

**SISTEM PAKAR DIAGNOSA PENYAKIT PADI DENGAN
METODE *CERTAINTY FACTOR***

Tugas Akhir

diajukan untuk memenuhi sebagian persyaratan mencapai
gelar Ahli Madya jenjang Diploma III
Jurusan Teknik Informatika



Oleh :

IRIN WINDIYATI

NIM. 1603073

**JURUSAN TEKNIK INFORMATIKA
POLITEKNIK NEGERI INDRAMAYU**

2019

HALAMAN PENGESAHAN

Tugas Akhir ini diajukan oleh :

Nama : Irin Windiyati
NIM : 1603073
Program Studi : Teknik Infotmatika
Judul : Sistem Pakar Diagnosa Penyakit Padi dengan Metode
Certainty Factor
Pembimbing : 1. Iryanto, S.Si.,M.Si
2. Esti Mulyani,S.Kom.,M.Kom

Telah berhasil dipertahankan dihadapan dewan penguji pada tanggal 16 Agustus 2019 dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Ahli Madya Program Studi Teknik Informatika, Jurusan Teknik Informatika, Politeknik Negeri Indramayu.

DEWAN PENGUJI

Nama	Jabatan	Tandatangan	Tanggal
1. Munengsih Sari Bunga, S.Kom., M.Eng	Ketua Penguji
2. Adi Suheryadi, S.ST.,M.Kom	Sekretaris Penguji
3. Iryanto, S.Si.,M.Si	Anggota

Indramayu, September 2019
Ketua Jurusan Teknik Informatika

Iryanto, S.Si.,M.Si

NIP. 199008012019031014

MOTTO

“Niscaya Allah akan mengangkat (derajat) orang-orang yang beriman
diantaramu dan orang yang diberi ilmu beberapa derajat”
(Q.S Al-mujaddalah : 11)

PERNYATAAN KEASLIAN

Saya menyatakan dengan sesungguhnya bahwa Tugas Akhir ini adalah asli hasil karya saya sendiri, dan sepanjang pengetahuan saya tidak terdapat karya atau pendapat yang pernah ditulis atau dipublikasikan oleh orang lain, kecuali yang secara tertulis disebutkan sumbernya dalam naskah dan dalam daftar pustaka.

Indramayu, 16 Agustus 2019
Yang Menyatakan,

IRIN WINDIYATI

NIM.1603073

ABSTRAK

Padi digunakan sebagai bahan pangan pokok sebagian orang terutama di Indonesia, dan terus meningkatnya kebutuhan jumlah padi membuat budidaya padi sangat diperlukan. Salah satu masalah yang dihadapi petani secara umum yaitu mengatasi serangan penyakit padi. Kenyataannya, saat ini banyak petani Indonesia yang membutuhkan bantuan para ahli untuk mengatasi masalah pertanian mereka, tetapi jumlah ahli dan penyebarannya terbatas menyebabkan permasalahan ini belum dapat diatasi dengan maksimal. Pada penelitian ini penulis membuat sistem pakar untuk mendiagnosa penyakit padi yang dapat memberikan solusi penanganan dari penyakit tersebut guna mengurangi atau memperkecil resiko kerusakan tanaman padi. Dengan dibuatnya sistem ini dapat membantu masyarakat luas khususnya para petani agar mengetahui cara membasmi penyakit pada padi tersebut sehingga tidak mengalami gagal panen. Dari semua hasil skema pengujian baik *test rule* maupun *test* gabungan sistem pakar diagnosa penyakit padi ini berjalan baik dengan akurasi tinggi. Berdasarkan hasil pengujian *black-box* fungsionalitas sistem pakar diagnosa penyakit padi dengan metode *Certainty Factor* ini berjalan baik, dan telah sesuai dengan daftar kebutuhan sistem, sedangkan berdasarkan pengujian kuisisioner yang dilakukan pada 20 orang penguji hasilnya adalah 84,9%, dengan persentase tersebut dapat disimpulkan bahwa sistem yang dibuat sudah baik dan dapat dengan mudah digunakan oleh *user*.

Kata kunci : sistem pakar, penyakit padi, *certainty factor*.

ABSTRACT

Rice is used as a staple food for some people, especially in Indonesia, and the increasing need for rice makes rice cultivation indispensable. One of the problems faced by farmers in general is overcoming attacks by rice disease. In fact, currently there are many Indonesian farmers who need the help of experts to overcome their agricultural problems, but the number of experts and their spread is limited causing this problem has not been resolved to the maximum. In this study the authors make an expert system to diagnose rice disease that can provide solutions to the handling of the disease in order to reduce or reduce the risk of damage to rice plants. With the creation of this system can help the wider community especially farmers to know how to eradicate the disease in rice so that it does not experience crop failure. Of all the results of the test scheme both the test rule and the combined test expert system of diagnosing rice disease is running well with high accuracy. Based on the results of the black-box testing of the functionality of the expert system of diagnosing rice disease with the Certainty Factor method is going well, and in accordance with the list of system requirements, and based on questionnaire testing conducted by 20 testers the results are 84.9%, with this percentage can be concluded that the system is good for the user.

Keywords: expert system, rice disease, certainty factor.

KATA PENGANTAR

Alhamdulillah Rabbil 'Alamin, puji syukur saya panjatkan kepada Allah *Subhanahu wa Ta'ala* atas rahmat dan karunia-Nya sehingga saya dapat menyelesaikan Tugas Akhir ini dengan tepat waktu.

Tugas akhir ini merupakan salah satu persyaratan memperoleh gelar Ahli Madya pada Jurusan Teknik Informatika Politeknik Negeri Indramayu. Tugas Akhir ini berjudul ‘Sistem Pakar Diagnosa Penyakit Padi dengan Metode *Certainty Factor*’.

Tugas Akhir yang dibuat oleh penulis bertujuan untuk membantu masyarakat luas khususnya para petani agar tahu cara membasmi penyakit padi sehingga tidak mengalami gagal panen tanpa harus menemui pakar/ahli/insinyur pertanian. Alur dari aplikasi yang telah dibuat penulis yakni dengan mendiagnosa gejala penyakit padi yang dialami kemudian dapat disimpulkan penyakitnya sehingga dapat diketahui pengendalian yang harus dilakukan agar tidak memperparah penyakit yang dialami melalui *platform* web.

Dalam penyusunan laporan Tugas Akhir ini penulis telah banyak mendapatkan bantuan dari berbagai pihak, oleh sebab itu penulis ingin mengungkapkan rasa terima kasih kepada :

1. Allah *Subhanahu wa Ta'ala*.
2. Kedua orang tua penulis (Ibu Sri Hayati & Bapak Ibrohim (bram)) yang telah memberikan doa dan semangat kepada penulis agar dapat menyelesaikan Tugas Akhir dengan lancar dan tepat waktu .
3. Bapak Casiman Sukardi, S.T., M.T selaku Direktur Politeknik Negeri Indramayu.
4. Bapak Iryanto,S.Si.,M.Si.,M.Sc selaku ketua jurusan Teknik Informatika Politeknik Negeri Indramayu sekaligus selaku Dosen Pembimbing I .
5. Ibu Esti Mulyani, S.Kom.,M.Kom selaku Dosen Pembimbing II.
6. Bapak Rifki Kadarachman.H, S.P selaku staff pelaksana di Badan Perlindungan Tanaman Pangan dan Hortikultura (BPTPH) Jawa Barat sub wilayah III Indramayu.

7. Tak lupa pula penulis mengucapkan banyak terima kasih kepada teman-teman sekalian yang tidak bisa disebutkan satu persatu yang telah membantu untuk menyelesaikan TA ini.

Penulis telah berusaha semaksimal mungkin untuk menyelesaikan Tugas Akhir ini dengan baik dan benar, tetapi penulis hanyalah manusia biasa tempatnya salah dan lupa. Penulis menyadari masih terdapat kekurangan di dalam penulisan laporan TA ini, baik dari analisa, metode, dan pembahasan laporan Tugas Akhir ini dari segi tanda baca, tata bahasa maupun isi. Sehingga penulis secara terbuka menerima segala kritik dan saran positif dari pembaca.

Demikian apa yang dapat penulis sampaikan. Semoga laporan Tugas Akhir ini dapat bermanfaat bagi kita semua.

Indramayu, 15 Agustus 2019

Penulis

DAFTAR ISI

HALAMAN PENGESAHAN.....	ii
MOTTO	iii
PERNYATAAN KEASLIAN.....	iv
ABSTRAK	v
<i>ABSTRACT</i>	vi
KATA PENGANTAR	vii
DAFTAR ISI.....	ix
DAFTAR TABEL.....	xii
DAFTAR GAMBAR	xii
DAFTAR LAMPIRAN.....	xv
BAB I PENDAHULUAN	1
1.1 Latar Belakang Masalah	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah.....	2
1.4 Tujuan Penelitian.....	3
1.5 Manfaat Penelitian.....	3
1.6 Sistematika Penulisan Laporan	3
BAB II LANDASAN TEORI	5
2.1 Penyakit Padi	5
2.2 Kecerdasan Buatan	6
2.1 Sistem Pakar	6
2.1.1 Kelebihan dan Kekurangan Sistem Pakar	6
2.1.2 Ciri-ciri Sistem Pakar	7
2.1.4 Perbandingan Pakar dan Sistem Pakar	8
2.1.5 Komponen Sistem Pakar	8
2.2 Metode <i>Certainty Factor</i>	10
2.3 <i>Unified Modeling Language (UML)</i>	11
2.3.1 Diagram <i>Use-Case (Usecase Diagram)</i>	11
2.3.2 Diagram Kelas (<i>Class Diagram</i>)	13

2.3.3 Diagram Aktivitas (<i>Activity Diagram</i>).....	15
2.4 Teknik Pengujian Perangkat Lunak.....	16
BAB III METODOLOGI PENELITIAN.....	18
3.1 Metode Penelitian	18
3.2 Pengumpulan Data.....	19
3.2.1 Studi Literatur	19
3.2.2 Wawancara.....	19
3.2.3 Analisis Data Penyakit	20
3.3 Analisis Kebutuhan Sistem.....	23
3.3.1 Kebutuhan <i>Hardware</i>	23
3.3.2 Kebutuhan <i>Software</i>	23
3.4 Perancangan Sistem.....	24
3.4.1 <i>Use Case Diagram</i>	24
3.4.2 <i>Activity Diagram</i>	26
3.4.3 Relasi Database	30
3.5 <i>Flowchart</i>	30
3.5.1 <i>Flowchart</i> Metode <i>Certainty Factor</i>	30
3.5.2 <i>Flowchart</i> Admin.....	32
3.5.3 <i>Flowchart</i> User	33
3.5.4 <i>Flowchart</i> Pakar	34
3.6 Perancangan <i>ERD (Entity Relationship Diagram)</i>	34
3.7 Perancangan <i>Database</i>	36
3.7.1 Tabel <i>tb_penyakit</i>	36
3.7.2 Tabel <i>tb_gejala</i>	36
3.7.3 Tabel <i>tb_data_rule</i>	37
3.7.4 Tabel <i>tb_detail_rule</i>	37
3.7.5 Tabel <i>tb_hasildiagnosa</i>	38
3.7.6 Tabel <i>tb_detail_diagnosa</i>	38
3.7.7 Tabel <i>tb_user</i>	39
3.7.8 Tabel <i>tb_level</i>	39
3.8 Perancangan Desain Antarmuka.....	40
3.8.1 Antarmuka <i>Landing Page</i>	40

3.8.2 Antarmuka Admin	42
3.8.3 Antarmuka <i>User</i>	46
3.8.4 Antarmuka Pakar	48
3.9 Perancangan Sistem Pakar	49
BAB IV HASIL DAN PEMBAHASAN	52
4.1 Hasil	52
4.2 Pembahasan	52
4.2.1 Struktur Direktori	52
4.2.2 Implementasi <i>Database</i>	55
4.2.3 Implementasi Desain Antarmuka (<i>Interface</i>)	58
4.2.4 Pengujian Sistem	68
BAB V PENUTUP	79
5.1 Kesimpulan	79
5.2 Saran	79
DAFTAR PUSTAKA	80

DAFTAR TABEL

Tabel 2.1 Perbandingan Sistem Pakar.....	8
Tabel 2.2 Simbol <i>Usecase Diagram</i>	11
Tabel 2.3 Simbol <i>Usecase Diagram</i> (Lanjutan)	12
Tabel 2.4 Simbol <i>Usecase Diagram</i> (Lanjutan)	13
Tabel 2.5 Simbol Class Diagram	14
Tabel 2.6 Simbol Class Diagram (Lanjutan).....	15
Tabel 2.7 Simbol Activity Diagram	16
Tabel 3.1 Tabel Analisis Data Penyakit.....	20
Tabel 3.2 Tabel Analisis Data Penyakit (Lanjutan)	21
Tabel 3.3 Tabel Analisis Data Penyakit (Lanjutan)	22
Tabel 3.4 Tabel Kebutuhan Hardware	23
Tabel 3.5 Tabel Kebutuhan <i>Software</i>	23
Tabel 3.6 Penjelasan Use Case Diagram.....	25
Tabel 3.7 Penjelasan Use Case Diagram (Lanjutan).....	26
Tabel 3.8 Penjelasan <i>ERD</i>	35
Tabel 3.9 Tabel tb_penyakit.....	36
Tabel 3.10 Tabel Tb_gejala	36
Tabel 3.11 Tabel Tb_rule.....	37
Tabel 3.12 Tabel Tb_detail_rule	37
Tabel 3.13 Tabel Tb_hasildiagnosa	38
Tabel 3.14 Tabel Tb_detail_diagnosa	38
Tabel 3.15 Tabel Tb_detail_diagnosa (Lanjutan)	39
Tabel 3.16 Tabel Tb_user	39
Tabel 3.17 Tabel Tb_level	39
Tabel 3.18 Tabel Nilai Bobot User	49
Tabel 4.1 Validasi Hasil.....	68
Tabel 4.2 <i>Black-Box Testing</i>	69
Tabel 4.3 Tabel Pertanyaan Kuisisioner	76
Tabel 4.4 Tabel Jawaban Kuisisioner	77

DAFTAR GAMBAR

Gambar 1.1 Grafik Tahan Pangan dan Gizi (Arthalia dkk, 2016)	1
Gambar 2.1 Komponen Sistem Pakar (Rachmawati dkk, 2012)	9
Gambar 3.1 Alur Metode Penelitian	18
Gambar 3.2 Use Case Diagram	24
Gambar 3.3 Diagram Aktivitas Admin	27
Gambar 3.4 Diagram Aktivitas User	28
Gambar 3.5 Diagram Aktivitas pakar	29
Gambar 3.6 Relasi Database	30
Gambar 3.7 Flowchart Metode Certainty Factor	31
Gambar 3.8 Flowchart Admin	32
Gambar 3.9 Flowchart User	33
Gambar 3.10 Flowchart Pakar	34
Gambar 3.11 Perancangan ERD	35
Gambar 3.12 Antarmuka Landing Page	40
Gambar 3.13 Antarmuka Menu Petunjuk	41
Gambar 3.14 Antarmuka Register	41
Gambar 3.15 Antarmuka Login	42
Gambar 3.16 Antarmuka Dashboard	43
Gambar 3.17 Antarmuka Data Penyakit	43
Gambar 3.18 Antarmuka Data Gejala	44
Gambar 3.19 Antarmuka Data Rule	44
Gambar 3.20 Antarmuka Tambah Bobot Rule	45
Gambar 3.21 Antarmuka Edit Bobot Rule	45
Gambar 3.22 Antarmuka Riwayat Diagnosa	46
Gambar 3.23 Antarmuka User	46
Gambar 3.24 Antarmuka Diagnosa	47
Gambar 3.25 Antarmuka Riwayat Diagnosa	47
Gambar 3.26 Antarmuka Data Gejala	48
Gambar 3.27 Antarmuka Data Gejala	48

Gambar 4.1 Struktur Direktori Folder Ta-pakar	52
Gambar 4.2 Struktur Direktori Folder Views	53
Gambar 4.3 Struktur Direktori Folder Controller	53
Gambar 4.4 Struktur Direktori Folder Models	54
Gambar 4.5 Struktur Direktori Folder Assets	54
Gambar 4.6 Database Pakar	55
Gambar 4.7 Tabel Tb_penyakit.....	56
Gambar 4.8 Tabel Tb_gejala.....	56
Gambar 4.9 Tabel Tb_data_rule	56
Gambar 4.10 Tabel Tb_detail_rule	57
Gambar 4.11 Tabel Tb_hasildiagnosa.....	57
Gambar 4.12 Tabel Tb_detail_diagnosa	57
Gambar 4.13 Tabel Tb_user	58
Gambar 4.14 Tabel Tb_level	58
Gambar 4.15 Antarmuka Landing Page.....	59
Gambar 4.16 Antarmuka Register.....	59
Gambar 4.17 Antarmuka Login	60
Gambar 4.18 Antarmuka Dashboard.....	60
Gambar 4.19 Antarmuka Data Penyakit	61
Gambar 4.20 Antarmuka Data Gejala	61
Gambar 4.21 Antarmuka Data Rule.....	62
Gambar 4.22 Antarmuka Tambah Bobot	62
Gambar 4.23 Antarmuka Edit Bobot	63
Gambar 4.24 Antarmuka Data User.....	63
Gambar 4.25 Antarmuka Riwayat Diagnosa	64
Gambar 4.26 Antarmuka Home	64
Gambar 4.27 Antarmuka Diagnosa.....	65
Gambar 4.28 Antarmuka Hasil Diagnosa	65
Gambar 4.29 Antarmuka Riwayat Diagnosa	66
Gambar 4.30 Antarmuka Cetak Hasil Diagnosa	66
Gambar 4.31 Antarmuka Validasi Data Pakar	67
Gambar 4.32 Antarmuka Data Pakar	67

DAFTAR LAMPIRAN

Lampiran 1 Source Code Program

Lampiran 2 Kuisisioner

Lampiran 3 Validasi Data Pakar

Lampiran 4 Validasi Hasil Pengujian Perhitungan Manual dengan Sistem

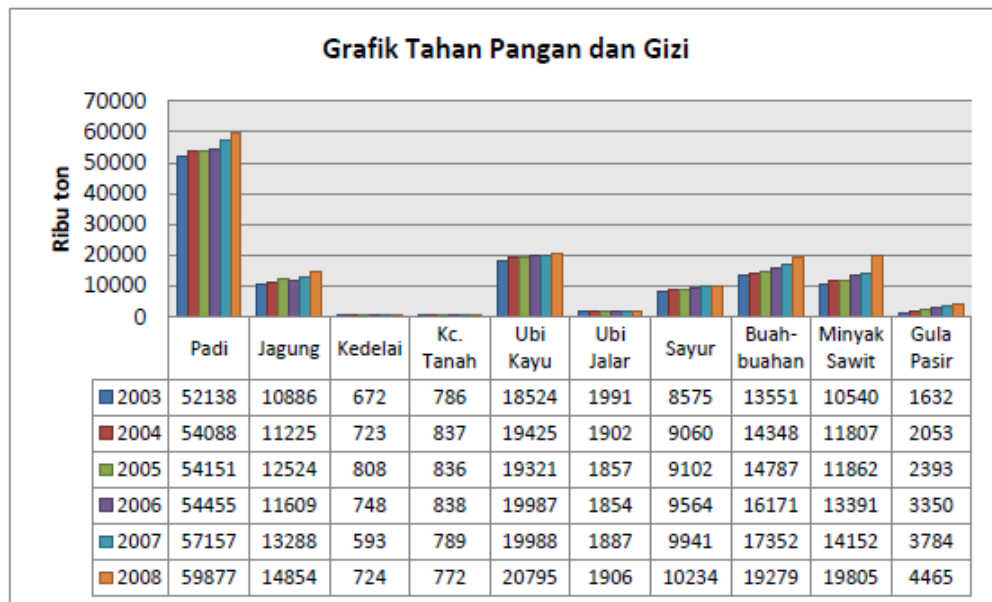
Lampiran 5 Biodata Penulis

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Padi digunakan sebagai bahan pangan pokok sebagian orang terutama di Indonesia, dan terus meningkatnya kebutuhan jumlah padi membuat budidaya padi sangat diperlukan. Berdasarkan Gambar 1.1 produksi bahan pangan yang terdiri dari: padi, jagung dan ubi kayu meningkat selama 2003 sampai dengan 2008 dan tanaman padi merupakan tanaman yang paling produktif di antara tumbuhan-tumbuhan sereal lain. Dikarenakan tingginya tingkat kebutuhan penduduk Indonesia akan padi, maka petani perlu dukungan yang maksimal untuk dapat menghasilkan padi yang berkualitas baik dengan kuantitas panen yang maksimal pula. Pada prosesnya budidaya padi tidak semuanya dapat dipanen ada pula padi yang gagal dipanen akibat terserang penyakit, akibatnya para petani dirugikan karena hal tersebut. Tanaman padi yang terlambat untuk didiagnosis dan telah mencapai tahap yang parah menjadi penyebab terjadinya gagal panen. Salah satu masalah yang dihadapi petani secara umum yaitu masalah dalam mengatasi serangan penyakit terhadap tanaman padi mereka. (Arthalia dkk, 2016)



Gambar 1.1 Grafik Tahan Pangan dan Gizi (Arthalia dkk, 2016)

Jika petani tersebut memiliki pengetahuan lebih mengenai serangan penyakit, maka serangan tersebut akan langsung dapat diatasi. Sebaliknya jika petani kurang memiliki pengetahuan mengenai serangan penyakit, maka petani tersebut cenderung membutuhkan bantuan orang yang lebih ahli untuk mengatasi masalah ini. Pada kenyataannya, saat ini banyak petani Indonesia yang membutuhkan bantuan para ahli untuk mengatasi masalah pertanian mereka, tetapi jumlah ahli dan penyebarannya terbatas menyebabkan permasalahan ini belum dapat diatasi dengan maksimal. (Arthalia dkk, 2016)

Ahli pertanian dalam hal ini mempunyai kemampuan untuk menganalisa gejala-gejala penyakit tanaman tersebut, tetapi untuk mengatasi semua persoalan yang dihadapi petani terkendala oleh waktu.

Pada penelitian ini penulis membuat sistem pakar untuk mendiagnosa penyakit padi yang dapat memberikan solusi pengendalian dari penyakit tersebut guna mengurangi atau memperkecil resiko kerusakan tanaman padi dengan menerapkan factor kepastian (*Certainty Factor*) dalam pengambilan keputusannya. Dengan dibuatnya sistem ini diharapkan dapat membantu masyarakat luas khususnya para petani padi agar mengetahui cara membasmi penyakit pada padi tersebut sehingga tidak mengalami gagal panen.

1.2 Rumusan Masalah

Rumusan masalah yang dapat diambil dari latar belakang yang telah dipaparkan diatas yaitu :

1. Bagaimana memudahkan petani dalam mendiagnosa penyakit padi ?
2. Bagaimana merancang dan membuat sistem pakar diagnosa penyakit padi dengan metode *Certainty Factor* ?

1.3 Batasan Masalah

Adapun batasan masalah dari penelitian ini adalah :

1. *Platform* yang digunakan berbasis *web*.
2. *Framework* yang digunakan untuk membuat sistem pakar diagnosa penyakit padi ini adalah *codeigniter* ?

3. *Database* yang digunakan untuk membuat sistem pakar diagnosa penyakit padi ini adalah *MySQL*.
4. Sistem pakar ini hanya dapat menyimpulkan satu penyakit dari setiap proses diagnosa.
5. Pada penelitian ini hanya terdapat lima rule yang masing-masing rule terdiri dari satu penyakit.
6. Pada sistem ini pakar hanya memberi validasi dari inputan admin.
7. Sistem harus terkoneksi dengan jaringan *internet*.

1.4 Tujuan Penelitian

Tujuan penelitian dari pembuatan sistem pakar diagnosa penyakit padi ini adalah :

1. Untuk memudahkan petani mendiagnosa penyakit padi.
2. Untuk merancang dan membuat sistem pakar diagnosa penyakit padi dengan metode *certainty factor*

1.5 Manfaat Penelitian

1. Manfaat bagi penulis :
Menerapkan ilmu pengetahuan yang didapat dari perkuliahan, melatih kreatifitas, menambah wawasan dan pengalaman untuk membuat suatu sistem.
2. Manfaat bagi pengguna :
Dapat mendiagnosa penyakit padi dan mengetahui solusi untuk penanggulangan dengan efisien melalui aplikasi tanpa harus mendatangi ahli pertanian secara langsung.
3. Manfaat bagi pembaca :
Sebagai referensi pembaca untuk membuat suatu aplikasi yang berkaitan dengan sistem pakar, *platform web* dengan *framework codeigniter*.

1.6 Sistematika Penulisan Laporan

Untuk memahami lebih jelas laporan ini, maka materi-materi yang tertera pada laporan tugas akhir ini dikelompokkan menjadi beberapa bab dengan sistematika penyampaian sebagai berikut :

BAB I PENDAHULUAN

Pada bab ini berisi pemaparan latar belakang masalah, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian dan sistematika penulisan laporan.

BAB II LANDASAN TEORI

Bab ini berisi teori dasar yang menguraikan tentang kecerdasan buatan, sistem pakar, metode *certainty factor*, *UML* dan teknik pengujian perangkat lunak.

BAB III METODOLOGI PENELITIAN

Bagian ini berisi penjelasan tentang metode penelitian, metode pengumpulan data, analisa data, analisa kebutuhan sistem, *flowchart*, perancangan sistem, perancangan *ERD* , perancangan *database*, perancangan desain antarmuka dan perancangan sistem pakar.

BAB IV HASIL DAN PEMBAHASAN

Pada bab ini berisi hasil dan pembahasan tentang implementasi *database*, implementasi desain antarmuka dan pengujian sistem.

BAB V PENUTUP

Bab ini berisi kesimpulan dan saran penulis.

BAB II

LANDASAN TEORI

2.1 Penyakit Padi

Penyakit padi tergolong menjadi tiga jenis yaitu penyakit yang disebabkan oleh jamur patogen seperti blas dan hawar pelepah daun busuk batang, penyakit yang disebabkan oleh virus seperti kerdil rumput dan tungro dan penyakit yang disebabkan oleh bakteri seperti hawar daun bakteri.

- 1) Penyakit blas disebabkan oleh jamur *Pyricularia oryzae* Cav. Jamur ini juga mudah mengadakan mutasi yang menyebabkan menjadi tahan terhadap fungisida.
- 2) Penyakit hawar pelepah daun (*Sheath Blight*) dan busuk batang merupakan penyakit penting terutama pada padi di lahan pasang surut bergambut. Apabila gejala penyakit sampai pada daun bendera, kerugian dapat mencapai lebih dari 20%.
- 3) Penyakit kerdil rumput disebabkan oleh *Rice Grass Stunt Virus*. Dimana pada tahun 1971 lebih dari 8 ribu ha padi di Tegal dan Klaten terjangkit. Virus penyebab penyakit ditularkan oleh WBC (*Nilaparvata lugens* Stal).
- 4) Penyakit tungro disebabkan oleh virus *Rice Tungro Bacilliform Virus* dan *Rice Tungro Spherica-form Virus*, virus ditularkan oleh serangga *Nephotettix nigropictus* dan *N. virescens*. Serangga yang menghisap pada tanaman yang sakit mendapatkan virus dalam waktu yang singkat (30 menit) segera setelah itu serangga dapat menularkan virus pada tanaman lain.
- 5) Penyakit hawar daun bakteri disebabkan oleh bakteri patogen *Xanthomonas campestris* pv. *Oryzae*. Tahun 1948 ditemukan penyakit baru di Bogor dan disebut penyakit “kresek” , setelah diidentifikasi penyakit tersebut identic dengan hawar daun bakteri.

(Direktorat Jenderal Tanaman Pangan, 2013)

2.2 Kecerdasan Buatan

Kecerdasan buatan atau “*Artificial Intelligence*” adalah cabang ilmu komputer yang bertujuan untuk membuat sebuah komputer dapat berpikir dan bernalar seperti manusia. Kecerdasan buatan dapat membantu manusia dalam membuat keputusan, mencari informasi secara lebih akurat, atau membuat komputer lebih mudah digunakan dengan tampilan menggunakan bahasa yang mudah dipahami. Salah satu bagian dari kecerdasan buatan adalah sistem pakar dimana sistem pakar adalah bagian dari ilmu kecerdasan buatan yang secara spesifik berusaha mengadopsi kepakaran seseorang di bidang tertentu ke dalam suatu sistem atau *program computer* (Handojo dkk, 2009).

2.1 Sistem Pakar

Sistem pakar merupakan cabang dari *Artificial Intelligence (AI)* yang dikembangkan pada pertengahan 1960. Sistem pakar berasal dari istilah *knowledge-based expert system*, yaitu sebuah sistem yang menggunakan pengetahuan manusia dimana pengetahuan tersebut dimasukkan kedalam komputer dan kemudian digunakan untuk menyelesaikan masalah-masalah yang biasanya membutuhkan kepakaran atau keahlian manusia (Sutojo dkk, 2011).

2.1.1 Kelebihan dan Kekurangan Sistem Pakar

Secara garis besar banyak kelebihan yang didapatkan dengan adanya sistem pakar, antara lain:

1. Menjadikan pengetahuan lebih mudah didapat.
2. Menyimpan kemampuan dan keahlian pakar.
3. Meningkatkan penyelesaian masalah
4. Memberikan respon (jawaban) yang cepat.
5. Merupakan panduan yang *intelligence* (cerdas).
6. Dapat mengolah informasi yang kurang lengkap dan mengandung ketidakpastian.

Selain kelebihan-kelebihan diatas, sistem pakar juga memiliki kekurangan, diantaranya adalah:

1. Masalah dalam mendapatkan pengetahuan di mana pengetahuan tidak selalu bisa didapatkan dengan mudah. Karena kadang kala pakar dari masalah yang dibuat tidak ada, dan walaupun ada kadang-kadang pendekatan yang dimiliki oleh pakar berbeda-beda.
2. Untuk membuat suatu sistem pakar yang benar-benar berkualitas tinggi sangatlah sulit dan memerlukan biaya yang sangat besar untuk pengembangan dan pemeliharaannya.
3. Ada kemungkinan sistem tak dapat membuat keputusan.
4. Sistem pakar tidaklah 100% sempurna, oleh karena itu perlu diuji secara teliti dan berulang sebelum digunakan.

Kelemahan-kelemahan atau kekurangan dari sistem pakar tersebut bukanlah sama sekali tidak bisa diatasi, namun dengan terus melakukan perbaikan dan pengolahan berdasarkan pengalaman yang telah ada maka hal tersebut diyakini akan dapat diatasi, walaupun dalam waktu yang panjang dan terus menerus (Arhami, 2004 : 9)

2.1.2 Ciri-ciri Sistem Pakar

Ciri-ciri sistem pakar adalah:

1. Terbatas pada keahlian tertentu.
2. Dapat memberikan penalaran-penalaran untuk data-data yang tidak lengkap atau tidak pasti.
3. Dapat mengemukakan rangkaian alasan-alasan yang diberikannya dengan cara yang dapat dipahami.
4. Berdasarkan pada kaidah/ketentuan/*rule* tertentu.
5. Dirancang untuk dapat dikembangkan secara bertahap.
6. Pengetahuan dan mekanisme penalaran (*inference*) jelas terpisah.
7. Keluarannya bersifat anjuran. (Sutojo, 2011).

2.1.3 Manfaat Sistem Pakar

Manfaat sistem pakar diantaranya:

1. Meningkatkan produktivitas, karena sistem pakar dapat bekerja lebih cepat daripada manusia.

2. Meningkatkan kualitas, dengan memberi nasihat yang konsisten dan mengurangi kesalahan.
3. Mampu menangkap pengetahuan dan kepakaran seseorang.
4. Memudahkan akses pengetahuan seorang pakar.
5. Handal, sistem pakar tidak pernah menjadi bosan dan kelelahan atau sakit.
6. Mampu bekerja dengan informasi yang tidak lengkap atau tidak pasti (Sutojo, 2011).

2.1.4 Perbandingan Pakar dan Sistem Pakar

Perbandingan kemampuan antara seorang pakar dengan sistem pakar disajikan pada Tabel 2.1.

Tabel 2.1 Perbandingan Sistem Pakar

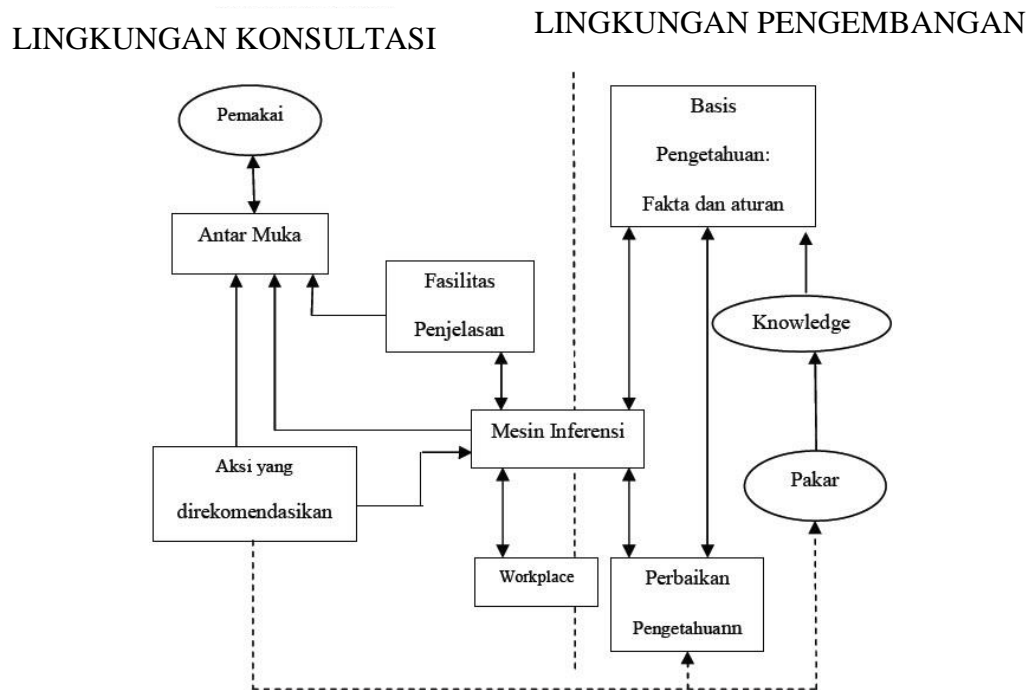
No	Pakar Manusia	Sistem Pakar
1	Memiliki waktu yang terbatas karena manusia membutuhkan istirahat	Waktu tidak terbatas karena dapat digunakan kapanpun
2	Tempat akses bersifat <i>local</i> pada suatu tempat saja dimana pakar berada	Dapat digunakan di berbagai tempat
3	Pengetahuan bersifat variabel dan dapat berubah tergantung situasi	Pengetahuan bersifat konsisten
4	Kecepatan untuk menemukan solusi bervariasi	Kecepatan untuk memberikan solusi bersifat konsisten
5	Biaya yang diperlukan untuk konsultasi sangat mahal	Biaya yang diperlukan untuk konsultasi lebih murah

Sumber : (Abdullah, 2016)

2.1.5 Komponen Sistem Pakar

Sistem pakar disusun oleh dua bagian utama, yaitu bagian lingkungan pengembangan (*development environment*) dan lingkungan konsultasi (*consultation environment*). Lingkungan pengembang digunakan oleh pembuat sistem pakar untuk membangun komponen-komponennya dan memperkenalkan

pengetahuan ke dalam *knowledge base* (basis pengetahuan). Sedangkan lingkungan konsultasi digunakan oleh pengguna untuk berkonsultasi sehingga pengguna mendapatkan pengetahuan dan nasihat dari sistem pakar layaknya berkonsultasi dengan seorang pakar. (Siswanto, 2010). Komponen-komponen sistem pakar (Rachmawati dkk, 2012) dapat dilihat pada Gambar2.1.



Gambar 2.1 Komponen Sistem Pakar (Rachmawati dkk, 2012)

Gambar 2.1 menjelaskan bahwa secara umum sistem pakar terdiri dari komponen penyusun sebagai berikut:

1. *Knowledge Base* (Basis Pengetahuan) Basis pengetahuan merupakan hasil akuisisi dan representasi pengetahuan dari seorang pakar. Basis pengetahuan berisi pengetahuan-pengetahuan dalam penyelesaian masalah. Basis pengetahuan terdiri dari dua elemen dasar yaitu fakta dan *rule* atau aturan.
2. *Inference Engine* (Mesin Inferensi) Mesin inferensi adalah sebuah program yang berisi metodologi yang digunakan untuk melakukan penalaran terhadap informasi-informasi dalam basis pengetahuan untuk memformulasikan konklusi.
3. *User Interface* (Antar Muka Pengguna) *User interface* adalah penghubung antar program sistem pakar dengan pengguna yang dapat dihubungkan via dekstop ataupun mobile. Antarmuka digunakan sebagai media komunikasi antara pengguna dan sistem pakar. (Rachmawati dkk, 2012)

2.2 Metode *Certainty Factor*

Teori *Certainty Factor* (CF) adalah untuk mengakomodasi ketidakpastian pemikiran (*inexact reasoning*) seorang pakar yang di usulkan oleh *Shortliffe* dan *Buchanan* pada tahun 1975. Seorang pakar (misalnya dokter) sering menganalisis informasi yang ada dengan ungkapan dengan ketidakpastian, untuk mengakomodasi hal ini maka digunakan *Certainty Factor* (CF) guna menggambarkan tingkat keyakinan pakar terhadap masalah yang sedang dihadapi. Dalam mengekspresikan derajat kepastian, *Certainty Faktor* untuk mengasumsikan derajat kepastian seorang pakar terhadap suatu data. Konsep ini kemudian diformulasikan dalam rumusan dasar sebagai berikut :

$$CF[H,E]1 = CF[H] * CF[E].....(2.1)$$

Dimana :

CF(E) = *certainty factor evidence* E yang dipengaruhi oleh *evidence* E

CF(H) = *certainty factor* hipotesa dengan asumsi *evidence* diketahui dengan pasti, yaitu ketika $CF(E,e) = 1$

CF(H,E) = *certainty factor* hipotesa yang dipengaruhi oleh *evidence* e diketahui dengan pasti

Certainty Factor untuk kaidah dengan kesimpulan yang serupa (*similarly concluded rules*) :

$$CF_{combine} CF[H,E]1,2 = CF[H,E]1 + CF[H,E]2 * [1 - CF[H,E]1].....(2.2)$$

$$CF_{combine} CF[H,E]old,3 = CF[H,E]old + CF[H,E]3 * (1 - CF[H,E]old).....(2.3)$$

Penggabungan kepercayaan dan ketidakpercayaan dalam bilangan yang tunggal memiliki dua kegunaan, yaitu pertama faktor kepastian digunakan untuk tingkat hipotesa di dalam urutan kepentingan (Sutojo, 2011).

Faktor kepastian menyatakan kepercayaan dalam sebuah kejadian (atau fakta atau hipotesis) berdasarkan bukti atau penilaian pakar. Faktor kepastian menggunakan suatu nilai untuk mengasumsikan derajat keyakinan seorang pakar terhadap suatu data (Turban, 2005).

Ada 2 macam faktor kepastian yang digunakan, yaitu:

1. Faktor kepastian yang diisikan oleh pakar bersama dengan aturan.
2. Faktor kepastian yang diberikan oleh pengguna (Kusrini, 2008).

2.3 Unified Modeling Language (UML)

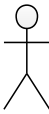

UML (*Unified Modeling Language*) adalah ‘bahasa’ pemodelan untuk sistem atau perangkat lunak yang berparadigma ‘berorientasi objek’. Pemodelan (*modeling*) sesungguhnya digunakan untuk penyederhanaan permasalahan-permasalahan yang kompleks sedemikian rupa sehingga lebih mudah dipahami. (Nugroho, 2009)

Berikut penjelasan dari beberapa diagram yang digunakan pada penelitian ini.

2.3.1 Diagram Use-Case (*Usecase Diagram*)






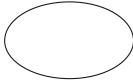
Usecase diagram menggambarkan fungsionalitas dari sebuah sistem. Sebuah *usecase* merepresentasikan sebuah interaksi antara *actor* dengan sistem. *Actor* adalah sebuah entitas manusia yang berinteraksi dengan sistem untuk melakukan pekerjaan tertentu. Simbol *Usecase Diagram* dapat dilihat pada Tabel 2.2.

Tabel 2.2 Simbol *Usecase Diagram*

No	Gambar	Nama	Keterangan
1		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i>
2		<i>Depedency</i>	Hubungan dimana perubahan yang terjadi pada satu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (<i>independent</i>)


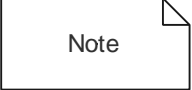
Sumber : (Abdullah, 2016)

Tabel 2.3 Simbol *Usecase Diagram* (Lanjutan)

No	Gambar	Nama	Keterangan
3		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>)
4		<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara eksplisit
5		<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan
6		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya
7		<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas
8		<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terstruktur bagi suatu aktor

Sumber : (Abdullah, 2016)

Tabel 2.4 Simbol *Usecase Diagram* (Lanjutan)

No	Gambar	Nama	Keterangan
9		<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi)
10		<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi

Sumber : (Abdullah, 2016)


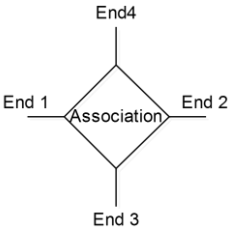
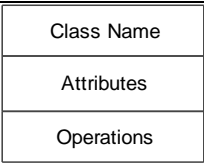

2.3.2 Diagram Kelas (*Class Diagram*)

Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi). *Class diagram* menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain. *Class* memiliki tiga area pokok, yaitu nama (dan *stereotype*), atribut, dan metoda. Atribut dan metoda dapat memiliki salah satu sifat berikut (Abdullah, 2016):

- 1) *Private*, tidak dapat dipanggil dari luar *class* yang bersangkutan
- 2) *Protected*, hanya dapat dipanggil oleh *class* yang bersangkutan dan anak-anak yang mewarisinya
- 3) *Public*, dapat dipanggil oleh siapa saja

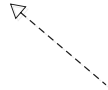
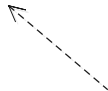
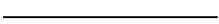
Simbol *Class Diagram* dapat dilihat pada Tabel 2.5.

Tabel 2.5 Simbol *Class Diagram*

No	Gambar	Nama	Keterangan
1		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>)
2		<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
3		<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
4		<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi)

Sumber : (Abdullah, 2016)

Tabel 2.6 Simbol *Class Diagram* (Lanjutan)

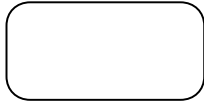
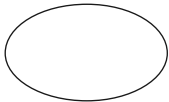



No	Gambar	Nama	Keterangan
5		<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan
6		Association	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempegaruhi elemen yang bergantung padanya elemen yang tidak mandiri
7		<i>Association</i>	Apa yang menghubungkan antara objek satu dnegan objek lainnya

Sumber : (Abdullah, 2016)

2.3.3 Diagram Aktivitas (*Activity Diagram*)

Activity diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. *Activity diagram* dapat dibagi menjadi beberapa *object swimlane* untuk menggambarkan objek mana yang bertanggung jawab untuk aktivitas tertentu. Simbol *Activity Diagram* dapat dilihat pada Tabel 2.7.

Tabel 2.7 Simbol *Activity Diagram*

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

Sumber : (Abdullah, 2016)

2.4 Teknik Pengujian Perangkat Lunak

Pengujian pada sistem pakar diagnosa penyakit padi dengan metode *Certainty Factor* ini terpadat dua teknik pengujian yaitu pengujian *Black-box* dan pengujian kuisisioner. Pengujian *Black-box* merupakan pengujian untuk mengetahui apakah semua fungsi perangkat lunak telah berjalan semestinya sesuai dengan kebutuhan fungsional yang telah didefinisikan (Jiang, 2012). Kasus ini bertujuan untuk menunjukkan fungsi perangkat lunak tentang cara beroperasinya. Teknik pengujian ini yaitu melakukan kasus uji dengan mempartisi domain *input* dan *output* program. Metode *Black-box* memungkinkan perekayasa perangkat lunak mendapatkan serangkaian kondisi *input* yang sepenuhnya menggunakan semua persyaratan

fungsional untuk suatu program. Pengujian ini berusaha menemukan kesalahan dalam kategori fungsi-fungsi yang tidak benar atau hilang, kesalahan *interface*, kesalahan kinerja, dan inisialisasi dan kesalahan terminal (Pressman, 2010). Sedangkan pengujian kuisisioner adalah pengujian yang dilakukan dengan memberikan pertanyaan kepada beberapa penguji/responden mengenai sistem untuk menentukan apakah sudah cukup baik digunakan.

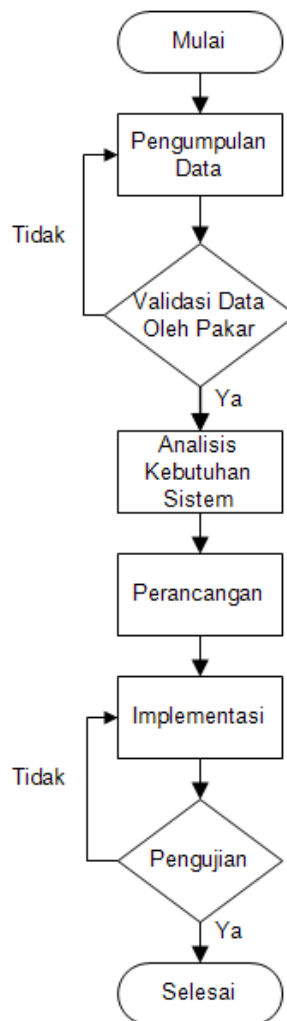
BAB III

METODOLOGI PENELITIAN

Pada bab metodologi penelitian ini berisikan langkah-langkah yang digunakan dalam pembuatan sistem pakar untuk mendiagnosa penyakit padi sebagai tugas akhir. Dengan adanya metodologi ini proses pembuatan aplikasi sehingga dapat dipahami oleh pembaca.

3.1 Metode Penelitian

Berdasarkan rumusan masalah dan tujuan penelitian yang ada pada bab I, pembuatan sistem pakar diagnosa penyakit padi menggunakan metode penelitian seperti yang terdapat pada Gambar 3.1 berikut.



Gambar 3.1 Alur Metode Penelitian

Metode penelitian pada sistem pakar diagnosa penyakit padi dengan metode *Certainty Factor* ini dimulai dari pengumpulan data kemudian dilakukan validasi data oleh pakar selanjutnya dilakukan analisis kebutuhan sistem berikutnya perancangan lalu implementasi dan tahap terakhir adalah pengujian.

3.2 Pengumpulan Data

Pengumpulan data pada sistem pakar diagnosa penyakit padi dengan metode *Certainty Factor* ini terdiri dari studi literatur, wawancara dan analisis data penyakit.

3.2.1 Studi Literatur

Pada pembuatan tugas akhir ini penulis menggunakan studi literatur yang bersumber dari buku Keputusan Direktur Jenderal Tanaman Pangan Nomor 53/HK.310/C.8/2012 tentang pedoman rekomendasi pengendalian organisme pengganggu tumbuhan (OPT) tanaman sereal, DIREKTORAT JENDERAL TANAMAN PANGAN tentang penyakit padi, jurnal Sistem Identifikasi Penyakit Tanaman Padi dengan Menggunakan Metode Forward Chaining tentang latar belakang masalah dan skripsi Sistem Pakar Mendiagnosa Penyakit Pada Ayam dengan Metode *Certainty Factor* Berbasis Android tentang sistem pakar dan UML yang menjadi referensi dalam pembuatan sistem pakar untuk mendiagnosa penyakit pada padi.

3.2.2 Wawancara

Penulis melakukan wawancara dengan Bapak Rifki Kadarachman.H, S.P selaku staff pelaksana di Badan Perlindungan Tanaman Pangan dan Hortikultura (BPTPH) Jawa Barat sub wilayah III Indramayu. Dari wawancara tersebut penulis memperoleh data serta penjelasan tentang apa saja penyakit, gejala serta pengendalian dari penyakit yang dialami tanaman padi, kemudian penulis menyimpulkan ke dalam proses yang terstruktur sehingga dapat diaplikasikan pada sistem pakar diagnosa penyakit padi dengan metode *Certainty Factor*.

3.2.3 Analisis Data Penyakit

Dalam mendiagnosa penyakit perlu diketahui terlebih dahulu gejala yang dialami oleh tanaman padi sehingga pakar dapat menyimpulkan penyakit yang menyerang tanaman padi tersebut. Pada Tabel 3.1 merupakan beberapa penyakit yang menyerang tanaman padi, beserta gejalanya, dan pembobotan dari masing-masing gejala yaitu :

Tabel 3.1 Tabel Analisis Data Penyakit

No	Penyakit		Gejala		
	Nama	Kode	Nama	Kode	Bobot
1.	Blas (Pyricularia Oryzae Cav.)	P01	1. Bercak coklat keputihan	G01	0.5
			2. Pelepah kering	G02	0.10
			3. Bercak pada malai	G03	0.50
			4. Bercak pada biji	G04	0.5
			5. Bulir padi hampa (kosong)	G05	0.50
			6. Daun busuk yang dimulai dengan adanya bercak berbentuk belah ketupat kemudian bercak meluas mengikuti urat tulang daun, kadang-kadang beberapa bercak daun bergabung menjadi satu seperti terbakar (malai belum keluar)	G06	0.75

Tabel 3.2 Tabel Analisis Data Penyakit (Lanjutan)

No	Penyakit		Gejala		
	Nama	Kode	Nama	Kode	Bobot
			7. Pangkal batang tanaman mengkerut, berwarna coklat kehitaman dan mudah rebah	G07	0.20
2	Hawar Pelepah daun dan busuk Batang (<i>Rhizoctonia solani kuhn</i>)	P02	1. Pelepah daun terlihat bercak basah berbentuk bulat, bercak membesar dengan bagian tengah berwarna abu-abu dan bagian tepi berwarna coklat	G08	0.25
			2. Bercak abu kehijauan pada pelepah daun dekat permukaan air	G09	0.40
			3. Tanaman mati	G10	0.5
3	Kerdil rumput (<i>Grassy stunt</i>)	P03	1. Tanaman menjadi kerdil	G11	0.70
			2. Pertumbuhan tidak normal	G12	0.70
			3. Daun-daun memendek, menyempit, kaku	G13	0.80
			4. Warna daun hijau kekuningan dipenuhi bercak seperti karat	G14	0.30

Tabel 3.3 Tabel Analisis Data Penyakit (Lanjutan)

No	Penyakit		Gejala		
	Nama	Kode	Nama	Kode	Bobot
4	Tungro	P04	1. Warna daun menjadi kuning sampai coklat yang dimulai dari ujung daun	G15	0.70
			2. Pembentukan dan perkembangan akar terhambat	G16	0.10
			3. Pembentukan bunga tertunda	G17	0.70
			4. Bercak pada daun warna hijau pucat	G18	0.50
5	Hawar daun bakteri (HDB) (<i>Xanthomonas campestris pv. oryzae.</i>)	P05	1. Tepi daun terdapat garis gelombang berwarna kuning	G20	0.70
			2. Pelepah daun menguning	G21	0.50
			3. Daun menjadi hijau kelabu dan menggulung dibagian ujung dan tepi daun	G22	0.30
			4. Terdapat bercak kuning pada daun yang dimulai dari ujung daun kemudian menjalar ke bawah	G23	0.70

3.3 Analisis Kebutuhan Sistem

Dalam pembuatan sistem pakar diagnosa penyakit padi ini terdapat kebutuhan *hardware* dan kebutuhan *software*. Berikut adalah kebutuhan-kebutuhan yang diperlukan :

3.3.1 Kebutuhan *Hardware*

Kebutuhan hardware yang diperlukan pada pembuatan sistem pakar diagnosa penyakit padi ini terdapat pada Tabel 3.4.

Tabel 3.4 Tabel Kebutuhan *Hardware*

No	Jenis <i>Hardware</i>	Kebutuhan <i>Hardware</i>	Keterangan
1	<i>Processor</i>	Intel Celeron N3060	Untuk mendukung berjalannya aplikasi pembuat sistem
2	<i>RAM</i>	4 GB atau lebih	Minimal
3	<i>Hardisk</i>	1 TB atau lebih	Untuk menyimpan data baik data aplikasi pembuat maupun pendukung lainnya

3.3.2 Kebutuhan *Software*

Adapun kebutuhan *software* dalam pembuatan aplikasi sistem pakar diagnosa penyakit padi ini terdapat pada Tabel 3.5.

Tabel 3.5 Tabel Kebutuhan *Software*

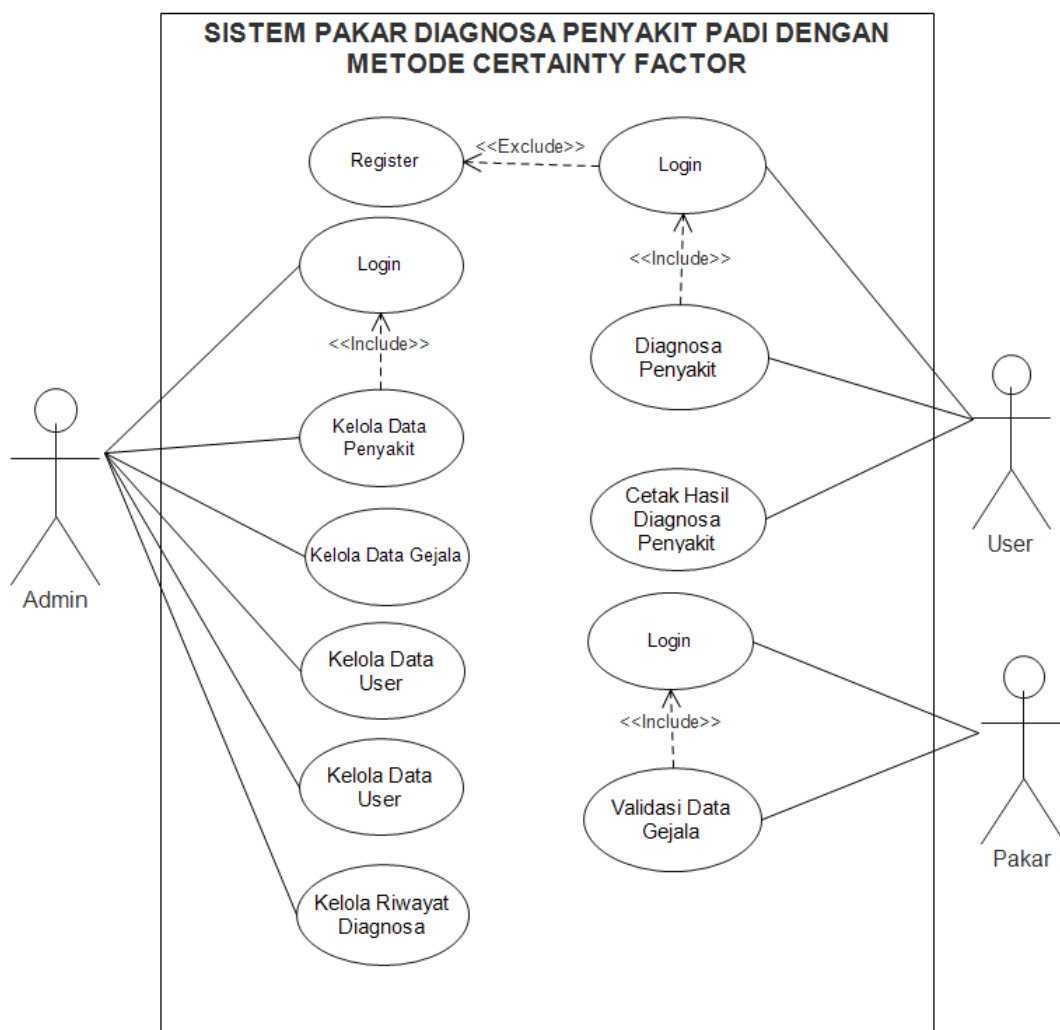
No	Jenis <i>Software</i>	Kebutuhan <i>Software</i>
1	Bahasa <i>Scripting</i>	PHP 5, HTML 5, CSS, JavaScript, JQuery
2	<i>Software</i> Pengolah	Sublime Text 3, Notepad++
3	Penyimpan Data	Database (MySQL)
4	Web Browser	Chrome, Firefox, dan lainnya

3.4 Perancangan Sistem

Untuk mendapatkan gambaran mengenai sistem yang dibuat, maka dimodelkan dengan menggunakan *functional modelling*. Proses dan data dari sistem dimodelkan dengan *use case diagram*, *activity diagram*, dan *class diagram*.

3.4.1 Use Case Diagram

Pada Gambar 3.2 merupakan *use case diagram* dari sistem pakar diagnosa penyakit padi dengan metode *Certainty Factor*.



Gambar 3.2 Use Case Diagram

Adapun penjelasan dari gambar 3.2 *Use Case Diagram* adalah pada Tabel 3.6 berikut ini :

Tabel 3.6 Penjelasan *Use Case* Diagram

<i>Actor</i>	<i>Nama Use Case</i>	Keterangan
Admin	<i>Login</i>	Admin dapat masuk ke sistem menggunakan akun yang telah terdaftar pada <i>database</i>
	Kelola data penyakit	Admin dapat menambah, melihat, mengubah, dan menghapus data penyakit yang ada pada sistem pakar diagnosa penyakit padi ini
	Kelola data gejala	Admin dapat menambah, melihat, mengubah, dan menghapus data gejala yang ada pada sistem pakar diagnosa penyakit padi ini
	Kelola data <i>rule</i>	Admin dapat menambah, melihat, mengubah, dan menghapus data <i>rule</i> yang ada pada sistem pakar diagnosa penyakit padi ini
	Kelola data <i>user</i>	Admin dapat melihat data <i>user</i> pada sistem pakar diagnosa penyakit padi ini
	Kelola data riwayat diagnosa	Admin dapat melihat riwayat diagnosa dari user

Tabel 3.7 Penjelasan Use Case Diagram (Lanjutan)

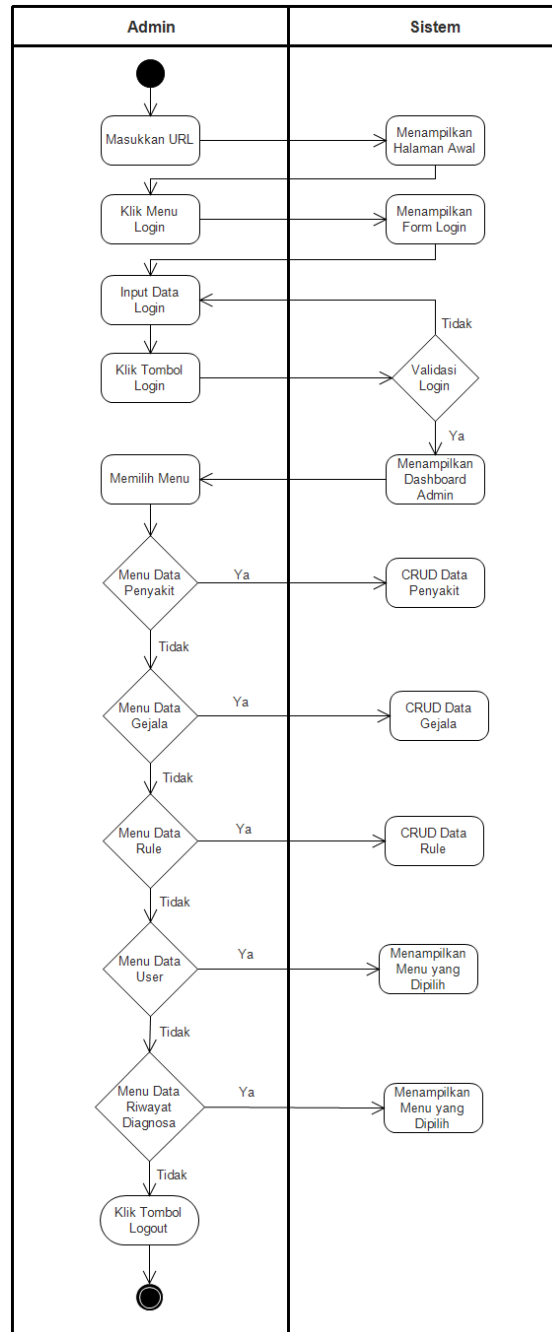
<i>Actor</i>	<i>Nama Use Case</i>	<i>Keterangan</i>
<i>User</i>	<i>Register</i>	<i>User</i> dapat mendaftarkan akun pada sistem pakar ini
	<i>Login</i>	<i>User</i> dapat masuk ke sistem menggunakan akun yang telah terdaftar pada <i>database</i>
	Diagnosa penyakit	<i>User</i> dapat melakukan diagnosa penyakit dengan menginputkan gejala yang dialami
	Cetak hasil diagnosa	<i>User</i> dapat mencetak hasil diagnosa yang telah dilakukan
Pakar	<i>Login</i>	Pakar dapat masuk ke sistem menggunakan akun yang telah terdaftar pada <i>database</i>
	Validasi data gejala	Pakar dapat melakukan validasi data pakar yang telah diinputkan oleh admin

3.4.2 Activity Diagram

Activity Diagram merupakan *state* diagram khusus yang menggambarkan berbagai aliran aktivitas di dalam sistem yang sedang dirancang, bagaimana masing-masing aliran berawal dan berakhir. Bahkan mungkin terjadi *decision* didalamnya. *Activity diagram* juga dapat menggambarkan proses parallel yang mungkin terjadi pada beberapa eksekusi dalam sistem.

3.4.2.1 Activity Diagram Admin

Pada diagram aktivitas admin ini menggambarkan aliran aktivitas yang dilakukan oleh admin dalam sistem yang dirancang. Adapun diagram aktivitas admin terdapat pada Gambar 3.3.



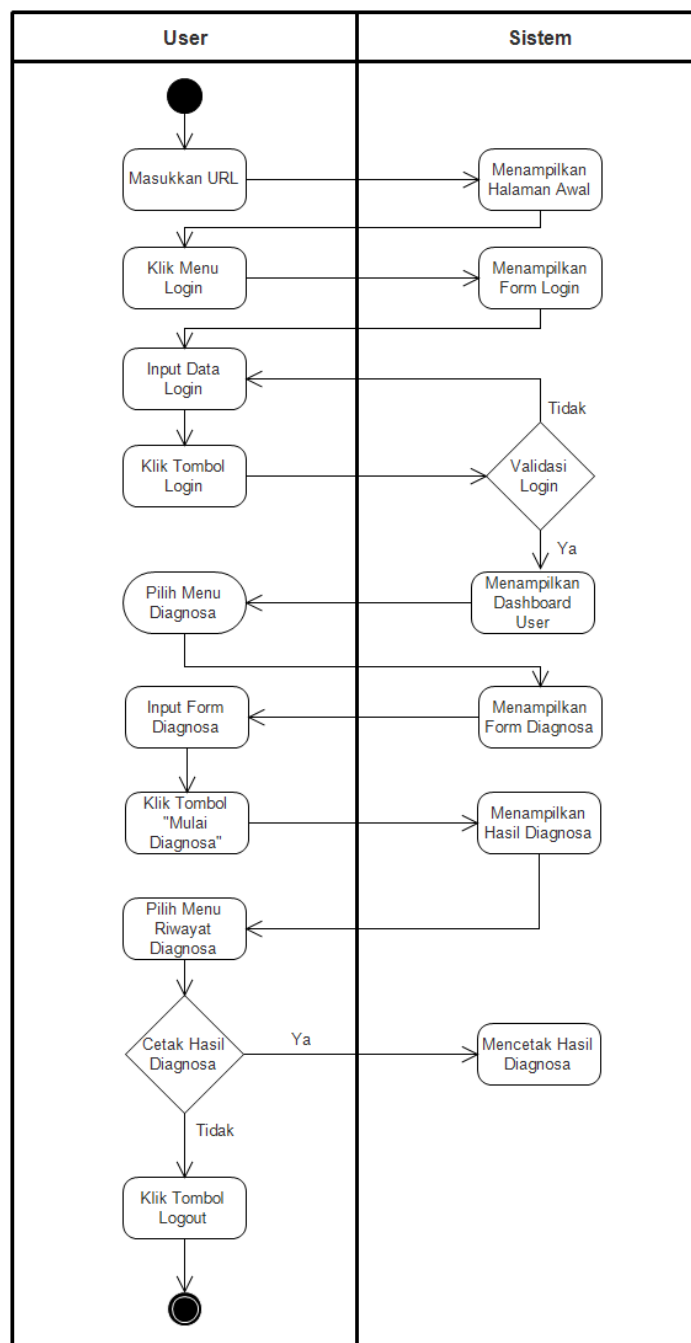
Gambar 3.3 Diagram Aktivitas Admin

Pada diagram aktivitas admin ini dimulai dari ini dimulai memasukkan *URL*, lalu klik *menu login*, selanjutnya yaitu input data *login* dan klik tombol *login*, selanjutnya admin dapat memilih *menu* dan melakukan *CRUD* pada data penyakit,

data gejala, data rule serta dapat memilih menu dan melihat data user dan data riwayat diagnosa.

3.4.2.2 Activity Diagram User

Pada diagram aktivitas *user* ini menggambarkan aliran aktivitas yang dilakukan oleh *user* dalam sistem yang dirancang. Adapun diagram aktivitas *user* terdapat pada Gambar 3.4.

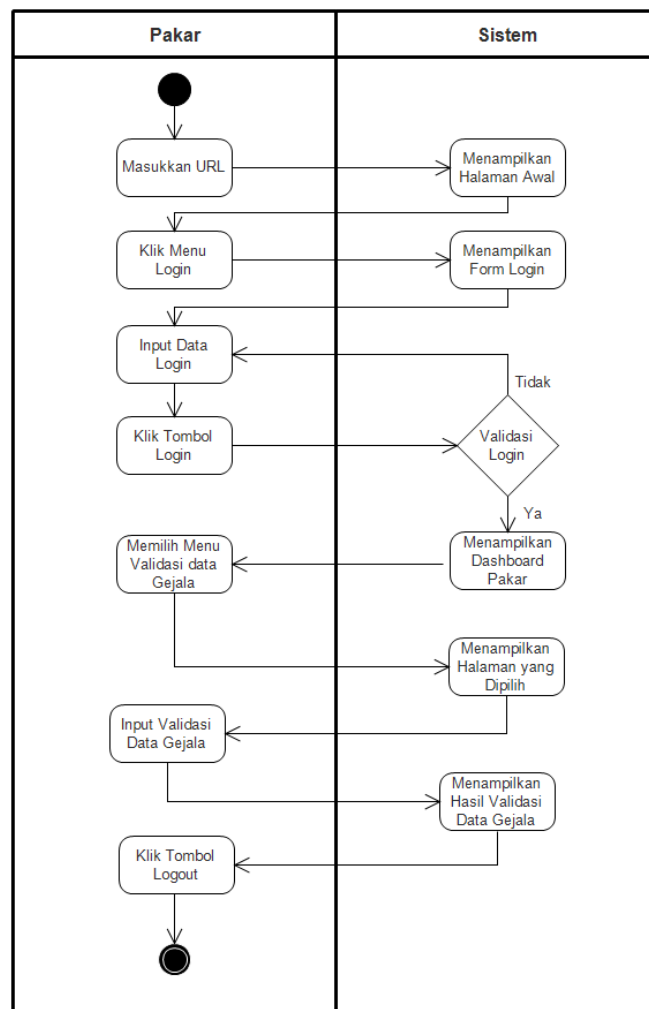


Gambar 3.4 Diagram Aktivitas *User*

Pada diagram aktivitas *user* ini dimulai memasukkan *URL*, lalu klik *menu login*, selanjutnya yaitu input data *login* dan klik tombol *login*, kemudian *user* dapat memilih menu diagnosa dan melakukan *input* pada *form* diagnosa, berikutnya *user* dapat memilih *menu* riwayat diagnosa dan dapat mencetak hasil diagnosa.

3.4.2.3 Activity Diagram Pakar

Pada diagram aktivitas pakar ini menggambarkan aliran aktivitas yang dilakukan oleh pakar dalam sistem yang dirancang. Pada aktivitas diagram ini dimulai dari login selanjutnya pakar dapat melakukan validasi gejala dari inputan admin. Adapun diagram aktivitas *user* terdapat pada Gambar 3.5.



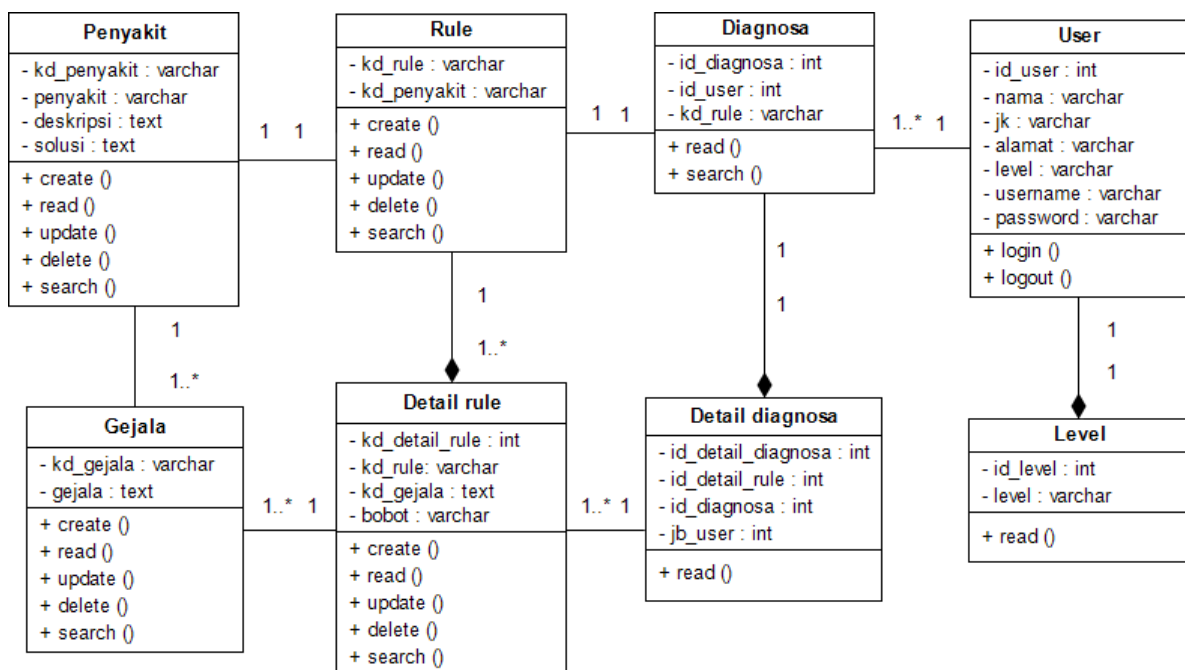
Gambar 3.5 Diagram Aktivitas pakar

Pada diagram aktivitas pakar ini dimulai memasukkan *URL*, lalu klik *menu login*, selanjutnya yaitu input data *login* dan klik tombol *login*, kemudian pakar

dapat memilih menu diagnosa dan melakukan *input* pada *form* diagnosa, berikutnya user dapat memilih *menu* riwayat diagnosa dan dapat mencetak hasil diagnosa.

3.4.3 Relasi Database

Pada Gambar 3.6 merupakan relasi database yang digunakan pada sistem pakar diagnosa penyakit padi dengan metode *Certainty Factor*. *Class diagram* ini memperlihatkan hubungan antar kelas dan penjelasan detail atribut dan method dari tiap kelas.



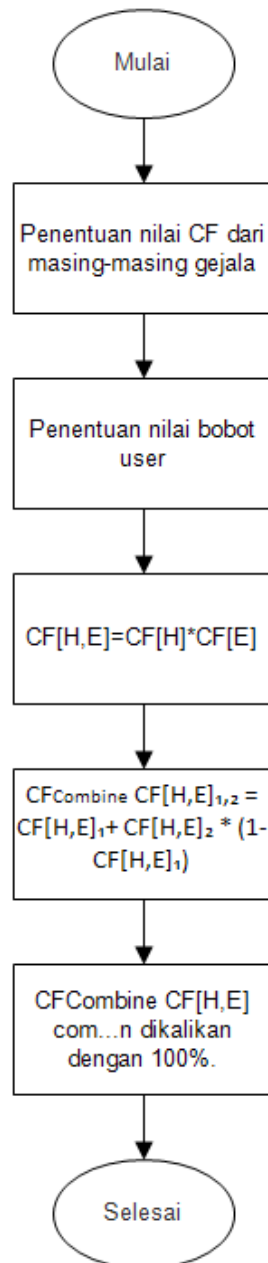
Gambar 3.6 Relasi Database

3.5 Flowchart

Flowchart yang dibuat untuk sistem pakar diagnosa penyakit padi dengan metode *certainty factor* ini terdiri dari *flowchart* metode *certainty factor*, *flowchart* admin, *flowchart* pakar dan *flowchart* user.

3.5.1 Flowchart Metode Certainty Factor

Flowchart ini adalah langkah-langkah penyelesaian masalah dari sisi metode. *Flowchart* metode dari sistem ini terdapat pada Gambar 3.7.

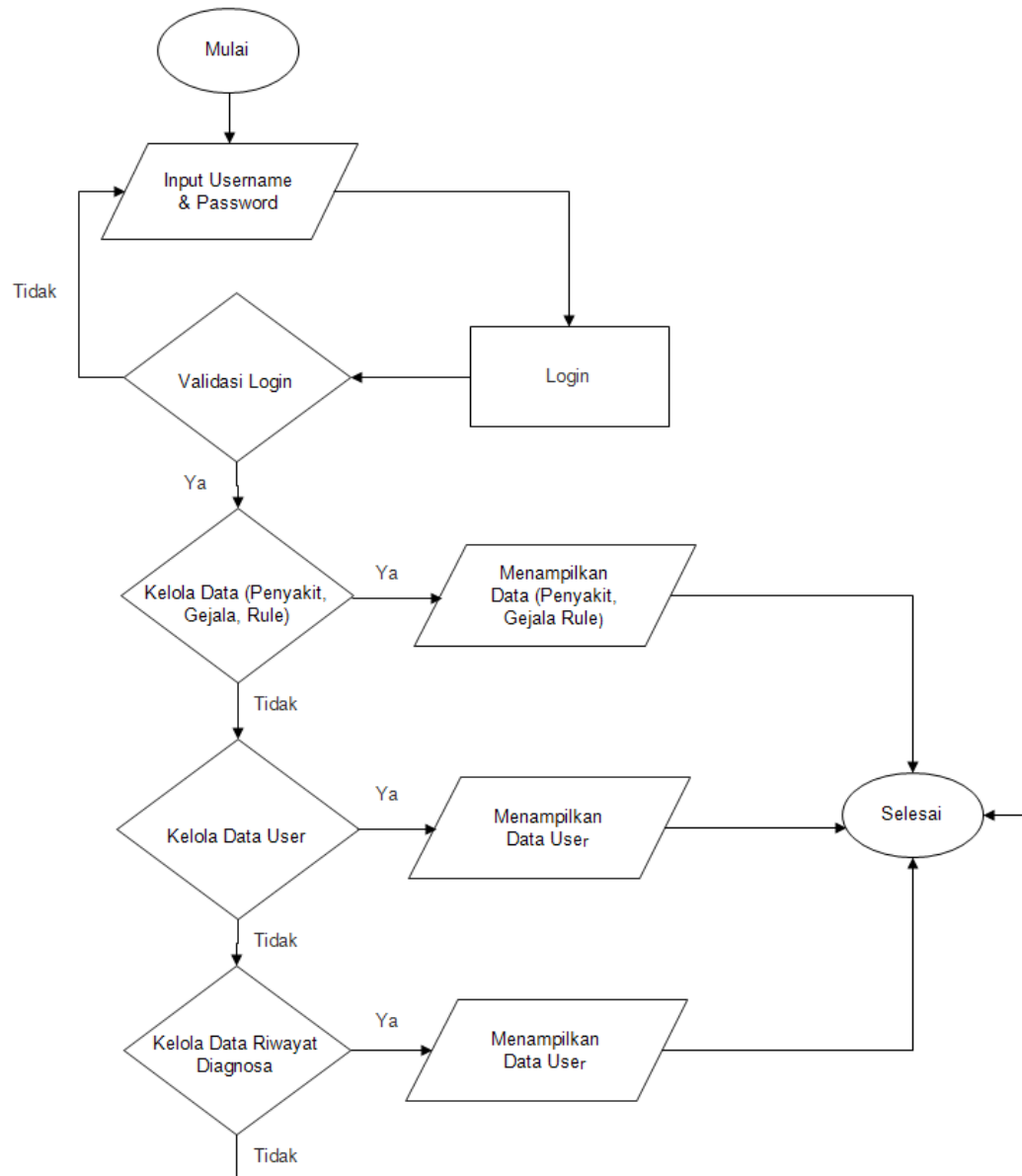


Gambar 3.7 Flowchart Metode Certainty Factor

Flowchart metode dimulai dari penentuan nilai *CF* masing-masing gejala yang dapat dilihat pada tabel 3.1, kemudian penentuan nilai bobot user yang dapat dilihat pada tabel 3.19, langkah berikutnya yaitu mencari hasil *CF* dengan mengalikan nilai gejala dengan bobot user yang dapat dilihat pada persamaan (2.2), langkah selanjutnya yaitu mencari hasil *CF* Combine yang dapat dilihat pada (2.3) dan langkah terakhir yaitu mengalikan hasil akhir *CF* Combine dengan 100%.

3.5.2 Flowchart Admin

Flowchart admin ini menggambarkan langkah-langkah penyelesaian masalah dari sisi admin. Adapun *flowchart* admin dari sistem ini terdapat pada Gambar 3.8.

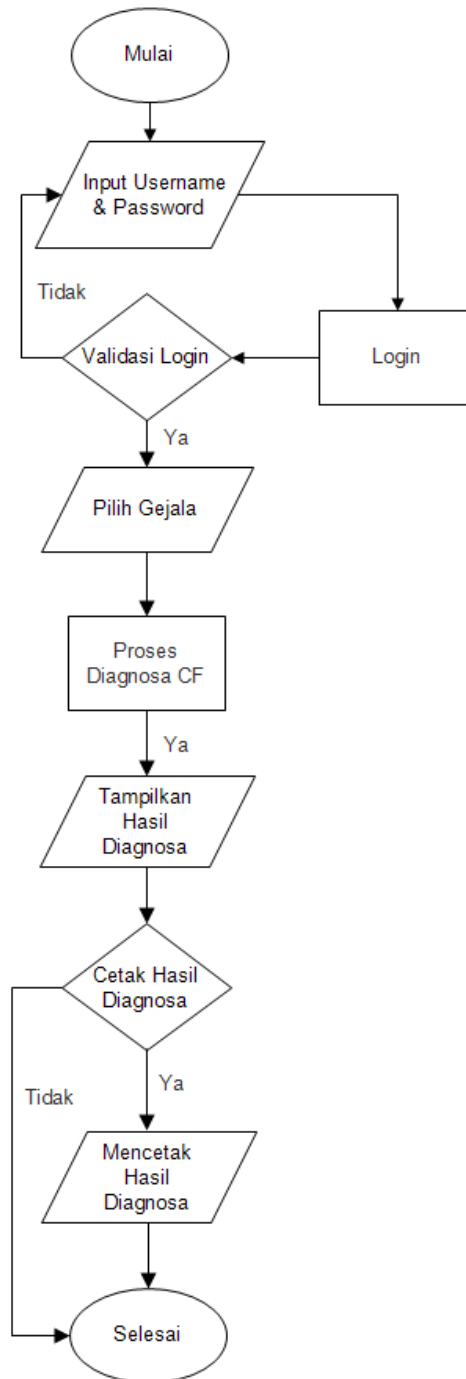


Gambar 3.8 *Flowchart* Admin

Flowchart admin dimulai dari input *username* dan *password* berikutnya dilakukan aksi login apabila validasi login berhasil maka selanjutnya admin dapat mengelola data penyakit, data gejala, data *rule*, data *user* dan data riwayat diagnosa.

3.5.3 Flowchart User

Flowchart user ini menggambarkan langkah-langkah penyelesaian masalah dari sisi *user*. Adapun *flowchart user* dari sistem ini terdapat pada Gambar 3.9.



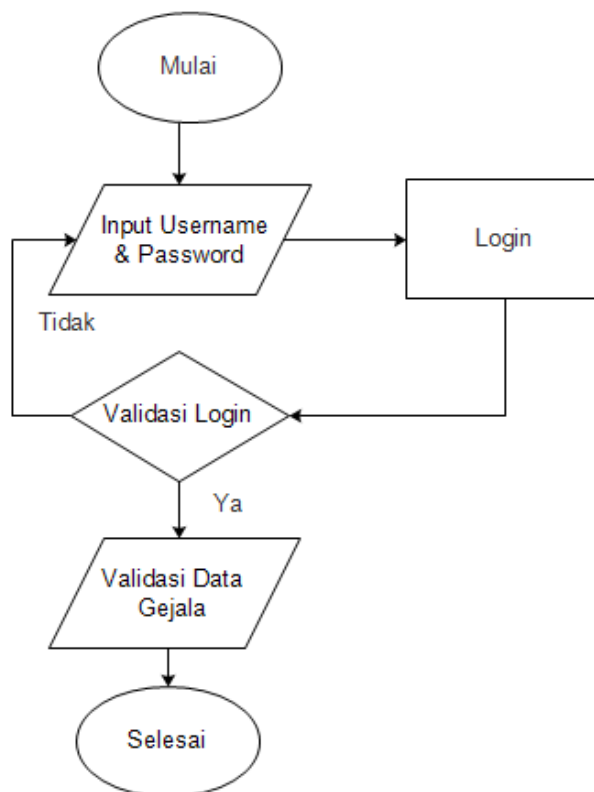
Gambar 3.9 *Flowchart User*

Flowchart user dimulai dari input *username* dan *password* berikutnya dilakukan aksi login apabila validasi login berhasil setelah itu *user* dapat memilih

gejala penyakit yang dialami, selanjutnya yaitu proses diagnosa penyakit, kemudian antarmuka menampilkan hasil diagnosa penyakit lalu user dapat mencetak hasil diagnosa penyakit tersebut.

3.5.4 Flowchart Pakar

Flowchart pakar ini menggambarkan langkah-langkah penyelesaian masalah dari sisi pakar. Adapun *flowchart* pakar dari sistem ini terdapat pada Gambar 3.10.



Gambar 3.10 *Flowchart* Pakar

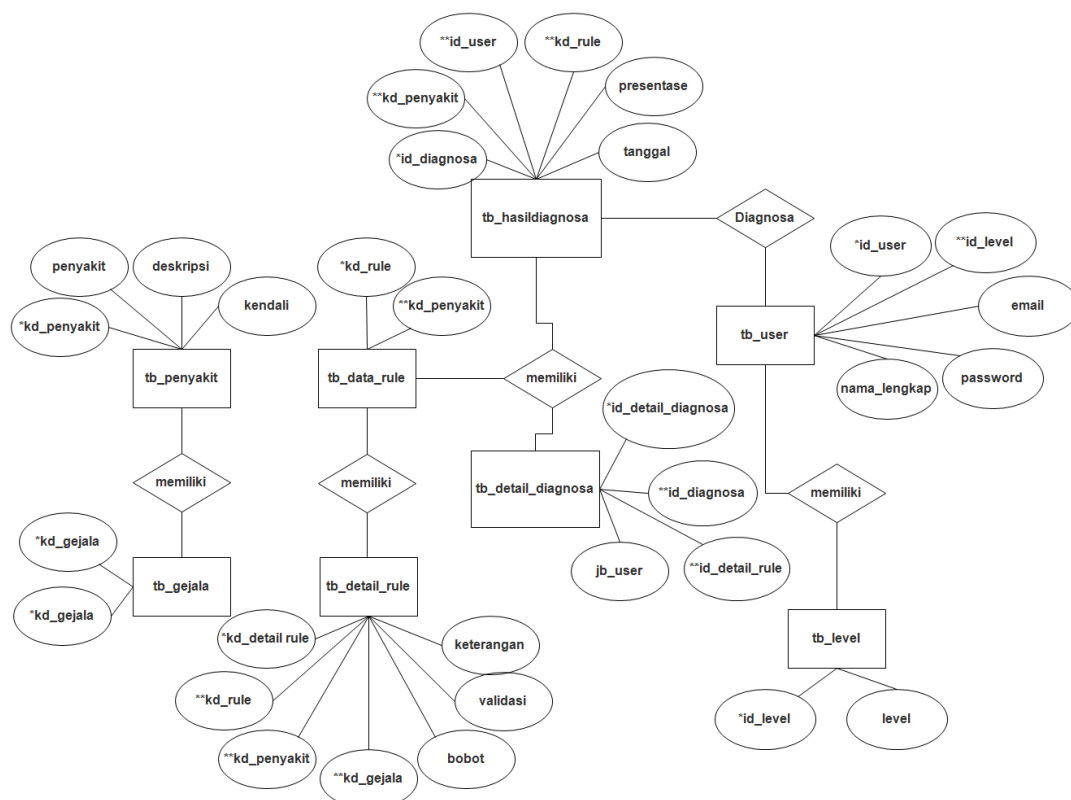
Flowchart pakar dimulai dari input *username* dan *password* berikutnya dilakukan aksi login apabila validasi login berhasil setelah itu pakar dapat melakukan validasi data gejala yang telah diinputkan admin.

3.6 Perancangan *ERD* (*Entity Relationship Diagram*)

Pada Gambar 3.11 merupakan gambar *Entity Relationship Diagram* (*ERD*) yang digunakan untuk proses Analisa perancangan dari sistem pakar diagnosa penyakit padi. Adapun penjelasan dari Gambar 3.11 adalah pada Tabel 3.8 berikut ini :

Tabel 3.8 Penjelasan *ERD*

Nama Tabel	Keterangan
tb_penyakit	Tabel yang digunakan untuk menyimpan data penyakit
tb_gejala	Tabel yang digunakan untuk menyimpan data gejala
tb_data_rule	Tabel yang digunakan untuk menyimpan data rule
tb_detail_rule	Tabel yang digunakan untuk menyimpan data detail rule
tb_hasildiagnosa	Tabel yang digunakan untuk menyimpan data hasil diagnosa
tb_detail_diagnosa	Tabel yang digunakan untuk menyimpan data detail hasil diagnosa
tb_user	Tabel yang digunakan untuk menyimpan data user
tb_level	Tabel yang digunakan untuk menyimpan level user



Gambar 3.11 Perancangan *ERD*

Pada sistem pakar diagnosa penyakit padi dengan metode *Certainty Factor* terdiri dari 8 tabel yaitu tb_penyakit, tb_gejala, tb_data_rule, tb_detail_rule, tb_hasildiagnosa, tb_detail_diagnosa, tb_user dan tb_level.

3.7 Perancangan Database

Berikut ini merupakan tabel dari perancangan *database* sistem pakar diagnosa penyakit padi dengan metode *Certainty Factor*.

3.7.1 Tabel tb_penyakit

Tabel tb_penyakit digunakan untuk menyimpan data penyakit, terdiri dari *field* kd_penyakit, penyakit, deskripsi, dan kendali. Adapun tabel tb_penyakit dapat dilihat pada Tabel 3.9.

Tabel 3.9 Tabel tb_penyakit

No	Nama <i>Field</i>	Tipe Data	Keterangan
1	kd_penyakit	Varchar (50)	Untuk menyimpan kode penyakit
2	penyakit	Text	Untuk menyimpan nama penyakit
3	deskripsi	Text	Untuk menyimpan deskripsi penyakit
4	kendali	Text	Untuk menyimpan solusi dari penyakit

3.7.2 Tabel tb_gejala

Tabel tb_gejala digunakan untuk menyimpan data gejala, terdiri dari *field* kd_gejala dan gejala dari suatu penyakit. Adapun tabel tb_gejala dapat dilihat pada Tabel 3.10.

Tabel 3.10 Tabel Tb_gejala

No	Nama <i>Field</i>	Tipe Data	Keterangan
1	kd_gejala	Varchar (50)	Untuk menyimpan kode gejala
2	gejala	Varchar (250)	Untuk menyimpan nama gejala

3.7.3 Tabel tb_data_rule

Tabel tb_data_rule digunakan untuk menyimpan data *rule* yang, terdiri dari *field* kd_rule dan kd_penyakit. Adapun tabel tb_rule dapat dilihat pada Tabel 3.11.

Tabel 3.11 Tabel Tb_rule

No	Nama Field	Tipe Data	Keterangan
1	kd_rule	Varchar (11)	Untuk menyimpan kode <i>rule</i> dari tiap-tiap penyakit
2	kd_penyakit	Varchar (25)	Untuk menyimpan nama kd_penyakit dari rule yang dibuat

3.7.4 Tabel tb_detail_rule

Tabel tb_detail_rule digunakan untuk menyimpan data detail *rule*, terdiri dari *field* kd_detail_rule, kd_rule, kd_penyakit, kd_gejala, bobot, validasi dan keterangan. Adapun tabel tb_detail_rule dapat dilihat pada Tabel 3.12.

Tabel 3.12 Tabel Tb_detail_rule

No	Nama Field	Tipe Data	Keterangan
1	kd_detail_rule	Int (50)	Untuk menyimpan kode detail <i>rule</i>
2	kd_rule	Varchar (50)	Untuk menyimpan kode <i>rule</i>
3	kd_penyakit	Varchar (50)	Untuk menyimpan nama kd_penyakit dari rule yang dibuat
4	kd_gejala	Varchar (50)	Untuk menyimpan kode gejala
5	bobot	Varchar (50)	Untuk menyimpan bobot dari tiap tiap gejala
6	validasi	Varchar (50)	Untuk menyimpan status validasi dari pakar
7	keterangan	Varchar (50)	Untuk menyimpan keterangan dari validasi pakar

3.7.5 Tabel tb_hasildiagnosa

Tabel tb_hasildiagnosa digunakan untuk menyimpan data hasil diagnosa yang dilakukan oleh tiap *user*, terdiri dari *field* id_diagnosa, id_user, kd_rule, kd_penyakit, tanggal, presentase. Adapun tabel tb_hasildiagnosa dapat dilihat pada Tabel 3.13.

Tabel 3.13 Tabel Tb_hasildiagnosa

No	Nama <i>Field</i>	Tipe Data	Keterangan
1	id_diagnosa	Int (11)	Untuk menyimpan id diagnosa
2	id_user	Int (11)	Untuk menyimpan user yang melakukan diagnosa
3	kd_rule	Varchar (11)	Untuk menyimpan kode <i>rule</i> dari hasil diagnosa
4	kd_penyakit	Varchar (25)	Untuk menyimpan kode penyakit dari hasil diagnosa
5	tanggal	Timestamp	Untuk menyimpan tanggal dari hasil diagnosa
6	presentase	Varchar (50)	Untuk menyimpan presentase penyakit dari hasil diagnosa

3.7.6 Tabel tb_detail_diagnosa

Tabel tb_detail_diagnosa digunakan untuk menyimpan data hasil diagnosa yang dilakukan oleh tiap user, terdiri dari *field* id_detail_diagnosa, id_diagnosa, id_detail_rule. Adapun tabel tb_detail_diagnosa dapat dilihat pada Tabel 3.14.

Tabel 3.14 Tabel Tb_detail_diagnosa

No	Nama <i>Field</i>	Tipe Data	Keterangan
1	id_detail_diagnosa	Int (11)	Untuk menyimpan id detail diagnosa
2	id_diagnosa	Varchar (50)	Untuk menyimpan id diagnosa

Tabel 3.15 Tabel Tb_detail_diagnosa (Lanjutan)

3	id_detail_rule	text	Untuk menyimpan id detail rule
4	jb_user	text	Untuk menyimpan jawaban user pada saat melakukan diagnosa

3.7.7 Tabel tb_user

Tabel tb_user digunakan untuk menyimpan data tiap *user*, terdiri dari *field* *id_user*, *id_level*, *email*, *password*, *nama_lengkap*. Adapun tabel tb_user dapat dilihat pada Tabel 3.16.

Tabel 3.16 Tabel Tb_user

No	Nama Field	Tipe Data	Keterangan
1	<i>id_user</i>	Int (11)	Untuk menyimpan id tiap user
2	<i>id_level</i>	Int (11)	Untuk menyimpan level dari tiap user
3	<i>email</i>	Varchar (50)	Untuk menyimpan <i>email user</i>
4	<i>password</i>	Varchar (50)	Untuk menyimpan <i>password user</i>
5	<i>nama_lengkap</i>	Varchar (50)	Untuk menyimpan <i>nama lengkap user</i>

3.7.8 Tabel tb_level

Tabel tb_level digunakan untuk menyimpan data level, terdiri dari *field* *id_level* dan *level*. Adapun tabel tb_user dapat dilihat pada Tabel 3.17.

Tabel 3.17 Tabel Tb_level

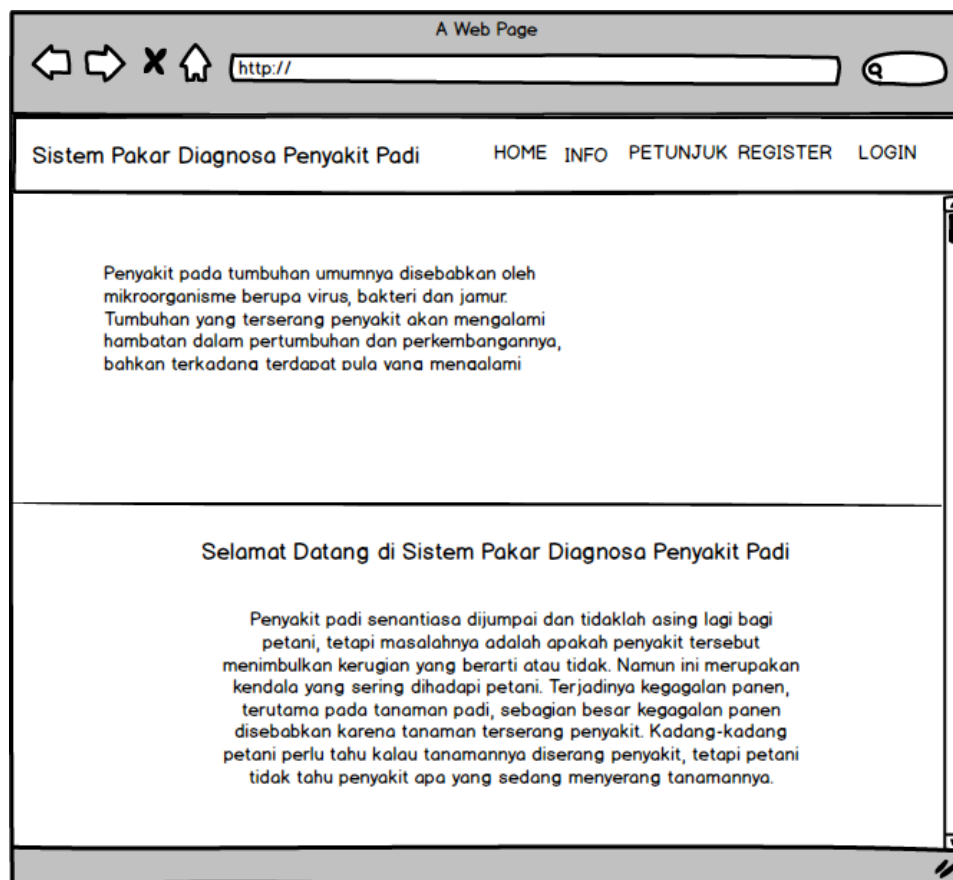
No	Nama Field	Tipe Data	Keterangan
1	<i>id_level</i>	Int (11)	Untuk menyimpan id level user
2	<i>level</i>	Varchar (50)	Untuk menyimpan level tiap user

3.8 Perancangan Desain Antarmuka

Ada beberapa rancangan desain antarmuka yang dibuat untuk sistem pakar diagnosa penyakit padi, berikut merupakan rancangannya :

3.8.1 Antarmuka *Landing Page*

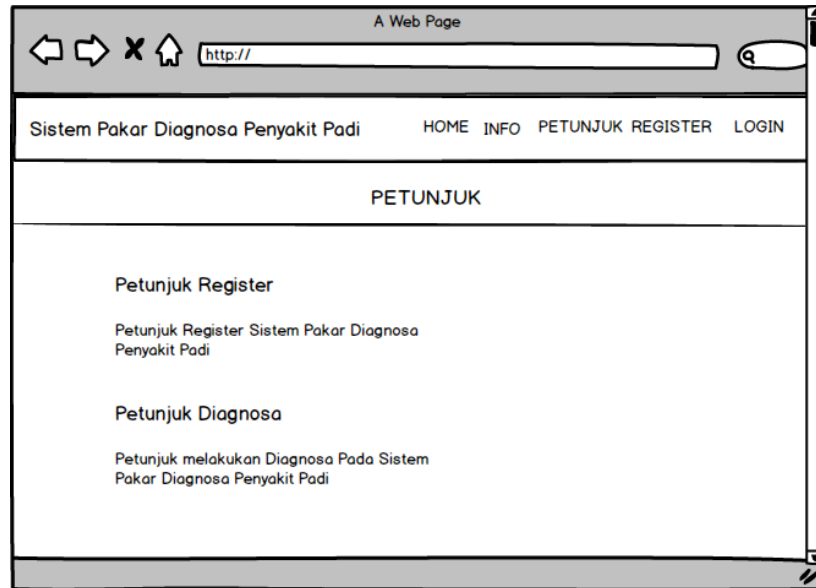
Antarmuka ini ditampilkan ketika *user* baru mengakses halaman sistem pakar diagnosa penyakit padi, pada halaman ini terdapat beberapa menu seperti *home*, *info*, *petunjuk*, *register* dan *login*. Adapun rancangan antarmuka *Landing Page* dapat dilihat pada Gambar 3.12.



Gambar 3.12 Antarmuka *Landing Page*

3.8.1.1 Antarmuka *Menu Petunjuk*

Antarmuka ini berisi petunjuk untuk melakukan pembuatan akun baru untuk melakukan diagnosa dan petunjuk untuk melakukan diagnosa. Adapun rancangan antarmuka *menu* petunjuk dapat dilihat pada Gambar 3.13.



Gambar 3.13 Antarmuka *Menu Petunjuk*

3.8.1.2 Antarmuka *Register*

Antarmuka ini ditampilkan ketika *user* baru akan melakukan pembuatan akun baru untuk melakukan diagnosa. Adapun rancangan antarmuka *register* dapat dilihat pada Gambar 3.14.

Gambar 3.14 Antarmuka *Register*

3.8.1.3 Antarmuka Login

Antarmuka ini ditampilkan ketika *user* akan masuk sistem dan sebelumnya telah memiliki akun. Adapun rancangan antarmuka *menu* petunjuk dapat dilihat pada Gambar 3.15.

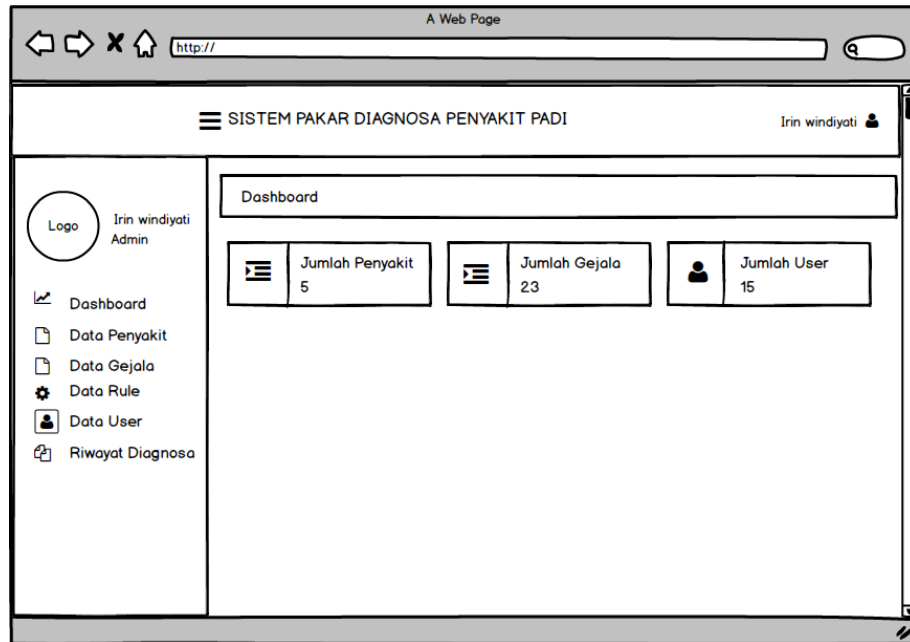
Gambar 3.15 Antarmuka *Login*

3.8.2 Antarmuka Admin

Antarmuka ini ditampilkan ketika admin berhasil login, pada antarmuka admin terdapat menu dashboard, data penyakit, data gejala, data rule, data user, dan riwayat diagnosa.

3.8.2.1 Antarmuka *Dashboard*

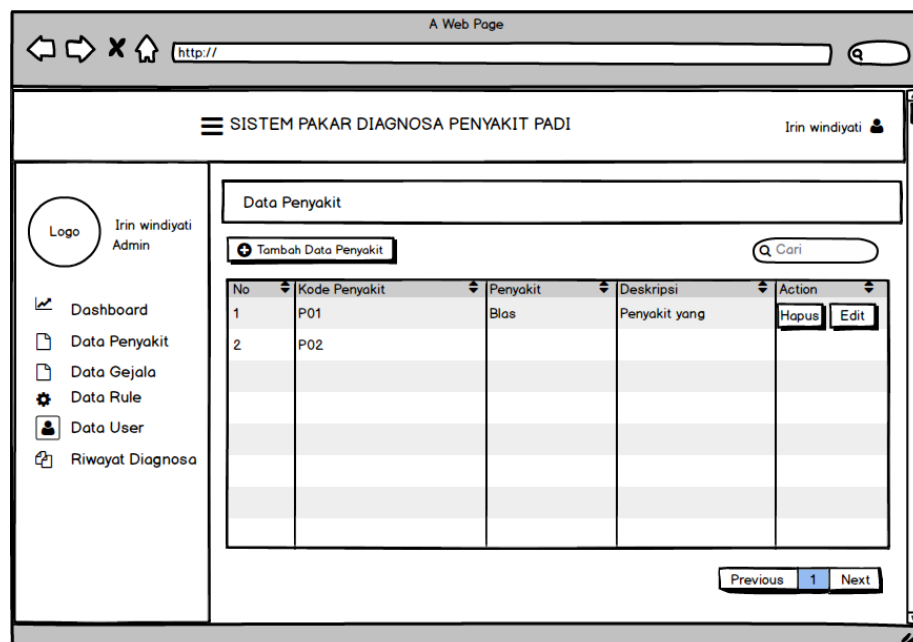
Antarmuka ini berisi info jumlah data penyakit, data jumlah gejala dan jumlah data *user*. Adapun rancangan antarmuka *dashboard* dapat dilihat pada Gambar 3.16.



Gambar 3.16 Antarmuka *Dashboard*

3.8.2.2 Antarmuka Data Penyakit

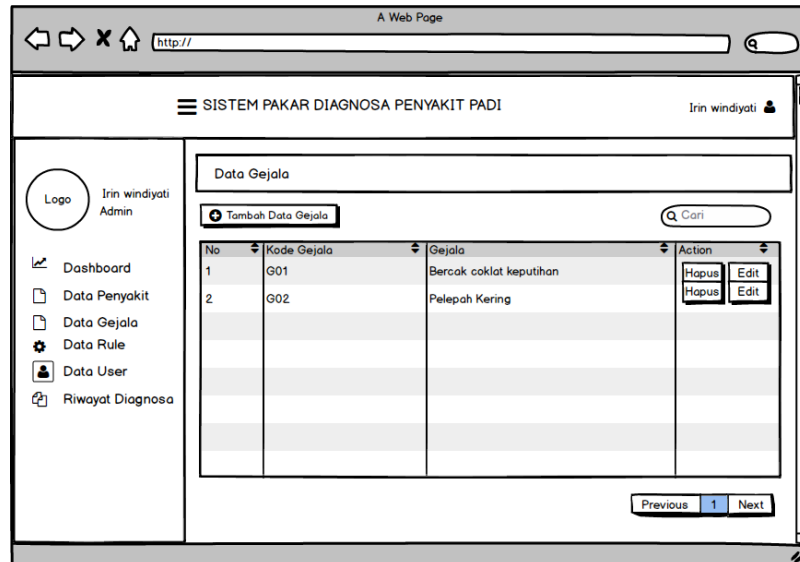
Antarmuka ini berisi data penyakit, pada halaman ini admin dapat menambah, menghapus dan mengedit data penyakit. Adapun rancangan antarmuka data penyakit dapat dilihat pada Gambar 3.17.



Gambar 3.17 Antarmuka Data Penyakit

3.8.2.3 Antarmuka Data Gejala

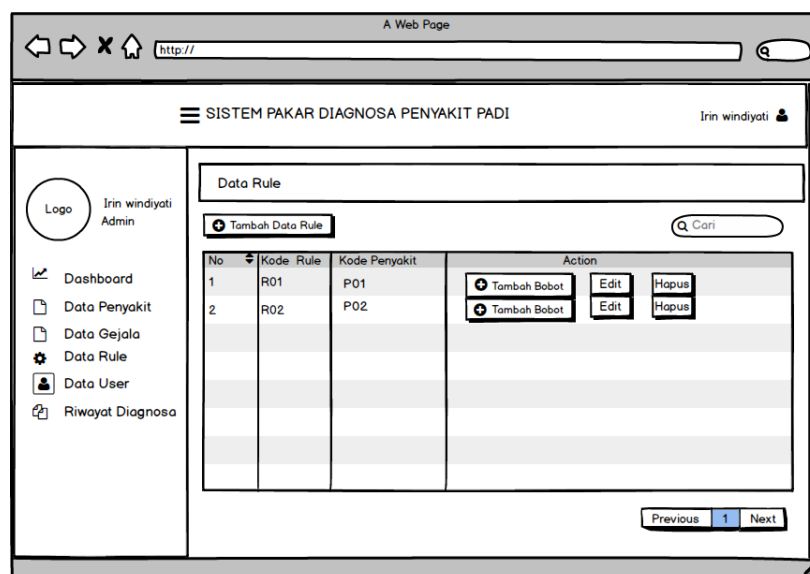
Antarmuka ini berisi data gejala, pada halaman ini admin dapat menambah, menghapus dan mengedit data gejala. Adapun rancangan antarmuka data gejala dapat dilihat pada Gambar 3.18.



Gambar 3.18 Antarmuka Data Gejala

3.8.2.4 Antarmuka Data Rule

Antarmuka ini berisi data *rule* dimana admin membuat dan merelasikan kode rule dengan kode penyakit, pada halaman ini admin dapat menambah, menghapus dan mengedit data *rule*. Adapun rancangan antarmuka data *rule* dapat dilihat pada Gambar 3.19.



Gambar 3.19 Antarmuka Data Rule

3.8.2.5 Antarmuka Tambah Bobot Rule

Antarmuka ini digunakan ketika admin menambahkan pembobotan gejala dari rule yang dibuat. Adapun rancangan antarmuka tambah bobot *rule* dapat dilihat pada Gambar 3.20.

No	Kode Gejala	Gejala	Bobot	Nilai Bobot
1	G01	Bercak Coklat Keputihan	<input checked="" type="checkbox"/>	0.75
2	G02	Pelepah Kering	<input type="checkbox"/>	

Gambar 3.20 Antarmuka Tambah Bobot *Rule*

3.8.2.6 Antarmuka *Edit Bobot Rule*

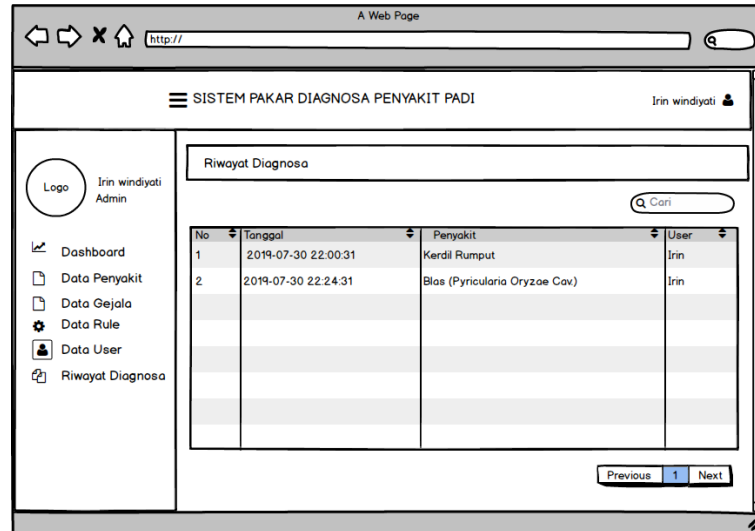
Antarmuka ini digunakan ketika admin mengedit pembobotan gejala dari *rule* yang dibuat. Adapun rancangan antarmuka *edit* bobot *rule* dapat dilihat pada Gambar 3.21.

No	Kode Gejala	Gejala	Validasi	Keterangan	Nilai Bobot
1	G01	Bercak Coklat Keputihan	Valid		0.75
2	G02	Pelepah Kering	Tidak Valid	Bukan gejala penya	0.9

Gambar 3. 21 Antarmuka *Edit Bobot Rule*

3.8.2.7 Antarmuka Riwayat Diagnosa

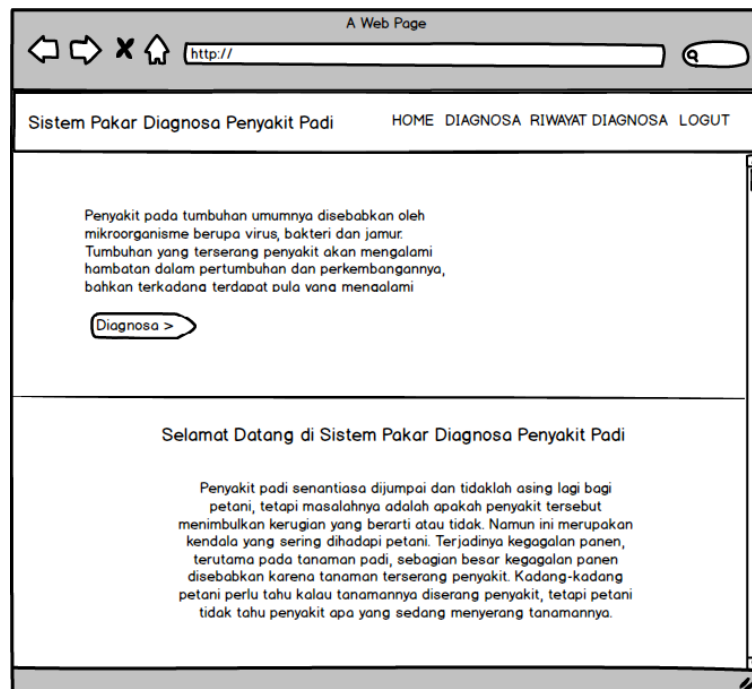
Antarmuka ini menampilkan riwayat diagnosa *user*. Adapun rancangan antarmuka data riwayat diagnosa dapat dilihat pada Gambar 3.22.



Gambar 3.22 Antarmuka Riwayat Diagnosa

3.8.3 Antarmuka User

Antarmuka ini ditampilkan ketika user berhasil login, pada antarmuka *user* terdapat *menu home*, diagnosa dan riwayat diagnosa. Adapun rancangan antarmuka *user* dapat dilihat pada Gambar 3.23.



Gambar 3.23 Antarmuka User

3.8.3.1 Antarmuka Diagnosa

Antarmuka ini ditampilkan ketika *user* akan melakukan diagnosa penyakit. Adapun rancangan antarmuka diagnosa penyakit dapat dilihat pada Gambar 3.24.

No	Gejala	Pilihan
1	Pelepah kering	<input type="radio"/> Ya <input type="radio"/> Tidak
2	Bercak pada biji	<input type="radio"/> Ya <input type="radio"/> Tidak
3	Bercak pada malai	<input type="radio"/> Ya <input type="radio"/> Tidak
4	Tanaman menjadi kerdil	<input type="radio"/> Ya <input type="radio"/> Tidak
5	Pertumbuhan tidak normal	<input type="radio"/> Ya <input type="radio"/> Tidak

Gambar 3.24 Antarmuka Diagnosa

3.8.3.2 Halaman Riwayat Diagnosa

Antarmuka ini berisi riwayat diagnosa penyakit yang dilakukan *user*. Adapun rancangan antarmuka riwayat diagnosa dapat dilihat pada Gambar 3.25.

NO	Tanggal	Penyakit	Aksi
1	2019-07-30 22:24:31	Kerdil Rumput	<button>Cetak</button>
2	2019-08-03 14:26:30	Blas (Pyricularia Oryzae Cav)	<button>Cetak</button>
3	2019-07-30 22:24:31	Kerdil Rumput	<button>Cetak</button>
4	2019-08-03 14:33:45	Blas (Pyricularia Oryzae Cav)	<button>Cetak</button>
5	2019-07-30 22:24:31	Kerdil Rumput	<button>Cetak</button>

Gambar 3.25 Antarmuka Riwayat Diagnosa

3.8.4 Antarmuka Pakar

Antarmuka ini ditampilkan ketika pakar berhasil login, pada antarmuka pakar terdapat dua menu yaitu menu validasi data gejala dan menu data gejala.

3.8.4.1 Antarmuka Validasi Data Gejala

Antarmuka ini ditampilkan ketika pakar akan melakukan validasi data gejala dimana pakar dapat memberi validasi data dari data yang telah diinputkan oleh admin. Adapun rancangan antarmuka diagnosa penyakit dapat dilihat pada Gambar 3.26.

No.	Penyakit	Gejala	Validasi	Keterangan
1	Blas	Bercak Coklat Keputihan	<input checked="" type="radio"/> Valid <input type="radio"/> Tidak Valid	Bukan gejala Penyakit Blas
2	Blas	Pelepah Kering	<input type="radio"/> Valid <input checked="" type="radio"/> Tidak Valid	

Gambar 3.26 Antarmuka Data Gejala

3.8.4.2 Antarmuka Data Gejala

Pada antarmuka ini pakar dapat melihat relasi gejala dengan penyakit yang telah tervalidasi. Adapun rancangan antarmuka diagnosa penyakit dapat dilihat pada Gambar 3.27.

No.	Penyakit	Gejala
1	Blas	Pelepah Kering
2	Blas	Bercak Coklat Keputihan

Gambar 3.27 Antarmuka Data Gejala

3.9 Perancangan Sistem Pakar

Sistem pakar yang dibangun merupakan *rule* yang menerapkan metode *Certainty Factor* yang digunakan untuk menghitung faktor kepastian dari gejala-gejala penyakit padi. Adapun logika metode *Certainty Factor* pada sesi diagnosa sistem, pengguna diberi pilihan jawaban yang masing masing memiliki bobot, nilai bobot 1 untuk jawaban ‘Ya’, nilai bobot 0 untuk jawaban ‘Tidak’.

Tabel 3.18 Tabel Nilai Bobot User

No	Keterangan	Nilai Bobot User
1	Ya	1
2	Tidak	2

Proses perhitungan persentase keyakinan diawali dengan pemecahan sebuah kaidah-kaidah yang memiliki premis majemuk menjadi kaidah-kaidah yang memiliki premis tunggal. Kemudian masing-masing aturan baru dihitung *Certainty Factor*-nya, sehingga diperoleh nilai *Certainty Factor* untuk masing-masing aturan, kemudian nilai *CF* tersebut dikombinasikan. Sebagai contoh, proses pemberian bobot pada setiap premis (gejala) hingga memperoleh keyakinan untuk penyakit padi.

Pada sistem pakar diagnosa penyakit padi dengan metode *Certainty Factor* ini terdapat 5 rule :

1. R01 untuk rule dari penyakit Blas (*Pyricularia Oryzae* Cav.)
2. R02 untuk rule dari penyakit Hawar Pelepah daun dan busuk Batang (*Rhizoctonia solani kuhn*)
3. R03 untuk rule dari penyakit Kerdil rumput (*Grassy stunt*)
4. R04 untuk rule dari penyakit Tungro
5. R04 untuk rule dari penyakit Hawar daun bakteri (HDB) (*Xanthomonas campestris pv.oryzae.*)

Berikut ini merupakan kaidah atau *rule* yang berkaitan dengan sistem pakar diagnosa penyakit padi dengan metode *Certainty Factor*, diantaranya sebagai berikut:

- R01 : **IF** bercak coklat keputihan, **AND** pelepah kering, **AND** bercak pada malai, **AND** bercak pada biji, **AND** bulir padi hampa (kosong), **AND** daun busuk yang dimulai dengan adanya bercak berbentuk belahketupat kemudian bercak maeluas menuruti urat tulang daun, kadang-kadang beberapa bercak daun bergabung menjadi satu seperti terbakar (malai belum keluar), **AND** pangkal batang tanaman mengkerut, berwarna coklat kehitaman dan mudah rebah, **THEN** Blas (*Pyricularia Oryzae* Cav.).
- R02 : **IF** pelepah daun terlihat bercak basah berbentuk bulat, bercak membesar dengan bagian tengah berwarna abu-abu dan bagian tepi berwarna coklat, **AND** bercak abu kehijauan pada pelepah daun dekat permukaan air, **AND** tanaman mati, **THEN** Hawar Pelepah daun dan busuk Batang (*Rhizoctonia solani kuhn*).
- R03 : **IF** tanaman menjadi kerdil, **AND** Pertumbuhan tidak normal, **AND** daun-daun memendek, menyempit, kaku, **AND** warna daun hijau kekuningan dipenuhi bercak seperti karat, **THEN** Kerdil rumput (*Grassy stunt*).
- R04 : **IF** warna daun menjadi kuning sampai coklat yang dimulai dari ujung daun, **AND** pembentukan dan perkembangan akar terhambat, **AND** pembentukan bunga tertunda, **AND** bercak pada daun warna hijau pucat, **AND** Waktu panen tertunda, **THEN** Tungro.
- R05 : **IF** tepi daun terdapat garis gelombang berwarna kuning, **AND** pelepah daun menguning, **AND** daun menjadi hijau kelabu dan menggulung dibagian ujung dan tepi daun, **AND** terdapat bercak kuning pada daun yang dimulai dari ujung daun kemudian menjalar ke bawah **THEN** Hawar daun bakteri (HDB) (*Xanthomonas campestris pv.oryzae*.)

Langkah pertama, pakar menentukan nilai *CF* dari masing-masing gejala. Lihat Tabel 3.1.

Langkah kedua, penentuan nilai bobot user berdasarkan gejala yang dipilih. Lihat tabel 3.14.

Contoh : pengguna memilih gejala pelepah kering, bercak pada malai, bulir padi hampa (kosong), daun busuk yang dimulai dengan adanya bercak berbentuk belah ketupat kemudian bercak meluas mengikuti urat tulang daun, kadang-kadang beberapa bercak daun bergabung menjadi satu seperti terbakar (malai belum keluar).

Langkah ketiga, kemudian hitung nilai CF nya dengan mengalikan nilai CF dengan nilai bobot user, dapat dilihat pada persamaan (2.1)

$$CF[H,E]_1 = 1 * 0.10 = 0.10$$

$$CF[H,E]_2 = 1 * 0.50 = 0.50$$

$$CF[H,E]_3 = 1 * 0.50 = 0.50$$

$$CF[H,E]_4 = 1 * 0.75 = 0.75$$

Langkah keempat adalah mengkombinasikan nilai CF dari masing-masing langkah, berikut adalah kombinasinya, dapat dilihat pada persamaan (2.2).

$$CFCombine\ CF[H,E]_{1,2} = 0.10 + 0.50 * (1 - 0.10) = 0.55$$

$$CFCombine\ CF[H,E]_{com2} = 0.55 + 0.50 * (1 - 0.55) = 0.775$$

$$CFCombine\ CF[H,E]_{com3} = 0.775 + 0.75 * (1 - 0.775) = 0.8875$$

Langkah kelima yaitu $CFCombine\ CF[H,E]_{com4}$ dikalikan dengan 100%.

Perhitungannya sebagai berikut :

$$CFCombine\ CF[H,E]_{com3} * 100\% = 88,75\%$$

Jadi, hasil dari perhitungan persamaan diatas menunjukkan bahwa nilai kepastian dari gejala yang telah dipilih yaitu penyakit Blas (*Pyricularia Oryzae* Cav.).

BAB IV

HASIL DAN PEMBAHASAN

4.1 Hasil

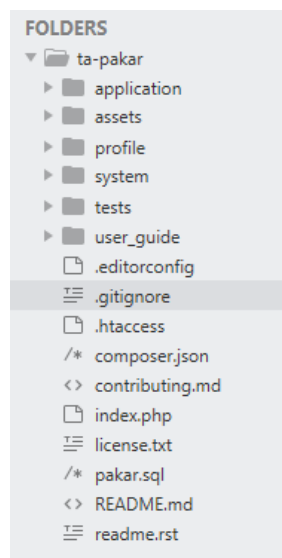
Penelitian ini menghasilkan sebuah sistem pakar untuk mendiagnosa penyakit padi. Sistem pakar ini digunakan untuk membantu mengatasi permasalahan yang terjadi di masyarakat terutama para petani padi dalam menganalisa penyakit yang menyerang padi mereka.

Penulis dapat menyimpulkan sistem pakar diagnosa penyakit padi dengan metode *Certainty Factor* yang telah dibuat adalah sistem yang digunakan untuk media konsultasi bagi masyarakat khususnya petani padi untuk mengetahui serta menangani penyakit yang menyerang padi mereka.

4.2 Pembahasan

Pada bagian ini merupakan tahap pembahasan dari hasil implementasi yang penulis kumpulkan, analisa dan rancang sehingga terciptalah sebuah sistem pakar diagnosa penyakit padi dengan metode *Certainty Factor*.

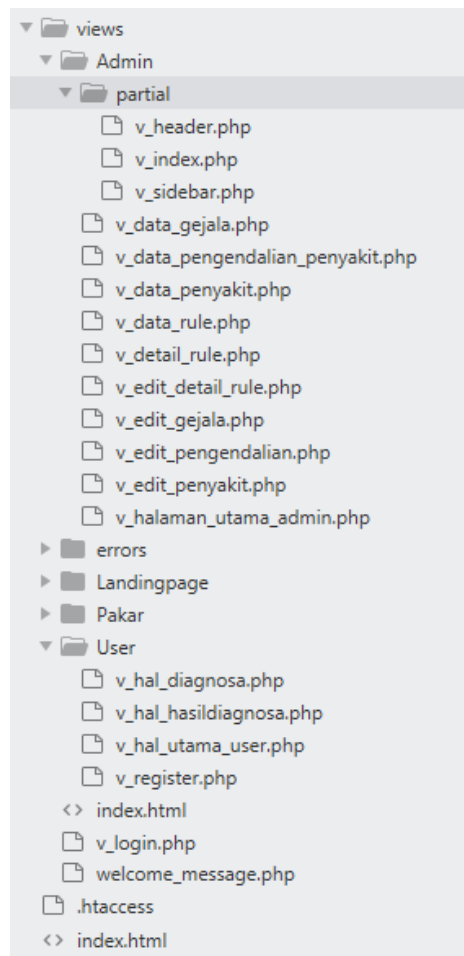
4.2.1 Struktur Direktori



Gambar 4.1 Struktur Direktori *Folder Ta-pakar*

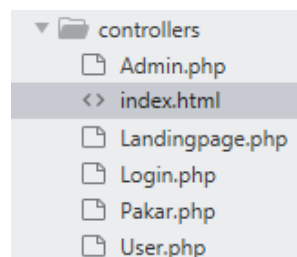
Pada Gambar 4.1 terdapat struktur direktori folder ta-pakar yang digunakan untuk membuat sistem pakar diagnosa penyakit padi dengan metode *Certainty*

Factor yang terdiri dari folder *application*, *assets*, *profile*, *system*, *tests*, dan *user_guide*.



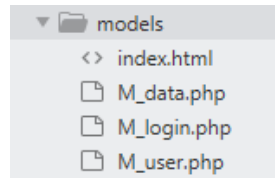
Gambar 4.2 Struktur Direktori *Folder Views*

Pada Gambar 4.2 terdapat struktur direktori *folder view* yang digunakan untuk membuat tampilan dari sistem yang dibuat. Pada *folder view* terdiri dari *folder Admin*, *errors*, *landing page*, *pakar* dan *user*. *Folder Admin* terdiri dari *folder partial* yang didalamnya terdapat file *v_header.php*, *v_index.php*, *v_sidebar.php*.



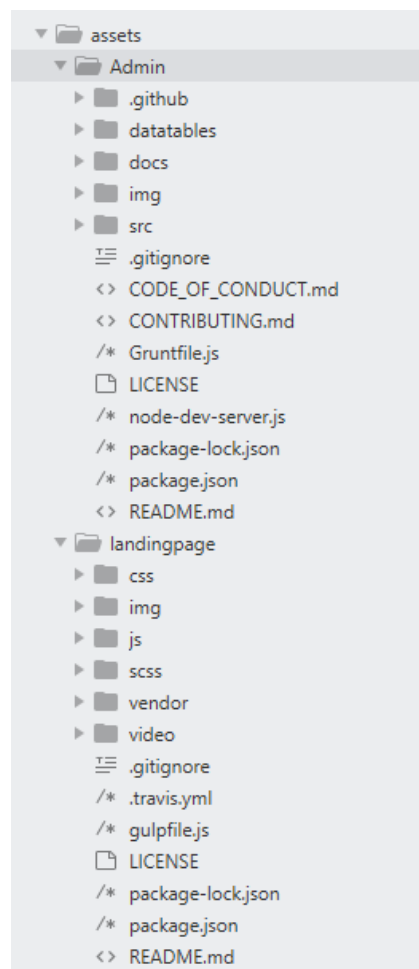
Gambar 4.3 Struktur Direktori *Folder Controller*

Pada Gambar 4.3 terdapat struktur direktori *folder controller* yang digunakan untuk membuat fungsi serta mengakses *view* dan *model*. Pada controller tersebut terdiri dari *file controller admin.php, landingpage.php, login.php, pakar.php, dan user.php*.



Gambar 4.4 Struktur Direktori *Folder Models*

Pada gambar 4.4 terdapat struktur direktori *folder models* yang digunakan untuk membuat query database dari setiap fungsi yang dibuat. Pada *models* tersebut terdiri dari *file model M_data.php, M_login.php, dan M_user.php*.



Gambar 4.5 Struktur Direktori Folder Assets

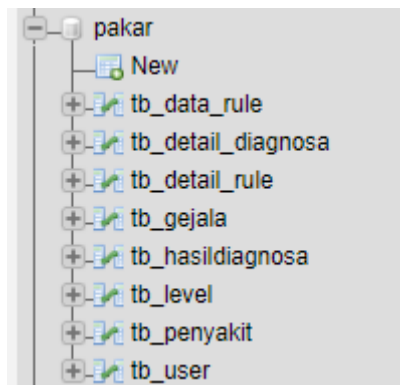
Pada Gambar 4.5 terdapat struktur direktori folder assets yang digunakan untuk menyimpan template tampilan. Pada *folder asses* tersebut terdiri dari *folder* admin dan *folder landingpage*. Pada *folder* admin terdiri dari *folder github*, *datatables*, *docs*, *img* dan *src*. Pada *folder landingpage* terdapat *folder css*, *img*, *js*, *scss*, *vendor* dan *video*.

4.2.2 Impementasi Database

Database sistem pakar diagnosa penyakit padi terdiri dari 8 tabel yaitu tabel *tb_penyakit*, tabel *tb_gejala*, tabel *tb_data_rule*, tabel *tb_detail_rule*, tabel *tb_hasil_diagnosa*, tabel *tb_detail_hasildiagnosa*, tabel *tb_user* dan tabel *tb_level*. Berikut implementasi pembuatan *database* untuk aplikasi sistem pakar diagnosa penyakit padi dengan metode *Certainty Factor*.

1. Database pakar

Database pakar merupakan nama *database* yang dibuat untuk menyimpan tabel-tabel *database* yang telah digunakan. *Database* pakar dapat dilihat pada Gambar 4.6.




Gambar 4.6 Database Pakar

Pada *database* pakar terdiri dari *tb_data_rule*, *tb_detail_diagnosa*, *tb_detail_rule*, *tb_gejala*, *tb_hasildiagnosa*, *tb_level*, *tb_penyakit*, dan *tb_user*.

2. Tabel tb_penyakit


Tabel ini digunakan untuk menyimpan data penyakit padi. Tabel *tb_penyakit* dapat dilihat pada Gambar 4.7. Pada tabel *tb_penyakit* terdiri dari *field* *kd_penyakit*, *penyakit*, *deskripsi*, dan *kendali*.

#	Nama	Jenis	Penyortiran
1	kd_penyakit 	varchar(50)	latin1_swedish_ci
2	penyakit	text	latin1_swedish_ci
3	deskripsi	text	latin1_swedish_ci
4	kendali	text	latin1_swedish_ci

Gambar 4.7 Tabel Tb_penyakit

3. Tabel tb_gejala


Tabel ini digunakan untuk menyimpan data gejala dari penyakit. Tabel tb_gejala dapat dilihat pada Gambar 4.8. Pada tabel tb_gejala terdiri dari *field* kd_gejala dan gejala.

#	Nama	Jenis	Penyortiran
1	kd_gejala 	varchar(50)	latin1_swedish_ci
2	gejala	varchar(250)	latin1_swedish_ci

Gambar 4.8 Tabel Tb_gejala

4. Tabel tb_data_rule

Tabel ini digunakan untuk menyimpan data *rule*. Tabel tb_data_rule dapat dilihat pada Gambar 4.9. Pada tabel tb_data_rule terdiri dari *field* kd_rule dan kd_penyakit.

#	Nama	Jenis	Penyortiran
1	kd_rule 	varchar(11)	latin1_swedish_ci
2	kd_penyakit	varchar(25)	latin1_swedish_ci

Gambar 4.9 Tabel Tb_data_rule

5. Tabel tb_detail_rule

Tabel ini digunakan untuk menyimpan data detail dari *rule* yang telah buat sebelumnya, tabel ini berisi pembobotan dari tiap gejala. Tabel tb_detail_rule dapat dilihat pada Gambar 4.10. Pada tabel tb_detail_rule terdiri dari *field* kd_detail_rule kd_rule, kd_gejala, dan bobot.

#	Nama	Jenis	Penyortiran
1	kd_detail_rule	int(50)	
2	kd_rule	varchar(50)	latin1_swedish_ci
3	kd_gejala	varchar(50)	latin1_swedish_ci
4	bobot	varchar(50)	latin1_swedish_ci

Gambar 4.10 Tabel Tb_detail_rule

6. Tabel tb_hasildiagnosa

Tabel ini digunakan untuk menyimpan data hasil diagnosa penyakit yang dilakukan oleh *user*. Tabel tb_hasil_diagnosa dapat dilihat pada Gambar 4.11. Pada tabel tb_hasildiagnosa terdiri dari *field* id_diagnosa, presentase, id_user, kd_rule, tanggal, dan kd_penyakit.

#	Nama	Jenis	Penyortiran
1	id_diagnosa	int(11)	
2	presentase	varchar(50)	latin1_swedish_ci
3	id_user	int(11)	
4	kd_rule	varchar(11)	latin1_swedish_ci
5	tanggal	timestamp	
6	kd_penyakit	varchar(25)	latin1_swedish_ci

Gambar 4.11 Tabel Tb_hasildiagnosa

7. Tabel tb_detail_diagnosa

Tabel ini digunakan untuk menyimpan data detail dari hasil diagnosa. Tabel tb_detail_diagnosa dapat dilihat pada Gambar 4.12. Pada tabel tb_detail_diagnosa terdiri dari *field* id_detail_diagnosa, id_diagnosa, id_detail_rule dan jb_user.

#	Nama	Jenis	Penyortiran
1	id_detail_diagnosa	int(11)	
2	id_diagnosa	int(11)	
3	id_detail_rule	int(11)	
4	jb_user	int(11)	

Gambar 4.12 Tabel Tb_detail_diagnosa

8. Tabel *tb_user*

Tabel ini digunakan untuk menyimpan data detail dari hasil diagnosa. Tabel *tb_detail_diagnosa* dapat dilihat pada Gambar 4.13. Pada tabel *tb_user* terdiri dari *field* *id_user*, *username*, *password*, *level*, *nama*, *jk*, dan *alamat*.

#	Nama	Jenis	Penyortiran
1	id_user 🔑	int(11)	
2	username	varchar(50)	latin1_swedish_ci
3	password	varchar(50)	latin1_swedish_ci
4	level	varchar(50)	latin1_swedish_ci
5	nama	varchar(50)	latin1_swedish_ci
6	jk	varchar(50)	latin1_swedish_ci
7	alamat	varchar(60)	latin1_swedish_ci

Gambar 4.13 Tabel *Tb_user*

9. Tabel *tb_level*

Tabel ini digunakan untuk menyimpan data *level* dari tiap user. Tabel *tb_level* dapat dilihat pada Gambar 4.14. Pada tabel *tb_level* terdiri dari *field* *id_level* dan *level*.

#	Name	Type	Collation
1	id_level 🔑	int(11)	
2	level	varchar(50)	latin1_swedish_ci

Gambar 4.14 Tabel *Tb_level*

4.2.3 Implementasi Desain Antarmuka (*Interface*)

Antarmuka atau *interface* merupakan cara untuk memudahkan pengguna dalam melakukan aktivitas dalam sistem. Terdapat empat pengelompokan antarmuka, yaitu pertama antarmuka untuk *landing page*, antarmuka ini digunakan pengunjung sistem saat pertama kali mengakses sistem dan tidak melakukan aksi login, antarmuka kedua yaitu admin, antarmuka ini digunakan admin untuk mengelola data pakar mulai dari membah, mengedit sampai menghapus data dari sistem, antarmuka ketiga yaitu antarmuka user, antarmuka ini digunakan user untuk melakukan diagnosa penyakit dan antarmuka keempat adalah antarmuka pakar, antarmuka ini digunakan pakar untuk melakukan validasi data yang telah diinputkan oleh admin.

4.2.3.1 Antarmuka *Landing Page*

Pada antarmuka *landing page* ditampilkan ketika pengunjung sistem pakar belum melakukan aksi *login*, pada antarmuka ini terdapat *menu home*, *menu info penyakit*, *menu petunjuk* yaitu *menu* untuk melihat petunjuk melakukan diagnosa, *register* yaitu *menu* untuk membuat akun bagi *user* baru yang akan melakukan diagnosa dan *menu login* yaitu *menu* untuk keluar akun dari sistem. Antarmuka *landingpage* dapat dilihat pada Gambar 4.15.



Gambar 4.15 Antarmuka *Landing Page*

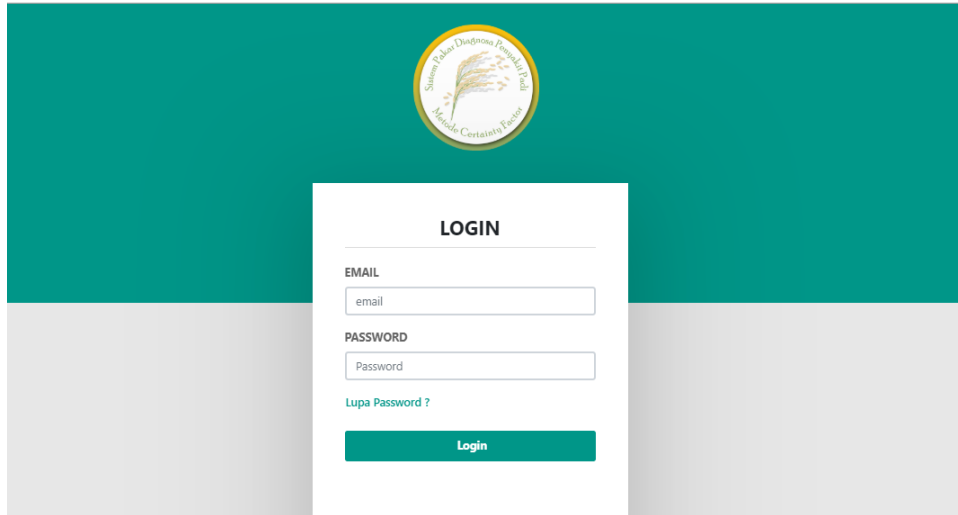
1. Antarmuka *Register*

Halaman ini merupakan halaman untuk membuat akun bagi user baru yang akan melakukan diagnosa penyakit, pada halamn ini user harus mengisi form *registrasi* yang telah disediakan oleh sistem, Antarmuka *landingpage* dapat dilihat pada Gambar 4.16.

Gambar 4.16 Antarmuka *Register*

2. Antarmuka *Login*

Halaman ini merupakan halaman untuk masuk ke sistem bagi *user* yang telah memiliki akun baik admin, *user*, maupun pakar. Pada halaman ini *user* harus mengisi *form login* yang telah disediakan oleh sistem, antarmuka *login* dapat dilihat pada Gambar 4. 17.



LOGIN

EMAIL

email

PASSWORD

Password

[Lupa Password ?](#)

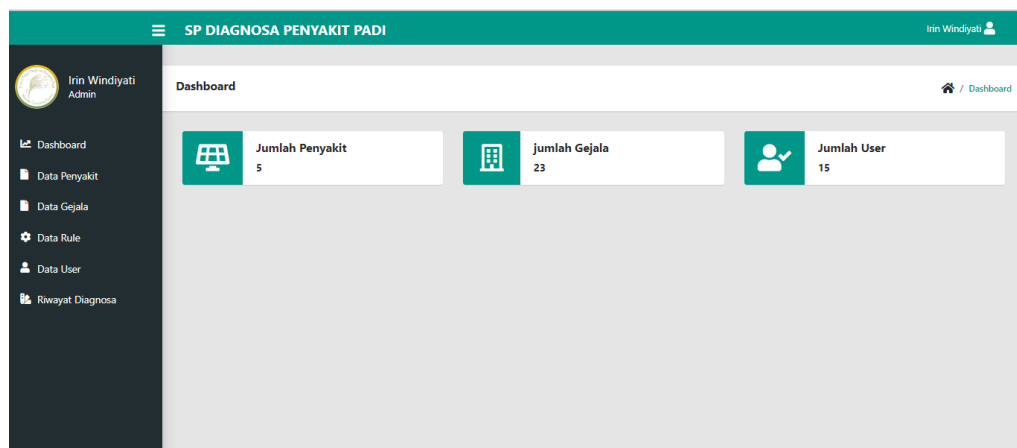
Login

Gambar 4.17 Antarmuka *Login*

4.2.3.2 Antarmuka Admin

Pada antarmuka admin terdapat *menu dashboard*, *menu data penyakit*, *menu data gejala*, *menu data rule*, *menu data user* dan riwayat diagnosa.

1. Antarmuka *Dashboard*



Gambar 4.18 Antarmuka *Dashboard*

Pada antarmuka dashboard admin dapat melihat jumlah penyakit, jumlah gejala dan jumlah user yang memiliki akun pada sistem pakar diagnosa penyakit padi ini. Antarmuka *dashboard* dapat dilihat pada Gambar 4.18.

2. Antarmuka Data Penyakit

No	Kode Penyakit	Nama Penyakit	Deskripsi	Pengendalian	Action
1	P01	Blas (Pyricularia Oryzae Cav.)	Penyakit blas disebabkan oleh jamur Pyricularia grisea. Awalnya penyakit ini berkembang di pertanaman padi gogo, tetapi akhir-akhir ini sudah menyebar di lahan sawah irigasi. Pada fase pertumbuhan generatif tanaman padi, gejala penyakit blas berkembang pada tangkai/leher malai disebut blas leher.	Penyakit blas disebabkan oleh jamur Pyricularia grisea. Awalnya penyakit ini berkembang di pertanaman padi gogo, tetapi akhir-akhir ini sudah menyebar di lahan sawah irigasi. Pada fase pertumbuhan generatif tanaman padi, gejala penyakit blas berkembang pada tangkai/leher malai disebut blas leher.	Hapus Edit
2	P02	Hawar Pelepah daun dan busuk Batang	Penyakit hawar daun Padi atau disebut juga sebagai busuk pelepah atau dikenal juga sebagai sheath blight yang disebabkan oleh jamur Rhizoctonia solani khun, adalah penyakit padi yang serius menyerang daun pelepah. Penyakit ini tergolong jamur tanah bersifat sebagai parasit	Disamping memperhatikan faktor2 penyebab berkembangnya penyakit ini, maka penggunaan fungisida yang berifat mencegah dan mengobatinya perlu dilakukan pada saat umur tanaman mulai	Hapus Edit

Gambar 4.19 Antarmuka Data Penyakit

Pada antarmuka data penyakit admin dapat menambahkan data penyakit, mengedit data penyakit, menghapus data penyakit dan mencari data penyakit yang telah ditambahkan sebelumnya. Antarmuka data penyakit dapat dilihat pada Gambar 4.19.

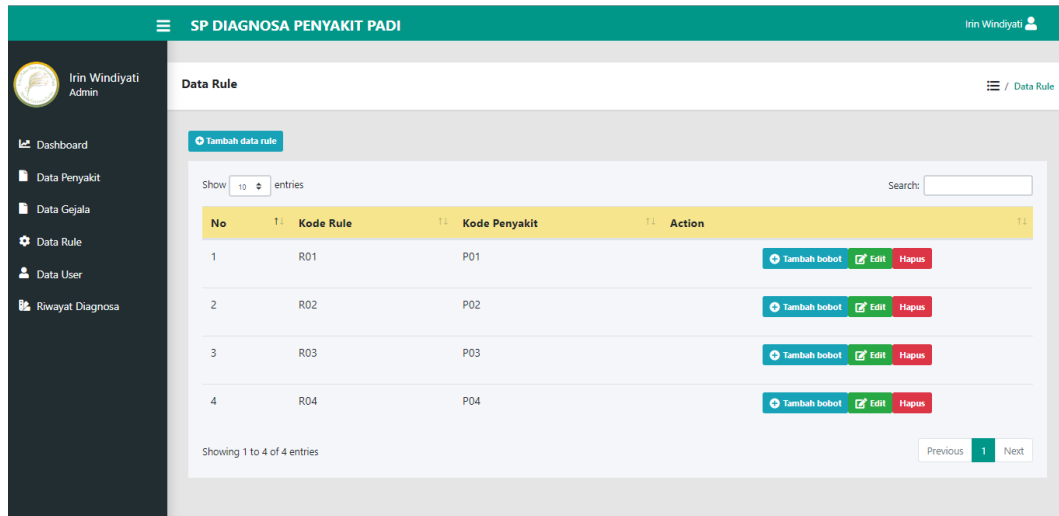
3. Antarmuka Data Gejala

No	Kode Gejala	Gejala	Action
1	G01	Bercak coklat keputihan	Hapus Edit
2	G02	Pelepah kering	Hapus Edit
3	G03	Bercak pada malai	Hapus Edit
4	G04	Bercak pada biji	Hapus Edit
5	G05	Bulir padi hampa (kosong)	Hapus Edit
6	G06	Daun busuk yang dimulai dengan adanya bercak berbentuk belahketupat kemudian bercak maeluas menurut urat tulang	Hapus Edit

Gambar 4.20 Antamuka Data Gejala

Pada antarmuka data penyakit admin dapat menambahkan data gejala, mengedit data penyakit, menghapus data gejala dan mencari data gejala yang telah ditambahkan sebelumnya. Antarmuka data penyakit dapat dilihat pada Gambar 4.20.

4. Antarmuka Data Rule



Gambar 4.21 Antarmuka Data Rule

Pada antarmuka data *rule* admin dapat menambahkan data *rule*, mengedit data penyakit, menghapus data *rule* dan mencari data *rule* yang telah ditambahkan sebelumnya. Antarmuka data penyakit dapat dilihat pada Gambar 4.21.

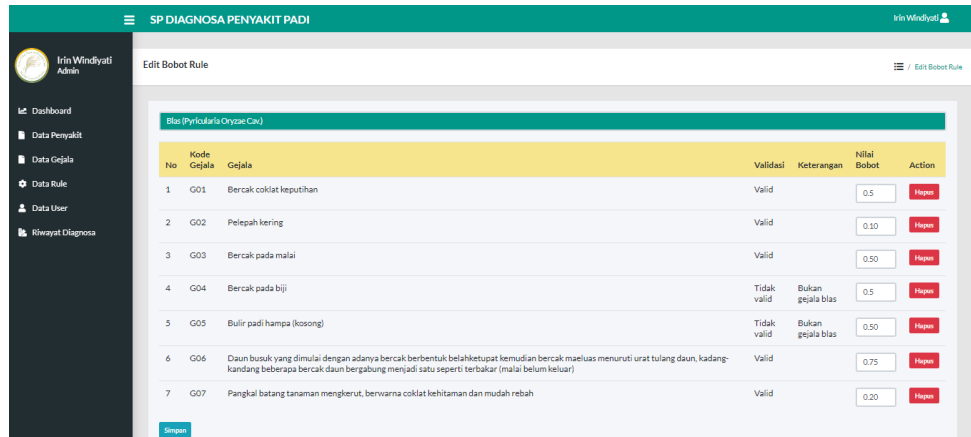
5. Antarmuka Tambah Bobot



Gambar 4.22 Antamuka Tambah Bobot

Pada antarmuka tambah bobot admin dapat menambahkan bobot masing-masing gejala dari tiap *rule* yang sebelumnya sudah ditambahkan. Antarmuka tambah bobot dapat dilihat pada Gambar 4.22.

6. Antarmuka Edit Bobot



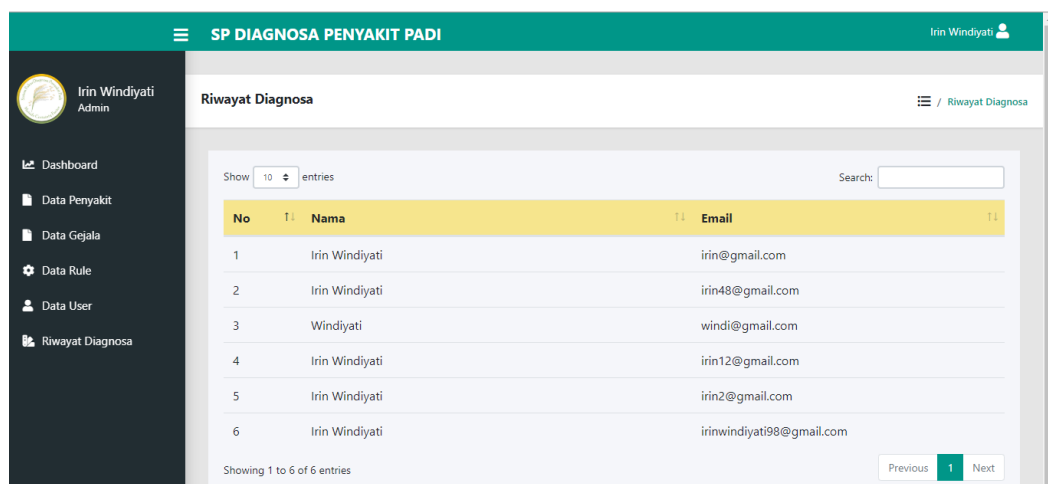
No	Kode Gejala	Gejala	Validasi	Keterangan	Nilai Bobot	Action
1	G01	Bercak coklat keputihan	Valid		0.5	Hapus
2	G02	Pelepah kering	Valid		0.10	Hapus
3	G03	Bercak pada malai	Valid		0.50	Hapus
4	G04	Bercak pada biji	Tidak valid	Bukan gejala bias	0.5	Hapus
5	G05	Bulir padi hampa (kosong)	Tidak valid	Bukan gejala bias	0.50	Hapus
6	G06	Daun busuk yang dimulai dengan adanya bercak berbentuk belahketupat kemudian bercak meluas menurut urat tulang daun, kadang-kadang beberapa bercak daun bergabung menjadi satu seperti terbakar (malai belum keluar)	Valid		0.75	Hapus
7	G07	Pangkal batang tanaman mengkerut, berwarna coklat kehitaman dan mudah rebah	Valid		0.20	Hapus

Gambar 4.23 Antarmuka Edit Bobot

Pada antarmuka edit bobot admin dapat mengedit bobot masing-masing gejala dari sebuah *rule*, pada antarmuka ini juga admin juga dapat melihat validasi gejala dari pakar. Antarmuka edit bobot dapat dilihat pada Gambar 4.23.

7. Antarmuka Data User

Pada antarmuka data *user* admin dapat melihat dan mencari data *user* yang telah melakukan registrasi. Antarmuka data *user* dapat dilihat pada Gambar 4.24.



No	Nama	Email
1	Irin Windiyati	irin@gmail.com
2	Irin Windiyati	irin48@gmail.com
3	Windiyati	windi@gmail.com
4	Irin Windiyati	irin12@gmail.com
5	Irin Windiyati	irin2@gmail.com
6	Irin Windiyati	irinwindiyati98@gmail.com

Gambar 4.24 Antarmuka Data User

8. Antarmuka Riwayat Diagnosa



Gambar 4.25 Antarmuka Riwayat Diagnosa

Pada antarmuka ini admin dapat melihat dan mencari data riwayat diagnosa user yang telah melakukan diagnosa penyakit. Antarmuka riwayat diagnosa dapat dilihat pada Gambar 4.25.

4.2.3.3 Antarmuka User

Pada antarmuka *user* terdapat *menu home*, *menu diagnosa*, *menu riwayat diagnosa* dan *menu logout*.

1. Antarmuka Home

Antarmuka *home* adalah antarmuka yang ditampilkan setelah *user* berhasil melakukan aksi *login*. Antarmuka *home* dapat dilihat pada Gambar 4.26.



Gambar 4.26 Antarmuka *Home*

2. Antarmuka Diagnosa

Sistem Pakar Diagnosa Penyakit Padi
HOME DIAGNOSA RIWAYAT DIAGNOSA LOGOUT

Show 10 entries
Search:

No	Gejala	Pilihan
1	Bercak coklat keputihan	<input type="radio"/> Ya <input type="radio"/> Tidak
2	Bercak pada malai	<input type="radio"/> Ya <input type="radio"/> Tidak
3	Bercak pada biji	<input type="radio"/> Ya <input type="radio"/> Tidak
4	Tanaman menjadi kerdil	<input type="radio"/> Ya <input type="radio"/> Tidak
5	Pertumbuhan tidak normal	<input type="radio"/> Ya <input type="radio"/> Tidak
6	Daun-daun memendek, menyempit, kaku	<input type="radio"/> Ya <input type="radio"/> Tidak
7	Warna daun hijau kekuningan dipenuhi bercak seperti karat	<input type="radio"/> Ya <input type="radio"/> Tidak
8	Warna daun menjadi kuning sampai coklat yang dimulai dari ujung daun	<input type="radio"/> Ya <input type="radio"/> Tidak
9	Pembentukan dan perkembangan akar terhambat	<input type="radio"/> Ya <input type="radio"/> Tidak
10	Pembentukan bunga tertunda	<input type="radio"/> Ya <input type="radio"/> Tidak

Showing 1 to 10 of 16 entries

[Previous](#)
[1](#)
[2](#)
[Next](#)

SIMPAN

Gambar 4.27 Antarmuka Diagnosa

Pada antarmuka ini user dapat melakukan diagnosa penyakit dengan mengisi tiap gejala sesuai dengan gejala yang dialami berupa inputan “Ya” atau “tidak”, antarmuka diagnosa dapat dilihat pada Gambar 4.27.

3. Antarmuka Hasil Diagnosa

Pada antarmuka ini user dapat melihat hasil diagnosa penyakit yang telah dilakukan sebelumnya, antarmuka diagnosa dapat dilihat pada Gambar 4.28.

Sistem Pakar Diagnosa Penyakit Padi
HOME DIAGNOSA RIWAYAT DIAGNOSA LOGOUT

Hasil Diagnosa

Nama	Tanggal	Penyakit	Deskripsi	Kendali	Persentase
Irin Windiyati	2019-08-11 22:22:16	Hawar daun bakteri (HDB) (Xanthomonas campestris pv.oryzae.)	Penyakit hawar daun bakteri (HDB) merupakan salah satu penyakit padi utama yang tersebar di berbagai ekosistem padi di negara-negara penghasil padi, termasuk di Indonesia. Penyakit disebabkan oleh bakteri Xanthomonas oryzae pv. oryzae (Xoo). Patogen ini dapat menginfeksi tanaman padi pada semua fase pertumbuhan tanaman dari mulai penanaman sampai menjelang panen.	Pupuk Nitrogen berkorelasi positif dengan keparahan penyakit HDB. Artinya pertanaman yang dipupuk Nitrogen dengan dosis tinggi menyebabkan tanaman menjadi lebih rentan dan keparahan penyakit lebih tinggi. Sebaliknya dengan pupuk Kalium menyebabkan tanaman menjadi lebih tahan terhadap penyakit hawar daun bakteri. Oleh karena itu agar perkembangan penyakit dapat ditekan dan diperoleh produksi yang tinggi disarankan menggunakan pupuk N	96.85 %

Gambar 4.28 Antarmuka Hasil Diagnosa

4. Antarmuka Riwayat Diagnosa

Sistem Pakar Diagnosa Penyakit Padi

HOME DIAGNOSA RIWAYAT DIAGNOSA LOGOUT

Riwayat Diagnosa

Show 10 entries

Search:

No	Tanggal	Penyakit	Aksi
1	2019-08-09 13:07:59	Blas (Pyricularia Oryzae Cav.)	CETAK
2	2019-08-09 13:08:34	Blas (Pyricularia Oryzae Cav.)	CETAK
3	2019-08-11 22:22:16	Blas (Pyricularia Oryzae Cav.)	CETAK
4	2019-08-09 13:07:59	Hawar Pelepah daun dan busuk Batang (Rhizoctonia solani kuhn)	CETAK
5	2019-08-09 13:08:34	Hawar Pelepah daun dan busuk Batang (Rhizoctonia solani kuhn)	CETAK
6	2019-08-11 22:22:16	Hawar Pelepah daun dan busuk Batang (Rhizoctonia solani kuhn)	CETAK
7	2019-08-09 13:07:59	Kerdil rumput (Grassy stunt)	CETAK

Gambar 4.29 Antarmuka Riwayat Diagnosa

Pada antarmuka ini user dapat melihat semua riwayat diagnosa penyakit yang telah dilakukan sebelumnya, antarmuka diagnosa dapat dilihat pada Gambar 4.29.

5. Antarmuka Cetak Hasil Diagnosa

1 / 1

Hasil Diagnosa

Nama Lengkap : Irin Windiyati

Tanggal : 2019-08-11 22:22:16

Penyakit : Hawar daun bakteri (HDB) (Xanthomonas campestris pv.oryzae.)

Persentase : 96.85

Deskripsi : Penyakit hawar daun bakteri (HDB) merupakan salah satu penyakit padi utama yang tersebar di berbagai ekosistem padi di negara-negara penghasil padi, termasuk di Indonesia.

Pengendalian : Pupuk Nitrogen berkorelasi positif dengan keparahan penyakit HDB. Artinya pertanaman yang dipupuk Nitrogen dengan dosis tinggi menyebabkan tanaman menjadi lebih rentan terhadap serangan penyakit HDB.

+

+

-

Gambar 4.30 Antarmuka Cetak Hasil Diagnosa

Pada antarmuka ini user dapat mencetak hasil diagnosa penyakit yang telah dilakukan sebelumnya, antarmuka diagnosa dapat dilihat pada Gambar 4.30.

4.2.3.4 Antarmuka Pakar

1. Antarmuka Validasi Data Pakar

No	Penyakit	Gejala	Validasi	Keterangan
1	Blas (Pyricularia Oryzae Cav.)	Daun busuk yang dimulai dengan adanya bercak berbentuk belahketupat kemudian bercak meluas mengikuti urat tulang daun, kadang-kadang beberapa bercak daun bergabung menjadi satu seperti terbakar (malai belum keluar)	<input type="radio"/> valid <input checked="" type="radio"/> Tidak Valid	
2	Blas (Pyricularia Oryzae Cav.)	Pangkal batang tanaman mengerut, berwarna coklat kehitaman dan mudah rebah	<input checked="" type="radio"/> valid <input type="radio"/> Tidak Valid	
3	Blas (Pyricularia Oryzae Cav.)	Pelepah daun terlihat bercak basah berbentuk bulat, bercak membesar dengan bagian tengah berwarna abu-abu dan bagian tepi berwarna coklat	<input checked="" type="radio"/> valid <input type="radio"/> Tidak Valid	
4	Kerdil rumput (Grassy stunt)	Bercak pada daun warna hijau pucat	<input type="radio"/> valid <input checked="" type="radio"/> Tidak Valid	

Gambar 4.31 Antarmuka Validasi Data Pakar

Pada antarmuka ini pakar dapat memberikan validasi dari suatu gejala penyakit. Antarmuka validasi data pakar dapat dilihat pada Gambar 4.31.

2. Antarmuka Data Pakar

Pada antarmuka ini ditampilkan relasi data antara gejala dengan penyakitnya yang telah divalidasi oleh pakar. Antarmuka data pakar dapat dilihat pada Gambar 4.32.

No	Penyakit	Gejala
1	Blas (Pyricularia Oryzae Cav.)	Bercak coklat keputihan
2	Blas (Pyricularia Oryzae Cav.)	Bercak pada malai
3	Blas (Pyricularia Oryzae Cav.)	Bercak pada biji
4	Kerdil rumput (Grassy stunt)	Tanaman menjadi kerdil
5	Kerdil rumput (Grassy stunt)	Pertumbuhan tidak normal
6	Kerdil rumput (Grassy stunt)	Daun-daun memendek, menyempit, kaku
7	Kerdil rumput (Grassy stunt)	Warna daun hijau kekuningan dipenuhi bercak seperti karat
8	Tungro	Warna daun menjadi kuning sampai coklat yang dimulai dari ujung daun

Gambar 4.32 Antarmuka Data Pakar

4.2.4 Pengujian Sistem

Setelah semua rancangan telah diimplementasikan tahap berikutnya adalah pengujian. Pengujian pada sistem pakar diagnosa penyakit padi ini terdapat pengujian metode yang berisi validasi hasil, pengujian *Black-Box* dan kuisisioner.

4.2.4.1 Validasi Hasil

Tabel 4.1 Validasi Hasil

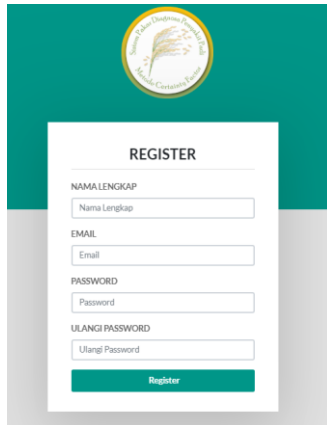
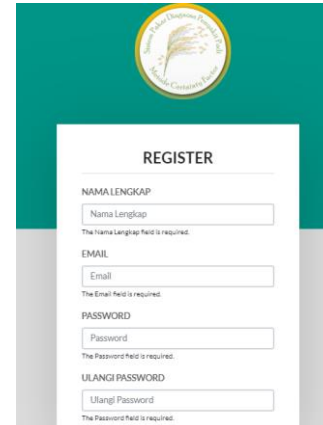
Skema		Hasil	
		Output Sistem	Hitung Manual
<i>Test Rule</i>	R01	98.875 %	98.875 %
	R02	77.5 %	77.5 %
	R03	98.74 %	98.74 %
	R04	98.38 %	98.38 %
	R05	96.85 %	96.85 %
<i>Test Gabungan (Random)</i>	<i>Random 1</i>	95,8% (R03)	95,8% (R03)
	<i>Random 2</i>	98,2% (R04)	98,2% (R04)

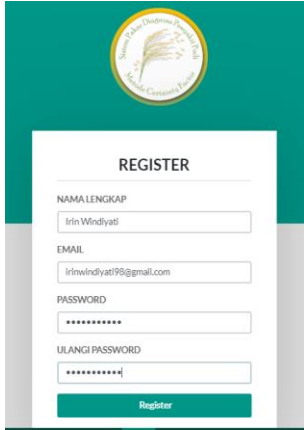

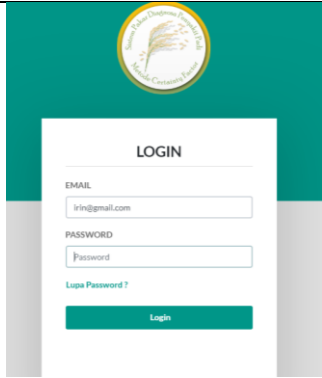
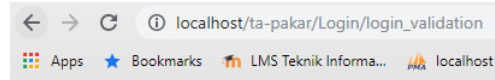
Dari semua hasil skema pengujian baik *test rule* maupun *test gabungan* sistem pakar diagnosa penyakit padi ini berjalan baik dengan akurasi tinggi. Pada skema *test rule* input jawaban *user* pada halaman diagnosa gejala yang dipilih adalah yang sesuai *rule* dari satu kali proses diagnosa baik R01, R02, R03, R04 dan R05, sedangkan pada skema gabungan input jawaban *user* pada halaman diagnosa gejala yang dipilih yaitu secara acak dari satu kali proses diagnosa. data validasi hasil terlampir pada lampiran 3. Validasi hasil sistem ini dapat dilihat pada tabel 4.1.

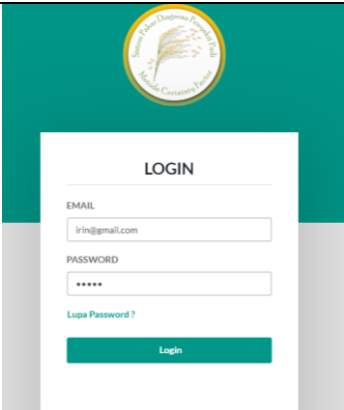
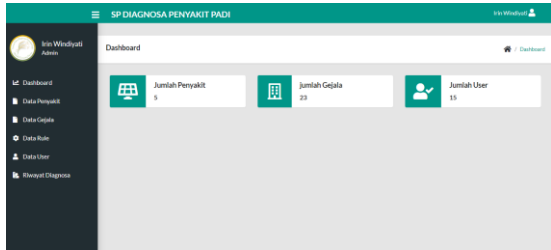
4.2.4.2 Black-Box Testing

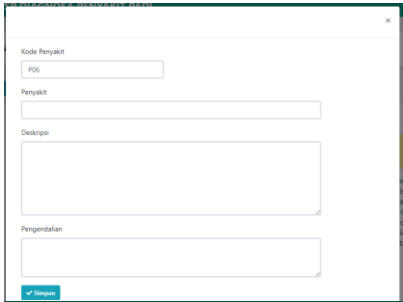
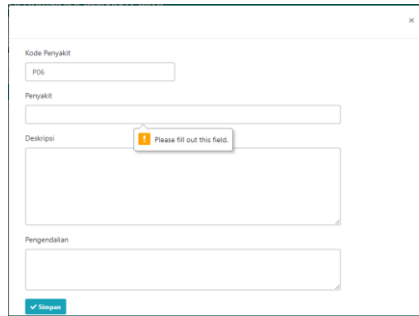
Black Box testing merupakan pengujian program yang mengutamakan pengujian terhadap kebutuhan fungsi suatu program. Tujuan dari metode *Black-Box testing* yaitu untuk menemukan kesalahan fungsi pada *program*. Pengujian dengan menggunakan metode *Black-Box testing* dilakukan hanya dengan mengamati hasil eksekusi melalui data uji dan memeriksa fungsional dari *software* dan fungsionalitasnya tanpa mengetahui yang terjadi dalam proses detail, melainkan hanya mengetahui *input* dan *output*. Pengujian *Black Box* pada sistem ini dapat dilihat pada tabel 4.2.

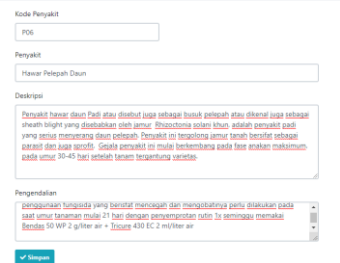

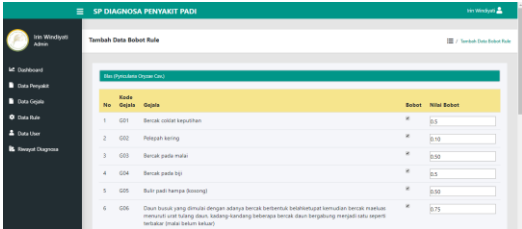

Tabel 4.2 *Black-Box Testing*

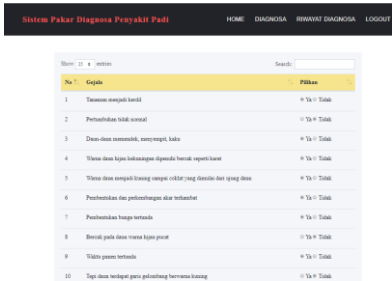

No	Skenario Pengujian	Test Case	Hasil Yang Diharapkan	Hasil Pengujian	Kesimpulan
Menu Register					
1	Mengosongkan kolom pada <i>form</i> lalu langsung klik tombol <i>register</i> .	 <p>The screenshot shows a 'REGISTER' form with the following fields: NAMA LENGKAP (Nama Lengkap), EMAIL (Email), PASSWORD (Password), and ULANGI PASSWORD (Ulangi Password). All fields are empty, and the 'Register' button is visible at the bottom.</p>	Sistem akan menolak registrasi dan antarmuka menampilkan pesan “ <i>field is required</i> ” pada kolom yang tidak terisi.	 <p>The screenshot shows the same 'REGISTER' form, but now with error messages below each empty field: 'The Nama Lengkap field is required.', 'The Email field is required.', 'The Password field is required.', and 'The Password field is required.'.</p>	<i>Valid</i>

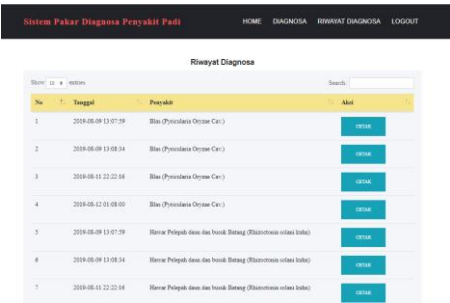
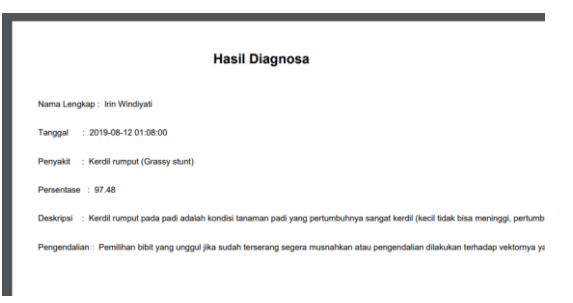
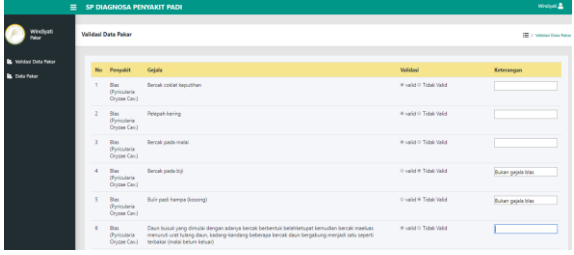
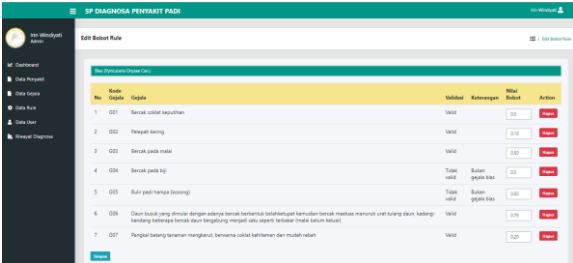
No	Skenario Pengujian	Test Case	Hasil Yang Diharapkan	Hasil Pengujian	Kesimpulan
2	Mengisi seluruh kolom pada form registrasi.		Sistem menyimpan data registrasi kemudian masuk ke <i>home user</i> .		<i>Valid</i>
Menu Login					
3	Mengosongkan salah satu kolom pada form lalu langsung klik tombol login.		Sistem akan menolak <i>login</i> dan antarmuka menampilkan pesan " <i>field is required</i> "	 Username dan password salah !	<i>Valid</i>

No	Skenario Pengujian	Test Case	Hasil Yang Diharapkan	Hasil Pengujian	Kesimpulan
			pada kolom yang tidak terisi.		
4	Mengisi seluruh kolom pada form <i>login</i> .		<p><i>Login validation</i></p> <p>berhasil dan sistem mengarahkan ke <i>dashboard</i> Admin</p>		<i>Valid</i>

No	Skenario Pengujian	Test Case	Hasil Yang Diharapkan	Hasil Pengujian	Kesimpulan
Tambah Data Penyakit					
5	Mengosongkan beberapa kolom pada <i>form</i> lalu langsung klik tombol simpan.		Sistem gagal menyimpan data dan antarmuka menampilkan pesan “ <i>please fill out this field</i> ” pada kolom yang tidak terisi.		<i>Valid</i>

No	Skenario Pengujian	Test Case	Hasil Yang Diharapkan	Hasil Pengujian	Kesimpulan
6	Mengisi seluruh kolom pada form tambah data penyakit.		Data penyakit berhasil ditambahkan.		Valid
Tambah Data Bobot Rule					
7	Memberi tanda centang pada gejala yang akan diberi bobot kemudian mengisi nilai bobot dari gejala tersebut.		Data berhasil ditambahkan.		Valid

No	Skenario Pengujian	Test Case	Hasil Yang Diharapkan	Hasil Pengujian	Kesimpulan
Diagnosa Penyakit					
8	Mengisi tiap gejala sesuai dengan gejala yang dialami berupa inputan “Ya” atau “tidak”.		Proses diagnosa berhasil, output pada antarmuka menampilkan nama, tanggal, penyakit, deskripsi, pengendalian dan persentase penyakit.		<i>Valid</i>

No	Skenario Pengujian	Test Case	Hasil Yang Diharapkan	Hasil Pengujian	Kesimpulan
Cetak Hasil Diagnosa					
9	Klik tombol cetak pada penyakit yang akan dicetak hasil diagnosanya.		antarmuka menampilkan data yang berhasil dicetak.		<i>Valid</i>
Validasi Data Pakar					
10	Mengisi validasi data dengan memberi jawaban valid/tidak valid pada semua gejala penyakit.		Validasi data pakar berhasil terkirim pada admin.		<i>Valid</i>

4.2.4.3 Pengujian Kuisisioner

Pengujian Kuisisioner adalah pengujian yang dilakukan dengan memberikan pertanyaan mengenai sistem untuk menentukan apakah sudah cukup baik digunakan.

a. Pertanyaan Kuisisioner

Tabel 4.3 Tabel Pertanyaan Kuisisioner

No	Pertanyaan	Penilaian			
		A	B	C	D
1	Bagaimana menurut anda, apakah sistem ini mudah digunakan?				
2	Bagaimana menurut anda tentang desain antarmuka dari sistem ini?				
3	Bagaimana menurut anda apakah sistem ini mudah dipahami?				
4	Bagaimana menurut anda, apakah sistem ini memudahkan pengguna dalam melakukan register?				
5	Bagaimana menurut anda, apakah fitur login pada sistem mudah digunakan?				
6	Bagaimana menurut anda, apakah sistem ini memudahkan pengguna untuk melakukan diagnosa penyakit?				
7	Bagaimana menurut anda, apakah menu pada admin mudah digunakan ?				
8	Bagaimana menurut anda, apakah menu pada pakar mudah digunakan ?				
9	Apakah semua fungsi pada aplikasi ini berjalan dengan baik?				
10	Bagaimana menurut anda tentang keseluruhan aplikasi ini?				

Pengujian kuisioner ini menggunakan sepuluh pertanyaan kepada dua puluh orang untuk mengisi pertanyaan dan mencoba sistem pakar diagnosa penyakit padi dengan metode *Certainty Factor* ini.

b. Jawaban Kuisioner

Tabel 4.4 Tabel Jawaban Kuisioner

No	Nama	Jawaban									
		1	2	3	4	5	6	7	8	9	10
1	Rifki K.	A	A	B	A	B	B	B	B	A	A
2	Diyah A.	A	B	B	A	A	B	B	B	B	A
3	Rudi T.	B	A	B	B	A	B	B	B	B	A
4	Inneke W.	B	A	B	A	A	A	A	A	A	B
5	Lulu M.	A	B	B	A	B	A	B	A	B	B
6	Noufal I.	B	B	A	A	A	A	A	A	B	A
7	M. Bimo	B	B	B	B	A	B	A	B	A	B
8	Shafa D.	B	B	B	A	B	A	B	B	B	B
9	Stephan D.	B	A	B	A	A	B	B	A	B	B
10	Widi I.	A	B	B	A	B	A	A	B	A	B
11	Ade K.	A	B	B	A	A	B	A	A	B	B
12	M. Faizal	A	B	B	B	B	A	B	B	B	B
13	Lufianti	B	B	B	B	B	B	B	B	A	A
14	Rizaluddin S.	B	A	A	A	A	B	A	A	B	A
15	Ari I.	B	B	B	A	A	B	A	A	B	B
16	Ari F.	A	B	B	A	A	A	B	B	A	A
17	Cica S.	A	A	A	B	A	B	A	B	B	B
18	Wulan D.	B	B	A	B	A	B	B	B	A	B
19	Firda F.	B	B	B	B	A	B	A	B	B	B
20	Eko P.	B	B	B	B	B	B	B	B	B	B

Dari hasil jawaban pengujian dengan kuisioner tersebut dapat disimpulkan dengan persamaan (4.1) dengan ketentuan skala sebagai berikut.

Sangat Baik (A) = 4

Baik (B) = 3

Cukup (C) = 2

Kurang (D) = 1

$$\text{Hasil Pengujian} = \sum_{i=0}^{i=4} H_i * 100\% \dots \dots \dots (4.1)$$

Dimana H_i didapat dari persamaan (4.2) berikut.

$$H_i = \frac{\sum_{j=1}^n x_j}{N_p} \dots \dots \dots (4.2)$$

Keterangan:

j = index soal ke j .

x_j = Nilai atau skor pertanyaan ke j untuk index i .

n = Maksimal soal.

N_p = Total poin

Berikut merupakan rumus untuk menghitung hasil N_p , dapat dilihat pada persamaan (4.3).

$$N_p = n * 4 * \text{jumlah responden} \dots \dots \dots (4.3)$$

Maka hasilnya,

$$N_p = 10 * 4 * 20 = 800$$

Jadi, rumus untuk menghitung hasil dari *persentase* dari jawaban kuisioner adalah pada persamaan (4.4) berikut.

$$\text{Skor} = (\text{Jumlah jawaban} * \text{nilai } N_p) * 100\% \dots \dots \dots (4.4)$$

$$A = (79 * \frac{4}{800}) * 100\% = 39.5 \%$$

$$B = (121 * \frac{3}{800}) * 100\% = 45,4 \%$$

$$C = (0 * \frac{4}{800}) * 100\% = 0 \%$$

$$D = (0 * \frac{3}{800}) * 100\% = 0 \%$$

Dari hasil pengujian kuisioner sistem pakar diagnosa penyakit padi dengan metode *Certainty Factor* ini hasilnya adalah 39.5% untuk perolehan nilai A atau yang menilai sistem ini sangat baik, dan 45.4% untuk perolehan nilai B atau yang menilai sistem ini baik, sehingga jika dijumlahkan dari nilai tersebut maka sistem ini mendapat total nilai 84,9%. Perolehan hasil persentase pengujian dengan kuisioner yang dilakukan oleh 20 orang penguji/responden hasilnya adalah 84,9%, dengan persentase tersebut dapat disimpulkan bahwa sistem yang dibuat sudah baik dan dapat dengan mudah digunakan oleh *user*.

BAB V

PENUTUP

5.1 Kesimpulan

Kesimpulan yang didapat setelah mengumpulkan, mengelola, menganalisa, mengimplementasi serta melakukan pengujian sistem pakar diagnosa penyakit padi ini adalah sebagai berikut :

1. Petani dapat dengan mudah mendiagnosa penyakit padi menggunakan fitur diagnosa yang ada pada sistem pakar diagnosa penyakit padi dengan metode *Certainty Factor* ini.
2. Hasil pengujian kuisioner yang dilakukan pada 20 orang penguji hasilnya adalah 84,9%, dengan persentase tersebut dapat disimpulkan bahwa sistem yang dibuat sudah baik dan dapat dengan mudah digunakan oleh *user*.
3. Hasil pengujian *black-box* fungsionalitas sistem pakar diagnosa penyakit padi dengan metode *Certainty Factor* ini berjalan baik, dan telah sesuai dengan daftar kebutuhan sistem
4. Pada sistem ini penerapan metode *Certainty Factor* terletak pada nilai bobot masing-masing gejala penyakit yang diinputkan oleh admin.

5.2 Saran

Pada sistem pakar diagnosa penyakit padi ini masih terdapat kekurangan, oleh karena itu penulis menyarankan untuk pengembang selanjutnya agar dapat mempertimbangkan saran yang diberikan agar sistem pakar diagnosa penyakit padi ini lebih lengkap dan semakin baik, saran-saran tersebut antara lain sebagai berikut:

1. Memperbaiki tampilan user, pakar maupun admin.
2. Dapat dibuat ke versi *mobile application* terutama bagi user agar lebih memudahkan akses diagnosa penyakit.
3. Dapat dikembangkan ke beberapa metode baru lainnya.

DAFTAR PUSTAKA

- Abdullah. P. 2016. *Sistem Pakar Mendiagnosa Penyakit Pada Ayam dengan Metode Certainty Factor Berbasis Android*. Skripsi. Universitas Lampung.
- Arhami, Muhammad. 2008. *Konsep Dasar Sistem Pakar*. Yogyakarta: Penerbit ANDI.
- Arthalia Ika, dkk. 2016. *Sistem Identifikasi Penyakit Tanaman Padi dengan Menggunakan Metode Forward Chaining*. Jurnal Komputasi. 4(1): 9-10.
- Direktorat Jenderal Tanaman Pangan. 2013. *Keputusan Direktur Jenderal Tanaman Pangan Nomor 53/HK.310/C.8/2012 tentang pedoman rekomendasi pengendalian organisme pengganggu tumbuhan (OPT) tanaman sereal* DIREKTORAT JENDERAL TANAMAN PANGAN. Jakarta : Direktorat Jenderal Tanaman Pangan
- Handojo, A. dan M. Isa Irawan. 2009. *Perancangan dan Pembuatan Aplikasi Sistem Pakar untuk Permasalahan Tindak Pidana terhadap Harta Kekayaan*. Universitas Kristen Petra
- Jiang, F., Y. Lu. 2012. Software testing model selection research based on yinyang testing theory. In: *IEEE Proceeding of International Conference on Computer Science and Information Processing (CISP)*, pp. 590-594.
- Kusrini. 2006. *Sistem Pakar Teori dan Aplikasi*. Yogyakarta: Penerbit ANDI.
- Nugroho, Adi. 2009. *Rekayasa Perangkat Lunak Menggunakan UML & Java*. Yogyakarta: Andi Offset.
- Pressman, R.S. 2010. *Software Engineering: A Practitioner's Approach, 7th Edition*. McGraw-Hill, New York.
- Rachmawati,dkk, 2012, *"Aplikasi Sistem Pakar Diagnosis Penyakit Asma"*, Garut : STT Garut.
- Siswanto, 2010. *Kecerdasan Tiruan Edisi 2*. Yogyakarta: Graha Ilmu.
- Sutojo, T., Edy M., dan Vincent S. 2011. *Kecerdasan Buatan*. Semarang:ANDI Yogyakarta.

Sutojo, T., Edy M., dan Vincent S. 2011. *Kecerdasan Buatan*. Semarang: ANDI Yogyakarta.

Turban, Efraim, 1995, "Decision Support Systems And Expert Systems", USA : Prentice Hall International Inc.