sesuai anjuran dan pemupukan K cenderung dapat menurunkan infeksi penyakit -gunakan <i>fungisida</i> (bila diperlukan) yang berbahan aktif <i>belerang</i> atau <i>difenokonazol</i></p>
<p>Busuk pelepah (<i>sheath rot</i>) <i>Sarocladium oryzae</i> (<i>sawada</i>) <i>Gums</i> dan <i>Hawksworth</i></p>	<p>-adanya bercak pada daun -bercak berwarna keabu-abuan -bercak bercak berbentuk bulat memanjang hingga tidak teratur dengan panjang 0,5-1,5cm -Infeksi terjadi pada pelepah daun paling atas yang menutupi malai muda pada akhir fase bunting -fase penyebaran, bercak membesar dan bersambung -bercak menutupi seluruh pelepah daun -bercak yang parah berwarna coklat jingga dan kering -perkembangan malai terhambat -mengurangi jumlah gabah yang terisi (hampa)</p>	<p>Bakar tunggul segera sesudah panen untuk mengurangi inokulum -atur jarak tanam agar tidak terlalu rapat -beri pupuk K pada fase anakan -gunakan fungisida (bila diperlukan) berbahan aktif karbendazim atau mankezob pada daun di fase bunting -gunakan <i>fungisida</i> (bila diperlukan) berbahan aktif <i>benomil</i></p>
<p>Bercak coklat (<i>brown spot</i>) <i>Helminthosporium oryzae</i></p>	<p>-adanya bercak pada daun - bercak berwarna coklat -bercak berbentuk oval -bercak berbentuk oval hingga bulat -bercak berukuran sebesar biji wijen -bercak dapat menyerang batang dan gabah</p>	<p>-Penyakit dapat dikendalikan secara efektif dengan varietas tahan melalui pemupukan dengan 250kg urea, 100kg SP3 dan 100 KCl</p>
<p>Bercak cerospora (<i>narrow brown leaf spot</i>) <i>Cercospora oryzae</i></p>	<p>-adanya bercak pada daun -bercak berwarna coklat -bercak berbentuk garis lurus dan menyempit pada helaian daun -bercak dapat menyerang pada</p>	<p>-Penyakit dikendalikan dengan pemupukan berimbang yang lengkap, dengan dosis 250kg urea, 100 kg SP36 dan 100kg KCl per ha</p>

	bagian batang dan gabah	
Hawar daun jingga (<i>red stripe</i>) Penyebab belum diketahui	<ul style="list-style-type: none"> -adanya bercak pada daun -bercak berwarna jingga di helaian daun -bercak berbentuk titik kecil -bercak membentuk garis lurus ke arah ujung daun (tidak pernah ke arah pangkal daun) 	<ul style="list-style-type: none"> -Hawar daun jingga dikendalikan secara kultur teknis. Pemberian dengan pupuk dengan dosis <i>250kg urea</i>, <i>100kg SP36</i>, dan <i>100kg KCl per ha</i> dapat menekan perkembangan penyakit -Penyakit ini juga dapat ditekan dengan mengeringkan lahan dan membuka kanopi pertanaman
Tungro Virus tungro	<ul style="list-style-type: none"> -daun-daun berwarna kuning -daun menguning hingga jingga -pertumbuhan tanaman terhambat -tanaman menjadi kerdil (seperti tidak berkembang) -gejala khas ini ditentukan oleh tingkat ketahanan varietas, kondisi lingkungan, dan fase tumbuh saat tanaman terinfeksi -penyakit tungro dibawa oleh wereng hijau 	<ul style="list-style-type: none"> -penyakit tungro dapat dikendalikan melalui pergiliran varietas tahan yang memiliki tetua berbeda, pengaturan waktu tanam, sanitasi dengan menghilangkan sumber tanaman sakit dan menekan populasi wereng hijau dengan insektisida -mengatur waktu tanam serempak minimal 20ha -menanam bibit pada saat yang tepat, yaitu dengan menanam bibit sebulan sebelum puncak kepadatan wereng hijau -menanam dengan cara jajar legowo -pada saat umur 2-3minggu setelah tanam bila dijumpai 2 tanaman bergejala lebih dari 10rumpun segera aplikasikan insektisida untuk mematikan wereng hijau -jangan keringkan sawah, biarkan kondisi air pada kapasitas lapang agar wereng hijau tidak aktif menyebarkan virus tungro
Kerdil rumput (<i>grassy stunt</i>)	<ul style="list-style-type: none"> -daun berwarna hijau pucat -adanya bercak pada daun -bercak seperti karat -pertumbuhan tanaman terhambat -tanaman menjadi kerdil -jumlah anakan meningkat -tanaman tampak seperti rumput 	<ul style="list-style-type: none"> -pengendalian dilakukan terhadap vektornya yaitu wereng coklat

	<ul style="list-style-type: none"> -Daun tanaman padi menjadi sempit, pendek dan kaku -penyakit ini disebabkan oleh virus yang ditularkan oleh wereng coklat dan tanaman inangnya hanya padi 	
Kerdil Hampa (<i>ragged stunt</i>)	<ul style="list-style-type: none"> -daun berwarna hijau tua -perkembangan tanaman terhambat -tanaman menjadi kerdil - daun bergerigi -daun melintir (seperti menggulung) - malai tanaman yang sakit hanya keluar sebagian -Gabah yang dihasilkan hampa -Kerdil hampa disebabkan oleh virus yang ditularkan oleh wereng coklat 	<ul style="list-style-type: none"> -penyakit dikendalikan melalui pengendalian wereng coklat antara lain dengan penanaman varietas tahan

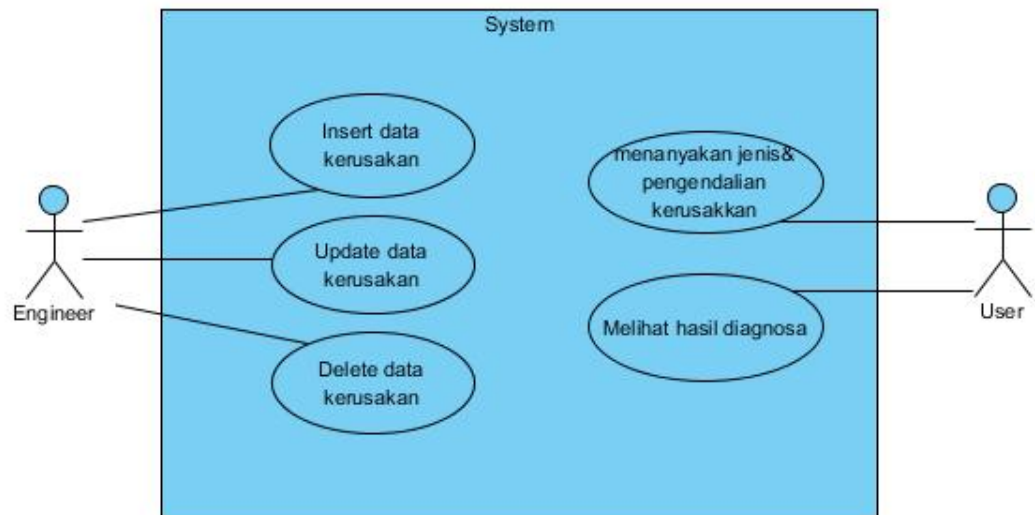
Informasi data tersebut kemudian dimasukkan ke dalam basis pengetahuan (*knowledge base*) yang nantinya akan diproses dalam bentuk bahasa pemrograman *prolog*. Pada bagian *knowledge base* ini mengandung semua fakta-fakta, baik fakta awal pada sistem mulai beroperasi maupun fakta-fakta yang didapatkan pada suatu pengambilan kesimpulan.

Sedangkan metode *interview* secara tak langsung yaitu diberikan rujukan buku berjudul "Masalah lapang Hama, Penyakit dan Hara pada padi" yang digunakan untuk referensi studi literatur.

3.6 Desain Sistem dan *Software*

Perancangan yang dilakukan yaitu perancangan desain. Pada tahap ini dilakukan pengumpulan informasi yang dibutuhkan pada sistem dan memulai perancangan konseptual dengan menggunakan *usecase* diagram.

Setelah itu pembuatan *algorithm top goal* dan kemudian *desain knowledge base* penyebab kerusakan yang diperlukan sistem nantinya. Berikut ini adalah gambar *usecase* diagram pada penelitian yang dilakukan :



Gambar 3.4 *Usecase* diagram

Dapat dilihat pada gambar 3.4 yang menunjukkan interaksi antara aktor dengan sistem yang dibangun. Pada sistem terdapat dua aktor yang berinteraksi dengan sistem yaitu *engineer* dan *user*. *Engineer* bertugas untuk memasukkan data, memperbarui dan menghapus data yang ada pada *knowledge base*, sedangkan *user* berperan sebagai pengguna yang mengoperasikan sistem untuk mendapatkan jawaban mengenai jenis dan pengendalian kerusakan dengan cara menjawab gejala-gejala yang diajukan oleh sistem.

a. Pembuatan algoritma

Pada bahasa pemrograman prolog tidak terdapat prosedur, tetapi hanya kumpulan data-data objek (fakta) yang akan diolah, dan relasi antar objek tersebut membentuk suatu jawaban programmer menentukan tujuan (*goal*), dan komputer menentukan bagaimana cara mencapai tujuan tersebut serta mencari jawabannya. Dilakukan pembuktian cocok tidaknya tujuan dengan data-data yang telah ada dan relasinya. Prolog ideal untuk memecahkan masalah yang tidak terstruktur, dan prosedur pemecahannya tidak diketahui, khususnya untuk memecahkan masalah *non-numeric*. Prolog bekerja seperti manusia, proses pemecahan masalah bergerak di dalam ruang masalah menuju suatu tujuan (jawaban tertentu).

Berikut algoritma *topgoal* pada sistem pakar menggunakan *software* amzi prolog secara umum :

```

prove(true,_) :- !.
prove((Goal,Rest),Hist) :-
    prov(Goal,[Goal|Hist]),
    prove(Rest,Hist).
prove(Goal,Hist) :-
    prov(Goal,[Goal|Hist]).

prov(true,_) :- !.
prov(ask(X),Hist) :- ask(X,Hist), !.
prov(Goal,Hist) :-
    clause(Goal,Body),
    prove(Body,Hist).

```

b. Knowledge base (Basis Pengetahuan)

Knowledge base adalah basis data yang berisi pengetahuan yang dibutuhkan untuk memahami, menformulasikan, dan memecahkan masalah. Basis pengetahuan tersusun atas dua elemen dasar :

1. Fakta

Salah satu contoh fakta yang terdapat pada sistem pakar yaitu sebagai berikut :

- a. Penyebab kerusakan dengan menunjukkan tanda-tanda kerusakan pada tanaman padi yaitu daun-daun padi menguning, daun yang terhisap hanya bagian pinggir, daun-daun mengering, daun tampak seperti terbakar, penyebaran serangan dimulai dari satu titik dan kemudian menyebar berbentuk lingkaran, perkembangan akar terganggu dan bagian bawah tanaman menjadi terlapisi oleh jamur. Gejala kerusakan tersebut merupakan fakta dari kondisi permasalahan kerusakan tanaman yang disebabkan oleh wereng cokelat.
- b. Penyebab kerusakan dengan menunjukkan tanda-tanda kerusakan pada tanaman padi yaitu adanya ngengat batang padi atau kupu-kupu, ngengat penggerek batang padi berwarna kuning kecokelatan, adanya titik hitam dibagian belakang sayap depan, adanya ulat, pangkal batang tanaman terpotong (sundep), matinya tunas-tunas padi dan matinya malai (daun padi) disebabkan oleh penggerek batang padi kuning
- c. Penyebab kerusakan dengan menunjukkan tanda-tanda kerusakan pada tanaman padi yaitu daun-daun menguning, kerusakan yang diserang mulai dari tengah petak meluas ke arah pinggir dan menyisakan 1-2 petakan di pinggir petakan, jumlah anakan menurun, adanya jejak kaki, adanya jalur jalan, adanya kotoran dan adanya lubang aktif.

2. Aturan

Aturan yang ada pada sistem pakar yaitu pertanyaan yang jawaban dari pertanyaan tersebut merupakan aturan untuk memecahkan masalah ke fakta yang spesifik dalam bidang yang khusus. Aturan / kaidah ini dituliskan dalam bentuk jika-maka (IF-THEN) yaitu bagian premise (jika) dan bagian konklusi (maka). Apabila bagian premise terpenuhi maka bagian konklusi juga akan bernilai benar. Berikut aturan dalam menganalisa gejala kerusakan tanaman padi :

Rule Masalah penyebab kerusakan wereng cokelat

IF apakah ciri-ciri yang terjadi pada tanaman padi Is Ya

AND adanya daun-daun menguning Is Ya

AND daun yang terhisap hanya bagian pinggir Is Ya

AND adanya titik hitam dibagian belakang sayap depan Is Ya

AND daun-daun mengering Is Ya

AND daun-daun tampak seperti terbakar Is Ya

AND kerusakan yang disebabkan dimulai dari satu titik, kemudian menyebar ke segala arah dalam bentuk lingkaran Is Ya

AND perkembangan akar terganggu Is Ya

AND bagian bawah tanaman menjadi terlapisi oleh jamur Is Ya

THEN gejala kerusakan disebabkan oleh wereng cokelat

Rule Masalah penyebab kerusakan Penggerek Batang Padi Kuning

IF apakah ciri-ciri yang terjadi pada tanaman padi Is Ya

AND adanya ngengat penggerek batang padi atau kupu-kupu Is Ya

AND ngengat penggerek batang padi berwarna kuning kecokelatan Is Ya

AND adanya titik hitam dibagian belakang sayap depan Is Ya

AND adanya ulat Is Ya

AND pangkal batang tanaman terpotong (sundep) Is Ya

AND matinya tunas-tunas padi Is Ya

AND matinya malai (daun padi) Is Ya

THEN gejala kerusakan disebabkan oleh penggerek batang padi kuning

Rule Masalah penyebab kerusakan Tikus

IF apakah ciri-ciri yang terjadi pada tanaman padi Is Ya

AND daun-daun menguning Is Ya

AND kerusakan yang diserang mulai dari tengah petak meluas ke arah pinggir dan menyisakan 1-2 petakan di pinggir petakan Is Ya

AND jumlah anakan menurun Is Ya

AND adanya jejak-jejak kaki Is Ya

AND adanya jalur jalan Is Ya

AND adanya kotoran Is Ya

AND adanya lubang aktif Is Ya

THEN gejala kerusakan disebabkan oleh tikus.

Berikut langkah-langkah dalam merancang *knowledge base* :

- a. Mendefinisikan *entity* yang dibutuhkan.

Entitas pada sistem yang dibangun adalah : Penyebab kerusakan pada tanaman padi.

- b. Menentukan *Attribute* setiap *Entity* berserta kata kuncinya

Tabel 3.2 Daftar *attribute* berserta kata kuncinya

<i>Entity</i>	<i>Attribute</i>
Hama dan Penyakit	adanya ngengat atau kupu-kupu
	ngengat berwarna kuning kecokelatan
	ngengat memiliki titik hitam dibagian belakang sayap depan
	adanya ulat
	pangkal batang tanaman terpotong
	matinya malai atau daun
	matinya tunas-tunas padi
*insektisida, berbahan aktif : Karbonfuran, bensultap, karbosulfan, dimenhipo, amitraz & fipronil. Kecuali, kupu-kupu, jangan memakai pestisida semprot untuk sundep dan beluk yang berterbangan.	hama penggerek batang padi kuning

c. Desain *Knowledge base*

Desain harus ditranslasikan ke dalam program perangkat lunak. Hasil dari tahap ini adalah pemrograman sesuai dengan perancangan yang telah dibuat pada tahap desain. Pada tahapan ini setelah mendapatkan semua kebutuhan *Requirement Analysis* yaitu dengan metode wawancara atau *interview* pada ahli/pakar pertanian mengenai data informasi mengenai gejala dan penyebab kerusakan hama dan penyakit tanaman padi yang telah dipaparkan pada tahapan analisa kebutuhan dan definisi. Pada pembuatan desain *knowledge base* terbagi menjadi dua yaitu *consult.pro* dan *database.kb*.

Berikut ini adalah gambaran umum contoh kode program yang ada pada *knowledge base*:

```
top_goal(X) :- kerusakan(X).
```

```

kerusakan('nama penyebab kerusakan') :-
    ask('gejala kerusakan 1'),
    ask('gejala kerusakan 2'),
    ask('gejala kerusakan 3'),
    ask('gejala kerusakan 4'),
    ask('gejala kerusakan 5'),
    ask('gejala kerusakan 6'),
    ask('gejala kerusakan 7').

penanggulangan('nama penyebab kerusakan') :-
    write('*solusi 1
          *solusi 2
          *solusi3').

```

Gambar 3.5 Kode program pada *Knowledge base*

Berikut ini adalah contoh kode program implementasi dari hama dan penyakit tanaman padi yang ada pada *knowledge base*:

```

top_goal(X) :- kerusakan(X).

kerusakan('hama penggerek batang padi kuning') :-
    ask('adanya ngengat seperti kupu-kupu'),
    ask('ngengat berwarna kuning kecokelatan'),
    ask('ngengat memiliki titik hitam dibagian belakang
        sayap depan'),
    ask('adanya ulat'),
    ask('pangkal batang tanaman terpotong'),
    ask('matinya malai atau daun'),
    ask('matinya tunas-tunas padi').

penanggulangan('hama penggerek batang padi kuning') :-
    write('*insektisida, berbahan aktif :
          *Karbonfuran, bensultap, karbosulfan, dimenhipo,
          amitraz & fipronil. Kecuali, kupu-kupu, jangan
          memakai pestisida semprot untuk sundep dan beluk
          yang berterbangan').

```

3.7 Implementasi

Tahap selanjutnya yaitu dirangkum dalam bentuk tabel data informasi yang selanjutnya data tersebut diubah (*konvers*) ke dalam bahasa program *prolog* dengan menggunakan aplikasi *Amzi Prolog*.

Pada *prolog* terdapat dua *file* yang dibuat, yaitu kode program yang berfungsi sebagai *inference engine* dan informasi tentang basis data kerusakan atau disebut juga *knowledge base*.

Berikut ini adalah bahasa pemrograman dari file *inference engine* dan *knowledge base* :

➤ Bahasa pemrograman *inference engine*

```
:-op(900,xfy, :).
:-initialization(consult).

consult :-
    opening,
    consult('database.kb'),
    repeat,
    write('> '),
    read(X),
    do(X),
    X == quit.

opening :-
    write('Program Diagnosa Hama& Penyakit Tanaman Padi'),
    nl,
    write('BY Dian Nindariansari'), nl,
    help_me.

do(diag) :- diagnosa, !.
do(mulai_baru) :- mulai_baru, !.
do(quit).
do(X) :-
    write(X),
    write(' Kata perintah tidak di mengerti.'), nl,
    help_me,
    fail.

help_me :-
    write('Ketik "diag." untuk memulai diagnosa penyakit
dan "quit." untuk keluar.'),nl,
    write('Ingat!! Setiap pernyataan harus diakhiri dengan
titik.'), nl, nl.

mulai_baru :-
    retractall(known(_,_)),          %hapus semua data
    jawaban (history) di variabel known
    help_me,
    diagnosa.                        %masuk ke
    pertanyaan

diagnosa :-
    retractall(known,2),            %hapus yang tidak
    berguna
```

```

        prove(top_goal(X),[]),          %masuk ke sesi
pertanyaan
        write('Penyebab Kerusakan '), write(X),nl,
        %kesimpulan jika pertanyaan menghasilkan jawaban
        write('Penanggulangan :'), penanggulangan(X), nl,
        %munculkan jawaban
        write('mulai diagnosa baru?'),
        repeat,
        read(M),
        proses_m(M),
        M==t.

diagnosa :-                               %kesimpulan jika
pertanyaan tidak menghasilkan jawaban
        write('Penyebab kerusakan Tidak di Temukan. '),nl,
        write('mulai diagnosa baru?'),
        repeat,
        read(M),
        proses_m(M),
        M==t.

proses_m(y) :-
        mulai_baru.

proses_m(t).

ask(Attribute,_) :-
        known(y,Attribute),
        !.
ask(Attribute,_) :-
        known(_,Attribute),
        !, fail.

ask(A,Hist) :-
        %fungsi untuk menampilkan menu pertanyaan
        write('apakah ciri-ciri yang terjadi pada tanaman padi
'),write(A),
        write('? (y. atau t.) '),
        get_user(Y,Hist),          % menunggu jawaban
user
        asserta(known(Y,A)),      % simpan jawababn
user di variabel known
        Y = y.

get_user(X,Hist) :-                               %fungsi
untuk menunggu input jawaban user
        repeat,
        write('> '),
        read(X),
        process_ans(X,Hist), !.

process_ans(X,_).

%FUNGSI BERIKUT INI BERFUNGSI UNTUK MEMILIH TOP GOAL.
prove(true,_) :- !.
prove((Goal,Rest),Hist) :-
        prov(Goal,[Goal|Hist]),
        prove(Rest,Hist).
prove(Goal,Hist) :-

```

```

prov(Goal,[Goal|Hist]).

prov(true,_) :- !.
prov(ask(X),Hist) :- ask(X,Hist), !.
prov(Goal,Hist) :-
    clause(Goal,Body),
    prove(Body,Hist).

```

➤ Bahasa pemrograman *knowledge base*

```

top_goal(X) :- kerusakan(X).

kerusakan('hama penggerek batang padi kuning') :-
    ask('adanya ngengat seperti kupu-kupu'),
    ask('ngengat berwarna kuning kecokelatan'),
    ask('ngengat memiliki titik hitam dibagian belakang
sayap depan'),
    ask('adanya ulat'),
    ask('pangkal batang tanaman terpotong'),
    ask('matinya malai atau daun'),
    ask('matinya tunas-tunas padi').

kerusakan('hama penggerek batang padi putih') :-
    ask('adanya ngengat seperti kupu-kupu'),
    ask('ngengat berwarna putih'),
    ask('adanya ulat'),
    ask('pangkal batang tanaman terpotong'),
    ask('matinya malai atau daun'),
    ask('matinya tunas-tunas padi').

kerusakan('hama penggerek batang padi bergaris') :-
    ask('adanya ngengat seperti kupu-kupu'),
    ask('ngengat berwarna kecokelatan'),
    ask('adanya ulat'),
    ask('pada ulat terdapat dua garis memanjang'),
    ask('pangkal batang tanaman terpotong'),
    ask('matinya malai atau daun'),
    ask('matinya tunas-tunas padi').

penanggulangan('hama penggerek batang padi kuning') :-
    write('*insektisida, berbahan aktif :
        *Karbonfuran, bensultap, karbosulfan, dimenhipo,
        amitraz & fipronil. Kecuali, kupu-kupu, jangan
        memakai pestisida semprot untuk sundep dan beluk
        yang berterbangan').

penanggulangan('hama penggerek batang padi putih') :-
    write('*insektisida, berbahan aktif :
        *Karbonfuran, bensultap, karbosulfan, dimenhipo,
        amitraz & fipronil. Kecuali, kupu-kupu, jangan
        memakai

```

```
pestisida semprot untuk sundep dan beluk yang
berterbangan').
```

```
penanggulangan('hama penggerek batang padi bergaris') :-
  write('*insektisida, berbahan aktif :
  *Karbonfuran, bensultap, karbosulfan, dimenhipo,
  amitraz & fipronil. Kecuali, kupu-kupu, jangan
  memakai pestisida semprot untuk sundep dan beluk
  yang berterbangan').
```

3.8 Pengujian (*testing*)

Setelah tahapan implementasi berhasil dilakukan, maka tahap selanjutnya yaitu pengujian sistem pakar diagnosa hama dan penyakit tanaman padi. Pada tahap ini pengujian mengutamakan ketepatan dari segi logik dan fungsional dengan tujuan untuk memastikan bahwa semua bagian sudah berjalan dengan baik. Selain itu tujuan dari pengujian itu sendiri yaitu untuk meminimalisir kesalahan (*error*) dan memastikan ketepatan dan kebenaran dari *output* yang dihasilkan sesuai dengan yang diharapkan.

Tahap pengujian dilakukan sebanyak dua kali dengan tujuan untuk mengetahui apakah sistem pakar yang dibangun berjalan dengan baik atau tidak. Pengujian keberhasilan sistem pakar dilakukan oleh engineer atau peneliti yang membangun sistem pakar tersebut. Selanjutnya dilakukan pengujian ketepatan mengenai *output* data informasi sistem pakar yang dibangun dimana tahap pengujian ini akan dilakukan dengan seorang pakar dan seorang petani.

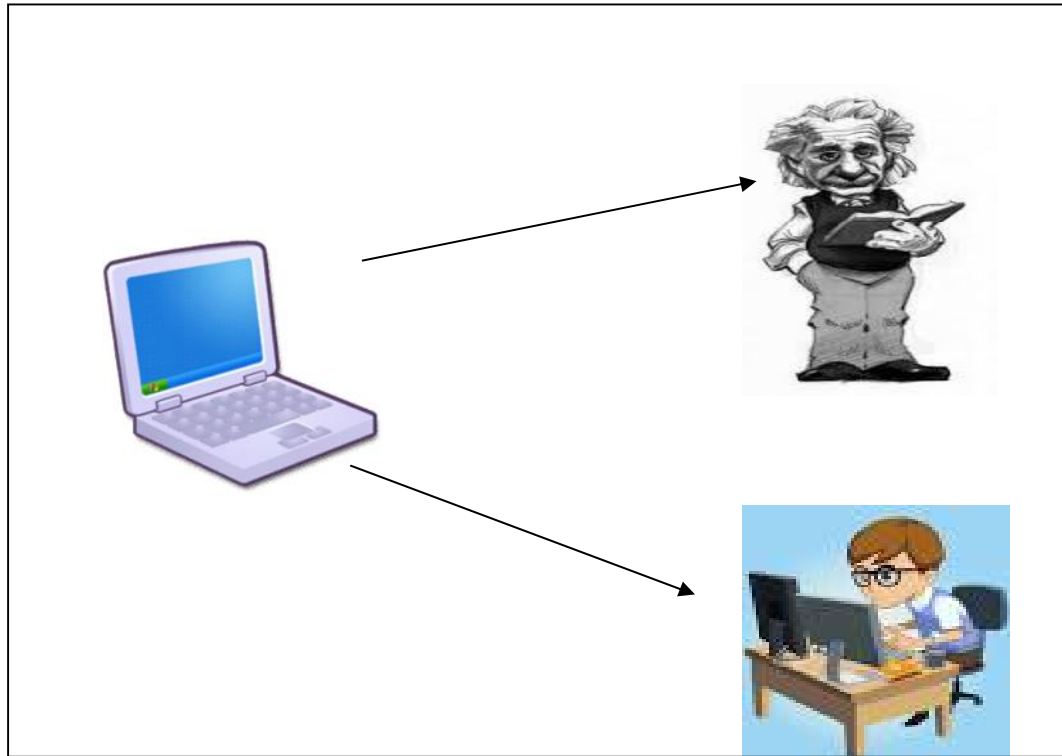
3.9 Pemeliharaan (*maintenance*)

Tahap terakhir dari *model waterfall* yaitu pemeliharaan. Tahap pemeliharaan itu sendiri perlu dilakukan mengingat tidak menutup kemungkinan sebuah perangkat lunak mengalami perubahan ketika sudah dikirimkan ke *user*.

Perubahan ini dapat terjadi karena adanya kesalahan yang muncul dan tidak terdeteksi saat pengujian atau perangkat lunak harus beradaptasi dengan lingkungan yang baru. Tahap pendukung atau pemeliharaan dapat mengulangi proses pengembangan mulai dari analisis spesifikasi untuk perubahan perangkat lunak yang sudah ada, tapi tidak untuk membuat perangkat lunak baru. Dari *maintenance* ini terdapat masukan *user* bahwa perangkat lunak diagnosa hama dan penyakit tanaman padi berbasis sistem pakar ini belum dilakukan mengingat pengaplikasian sistem pakar ini masih mengutamakan ketepatan kerja dan data sistem pakar. Dan, dikemudian hari perlu adanya peningkatan bahasa program berbasis teks ke dalam bentuk bahasa program berbasis GUI (*Graphical User Interface*) agar lebih mudah dan menarik dalam menggunakan program tersebut.

3.10 Gambaran Umum Penelitian

Gambaran umum penelitian mengenai sistem pakar ini dapat digambarkan sebagai berikut:



Gambar 3.6 Gambaran umum sistem

Berdasarkan gambar 3.6 dapat dilihat bahwa PC (*Personal Computer*) yang digunakan untuk menjalankan sistem ini dapat diakses di berbagai tempat, termasuk *notebook* yang bersifat *mobile* dan praktis digunakan setiap orang. *Notebook* yang digunakan telah terpasang perangkat lunak *Amzi Prolog* agar dapat menjalankan simulasi program sebelum dibuat menjadi perangkat lunak pendiagnosa penyakit tanaman padi berbasis sistem pakar.

BAB V

KESIMPULAN DAN SARAN

5.1.1 KESIMPULAN

1. Telah direalisasikan perancangan sistem pakar untuk mendiagnosa gejala-gejala kerusakan tanaman padi akibat serangan hama dan penyakit dengan menggunakan metode wawancara (*interview*) dan studi literatur.
2. Telah dibuktikan bahwa sistem pakar yang dibangun merupakan bagian dari ilmu kecerdasan buatan (*Artificial Intelligence*).
3. Sistem yang dibangun dapat dijadikan sebagai bahan rujukan bagi para petani untuk pengambilan keputusan serta menghasilkan kesimpulan dan memudahkan pengguna untuk mengetahui penyebab, akibat dan gejala-gejala kerusakan yang ditimbulkan pada tanaman padi.
4. Sistem pakar ini telah melalui tahap pengujian ketepatan *knowledge base* oleh dua orang ahli/pakar pertanian dan seorang petani yaitu Bapak Ir. Iwan Gunawan, M.P., Ibu Ir. Raida Kartina, M.P. dengan hasil persentasi keberhasilan sebesar 90% dan 10% kegagalan.

5. Sistem pakar ini telah melalui tahap pengujian kelayakkan sistem oleh dua orang ahli /pakar pertanian dan seorang petani yaitu Bapak Ir. Iwan Gunawan, M.P., Ibu Ir. Raida Kartina, M.P. dengan menghasilkan tingkat kepuasan 80%.
6. Sistem pakar ini telah melalui tahap pengujian kelayakan sistem oleh seorang petani padi yaitu Bapak Deroni dan menghasilkan tingkat kepuasan 80%.

5.1.1 SARAN

Diharapkan untuk pengembangan selanjutnya dapat membuat aplikasi yang lebih mempermudah *user* dalam penggunaannya, yaitu dengan pengembangan bahasa program berbasis teks menjadi berbasis GUI (*Graphical User Interface*) bahkan berbasis android agar dapat lebih mudah diakses dan lebih praktis.

DAFTAR PUSTAKA

- [1] Kusuma dewi, Sri. *Artificial Intellingence*, Yogyakarta: Graha Ilmu, 2003.
- [2] Ladjamudin B, Al-bahra. *Rekayasa Perangkat Lunak*. Yogyakarta: Graha Ilmu, 2006.
- [3] Suyamto. *Masalah Lapang Hama, Penyakit dan hara pada padi*. Balai Pengkajian Teknologi Pertanian Lampung. Bandar Lampung.
- [4] Siswanto. *Kecerdasan Tiruan Edisi I*, Yogyakarta: Graha Ilmu, 2005.
- [5] Siswanto. *Kecerdasan Tiruan Edisi II*, Yogyakarta: Graha Ilmu, 2005.
- [6] Kristanto, Andri. *Kecerdasan Buatan*, Yogyakarta: Graha Ilmu, 2004.
- [7] Merlina, Nita dan Hidayat, Rahmat. *Perancangan Sistem Pakar*, Bogor: Ghalia Indonesia, 2012.
- [8] Ungkawa, Uung. *Bahasa Pemrograman Logika Turbo Prolog*, Yogyakarta: Andi offset, 1992.
- [9] Amble, T. *Logic Programming And Knowledge Engineerin*, United states: Adison Wesley Publishing Co, 1987.
- [10] Simon, G. *Introducing Artificial Intellingence*, NCC Publication, 1984.
- [11] Roni Yanuar Nainggolan. *Perangkat Lunak Diagnosa Kerusakan Sepeda Motor Berbasis Sistem Pakar*, pp. 21-27, 2016.
- [12] Adam Hussein. *Perangkat Lunak Diagnosa Penyakit Pada Manusia*, pp. 19-25, 2015.

[13] Sudarmo, Subiyakto. Pengendalian Serangga Hama Penyakit Dan Gulma Padi, Yogyakarta: Kanisius, 1993.

[14] Harahap, Idham Sakti. Pengendalian Hama Penyakit Padi, Jakarta: Swadaya, 2004.