

**SISTEM PAKAR MENDIAGNOSA PENYAKIT  
PADA TANAMAN JAGUNG**

**SKRIPSI**



**Oleh:  
Helni Juspita  
140210227**

**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS TEKNIK DAN KOMPUTER  
UNIVERSITAS PUTERA BATAM  
2019**

# **SISTEM PAKAR MENDIAGNOSA PENYAKIT PADA TANAMAN JAGUNG**

**SKRIPSI**

**Untuk memenuhi salah satu syarat  
guna memperoleh gelar sarjana**



**Oleh:  
Helni Juspita  
140210227**

**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS TEKNIK DAN KOMPUTER  
UNIVERSITAS PUTERA BATAM  
2019**

## PERNYATAAN

Dengan ini saya menyatakan bahwa :

1. Skripsi ini adalah asli dan belum pernah diajukan untuk mendapatkan gelar akademik (sarjana, dan/atau magister), baik di Universitas Putera Batam maupun di perguruan tinggi lain.
2. Skripsi ini adalah murni gagasan, rumusan, dan penelitian saya sendiri, tanpa bantuan pihak lain, kecuali arahan pembimbing.
3. Dalam skripsi ini tidak terdapat karya atau pendapat yang telah ditulis atau dipublikasikan orang lain, kecuali secara tertulis dengan jelas dicantumkan sebagai acuan dalam naskah dengan disebutkan nama pengarang dan dicantumkan dalam daftar pustaka.
4. Pernyataan ini saya buat dengan sesungguhnya dan apabila di kemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini, maka saya bersedia menerima sanksi akademik berupa pencabutan gelar yang telah diperoleh, serta sanksi lainnya sesuai dengan norma yang berlaku di perguruan tinggi.

Batam, 16 Februari 2019  
Yang membuat pernyataan,

Helni Juspita  
140210227

# **SISTEM PAKAR MENDIAGNOSA PENYAKIT PADA TANAMAN JAGUNG**

**Oleh:  
Helni Juspita  
140210227**

**SKRIPSI**

**Untuk memenuhi salah satu syarat  
guna memperoleh gelar sarjana**

**Telah disetujui oleh pembimbing pada tanggal  
Seperti tertera dibawah ini**

**Batam, 16 Februari 2019**

**Yulia S.Kom., M.Kom  
Pembimbing**



## ABSTRAK

Jagung merupakan salah satu komoditas unggulan pertanian tanaman pangan yang multi guna dan bernilai strategis untuk dikembangkan. Pada saat ini, jagung tidak hanya dimanfaatkan untuk bahan pangan saja tetapi juga untuk pakan ternak dan juga bahan bakar. Berdasarkan hasil observasi di lapangan, terkadang petani sering salah mendiagnosis tanaman jagung yang sakit. Kesalahan petani mendiagnosis tanaman jagung yang sakit menyebabkan pengendalian yang dilakukan menjadi kurang tepat dan dapat menjadi salah satu penyebab menurunnya kualitas dan kuantitas produksi jagung. Sistem pakar merupakan sistem yang berusaha mengadopsi pengetahuan dari pakar ke komputer, agar komputer dapat menyelesaikan masalah seperti yang biasa dilakukan oleh para ahli. Dalam penelitian ini peneliti membuat sebuah sistem pakar untuk mendiagnosa penyakit pada tanaman jagung. Metode yang digunakan dalam penelitian ini adalah metode penalaran mundur (*backward chaining*). Sehingga dengan pembuatan sistem pakar ini berharap dapat membantu para pengguna sistem, khususnya petani untuk mengatasi permasalahan dan memberikan solusi yang baik. Rule yang ada pada basis pengetahuan sangat berpengaruh dalam menentukan hasil diagnosa. Jika terdapat rule yang tidak sesuai atau tidak cocok dengan gejala yang dipilih oleh user maka sistem tidak dapat mengidentifikasi jenis penyakit apa yang menyerang tanaman jagung. Disini peneliti melakukan penelitian pada kantor Dinas Ketahanan Pangan dan Pertanian di kota Batam, hasil dari penelitian ini adalah Peneliti merancang aplikasi mendiagnosa penyakit pada tanaman jagung berbasis website dengan menggunakan *database MySQL*. Dengan ini pendiagnosa penyakit tanaman jagung menjadi lebih mudah dan tidak terjadi kesalahan dalam penanggulangan penyakit pada tanaman jagung.

**Kata kunci:** Diagnosa, Penyakit Pada Jagung, Sistem Pakar, *Website*.

## ***ABSTRACT***

*Corn is one of the most strategic and valuable commodities for food crop agriculture to be developed. At present, corn is not only used for food but also for animal feed and fuel. Based on observations in the field, sometimes farmers often misdiagnose sick corn plants. The fault of farmers in diagnosing sick corn plants causes controls to be made inaccurate and can be one of the causes of declining quality and quantity of corn production. Expert system is a system that seeks to adopt knowledge from experts to computers, so that computers can solve problems as is usually done by experts. In this study researchers created an expert system for diagnosing diseases in corn plants. The method used in this study is the backward chaining method. So by making this expert system hope that it can help system users, especially farmers, to overcome problems and provide good solutions. Rules that exist on the knowledge base are very influential in determining a diagnosis. If there are rules that do not match or do not match the symptoms chosen by the user, the system cannot identify the type of disease that attacks the corn plant. Here the researchers conducted a study in the Office of Food and Agriculture Resilience in the city of Batam, the results of this study were that the researchers designed an application to diagnose diseases on corn-based websites using a MySQL database. With this, the diagnosis of corn disease is easier and there are no errors in preventing disease in corn plants.*

**Keywords:** *Diagnose, Diseases In Corn, Expert System, Website.*

## KATA PENGANTAR

Puji dan syukur penulis panjatkan kepada Tuhan yang maha kuasa yang telah melimpahkan segala rahmat dan karuniaNya, sehingga penulis dapat menyelesaikan laporan tugas akhir yang merupakan salah satu persyaratan untuk menyelesaikan program studi strata satu (S1) pada Program Studi Teknik Informatika Universitas Putera Batam.

Penulis menyadari bahwa skripsi ini masih jauh dari sempurna. Karena itu, kritik dan saran akan senantiasa penulis terima dengan senang hati.

Dengan segala keterbatasan, penulis menyadari pula bahwa skripsi ini takkan terwujud tanpa bantuan, bimbingan, dan dorongan dari berbagai pihak. Untuk itu, dengan segala kerendahan hati, penulis menyampaikan ucapan terima kasih kepada:

1. Rektor Universitas Putera Batam Ibu Dr. Nur Elfi Husda,S.Kom.,M.SI
2. Dekan Fakultas Teknik dan Komputer Bapak Amrizal, S.Kom.,M.SI
3. Ketua Program Studi Teknik Informatika Andi Maslan,S.T.,M.SI.
4. Yulia S.Kom.,M.Kom selaku pembimbing Skripsi pada Program Studi Teknik Informatika Universitas Putera Batam.
5. Dosen dan Staff Universitas Putera Batam
6. Bapak Devi Januardi Sartely, S.P selaku narasumber yang telah bersedia meluangkan waktunya untuk memberikan data yang dibutuhkan oleh penulis dalam rangka pembuatan skripsi ini.

7. Kepada kedua orang tua serta seluruh keluarga yang telah memberikan dukungan baik secara materi maupun secara moril. Yang selalu memberikan motivasi untuk menyelesaikan kuliah dengan baik serta mendoakan keberhasilan penulis menyelesaikan skripsi ini.
8. Rekan-rekan mahasiswa mahasiswi Universitas Putera Batam terutama khususnya teman-teman fakultas teknik dan komputer Tiban yang turut memberikan doa dan dukungannya.
9. Serta pihak-pihak lain yang tidak dapat disebutkan satu persatu.

Semoga Allah SWT membalas kebaikan dan selalu mencurahkan hidayah serta taufikNya, Amin.

Batam, 16 Februari 2019

Helni Juspita

## DAFTAR ISI

<b>HALAMAN SAMPUL DEPAN</b> .....	i
<b>HALAMAN JUDUL</b> .....	ii
<b>PERNYATAAN</b> .....	iii
<b>HALAMAN PENGESAHAN</b> .....	iv
<b>ABSTRAK</b> .....	v
<b>ABSTRACT</b> .....	vi
<b>KATA PENGANTAR</b> .....	vii
<b>DAFTAR ISI</b> .....	ix
<b>DAFTAR GAMBAR</b> .....	xi
<b>DAFTAR TABEL</b> .....	xiii
<b>BAB I PENDAHULUAN</b> .....	1
1.1 Latar Belakang Penelitian .....	1
1.2 Identifikasi Masalah .....	5
1.3 Pembatasan Masalah .....	5
1.4 Perumusan Masalah .....	6
1.5 Tujuan Penelitian .....	6
1.6 Manfaat Penelitian .....	7
1.6.1 Bagi Petani .....	7
1.6.2 Bagi Penulis .....	7
<b>BAB II KAJIAN PUSTAKA</b> .....	8
2.1 Teori Dasar .....	8
2.1.1 <i>Artificial Intelligence</i> (kecerdasan buatan) .....	8
2.1.1.1 Logika Fuzzy ( <i>fuzzy logic</i> ) .....	9
2.1.1.2 Jaringan Syaraf Tiruan ( <i>Artificial Neural Network</i> ) .....	11
2.1.1.3 Sistem Pakar ( <i>Expert System</i> ) .....	13
2.1.2 Web .....	23
2.1.3 <i>Database</i> (basis data) .....	24
2.2 Variabel Penelitian .....	25
2.2.1 Penyakit Busuk Batang/Busuk Akar .....	26
2.2.2 Penyakit Layu <i>Fusarium</i> .....	27
2.2.3 Penyakit Hawar Daun .....	28
2.2.4 Penyakit Bulai .....	29
2.3 Software Pendukung .....	30

2.3.1 Xampp .....	37
2.3.2 PHP ( <i>Hypertext Processor</i> ) .....	37
2.3.3 HTML ( <i>Hyper Text Markup Language</i> ).....	38
2.3.4 MySQL .....	39
2.3.5 Notepad ++ .....	40
2.4 Penelitian Terdahulu .....	40
2.5 Kerangka Pemikiran .....	44
<b>BAB III METODE PENELITIAN</b> .....	44
3.1 Desain Penelitian .....	44
3.2 Teknik Pengumpulan Data .....	46
3.3 Operasional Variabel .....	47
3.4 Perancangan Sistem .....	50
3.4.1 Desain Basis Pengetahuan .....	50
Penyakit Layu Fusarium .....	50
3.4.2 UML (Unified Modeling Language) .....	56
3.4.3 Desain Database .....	81
3.4.4 Desain Antarmuka .....	84
3.5 Lokasi dan Jadwal Penelitian.....	95
3.5.1 Lokasi Penelitian .....	95
3.5.2 Jadwal Penelitian .....	95
<b>BAB IV HASIL DAN PEMBAHASAN</b> .....	96
4.1. Hasil Penelitian .....	96
4.2. Pembahasan.....	109
4.2.1 Pengujian Validasi .....	110
4.2.2 Pengujian Akurasi .....	112
<b>BAB V SIMPULAN DAN SARAN</b> .....	114
5.1 Simpulan.....	114
5.2 Saran .....	115
<b>DAFTAR PUSTAKA</b>	
<b>LAMPIRAN 1 PENDUKUNG PENELITIAN</b>	
<b>LAMPIRAN 2 DAFTAR RIWAYAT HIDUP</b>	
<b>LAMPIRAN 3 SURAT PENELITIAN</b>	

## DAFTAR GAMBAR

<b>Gambar 2.2</b> struktur sistem pakar berbasis kaidah produksi .....	20
<b>Gambar 2.3</b> Pohon keputusan .....	22
<b>Gambar 2.4</b> Tanaman jagung .....	25
<b>Gambar 2.5</b> Penyakit busuk batang dan busuk akar .....	26
<b>Gambar 2.6</b> Penyakit Layu <i>Fusarium</i> .....	27
<b>Gambar 2.7</b> Penyakit Hawar Daun .....	28
<b>Gambar 2.8</b> Penyakit Bulai .....	29
<b>Gambar 2.9</b> Logo <i>Xampp</i> .....	37
<b>Gambar 2.10</b> Logo PHP .....	37
<b>Gambar 2.11</b> Logo <i>HTML</i> .....	38
<b>Gambar 2.12</b> Logo <i>MySQL</i> .....	39
<b>Gambar 2.13</b> Logo Notepad++ .....	40
<b>Gambar 2.14</b> Kerangka Pemikiran .....	44
<b>Gambar 3.1</b> Desain Penelitian .....	44
<b>Gambar 3.2</b> Pohon Keputusan .....	55
<b>Gambar 3.3</b> <i>Use case Diagram</i> .....	56
<b>Gambar 3.4</b> <i>Activity Diagram</i> Data Login .....	58
<b>Gambar 3.5</b> <i>Activity Diagram</i> Data Profile .....	59
<b>Gambar 3.6</b> <i>Activity Diagram</i> Data Gejala .....	60
<b>Gambar 3.7</b> <i>Activity Diagram</i> Data Penyakit .....	61
<b>Gambar 3.8</b> <i>Activity Diagram</i> Data Pengetahuan .....	63
<b>Gambar 3.9</b> <i>Activity Diagram</i> Data Riwayat Diagnosa .....	64
<b>Gambar 3.10</b> <i>Activity Diagram</i> Ubah Password .....	65
<b>Gambar 3.11</b> <i>Activity Diagram</i> Logout .....	65
<b>Gambar 3.12</b> <i>Activity Diagram</i> Data Riwayat .....	66
<b>Gambar 3.13</b> <i>Activity Diagram</i> Data Diagnosa .....	67
<b>Gambar 3.14</b> <i>Activity Diagram</i> Data Pengetahuan .....	68
<b>Gambar 3.15</b> <i>Sequence Diagram</i> Data Login .....	69
<b>Gambar 3.16</b> <i>Sequence Diagram</i> Data Profile .....	70
<b>Gambar 3.17</b> <i>Sequence Diagram</i> Data Gejala .....	71
<b>Gambar 3.18</b> <i>Sequence Diagram</i> Data Penyakit .....	72
<b>Gambar 3.19</b> <i>Sequence Diagram</i> Data Pengetahuan .....	73
<b>Gambar 3.20</b> <i>Sequence Diagram</i> Data Riwayat Diagnosa .....	74
<b>Gambar 3.21</b> <i>Sequence Diagram</i> Ubah Password .....	75
<b>Gambar 3.22</b> <i>Sequence Diagram</i> Logout .....	76
<b>Gambar 3.23</b> <i>Sequence Diagram</i> Data Riwayat .....	77
<b>Gambar 3.24</b> <i>Sequence Diagram</i> Data Diagnosa .....	78
<b>Gambar 3.25</b> <i>Sequence Diagram</i> Data Pengetahuan .....	79
<b>Gambar 3.26</b> <i>Class Diagram</i> .....	80
<b>Gambar 3.27</b> Halaman Utama Web .....	85
<b>Gambar 3.28</b> Tampilan Halaman Diagnosa Pendaftaran .....	85

<b>Gambar 3.29</b> Tampilan Halaman Diagnosa Pilihan Penyakit .....	86
<b>Gambar 3.30</b> Tampilan Halaman Diagnosa Pertanyaan.....	86
<b>Gambar 3.31</b> Tampilan Halaman Diagnosa Hasil .....	87
<b>Gambar 3.32</b> Tampilan Halaman Diagnosa Cetak .....	87
<b>Gambar 3.33</b> Halaman Data Riwayat .....	88
<b>Gambar 3.34</b> Halaman Basis Pengetahuan.....	89
<b>Gambar 3.35</b> Tampilan Halaman <i>Login Pakar</i> .....	89
<b>Gambar 3.36</b> Tampilan Menu Utama <i>Admin</i> .....	90
<b>Gambar 3.37</b> Tampilan Menu <i>Profile Admin</i> .....	90
<b>Gambar 3.38</b> Tampilan Menu Gejala <i>Admin</i> .....	91
<b>Gambar 3.39</b> Tampilan Menu Gejala <i>Admin</i> Tambah dan Edit .....	91
<b>Gambar 3.40</b> Tampilan Menu Penyakit <i>Admin</i> .....	92
<b>Gambar 3.41</b> Tampilan Menu Penyakit <i>Admin</i> Tambah dan Edit.....	92
<b>Gambar 3.42</b> Tampilan Menu Basis Pengetahuan <i>Admin</i> .....	93
<b>Gambar 3.43</b> Tampilan Menu Atur Basis Pengetahuan <i>Admin</i> .....	93
<b>Gambar 3.44</b> Tampilan Menu Basis Pengetahuan <i>Admin</i> Tambah & Edit .....	94
<b>Gambar 3.45</b> Tampilan Menu Riwayat Diagnosa <i>Admin</i> .....	94
<b>Gambar 3.46</b> Tampilan Menu Ubah <i>Password Admin</i> .....	95
<b>Gambar 4.1</b> Halaman Beranda.....	97
<b>Gambar 4.2</b> <i>Form</i> Pendaftaran .....	97
<b>Gambar 4.3</b> Daftar Penyakit Pada Tanaman Jagung .....	97
<b>Gambar 4.4</b> Pertanyaan Diagnosa .....	98
<b>Gambar 4.5</b> Hasil Diagnosa.....	98
<b>Gambar 4.6</b> Cetak Hasil Diagnosa .....	99
<b>Gambar 4.7</b> Halaman Data Riwayat .....	99
<b>Gambar 4.8</b> Data Gejala .....	100
<b>Gambar 4.9</b> Data Penyakit.....	100
<b>Gambar 4.10</b> Basis Pengetahuan .....	101
<b>Gambar 4.11</b> Detail Basis Pengetahuan .....	101
<b>Gambar 4.12</b> Halaman <i>Login Admin</i> .....	102
<b>Gambar 4.13</b> Halaman Beranda <i>admin</i> .....	102
<b>Gambar 4.14</b> Halaman <i>Profile</i> .....	103
<b>Gambar 4.15</b> Menampilkan Daftar Data Gejala.....	103
<b>Gambar 4.16</b> <i>Form</i> Tambah Data Gejala .....	104
<b>Gambar 4.17</b> <i>Form</i> Edit Data Gejala.....	104
<b>Gambar 4.18</b> Menampilkan Daftar Data Penyakit .....	105
<b>Gambar 4. 19</b> <i>Form</i> Tambah Data Penyakit.....	105
<b>Gambar 4. 20</b> <i>Form</i> Edit Data Penyakit .....	106
<b>Gambar 4. 21</b> Menampilkan Daftar Penyakit Basis Pengetahuan.....	106
<b>Gambar 4. 22</b> Menampilkan Daftar Basis Pengetahuan Berdasarkan Penyakit	107
<b>Gambar 4. 23</b> <i>Form</i> Tambah Basis Pengetahuan .....	107
<b>Gambar 4. 24</b> <i>Form</i> Edit Basis Pengetahuan.....	108
<b>Gambar 4. 25</b> Halaman Data Riwayat Diagnosa.....	108
<b>Gambar 4. 26</b> Halaman <i>Profile</i> .....	109



## DAFTAR TABEL

<b>Tabel 2.1</b> Tabel keputusan.....	22
<b>Tabel 2.2</b> simbol <i>use case diagram</i> .....	32
<b>Tabel 2.3</b> simbol <i>actifity diagram</i> .....	33
<b>Tabel 2.4</b> simbol <i>sequence diagram</i> .....	34
<b>Tabel 2.5</b> simbol <i>class diagram</i> .....	36
<b>Tabel 3.1</b> Operasional Variabel.....	47
<b>Tabel 3.2</b> Kriteria Jenis Penyakit, Gejala, dan Solusi .....	48
<b>Tabel 3.3</b> Tabel Penyakit.....	50
<b>Tabel 3.4</b> Tabel Gejala .....	51
<b>Tabel 3.5</b> Tabel Pengetahuan .....	52
<b>Tabel 3.6</b> Tabel Keputusan.....	53
<b>Tabel 3.7</b> Tabel Pakar.....	81
<b>Tabel 3.8</b> Tabel Gejala .....	82
<b>Tabel 3.9</b> Tabel Penyakit.....	82
<b>Tabel 3.10</b> Tabel Pengetahuan .....	83
<b>Tabel 3.11</b> Tabel <i>User</i> .....	83
<b>Tabel 3. 12</b> Tabel Diagnosa.....	84
<b>Tabel 3. 13</b> Jadwal Kegiatan .....	96
<b>Tabel 4.1</b> Tabel Pengujian Validasi Halaman <i>User</i> .....	110
<b>Tabel 4.2</b> Tabel Pengujian Validasi Halaman <i>Admin</i> .....	111
<b>Tabel 4.3</b> Hasil Diagnosa Pakar dan Sistem.....	113

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang Penelitian**

Saat ini teknologi informasi mengalami perkembangan semakin pesat dan semakin akrab menyentuh kehidupan manusia. Manusia yang dalam hal ini sebagai user menginginkan untuk dapat memperoleh informasi yang lengkap serta *up to date*. Disinilah peranan teknologi yang semakin maju sangat dibutuhkan, yaitu dalam upaya mewujudkan keinginan manusia tersebut, karena informasi dirasa sangat penting dalam pengambilan keputusan dan dalam pencapaian tujuan.

Tanaman pangan adalah segala jenis tanaman yang dapat menghasilkan karbohidrat dan protein, oleh karena itu tanaman pangan menjadi sumber utama makanan pokok sebagian besar penduduk Indonesia. Salah satu tanaman pangan yaitu jagung. Salah satu faktor yang mempengaruhi produktifitas tanaman jagung yaitu adanya serangan penyakit pada tanaman jagung.

Jagung merupakan salah satu komoditas unggulan pertanian dari sub sektor pertanian tanaman pangan yang multi guna dan bernilai strategis untuk dikembangkan. Pada saat ini, jagung tidak hanya dimanfaatkan untuk bahan pangan saja tetapi juga untuk pakan ternak dan juga bahan bakar. Jumlah produksi, produktivitas dan harga jagung selalu mengalami fluktuasi karena pengaruh jumlah permintaan dan penawaran yang selalu berubah-ubah. Tingginya permintaan jagung di pasar domestik merupakan salah satu peluang bagi

Indonesia untuk menyeimbangkan antara permintaan dan penawaran jagung. (Syarifudin, Hidayat, & Fanani, 2018).

Di Indonesia, tenaga tani yang ahli pada bidang penyakit jagung masih terbatas, baik dari segi jumlah maupun waktu kerja. Dalam menyelesaikan serangan penyakit yang menyerang tidak sedikit dari petani melakukan kesalahan dalam mengatasi permasalahan yang dihadapi. (Sihotang, 2018).

Berdasarkan data statistik yang diperoleh oleh peneliti dari kantor Dinas Ketahanan Pangan dan Pertanian kota Batam, sektor pertanian di kota Batam dapat dikembangkan di daerah pedalaman. Potensi lahan pertanian di delapan Kecamatan di kota Batam, terbagi menjadi luas tanam palawija 1.016,6 Ha, luas tanam sayuran 5.157 Ha dan luas tanam buah-buahan 1.568 Ha. Dengan jumlah penduduk kota Batam tahun 2017 sebesar 1,2 juta jiwa, jumlah kebutuhan sayuran sebesar 79.200 Ton/tahun, produksi tanaman sayuran sebesar 31.838,79 Ton/tahun serta kebutuhan buah-buahan 25.240 Ton/tahun sedangkan produksi buah-buahan 9.111,71 Ton/tahun, mempunyai potensi ekonomi yang sangat strategis, dengan pangsa pasar yang sangat potensial dan harga yang sangat tinggi. Penyajian data pertanian terdiri dari banyaknya rumah tangga dan penduduk yang bekerja di sektor pertanian menurut Kecamatan di kota Batam tahun 2017.

Pekembangan tanaman Palawija di kota Batam khususnya untuk tanaman jagung pada tahun 2017 yaitu dengan luas tanam 406,6 Ha, luas panen 392 Ha, produksi 2.440,00 Ton dan dengan produktivitasnya 62,24 Kw/Ha. Sedangkan untuk target dan realisasi produksi tanaman Palawija di kota Batam pada tahun 2017 khususnya tanaman jagung yaitu dengan target 2.810,5 Ton, realisasi

22.058,80 Ton dan dengan persentase 78,49 %. Dan untuk harga rata-rata Komoditi Palawija di kota Batam tahun 2017 untuk tanaman jagung sendiri yaitu dengan harga eceran Rp.10.000, pengumpul Rp.8.000 dan untuk petani Rp.6.000.

Berdasarkan hasil observasi yang dilakukan di lapangan, terkadang petani salah mendiagnosis tanaman jagung yang terserang penyakit. Kesalahan petani mendiagnosis tanaman jagung yang sakit menyebabkan pengendalian yang dilakukan menjadi kurang tepat dan dapat menjadi salah satu penyebab menurunnya kualitas dan kuantitas produksi jagung. Salah satu petani di kota Batam yang berlokasi di Sei Temiang, selama tiga tahun belakangan ini menggeluti tanaman jagung, dengan luas lahan perkebunannya sekitar 1 Ha. Dari hasil panen tanaman jagung setiap tahunnya tidak sedikit yang terkena penyakit. Ada berbagai macam penyakit yang sering menyerang tanaman jagung milik petani tersebut antara lain, penyakit bulai. Gejala yang ditimbulkan oleh penyakit bulai ini pada permukaan daun terdapat garis-garis berwarna putih. Bila tanaman jagung terinfeksi penyakit bulai lebih awal akan menyebabkan tanaman jagung kerdil. Lalu penyakit Hawar daun, penyakit ini juga menyerang bagian daun tanaman jagung dengan gejala mula-mula terlihat bercak daun yang tidak teratur pada ujung daun. Biasanya gejala ini akan menyebar dengan cepat pada cuaca yang lembab dan curah hujan yang tinggi. Penyakit busuk tongkol, gejalanya pada pangkal batang busuk sehingga bagian atas layu dan mengering. Dari hasil observasi tersebut, maka disini peneliti membuat sebuah sistem pakar untuk mendiagnosa penyakit pada tanaman jagung. Sehingga dengan pembuatan sistem pakar ini berharap dapat membantu para pengguna sistem, khususnya petani

untuk mengatasi permasalahan dan memberikan solusi yang baik. Disini peneliti melakukan penelitian pada kantor Dinas Ketahanan Pangan dan Pertanian di kota Batam, karena masih banyak para petani setempat yang kekurangan informasi mengenai penyakit tanaman jagung.

Sistem pakar merupakan sebuah sistem yang berusaha mengadopsi pengetahuan dari pakar ke komputer, agar komputer dapat menyelesaikan masalah layaknya yang biasa dilakukan oleh para ahli. Sistem pakar juga terbukti sangat membantu karena masyarakat menjadi mempunyai kemampuan untuk menyelesaikan suatu permasalahan. Sebagai contoh, petani dapat mengidentifikasi penyakit yang menyerang tanamannya tanpa harus menunggu kedatangan penyuluh pertanian sehingga antisipasi dapat dilakukan sedini mungkin. (Cahyono, 2017). Untuk mendapatkan sebuah kesimpulan berupa jenis penyakit dan cara pengendalian berdasarkan gejala yang menyerang tanaman jagung agar petani tidak salah dalam mendiagnosis tanaman yang terserang penyakit melalui sistem diagnosa, maka dari itu penelitian ini akan menggunakan metode *backward chaining* yang dirasa cocok oleh peneliti untuk digunakan dalam sistem diagnosa penyakit pada tanaman jagung.

Peneliti merancang aplikasi mendiagnosa penyakit pada tanaman jagung berbasis website dengan menggunakan *database MySQL*. Dengan ini pendiagnosa penyakit tanaman jagung menjadi lebih mudah dan tidak terjadi kesalahan dalam penanggulangan penyakit pada tanaman jagung. Berdasarkan uraian di atas, agar dapat memberikan solusi terhadap suatu permasalahan yang telah diuraikan tersebut maka peneliti mengangkat sebuah judul penelitian dengan judul **“Sistem**

**Pakar Mendiagnosa Penyakit Pada Tanaman Jagung”** dengan harapan dengan adanya sistem pakar ini bermanfaat bagi pengguna.

## **1.2 Identifikasi Masalah**

Berdasarkan latar belakang yang diuraikan di atas, dapat diidentifikasi sebagai berikut :

1. Kurangnya pengetahuan para petani dalam mengidentifikasi penyakit tanaman jagung.
2. Sering terjadinya kesalahan atau kekeliruan saat pengendalian penyakit tanaman jagung.
3. Belum adanya sebuah sistem dalam mendiagnosa penyakit tanaman jagung.

## **1.3 Pembatasan Masalah**

Suatu perancangan memerlukan adanya pembatasan masalah agar lebih terarah dan memudahkan dalam pembahasan, sehingga tujuan penelitian dapat tercapai sesuai kebutuhan. Berikut ini adalah pembatasan masalahnya, yaitu :

1. Metode yang digunakan dalam penelitian ini adalah metode *backward chaining*.
2. Sistem pakar yang dibuat pada penelitian ini berbasis *web*.
3. Sistem pakar yang dibuat menggunakan bahasa pemrograman HTML.
4. *Database* pada penelitian ini menggunakan *MySQL*.

5. Penelitian dilakukan di kantor Dinas Ketahanan Pangan dan Pertanian kota Batam.

#### **1.4 Perumusan Masalah**

Berdasarkan latar belakang yang telah diuraikan, maka adapun yang menjadi perumusan masalahnya adalah sebagai berikut :

1. Bagaimana mendiagnosa penyakit pada tanaman jagung ?
2. Bagaimana penerapan metode *backward chaining* untuk mendiagnosa penyakit pada tanaman jagung ?
3. Bagaimana merancang aplikasi dalam mendiagnosa penyakit pada tanaman jagung dan memberikan solusi yang tepat dalam diagnosa penyakit tanaman jagung ?

#### **1.5 Tujuan Penelitian**

Adapun tujuan yang terdapat dalam penelitian ini adalah :

1. Untuk mengetahui bagaimana mendiagnosa penyakit pada tanaman jagung.
2. Untuk menerapkan sebuah sistem pakar dengan metode *backward chaining* dalam mendiagnosa penyakit pada tanaman jagung.
3. Untuk mengetahui bagaimana sistem yang dirancang dalam mendiagnosa penyakit pada tanaman jagung.

## **1.6 Manfaat Penelitian**

Adapun manfaat penelitian yaitu sebagai berikut :

### **1.6.1 Bagi Petani**

1. Menambah pengetahuan para petani tentang penyakit pada tanaman jagung.
2. Mempermudah para petani dalam mengidentifikasi penyakit tanaman jagung.
3. Para petani dapat menanggulangi penyakit pada tanaman jagung tanpa harus membeli insektisida pembasmi hama penyakit tanaman jagung yang mahal.

### **1.6.2 Bagi Penulis**

1. Menambah pengetahuan tentang sistem pakar dalam diagnosa penyakit tanaman jagung.
2. Sebagai salah satu syarat untuk mendapatkan gelar sarjana.
3. Dapat memberikan gambaran terhadap pemecahan dari suatu permasalahan yang sedang dihadapi.



## **BAB II**

### **KAJIAN PUSTAKA**

#### **2.1 Teori Dasar**

Deskripsi teori berisi tentang penjelasan variabel-variabel yang diteliti melalui pendefinisian, dan uraian yang lengkap serta mendalam dari berbagai referensi, sehingga ruang lingkup dan prediksi terhadap hubungan antara variabel yang akan diteliti menjadi lebih jelas.

##### **2.1.1 *Artificial Intelligence* (kecerdasan buatan)**

Kecerdasan buatan atau *artificial intelligence* adalah merujuk pada mesin yang mampu berpikir, menimbang tindakan yang akan diambil, dan mampu mengambil keputusan seperti yang dilakukan oleh manusia (T.sutojo, Edy Mulyanto, 2011:1)

Menurut (T.sutojo, Edy Mulyanto, 2011:1-2) Alan Turing, ahli matematika berkebangsaan inggris yang dijuluki bapak komputer modern dan pembongkar sandi Nazi dalam era Perang Dunia II 1950, menetapkan defenisi *artificial intelligence*. “Jika komputer tidak bisa dibedakan dengan manusia saat berbincang melalui terminal komputer, maka bisa dikatakan komputer itu cerdas, mempunyai kecerdasan”.

Kombinasi antara *AI* dengan bidang ilmu yang lainnya melahirkan subdisiplin ilmu dalam *AI*. Beberapa diantaranya adalah logika *fuzzy* (*fuzzy logic*), jaringan syaraf tiruan (*artificial neural network*), dan sistem pakar (*expert system*) (T.sutojo, Edy Mulyanto, 2011:12).

#### **2.1.1.1 Logika Fuzzy (*fuzzy logic*)**

Logika *fuzzy* adalah metodologi sistem kontrol pemecahan masalah, yang sesuai diimplementasikan pada sistem, mulai dari sistem yang sederhana, sistem kecil, *embedded system*, jaringan komputer, *multi-channel* atau *workstation* berbasis akuisisi data, dan sistem kontrol. Dalam logika *fuzzy* memungkinkan nilai keanggotaan berada di antara 0 dan 1, artinya suatu keadaan memungkinkan mempunyai dua nilai “Ya” dan “Tidak” secara bersamaan, namun besar nilainya tergantung pada bobot keanggotaan yang dimilikinya. Logika *fuzzy* dapat digunakan diberbagai bidang seperti pada sistem diagnosis penyakit (dalam bidang kedokteran), pemodelan sistem pemasaran, sistem operasi (dalam bidang ekonomi), kendali kualitas air, prediksi adanya gempa bumi, klasifikasi dan pencocokan pola (T.sutojo, Edy Mulyanto, 2011:211-212).

Kelebihan logika *fuzzy* adalah kemampuannya dalam proses penalaran secara bahasa sehingga dalam perancangannya tidak memerlukan persamaan matematik yang rumit.

Beberapa metode yang dapat digunakan dalam sistem inferensi *fuzzy* adalah (T.sutojo, Edy Mulyanto, 2011:233-237) :

1. Metode Tsukamoto

Dalam inferensinya, metode Tsukamoto menggunakan tahapan berikut :

- a. *Fuzzyfikasi*
- b. Pembentukan basis pengetahuan *fuzzy* (*Rule* dalam bentuk IF...THEN).
- c. Mesin *inferensi*.
- d. Menggunakan fungsi implikasi Min.

2. Metode Mamdani

Metode mamdani paling sering digunakan dalam aplikasi-aplikasi karena strukturnya yang sederhana, yaitu menggunakan operasi MIN-MAX atau MAX-PRODUCT. Untuk mendapatkan output, diperlukan 4 tahapan berikut:

- a. *Fuzzyfikasi*
- b. Pembentukan baris pengetahuan *fuzzy* (*rule* dalam bentuk IF...THEN
- c. Aplikasi fungsi implikasi menggunakan fungsi MIN dan komposisi antar *rule* menggunakan fungsi MAX (menghasilkan himpunan *fuzzy* baru).
- d. Defuzzifikasi menggunakan metode *centroid*.

3. Metode Sugeno

Bila *output* dari penalaran dengan metode mamdani berupa himpunan *Fuzzy*, tidak demikian dengan metode Sugeno. Dalam metode Sugeno, output sistem berupa konstanta, atau persamaan linear. Dalam inferensinya, metode Sugeno menggunakan tahapan berikut :

- a. *Fuzzyfikasi*
- b. Pembentukan basis pengetahuan *fuzzy* (rule dalam bentuk IF...THEN)
- c. Mesin inferensi, menggunakan fungsi implikasi MIN
- d. *Defuzzyfikasi*, menggunakan metode rata-rata.

### 2.1.1.2 Jaringan Syaraf Tiruan (*Artificial Neural Network*)

Jaringan syaraf tiruan adalah paradigma pengolahan informasi yang terinspirasi oleh sistem saraf secara biologis, seperti proses informasi pada otak manusia. Cara kerja JST seperti cara kerja manusia, yaitu belajar melalui contoh. Sebuah JST dikonfigurasi untuk aplikasi tertentu, seperti pengenalan pola atau klasifikasi data, melalui proses pembelajaran. Belajar dalam sistem biologis melibatkan penyesuaian terhadap koneksi *synaptic* yang ada antara *neuron* (T.sutojo, Edy Mulyanto, 2011:283-285).

Kelebihan-kelebihan yang diberikan oleh JST antara lain :

1. Belajar *adaptive* : kemampuan untuk mempelajari bagaimana melakukan pekerjaan berdasarkan data yang diberikan untuk pelatihan atau pengalaman awal.
2. *Self-organisation* : sebuah JST dapat membuat organisasi sendiri atau representasi dari informasi yang diterimanya selama waktu belajar.
3. *Real time operation* : perhitungan JST dapat dilakukan secara paralel sehingga perangkat keras yang dirancang dan diproduksi secara khusus dapat mengambil keuntungan dari kemampuan ini.

Selain mempunyai kelebihan-kelebihan tersebut, JST juga mempunyai kelemahan-kelemahan berikut :

1. Tidak efektif jika digunakan untuk melakukan operasi-operasi numerik dengan presisi yang tinggi.
2. Tidak efisien jika digunakan untuk melakukan operasi algoritma aritmatik, operasi logika, dan simbolis.
3. Untuk beroperasi JST butuh pelatihan sehingga bila jumlah datanya besar, waktu yang digunakan untuk proses pelatihan sangat lama.

Baik tidaknya suatu model JST ditentukan oleh hubungan antar neuron atau yang biasa disebut sebagai arsitektur jaringan. *Neuron-neuron* tersebut terkumpul dalam lapisan-lapisan yang disebut *neorun layer*. Lapisan-lapisan penyusun JST terbagi menjadi tiga, yaitu (T.sutojo, Edy Mulyanto, 2011:292) :

1. Lapisan Input (*Input Layer*)  
Unit-unit dalam lapisan *input* disebut unit-unit input yang bertugas menerima pola inputan dari luar yang menggambarkan suatu permasalahan.
2. Lapisan Tersembunyi (*Hidden Layer*)  
Unit-unit dalam lapisan tersembunyi disebut unit-unit tersembunyi yang mana nilai *output*nya tidak dapat diamati secara langsung.
3. Lapisan Output (*Output Layer*)  
Unit-unit dalam lapisan output disebut unit-unit *output* yang merupakan solusi JST terhadap suatu permasalahan.

Beberapa arsitektur jaringan yang sering digunakan dalam jaringan saraf tiruan antara lain (T.sutojo, Edy Mulyanto, 2011:292-295) :

1. Jaringan Lapisan Tunggal (*single layer net*)

Jaringan dengan lapisan tunggal terdiri dari 1 lapisan *input* dan 1 lapisan *output*. Jaringan ini menerima input kemudian mengolahnya menjadi output tanpa melewati lapisan tersembunyi.

2. Jaringan Lapisan Banyak (*multilayer net*)

Jaringan lapisan banyak mempunyai 3 jenis lapisan, yaitu lapisan *input*, lapisan tersembunyi, dan lapisan *output*. Jaringan ini dapat menyelesaikan permasalahan yang lebih kompleks dibandingkan dengan jaringan lapisan tunggal.

3. Jaringan dengan Lapisan Kompetitif (*competitive layer net*)

Jaringan ini digunakan untuk mengetahui neuron pemenang dari sejumlah *neuron* yang ada. Akibatnya, pada jaringan ini sekumpulan *neuron* bersaing untuk mendapatkan hak menjadi aktif. Nilai bobot setiap *neuron* untuk dirinya sendiri adalah 1, sedangkan untuk *neuron* lainnya bernilai random negatif.

### **2.1.1.3 Sistem Pakar (*Expert System*)**

Sistem pakar pertama kali dikembangkan pada pertengahan tahun 1960. Sistem pakar yang muncul pertama kali adalah *General-purpose problem solver* (GPS) yang dikembangkan oleh Newel dan Simon. Sampai saat ini sudah banyak sistem pakar yang dibuat, seperti MYCIN untuk diagnosis penyakit, DENDRAL

untuk mengidentifikasi struktur molekul campuran yang tak dikenal, XCON & XSEL untuk membantu konfigurasi sistem komputer besar, SHOPIE untuk analisis sirkuit elektronik, prospector digunakan dibidang geologi untuk membantu mencari dan menemukan deposit, FOLIO digunakan untuk membantu keputusan bagi seorang manager dalam stok dan investasi, DELTA dipakai untuk pemeliharaan lokomotif listrik diesel (T.sutojo, Edy Mulyanto, 2011:159-160).

Menurut (T.sutojo, Edy Mulyanto, 2011:13) sistem pakar adalah suatu sistem yang dirancang untuk dapat menirukan keahlian seorang pakar dalam menjawab pertanyaan dan memecahkan suatu masalah. Sistem pakar akan memberikan pemecahan suatu masalah yang didapat dari dialog dengan pengguna. Dengan bantuan sistem pakar seorang yang bukan pakar/ahli dapat menjawab pertanyaan, menyelesaikan masalah serta mengambil keputusan yang bisanya dilakukan oleh seorang pakar.

Sistem pakar menjadi sangat populer karena sangat banyak kemampuan dan manfaat yang diberikan, diantaranya (T.sutojo, Edy Mulyanto, 2011:160-161) :

1. Meningkatkan produktivitas, karena sistem pakar dapat bekerja lebih cepat dari pada manusia.
2. Membuat seorang yang awam bekerja seperti layaknya seorang pakar.
3. Meningkatkan kualitas, dengan memberi nasehat yang konsisten dan mengurangi kesalahan.
4. Mampu menangkap pengetahuan dan kepakaran seorang.
5. Dapat beroperasi di lingkungan yang berbahaya.
6. Memudahkan akses pengetahuan seorang pakar.

7. Andal, sistem pakar tidak pernah menjadi bosan dan kelelahan atau sakit.
8. Meningkatkan kapabilitas sistem komputer.
9. Mampu bekerja dengan informasi yang tidak lengkap atau tidak pasti.
10. Bisa digunakan sebagai media pelengkap dalam pelatihan.
11. Meningkatkan kemampuan untuk menyelesaikan masalah karena sistem pakar mengambil sumber pengetahuan dari banyak pakar.

Sistem pakar juga mempunyai beberapa kelemahan selain banyaknya kemampuan dan manfaat yang diberikan, antara lain (Andriani, 2016:12) :

1. Biaya yang diperlukan untuk membuat, memelihara, dan mengembangkan sistem pakar sangat mahal.
2. Sulit kembangkan karena ketersediaan pakar dibidangnya dan kepakaran sulit diekstrak dari manusia karena terkadang sulit bagi seorang pakar untuk menjelaskan langkah mereka dalam menangani masalah.
3. Sistem pakar tidak 100% benar karena seseorang yang terlibat dalam pembuatan sistem pakar tidak selalu benar. Oleh karena itu setelah pembuatan sistem pakar harus dilakukan pengujian terlebih dahulu secara teliti sebelum digunakan.
4. Pendekatan oleh setiap pakar untuk suatu situasi atau *problem* bisa berbeda-beda, meskipun sama-sama benar.
5. *Transfer* pengetahuan dan bersifat subjektif dan bias.
6. Kurangnya rasa percaya pengguna dapat menghalangi pemakaian sistem pakar.



Suatu sistem dikatakan sebagai sistem pakar jika memiliki ciri-ciri sebagai berikut (T.sutojo, Edy Mulyanto, 2011:162) :

1. Terbatas pada domain keahlian tertentu.
2. Dapat memberikan penalaran untuk data-data yang tidak lengkap atau tidak pasti.
3. Dapat menjelaskan alasan-alasan dengan cara yang dapat dipahami.
4. Bekerja berdasarkan kaidah/*rule* tertentu.
5. Mudah dimodifikasi.
6. Basis pengetahuan dan mekanisme inferensi terpisah.
7. Keluarannya bersifat anjuran.
8. Sistem dapat mengaktifkan kaidah secara searah yang sesuai, dituntun oleh dialog dengan pengguna.

*Representasi* pengetahuan merupakan metode yang digunakan untuk mengodekan pengetahuan dalam sebuah sistem pakar yang berbasis pengetahuan. Hal ini dimaksudkan untuk menangkap sifat-sifat penting dari suatu masalah, sehingga suatu informasi itu dapat diakses oleh prosedur pemecahan masalah. Bahasa *representasi* harus dirancang agar fakta-fakta dan pengetahuan lain yang terkandung didalamnya dapat digunakan untuk penalaran.

Pada sistem pakar berbasis *rule*, domain pengetahuan direpresentasikan dalam sebuah kumpulan *rule* berbentuk IF-THEN, sedangkan data direpresentasikan dalam sebuah kumpulan fakta-fakta tentang kejadian saat ini. Mesin inferensi membandingkan masing-masing *rule* yang tersimpan dalam basis pengetahuan dengan fakta-fakta yang terdapat dalam database. Jika bagian IF

(kondisi) dari *rule* cocok dengan fakta, maka *rule* dieksekusi dan bagian THEN (aksi) diletakkan dalam database sebagai fakta baru yang ditambahkan (T.sutojo, Edy Mulyanto, 2011:171-178).

Sistem pakar mempunyai komponen utama pada strukturnya, antara lain (Andriani, 2016:14-15) :

1. Basis Pengetahuan (*Knowledge Base*)

Basis pengetahuan yang merupakan representasi pengetahuan yang dimiliki oleh seorang pakar yang tersusun atas fakta dan kaidah. Fakta merupakan informasi tentang objek, peristiwa, dan situasi. Sedangkan kaidah merupakan suatu cara untuk memunculkan fakta baru berdasarkan fakta yang sudah ada dan sudah diketahui.

2. Mesin Inferensi (*Inference Engine*)

Otak dari sebuah sistem pakar adalah mesin inferensi yang berfungsi untuk memandu proses penalaran terhadap suatu kondisi berdasarkan pada basis pengetahuan yang tersedia. Di dalam mesin inferensi terjadi proses untuk memanipulasi dan mengarahkan kaidah, model, dan fakta yang disimpan dalam basis pengetahuan dalam rangka mencapai solusi atau kesimpulan.

Terdapat dua konsep penalaran yang dapat dilakukan dalam melakukan inferensi, yaitu :

1. Penalaran maju (*forward chaining*)

Penalaran maju merupakan cara penalaran dengan memulai dari fakta terlebih dahulu untuk menguji kebenaran hipotesis atau mencocokkan fakta atau pernyataan dimulai dari bagian sebelah kiri dulu (*IF*). *Forward chaining*

merupakan grup dari *multiple* inferensi yang melakukan pencarian dari suatu masalah kepada solusinya. Menurut (T.sutojo, Edy Mulyanto, 2011:171) bila ada fakta yang cocok dengan bagian *IF*, maka *rule* tersebut dieksekusi. Bila sebuah *rule* dieksekusi, maka sebuah fakta baru (bagian *THEN*) ditambahkan kedalam *database*. Setiap kali pencocokan dimulai dari *rule* teratas. Setiap *rule* hanya boleh dieksekusi sekali saja. Proses pencocokan berhenti bila tidak ada lagi *rule* yang bisa dieksekusi.

Pada metode *forward chaining*, pencarian dapat dilakukan dengan dua cara, yaitu (Andriani, 2016:15) :

- a. Menginputkan semua data kedalam sistem pakar dalam sesi konsultasi. Cara seperti ini tepat dan berguna pada sistem pakar dimana proses di dalamnya terotomatisasi dan langsung menerima data dari *database* atau dari suatu set sensor.
  - b. Memberikan elemen spesifik dari data yang diperoleh selama sesi konsultasi dalam sistem pakar. Cara ini mengurangi jumlah data yang diminta, sehingga data yang diminta hanya data yang benar-benar dibutuhkan oleh sistem pakar tersebut yang nantinya akan digunakan untuk mengambil keputusan.
2. Penalaran mundur (*backward chaining*)

Penalaran mundur merupakan cara penalaran dengan memulai dari hipotesis (ekspektasi apa yang diinginkan terjadi) terlebih dahulu, dan untuk menguji kebenaran hipotesis tersebut harus dicari fakta-fakta yang ada dalam basis pengetahuan. *Backward chaining* adalah metode inferensi yang bekerja mundur

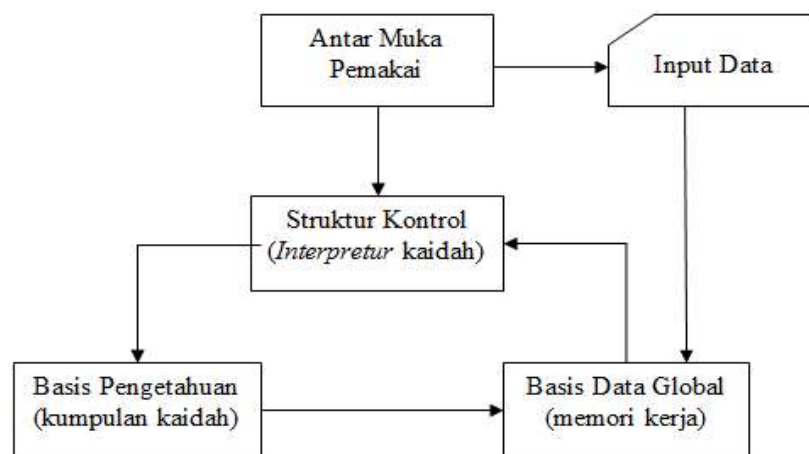
ke arah kondisi awal. Proses diawali dari *Goal* (yang berada dibagian *THEN* dari rule *IF-THEN*), kemudian pencarian mulai dijalankan untuk mencocokkan apakah fakta-fakta yang ada cocok dengan premis-premis dibagian *IF*. Jika cocok, *rule* dieksekusi, kemudian hipotesis di bagian *THEN* ditempatkan di basis data sebagai fakta baru. Jika tidak cocok, simpan premis di bagian *IF* ke dalam *stack* sebagai *subGoal*. *Backward chaining* cocok digunakan untuk suatu aplikasi yang menghasilkan *tree* yang sempit dan cukup dalam.

Ada dua bagian penting dari sistem pakar, yaitu lingkungan pengembangan (*development environment*) dan lingkungan konsultasi (*consultation environment*). Lingkungan pengembangan digunakan oleh pembuat sistem pakar untuk membangun komponen-komponennya dan memperkenalkan pengetahuan ke dalam basis pengetahuan. Sedangkan lingkungan konsultasi digunakan oleh pengguna untuk berkonsultasi sehingga pengguna mendapatkan pengetahuan dan nasihat dari sistem pakar layaknya berkonsultasi dengan seorang pakar (T.sutojo, Edy Mulyanto, 2011:160).

Struktur sistem pakar berbasis kaidah produksi terdiri dari empat komponen, yaitu (Sri Hartati, 2008:10) :

1. Antar muka pemakai, antar muka penghubung antara pemakai dengan sistem pakar.
2. Basis pengetahuan, berisi sekumpulan kaidah yang berasal dari pengetahuan dalam domain tertentu. Kaidah ini secara umum disajikan dalam bentuk kaidah produksi (IF...THEN...)

3. Struktur kontrol, merupakan interpreter kaidah atau mesin inferensi yang menggunakan pengetahuan-pengetahuan yang tersimpan dalam basis pengetahuan untuk memecahkan/menyelesaikan permasalahan yang ada.
4. *Working memory* atau basis data global, mencatat status problem saat ini dan histori solusi.



**Gambar 2.1** struktur sistem pakar berbasis kaidah produksi  
(Sumber : Firebaugh, 1998 daam Hartati. Sri, Iswanti 2008:10)

Menurut (Sri Hartati, 2008: 25) menjelaskan bahwa kaidah produksi menyediakan cara formal untuk merepresentasikan rekomendasi, arahan, atau strategi. Kaidah produksi dituliskan dalam bentuk jika-maka (*if-then*). Kaidah *if-then* menghubungkan antesenden (*antecedent*) dengan konsekuensi yang diakibatkannya. Berikut ini adalah berbagai struktur kaidah *if-then* yang menghubungkan obyek atau atribut Adedeji,1992 dalam (Sri Hartati, 2008:25) :

IF premis THEN konklusi

IF masukan THEN keluaran

IF kondisi THEN tindakan

IF antesenden THEN konsekuen

IF data THEN hasil

IF tindakan THEN tujuan

IF aksi THEN reaksi

IF sebab THEN akibat

IF gejala THEN diagnosa

*Premis* mengacu pada fakta yang harus benar sebelum konklusi tertentu dapat diperoleh. Masukan mengacu pada data yang harus tersedia sebelum keluaran dapat diperoleh. Kondisi mengacu pada keadaan yang harus berlaku sebelum tindakan dapat diambil. *Antesenden* mengacu situasi yang terjadi sebelum konsekuensi dapat diamati. Data mengacu pada informasi yang harus tersedia sehingga sebuah hasil dapat diperoleh. Tindakan mengacu pada kegiatan yang harus dilakukan sebelum hasil dapat diharapkan. Aksi mengacu pada kegiatan yang menyebabkan munculnya efek dari tindakan tersebut. Sebab mengacu pada keadaan tertentu yang menimbulkan akibat teretntu. Gejala mengacu pada keadaan yang menyebabkan adanya kerusakan atau keadaan tertentu yang mendorong adanya pemeriksaan (Sri Hartati, 2008:25-26).

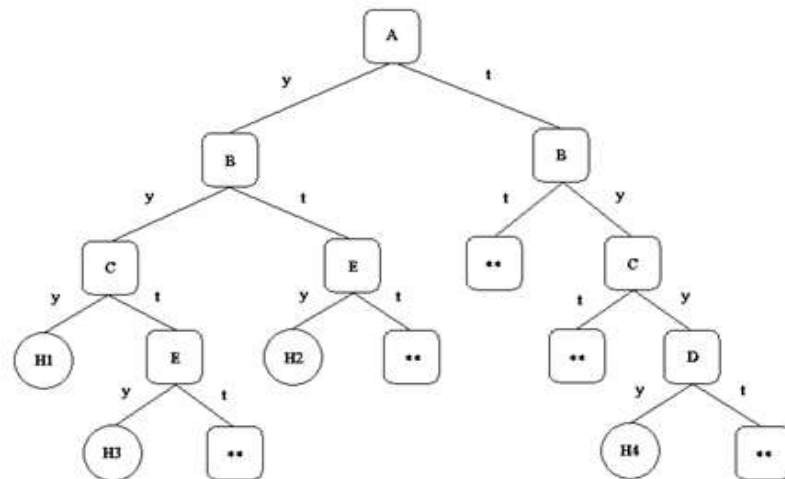
Langkah-langkah sebelum sampai pada bentuk kaidah produksi adalah menyajikan pengetahuan yang berhasil didapatkan dalam bentuk tabel keputusan kemudian dari tabel keputusan dibuat pohon keputusan. Tabel keputusan merupakan matrik kondisi yang dipertimbangkan dalam pendeskripsian kaidah (Sri Hartati, 2008:26).

**Tabel 2.1** Tabel Keputusan

Hipotesa <i>Evidence</i>	Hipotesa 1	Hipotesa 2	Hipotesa 3	Hipotesa 4
<i>Evidence A</i>	Ya	Ya	ya	tidak
<i>Evidence B</i>	Ya	Tidak	ya	ya
<i>Evidence C</i>	Ya	Tidak	tidak	ya
<i>Evidence D</i>	tidak	Tidak	tidak	ya
<i>Evidence E</i>	tidak	Ya	ya	tidak

(Sumber : Sri Hartati 2008:32)

Mengacu tabel 2.1, maka dapat dibuat pohon keputusan sebagai berikut :



**Gambar 2.2** Pohon keputusan  
(Sumber : Hartati. Sri, Iswanti 2008:33)

Keterangan :

A = *evidence A*, H1 = hipotesa 1, y = ya

B = *evidence B*, H2 = hipotesa 2, t = tidak

C = *evidence C*, H3 = hipotesa 3, \*\* = tidak menghasilkan

D = *evidence D*, H4 = hipotesa 4, hipotesa tertentu

Dari gambar 2.2 dapat diketahui bahwa hipotesa H1 terpenuhi jika memenuhi *evidence A*, B, dan C. Hipotesa H2 terpenuhi jika memiliki *evidence A* dan *evidence E*. Hipotesa H3 akan terpenuhi jika memiliki *evidence A*, B, C dan E.

Demikian juga untuk hipotesa H4 akan dihasilkan jika memenuhi *evidence* B,C, dan D. Notasi “y” mengandung arti memenuhi node (*evidence*) di atasnya, notasi “t” artinya tidak memenuhi.

Model representasi pengetahuan kaidah produksi banyak digunakan pada aplikasi sistem pakar karena representasi pengetahuan ini mudah dipahami dan bersifat deklaratif, sesuai dengan jalan pikiran manusia dalam menyelesaikan suatu masalah dan mudah diinterpretasikan (Sri Hartati, 2008:39).

Menurut (Andriani, 2016:13) pengembangan aplikasi sistem pakar yang dapat digunakan untuk menggantikan seorang pakar, dilandasi oleh beberapa alasan antara lain :

1. Ilmu pengetahuan dari sistem pakar dapat digunakan setiap waktu dan di berbagai lokasi.
2. Biaya untuk mendatangkan seorang pakar tergolong tinggi.
3. Seorang pakar suatu saat akan pergi atau pensiun.
4. Program dari sistem pakar dapat mengerjakan tugas-tugas rutin yang membutuhkan seorang pakar secara otomatis.
5. Terkadang kepaaran dibutuhkan pada lingkungan yang tidak bersahabat.

### **2.1.2 Web**

Menurut (Murya, 2012:3) *web* merupakan salah satu layanan yang didapat oleh pengguna komputer yang terhubung dengan internet. *Web* adalah suatu layanan sajian informasi yang menggunakan konsep *hyperlink* (tautan), yang memudahkan *surfer* (sebutan para pemakai komputer yang melakukan *browsing*



atau penelusuran informasi melalui internet). Sekarang *web* menjadi standar *interface* pada layanan-layanan yang ada di internet seperti komunikasi melalui *e-mail*, *chatting*, transaksi bisnis, pencarian informasi, dan lain sebagainya. Keistimewaan inilah yang telah menjadikan *Web* sebagai *service* yang paling cepat pertumbuhannya.

### 2.1.3 *Database* (basis data)

Menurut (Rosa A.S, 2013:43-44) *database* adalah sistem terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah diolah atau informasi dan membuat informasi tersedia saat dibutuhkan. *Database* merupakan media untuk menyimpan data agar dapat diakses dengan mudah dan cepat. Kebutuhan *database* dalam sistem informasi yaitu meliputi memasukkan, menyimpan, dan mengambil data, serta membuat laporan berdasarkan data yang telah disimpan. Salah satu bentuk yang dibutuhkan dalam basis data dalam bentuk sistem yaitu *database management system* (DBMS) adalah suatu sistem aplikasi yang digunakan untuk menyimpan, mengelola, dan menampilkan data.

Suatu sistem aplikasi disebut DBMS jika memenuhi syarat-syarat sebagai berikut (Rosa A.S, 2013:45) :

- a. Menyediakan fasilitas untuk mengelola akses data
- b. Mampu menangani integritas data
- c. Mampu menangani akses data
- d. Mampu menangani *backup* data.

## 2.2 Variabel Penelitian

Variabel penelitian merupakan segala sesuatu yang berbentuk apa saja yang ditetapkan oleh peneliti untuk dipelajari sehingga diperoleh informasi tentang hal tersebut, kemudian ditarik kesimpulannya (Sugiyono, 2014:38). Pada penelitian ini yang menjadi variabel adalah penyakit tanaman jagung, penyakit tanaman jagung ini juga memiliki beberapa jenis, yaitu penyakit busuk batang/busuk akar, penyakit layu fusarium, penyakit hawar daun, penyakit bulai.



**Gambar 2.3** Tanaman jagung  
(Sumber : Data Olahan, 2018)

### 2.2.1 Penyakit Busuk Batang/Busuk Akar



**Gambar 2.4** Penyakit busuk batang dan busuk akar  
(Sumber : Data Olahan, 2018)

*Cendawan Fusarium* sp merupakan salah satu patogen penyebab penyakit busuk batang/busuk akar pada tanaman jagung. Cendawan ini memproduksi konidia pada permukaan tanaman induknya. Konidia dapat disebarkan oleh angin, air hujan, ataupun serangga. Konidia yang terbawa angin dapat menginfeksi ke tongkol jagung, akibatnya jika biji yang terinfeksi ditanam maka dapat menyebabkan penyakit busuk batang.

Gejala yang terdapat pada penyakit busuk tongkol/busuk akar ini adalah sebagai berikut :

1. Daun layu pada siang hari.
2. Daun menggulung.
3. Biji jagung yang terinfeksi berwarna merah muda hingga coklat.

Solusi untuk penyakit busuk tongkol/busuk akar ini adalah sebagai berikut :

1. Membuat parit yang lancar disekitar tanaman jagung.

2. Membuat bedengan dengan ukuran minimal 20 cm dan tidak boleh terlalu rendah sehingga air tidak langsung mengenai akar dan agar langsung menyerap ke tanah.
3. Pastikan bulma (tanaman pengganggu) selalu bersih.

### 2.2.2 Penyakit Layu *Fusarium*



**Gambar 2.5** Penyakit Layu *Fusarium*  
(Sumber : Data Olahan, 2018)

Penyakit layu *fusarium* adalah salah satu patogen penyebab penyakit pada tanaman jagung. Patogen ini menyebabkan pembusukan pada batang dan tongkol jagung.

Gejala-gejala yang terdapat pada penyakit ini adalah :

1. Batang menguning.
2. Batang mengecil.
3. Batang patah.
4. Busuk pada bagian tongkol.
5. Adanya jamur pada tanaman jagung.
6. Jarak tanam terlalu dekat.

Adapun solusi untuk penyakit layu *fusarium* ini adalah :

1. Pastikan jarak tanamnya minimal 40 x 40 cm.
2. Menggunakan pestisida.

### 2.2.3 Penyakit Hawar Daun



**Gambar 2.6** Penyakit Hawar Daun  
(Sumber : Data Olahan, 2018)

Penyakit hawar daun yang disebabkan *Helminthosporium sp*, adalah salah satu penyebab penyakit yang terdapat pada tanaman jagung. Satu gejala bercak yang semakin melebar dapat bersatu dengan bercak yang lain sehingga menyebabkan jaringan daun mati.

Gejala-gejala yang muncul pada penyakit hawar daun ini adalah :

1. Daun layu tidak sehat.
2. Bercak kecil pada daun berwarna coklat kehijauan berbentuk bulat memanjang.
3. Bercak berkembang besar berbentuk oval dengan lebar 5-15 cm.



Solusi untuk penyakit hawar daun ini adalah dengan menyiram tanaman jagung pada pagi hari

#### 2.2.4 Penyakit Bulai



**Gambar 2.7** Penyakit Bulai  
(Sumber : Data Olahan, 2018)

Penyakit bulai pada tanaman jagung disebabkan oleh *oomycete* patogen *Peronosclerospora* spp merupakan salah satu faktor penghambat produktivitas tanaman jagung. Penyakit bulai ini berasal dari genetika atau dari benih, namun jarang terjadi pada tanaman jagung, dalam 1 Ha tanaman jagung yang terkena penyakit bulai ini hanya sekitar 1%. Penyakit bulai juga dapat menyebabkan hasil buah tidak maksimal dan kehilangan hasil produksi tanaman jagung.

Gejala-gejala penyakit bulai, yaitu :

1. Daun berwarna kuning kehijauan.
2. Pada daun permukaan atas dan bawah terdapat warna putih seperti tepung.

3. Adanya warna khlorotik memanjang sejajar tulang daun dengan batas yang jelas antara daun sehat.

Solusi untuk penyakit bulai sebagai berikut :

1. Penanaman jagung secara serentak.
2. Pemusnahan seluruh bagian tanaman sampai ke akarnya pada tanaman yang terserang penyakit bulai.

### 2.3 Software Pendukung

*Unified Modeling language* (UML) merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung. UML hanya berfungsi untuk melakukan pemodelan. UML tidak terbatas pada metodologi tertentu, meskipun pada kenyataannya UML paling banyak digunakan pada metodologi berorientasi objek (Rosa A.S, 2013:137-138).

Terdapat 13 macam diagram dalam UML yang dibagi menjadi tiga kategori, yaitu (Rosa A.S, 2013:141-142).

1. *Structure diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan. Diagram UML yang termasuk dalam kategori ini adalah *class diagram*, *object diagram*, *component diagram*, *composite structure diagram*, *package diagram*, *deployment diagram*.
2. *Behavior diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi

pada sebuah sistem. *UML* yang termasuk dalam kategori ini adalah *use case diagram*, *activity diagram*, *state machine diagram*.

3. *Interaction diagrams* yaitu kumpulan digram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun interaksi antar subsistem pada suatu sistem. *UML* yang termasuk dalam kategori ini adalah *sequence diagram*, *communication diagram*, *timing diagram*, *interaction overview diagram*.

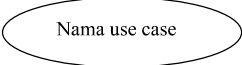

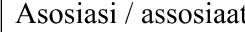
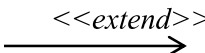
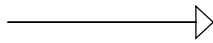
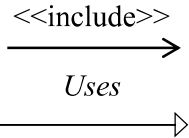
Menurut (Rosa A.S, 2013:155-156) *use case* dan *sequence diagram* merupakan bagian dari desain sistem. Dalam penelitian ini, diagram yang akan digunakan untuk desain sistem, yaitu :

1. *Use case diagram*

*Use case diagram* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada didalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu. Berikut adalah simbol-simbol yang ada pada diagram *Use case* :



**Tabel 2.2** Simbol *Use Case* Diagram

Simbol	Deskripsi
<p><i>Use case</i></p> 	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor. Biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama <i>use case</i> .
<p>Aktor / <i>actor</i></p> 	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah orang, tapi aktor belum tentu merupakan orang.
<p>Asosiasi / <i>assosiaation</i></p> 	Komunikasi antara actor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> memiliki interaksi dengan aktor.
<p>Ekstensi / <i>extend</i></p> 	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu, mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek.
<p>Generalisasi / <i>generalization</i></p> 	Hubungan generalisasi dan spesialisasi (umum-khusus antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
<p>Menggunakan / <i>include</i> / <i>uses</i></p> 	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini


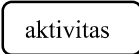
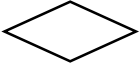


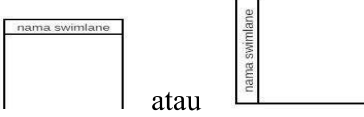
(Sumber : A.S dan Shalahuddin, 2013)

## 2. Activity diagram

Diagram aktivitas menggambarkan aliran kerja atau aktivitas dari sebuah sistem atau proses bisnis yang ada pada perangkat lunak. Pada diagram aktivitas

ini yang perlu diperhatikan adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem. Berikut adalah simbol-simbol yang ada pada diagram aktivitas (Rosa A.S, 2013:161-163) :

**Tabel 2.3** Simbol *Activity Diagram*

Simbol	Deskripsi
Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki awal sebuah status awal
Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja
Percabangan / <i>decision</i> 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu
Penggabungan / <i>join</i> 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu
Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir
<i>Swimlane</i> 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi


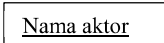
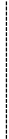
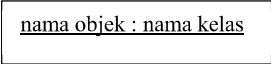

(Sumber : A.S dan Shalahuddin, 2013)

### 3. Sequence diagram

Diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek (Rosa A.S, 2013:165-167).

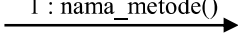
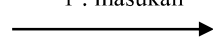
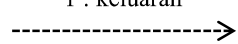
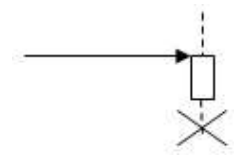
Berikut adalah simbol-simbol yang ada pada diagram sekuen :

**Tabel 2.4** Simbol *Sequence Diagram*

Simbol	Deskripsi
<p>Aktor</p>  <p>nama aktor</p> <p>atau</p>  <p>Nama aktor</p> <p>Tanpa waktu aktif</p>	<p>Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata berada di awal <i>frase</i> nama aktor</p>
<p>Garis hidup / <i>lifeline</i></p> 	<p>Menyatakan kehidupan suatu objek</p>
<p>Objek</p>  <p>nama objek : nama kelas</p>	<p>Menyatakan objek yang berinteraksi pesan</p>
<p>Waktu aktif</p> 	<p>Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya.</p>

(Sumber : A.S dan Shalahuddin, 2013)

Tabel 2.4 Lanjutan

Simbol	Deskripsi
Pesan tipe <i>call</i> 1 : nama_metode() 	Menyatakan suatu objek memanggil operasi / metode yang ada pada objek lain atau dirinya sendiri
Pesan tipe <i>send</i> 1 : masukan 	Menyatakan bahwa suatu objek mengirimkan data / masukan / informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim
Pesan tipe <i>return</i> 1 : keluaran 	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian
Pesan tipe <i>destroy</i> <<destroy>> 	Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada <i>create</i> maka ada <i>destroy</i>

(Sumber : A.S dan Shalahuddin, 2013)

e

#### 4. *Class Diagram*

*Class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Diagram kelas dibuat agar *programmer* membuat membuat kelas-kelas sesuai rancangan di dalam diagram kelas agar antara dokumentasi perancangan dan perangkat lunak sinkron (Rosa A.S, 2013:141-146).

Berikut ini adalah simbol-simbol yang ada pada diagram kelas :

**Tabel 2.5** Simbol *Class Diagram*

<b>Simbol</b>	<b>Deskripsi</b>			
<p>Kelas</p> <table border="1" style="margin-left: 20px;"> <tr><td>Nama_kelas</td></tr> <tr><td>+atribut</td></tr> <tr><td>+operasi()</td></tr> </table>	Nama_kelas	+atribut	+operasi()	Kelas pada struktur sistem
Nama_kelas				
+atribut				
+operasi()				
<p>Antarmuka / <i>interface</i></p> <p style="text-align: center;">○ Nama_interface</p>	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek			
<p>Asosiasi / <i>association</i></p> <p style="text-align: center;">—————</p>	Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>			
<p>Asosiasi berarah / <i>directed association</i></p> <p style="text-align: center;">—————&gt;</p>	Relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i>			
<p>Generalisasi</p> <p style="text-align: center;">—————▷</p>	Relasi antarkelas dengan makna generalisasi-spesialisasi (umum-khusus)			
<p>Kebergantungan / <i>dependency</i></p> <p style="text-align: center;">—————&gt;</p>	Kebergantungan antarkelas			
<p>Agregasi / <i>aggregation</i></p> <p style="text-align: center;">—————◊</p>	Relasi antarkelas dengan makna semua-bagian ( <i>whole-part</i> )			

(Sumber : A.S dan Shalahuddin, 2013)

### 2.3.1 Xampp



**Gambar 2.8** Logo *Xampp*  
(Sumber : Reza Palevi, 2013)

*Xampp* merupakan sebuah perangkat lunak *web server apache* yang terdiri dari kumpulan paket program yang berhubungan dengan *database server*, *web server*, dan lain sebagainya. Didalam *xampp* terdapat *apache* sebagai *server web*, *MySQL* sebagai sever basis data, *Filezila* sebagai *FTP server*, dan beberapa fitur tambahan seperti *Mercury* dan *Tomcat*. *Xampp* merupakan software yang mudah digunakan, gratis, dan mendukung instalasi di *Linux* dan *Windows* (Reza Palevi, 2013).

### 2.3.2 PHP (*Hypertext Processor*)



**Gambar 2.9** Logo PHP  
(Sumber : Amin Miftakul, 2010)

PHP (*Hypertext Processor*) merupakan sebuah bahasa pemrograman server side scripting yang lahir sejalan dengan perkembangan internet. PHP adalah sebuah script yang telah terintegrasi dengan HTML dan mampu menyajikan

informasi yang dinamis. PHP dapat bekerjasama dengan layanan-layanan yang ada di internet menggunakan protokol seperti IMAP, SMTP, HTTP, POP3 dan protokol lain. Selain itu PHP juga menyediakan beragam pilihan *database* yang dapat dipergunakan untuk membuat aplikasi yang berjalan di *web* yang sumber datanya berasal dari *database* (Amin, 2010:1-2).

### 2.3.3 HTML (*Hyper Text Markup Language*)



**Gambar 2.10** Logo *HTML*  
(Sumber : Agus Saputra, 2012)

*Hyper Text Markup Language* (HTML) adalah bahasa paling dasar dan penting yang digunakan untuk menampilkan dan mengelola tampilan pada halaman *website*, digunakan untuk menampilkan berbagai informasi didalam sebuah penjelajah *web* internet dan *formatting hypertext* sederhana yang ditulis ke dalam berkas format ASCII agar dapat menghasilkan tampilan wujud yang terintegrasi. Dengan kata lain, berkas yang dibuat dalam format ASCII normal sehingga menjadi *homepage* dengan perintah-perintah HTML (Saputra, 2012:1).

Pada tahun 1986, ISO mengeluarkan standarisasi bahasa markup berdasarkan GML dengan nama *Standart Generalized Markup Language* (SGML). HTML dibuat oleh kalaborasi *Caillau Tim* dengan *Banners lee Robert* ketika mereka bekerja di CERN pada tahun 1989 yang kini digunakan untuk

membuat halaman *website*. CERN adalah sebuah lembaga penelitian fisika energi tinggi di jenewa, sejarah HTML (Saputra, 2012:4)

### 2.3.4 MySQL



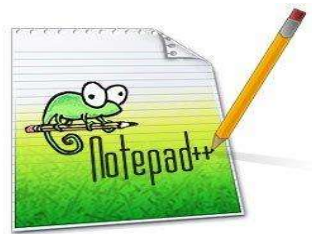
**Gambar 2.11** Logo *MySQL*  
(Sumber : Agus Saputra, 2012)

Menurut (Saputra, 2012:77) *MySQL* merupakan salah satu *database* kelas dunia yang sangat cocok bila dipadukan dengan bahasa pemrograman PHP. *MySQL* bekerja menggunakan bahasa *SQL* (*Structure Query Language*) yang merupakan bahasa standar yang digunakan untuk manipulasi *database*. Perintah yang paling sering digunakan dalam *MySQL* adalah *SELECT* (mengambil), *INSERT* (menambah), *UPDATE* (mengubah), dan *DELETE* (menghapus). *SQL* juga menyediakan perintah untuk membuat *database*, *field*, ataupun *index* untuk menambah atau menghapus data.

*MySQL* dilepaskan dengan suatu lisensi *open source*, dan tersedia secara cuma-cuma. *MySQL* bekerja pada berbagai sistem operasi, dan banyak bahasa. *MySQL* bekerja dengan cepat dan baik dengan data yang besar. *PHP* menyediakan banyak fungsi untuk mendukung *database MySQL*



### 2.3.5 Notepad ++



**Gambar 2.12** Logo Notepad++  
(Sumber : Reza Palevi, 2013)

Notepad++ merupakan sebuah aplikasi yang berguna untuk mengedit text dan skrip kode pemrograman. Notepad++ sebuah aplikasi yang bersifat gratis. Notepad++ menitikberatkan kegunaan aplikasi untuk editing text dalam waktu yang cepat dan praktis. Notepad++ mendukung banyak format bahasa pemrograman seperti PHP, HTML, JavaScript, dan CSS (Reza Palevi, 2013).

### 2.4 Penelitian Terdahulu

Untuk mendukung teori yang berkaitan dengan penelitian, peneliti mencantumkan beberapa penilийan terdahulu di bidang sistem pakar.

1. Nama Penulis: (Syarifudin et al., 2018)

Judul Jurnal : **Sistem Pakar Diagnosis Penyakit Pada Tanaman Jagung Menggunakan Metode Naïve Bayes Berbasis Web**

ISSN : 2548-964X

Sistem pakar ini bertujuan untuk mendiagnosis penyakit yang terdapat pada tanaman jagung. Sistem pakar ini dirancang menggunakan metode naive bayes dengan menggunakan bahasa pemrograman berbasis android. Pada sistem pakar

ini dilakukan tiga pengujian yaitu *blackbox testing*, *usability testing* dan pengujian akurasi. Tingkat akurasi dengan menggunakan metode naive bayes pada penelitian ini menghasilkan tingkat akurasi sebesar 96%.

2. Nama penulis : (Armansyah, 2016)

Judul Jurnal : **Sistem Pakar Identifikasi Hama Dan Penyakit Tanaman Jagung Berbasis Web**

ISSN : 2302-8149

Sistem pakar ini menggunakan mesin inferensi dengan metode *forward chaining*. Penalaran dilakukan berdasarkan gejala-gejala yang tampak secara fisik terhadap tanaman jagung lalu kemudian disimpulkan nama penyakit yang menyerang tanaman jagung tersebut. Aplikasi sistem pakar ini dapat memudahkan pengguna dalam menentukan jenis hama dan penyakit yang ada pada tanaman jagung.

3. Nama Penulis : (Nusantara, Pamungkas, & Syaifudin, 2017)

Judul Jurnal : **Sistem Pakar Analisa Penyakit Tanaman Cabai merah Menggunakan Metode Backward Chaining**

ISSN :2302-3805

Sistem pakar ini menggunakan metode *backward chaining* yang telah disesuaikan dengan penyakit cabai merah dengan berbagai macam gejala, penyakit, beserta solusinya. Berdasarkan gejala-gejala tersebut maka dibuatlah *rule-rule* yang akan di terapkan dalam mesin inferensi untuk mengetahui penyakit apa yang dialami tanaman cabai merah tersebut. Pada penelitian ini menggunakan bahasa pemrograman *PHP* dengan berbasis *web*.

---

4. Nama Penulis: (Sihotang, 2018)

Judul Jurnal : **Sistem Pakar Untuk Mendiagnosa Penyakit Pada  
Tanaman Jagung Dengan Metode Bayes**

ISSN : 2541-3724

Proses pembuatan sistem pakar ini metode kepastiannya teorema bayes dimana metode ini didasarkan dari kondisi awal dimana kondisi awal merupakan kondisi gejala-gejala yang ada kemudian dikenakan aturan yang sudah ditentukan lalu diambil nilai kebenaran yang paling besar untuk menentukan kesimpulan dan solusi dari gejala yang ada. Hasil yang dihasilkan berupa pengurutan data penyakit pada tanaman jagung yang dijadikan sebagai alat bantu dalam pengambilan keputusan bagi petani.

5. Nama Penulis : (Cahyono, 2017)

Judul Jurnal : **Metode Penalaran Sistem Pakar Menggunakan Model  
Hibrid Fuzzy Dempster Shafer Untuk Identifikasi  
Hama Dan Penyakit Tanaman Jagung**

ISSN : 2442-7764

Pada sistem pakar ini penelitian menggunakan model *hybrid fuzzy Dempster*. Penelitian ini akan menentukan nilai densitas (m) dengan menggunakan logika fuzzy. Sistem pakar yang dirancang ini dapat memberikan hasil identifikasi berdasarkan pada gejala yang diinputkan. Penentuan jenis hama dan penyakit yang menyerang didasarkan pada proses pencarian nilai kepercayaan terhadap suatu diagnosa.

6. Nama penulis : (Guoqi, Yuanxun, Sheng, & Bin, 2014)

Judul Jurnal : *An IPC-Based Prolog Design Pattern For Integrating Backward Chaining Inference Into Applications Or Embedded System*

ISSN : 1000-9361

Prolog adalah salah satu yang paling penting untuk membangun sistem pakar dan AI terkait program dan memiliki aplikasi potensial dalam sistem penanaman. Pola desain prolog berasal dari metode mundur yang prinsip dan defenisinya disediakan secara rinci.

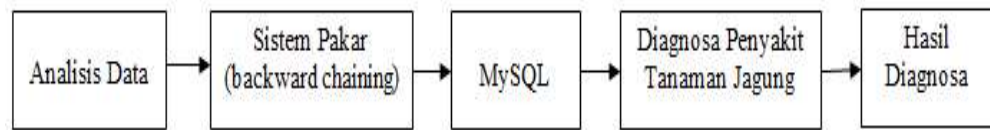
7. Nama penulis : (Abu-nasser et al., 2018)

Judul Jurnal : **Rule Based System For Watermelon Diseases and Treatment**

ISSN : 2000-002X

Tujuan utama sistem pakar ini untuk membantu petani dalam mendeteksi penyakit dan solusi semangka. Metode dalam makalah ini di desain sistem pakar yang diusulkan yang diproduksi untuk membantu petani dalam mendiagnosis penyakit semangka. Sistem pakar yang diusulkan menyajikan gambaran tentang penyakit semangka yang diberikan, penyebabnya penyakit diuraikan dan pengobatan penyakit bila memungkinkan diberikan.

## 2.5 Kerangka Pemikiran



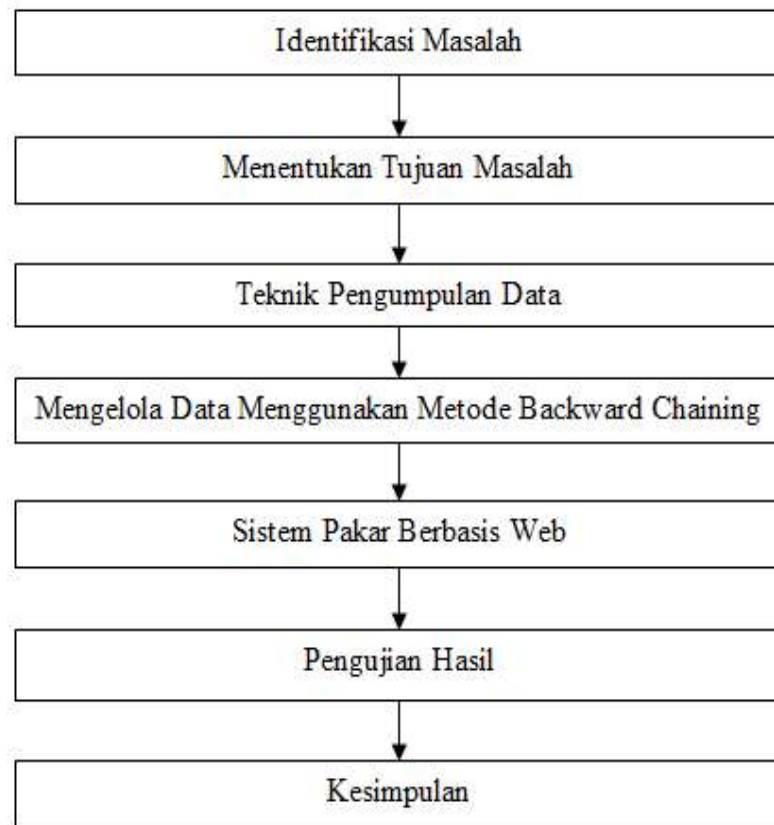
**Gambar 2.13** Kerangka Pemikiran  
(Sumber : Data Olahan, 2018)

Data-data yang dibutuhkan berkaitan dengan penyakit tanaman jagung dianalisis terlebih dahulu agar lebih sederhana dan mudah dilakukan proses pengolahan datanya. Data-datanya tersebut kemudian diolah menggunakan sistem pakar menggunakan metode *backward chaining*. Sistem pakar yang menggunakan metode *backward chaining* ini menggunakan *database* MySQL yang dapat digunakan untuk mendiagnosa penyakit tanaman jagung dan menghasilkan *output* (hasil diagnosa).

## BAB III METODE PENELITIAN

### 3.1 Desain Penelitian

Tahapan penelitian yang dilakukan pada penelitian ini ditunjukkan dalam Gambar 3.1.



**Gambar 3.1** Desain Penelitian  
(Sumber: Data Penelitian 2018)

Adapun penjelasan pada gambar 3.1 adalah:

1. Identifikasi permasalahan

Penelitian diawali dengan mengidentifikasi permasalahan yang berkaitan dengan topik penelitian agar peneliti mendapatkan apa yang sesungguhnya menjadi masalah untuk dipecahkan.

2. Menentukan tujuan penelitian

Tujuan penulisan ini adalah merancang sebuah sistem yang dapat mendiagnosa penyakit pada tanaman jagung menggunakan metode *backward chaining* berbasis *web*.

3. Teknik Pengumpulan Data

Pengumpulan data yang dilakukan adalah studi pustaka dan wawancara di Kantor dinas ketahanan pangan dan pertanian kota Batam dengan Bapak Devi Januardi Sartely, S.P, yang beralamat di Jln. Raja Haji No. 03, Batam Kepulauan Riau.

4. Mengolah data menggunakan metode *Backward Chaining*

Data yang diolah adalah tentang penyakit tanaman jagung menggunakan metode *backward chaining*, hasil dari data yang telah diolah tersebut akan dipilih untuk diproses ke web, dimana terdapat variabel penyakit diantaranya yaitu prnyakit busuk batang/busuk akar, penyakit layu *fusarium*, penyakit hawar daun dan penyakit bulai.

5. Sistem Pakar berbasis *web*

Pada tahap ini, peneliti melakukan kegiatan perancangan mulai dari desain *UML* dan Perancangan Sistem. Setelah itu dilakukan pengkodean untuk mentranslasikan desain yang telah dibuat ke dalam program perangkat lunak sehingga menghasilkan sebuah program komputer. Pengkodean dilakukan menggunakan bahasa pemrograman *PHP* yang dikombinasikan dengan bahasa pemrograman *HTML* dan *database SQL* melalui Web.

6. Pengujian hasil

Proses ini bertujuan untuk meminimalisir kesalahan dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan. Melakukan uji coba untuk mengetahui apakah setiap tahapan dari sistem aplikasi berbasis *web* ini telah berfungsi sebagaimana mestinya atau tidak.

7. Kesimpulan

Tahapan ini merupakan tahapan akhir dalam penelitian ini yaitu menyimpulkan hasil penelitian yang berisi jawaban singkat terhadap rumusan masalah berdasarkan data-data yang ada.

### **3.2 Teknik Pengumpulan Data**

Dalam penelitian ini, teknik yang digunakan untuk pengumpulan data antara lain:



a. Studi Pustaka

Pengumpulan data dengan cara membaca dan mengkaji buku-buku secara teoritis yang berkaitan dengan metode yang dibahas, seperti jurnal, buku referensi tentang sistem pakar dan internet.

b. Studi Lapangan

Melakukan wawancara berupa tanya jawab dari seorang pakar dan pengambilan data secara langsung ke lokasi penelitian yang akan menjadi sumber pengetahuan dan objek penelitian.

### 3.3 Operasional Variabel

Definisi operasional variabel adalah pengertian variabel (yang diungkap dalam definisi konsep) tersebut, secara operasional, secara praktik, secara nyata dalam lingkup obyek penelitian/obyek yang diteliti. Variabel penelitian adalah suatu atribut atau sifat atau nilai dari orang, obyek atau kegiatan yang mempunyai variasi tertentu yang ditetapkan oleh peneliti untuk dipelajari dan ditarik kesimpulan. Berikut penjelasan dari operasional variabel penelitian melalui tabel berikut ini.

**Tabel 3.1** Operasional Variabel

Variabel	Kriteria Penyakit
PENYAKIT TANAMAN JAGUNG	1. Penyakit busuk batang/busuk akar
	2. Penyakit layu <i>fusarium</i>
	3. Penyakit hawar daun
	4. Penyakit bulai

(Sumber: Data Penelitian 2018)

Setelah operasional variabelnya dibuat dalam tabel seperti terlihat diatas, selanjutnya dibuatlah kriteria jenis penyakit, penyebab, dan solusi pada tanaman jagung seperti terlihat pada tabel berikut ini:

**Tabel 3.2** Kriteria Jenis Penyakit, Gejala, dan Solusi

Jenis Penyakit	Gejala	Solusi
Penyakit busuk batang/busuk akar	<ol style="list-style-type: none"> <li>1. Daun layu pada siang hari,</li> <li>2. Daun menggulung,</li> <li>3. Biji jagung yang terinfeksi berwarna merah muda hingga coklat.</li> </ol>	<ol style="list-style-type: none"> <li>1. Membuat parit yang lancar disekitar tanaman jagung.</li> <li>2. Membuat bedengan dengan ukuran minimal 20 cm dan tidak boleh terlalu rendah sehingga air tidak langsung mengenai akar dan agar langsung menyerap ke tanah.</li> <li>3. Pastikan bulma (tanaman pengganggu) selalu bersih.</li> </ol>
Penyakit Layu <i>Fusarium</i>	<ol style="list-style-type: none"> <li>1. Batang menguning,</li> <li>2. Batang mengecil,</li> <li>3. Batang patah,</li> <li>4. Busuk pada bagian tongkol,</li> <li>5. Adanya jamur pada tanaman,</li> <li>6. Jarak tanam terlalu dekat.</li> </ol>	<ol style="list-style-type: none"> <li>3. Pastikan jarak tanamnya minimal 40 x 40 cm.</li> <li>4. Menggunakan pestisida.</li> </ol>

**Tabel 3.2** lanjutan

Jenis Penyakit	Gejala	Solusi
Penyakit hawar daun	<ol style="list-style-type: none"> <li>1. Daun layu tidak sehat,</li> <li>2. Bercak kecil pada daun berwarna coklat kehijauan berbentuk bulat memanjang,</li> <li>3. Bercak berkembang besar berbentuk oval dengan lebar 5-15.</li> </ol>	Menyiram tanaman jagung pada pagi hari.
Penyakit Bulai	<ol style="list-style-type: none"> <li>1. Daun berwarna kuning kehijuan,</li> <li>2. Pada daun permukaan atas dan bawah terdapat warna putih seperti tepung,</li> <li>3. Adanya warna khlorotik memanjang sejajar tulang daun dengan batas yang jelas dengan daun sehat.</li> </ol>	<ol style="list-style-type: none"> <li>1. Penanaman jagung secara serentak,</li> <li>2. Pemusnahan seluruh bagian tanaman sampai ke akarnya pada tanaman yang terserang penyakit bulai,</li> <li>3. Menanam varietas tahan penyakit bulai.</li> </ol>

(Sumber : Data Penelitian 2018)

Tabel 3.2 diatas menjelaskan tentang semua kriteria jenis penyakit tanaman jagung yang akan digunakan pada sistem pakar ini dan dilengkapi juga dengan penyebab masalah dan solusinya dijelaskan secara terperinci pada tabel di atas.

### 3.4 Perancangan Sistem

Perancangan sistem adalah proses mengimplementasikan hasil-hasil dari analisis sistem ke dalam suatu rancangan sistem yang baru (Hendarti, Haryanto, & Akuntansi, 2009).

#### 3.4.1 Desain Basis Pengetahuan

Sebelum melakukan desain basis pengetahuan, telah dilakukan proses akuisisi pengetahuan dengan mengumpulkan fakta melalui wawancara dengan pakar.

##### 1. Data Kriteria Penyakit Tanaman Jagung

Data kriteria penyakit tanaman jagung merupakan data yang pernah dialami petani. Dalam pengkodean penulis memberikan kode “P” untuk kriteria penyakit tanaman jagung yang dimulai dari urutan ”P001” sampai “P004” secara berurutan. Pengetahuan dan fakta tersebut ditampilkan dalam tabel sebagai berikut:

**Tabel 3.3** Tabel Penyakit

Kode Penyakit	Nama Penyakit
P001	Penyakit Busuk Batang/Busuk Akar
P002	Penyakit Layu Fusarium
P003	Penyakit hawar daun
P004	Penyakit bulai

(Sumber: Data Penelitian 2018)

## 2. Data Gejala

Data gejala adalah daftar gejala penyakit tanaman jagung. Dalam pengkodean penulis memberikan kode “G” untuk gejala penyakit tanaman jagung yang dimulai dari urutan ”G001” sampai “G015” secara berurutan. Pengetahuan dan fakta tersebut ditampilkan dalam tabel sebagai berikut:

**Tabel 3.4** Tabel Gejala

Kode Gejala	Nama Gejala
G001	Daun layu pada siang hari
G002	Daun menggulung
G003	Biji jagung yang terinfeksi berwarna merah muda hingga coklat
G004	Batang menguning
G005	Batang mengecil
G006	Batang patah
G007	Busuk pada bagian tongkol
G008	Daun layu tidak sehat
G009	Bercak kecil pada daun berwarna coklat kehijauan berbentuk bujur memanjang
G010	Bercak berkembang besar berbentuk oval dengan lebar 5-15
G011	Daun berwarna kuning kehijauan
G012	Pada daun permukaan atas dan bawah terdapat warna putih seperti tepung
G013	Adanya warna khlorotik memanjang sejajar tulang daun dengan batas yang jelas antara daun sehat
G014	Adanya jamur pada tanaman jagung
G015	Jarak tanam terlalu dekat

(Sumber: Data Penelitian 2018)

### 3. Data Pengetahuan

Data Pengetahuan merupakan data yang berisi relasi antara data-data bagian Penyakit dan Gejala penyakit tanaman jagung yang telah diberikan kode sebelumnya. Relasi antara data tersebut disusun berdasarkan sumber pengetahuan dan fakta dibuat sebelumnya. Data pengetahuan ini disusun untuk memudahkan peneliti dalam menyusun kaidah yang akan digunakan sebagai basis pengetahuan dalam sistem pakar pada penelitian ini. Susunan data pengetahuan yang digunakan dapat dilihat pada tabel sebagai berikut:

**Tabel 3.5 Tabel Pengetahuan**

Kode Penyakit	Kode Pengetahuan
P001	G001, G002, G003
P002	G004, G005, G006, G007, G014, G015
P003	G008, G009, G010
P004	G011, G012, G013

(Sumber: Data Penelitian 2018)

Berdasarkan data pengetahuan yang telah di susun dalam tabel 3.5, maka kaidah aturan (*rule*) yang akan digunakan dalam sistem pakar adalah sebagai berikut:

1. Kaidah 1: *IF G001 AND G002 AND G003 THEN P001*
2. Kaidah 2: *IF G004 AND G005 AND G006 AND G007 AND G014 AND G015 THEN P002*
3. Kaidah 3: *IF G008 AND G009 AND G010 THEN P003*
4. Kaidah 4: *IF G0011 AND G0012 AND G0013 THEN P004*

Berdasarkan kaidah (*rule*) yang telah dibuat maka dapat dijelaskan sebagai berikut:

1. Kaidah 1: *IF* Daun layu pada siang hari *AND* Daun menggulung *AND* Biji jagung yang terinfeksi berwarna merah muda hingga coklat *THEN* Penyakit Busuk Batang/Busuk Akar.
2. Kaidah 2: *IF* Batang menguning *AND* Batang mengecil *AND* Batang patah *AND* Busuk pada bagian tongkol *AND* Adanya jamur pada tanaman jagung *AND* Jarak tanam terlalu dekat *THEN* Penyakit Layu *Fusarium*.
3. Kaidah 3: *IF* Daun layu tidak sehat *AND* Bercak kecil pada daun berwarna coklat kehijauan berbentuk bulat memanjang *AND* Bercak berkembang besar berbentuk oval dengan lebar 5-15 cm *THEN* Penyakit Hawar Daun.
4. Kaidah 4: *IF* Daun berwarna kuning kehijauan *AND* Pada daun permukaan atas dan bawah terdapat warna putih seperti tepung *AND* Adanya warna khlorotik memanjang sejajar tulang daun dengan batas yang jelas antara daun sehat *THEN* Penyakit Bulai.

Berdasarkan kaidah yang telah dibuat tersebut, maka tabel keputusannya adalah sebagai berikut:

**Tabel 3.6** Tabel Keputusan

GEJALA	KRITERIA PENYAKIT			
	P001	P002	P003	P004
G001	√			
G002	√			
G003	√			
G004		√		
G005		√		

**Tabel 3.6** Lanjutan

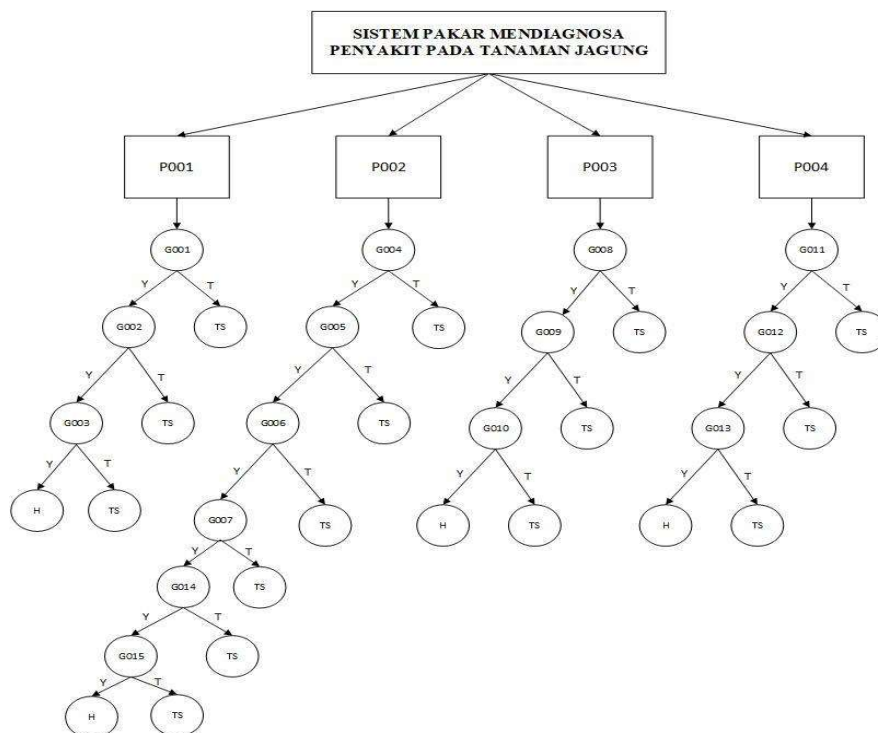
GEJALA	KRITERIA PENYAKIT			
	P001	P002	P003	P004
G006		√		
G007		√		
G008			√	
G009			√	
G010			√	
G011				√
G012				√
G013				√
G014		√		
G015		√		

(Sumber: Data Penelitian 2018)

Dari tabel 3.6 Diatas menjelaskan tentang gejala penyakit tanaman jagung apa saja yang terdapat dalam suatu kriteria penyakit. Dalam sistem pakar penyakit tanaman jagung ini terdapat gejala/ sifat yang kemudian digunakan untuk memberikan solusi.

Berdasarkan tabel keputusan tersebut maka pohon keputusannya adalah sebagai berikut:





**Gambar 3. 2** Pohon Keputusan  
(Sumber: Data Penelitian 2018)

Pohon Keputusan pada gambar 3.2 digunakan untuk memperlihatkan hubungan terkait antara kriteria dengan gejala penyakit tanaman jagung. Alur penelusuran dimulai dari G001. Proses penelusuran selanjutnya tergantung bagaimana jawaban yang diberikan pengguna. Jika pengguna memberikan jawaban “Y”, maka penelusuran menuju pada level berikutnya G002. Begitulah seterusnya sampai penelusuran menemukan simpul “H” (Hasil kriteria penyakit tanaman jagung). Jika sampai pada simpul “TS” (Tidak Sesuai) maka proses berhenti dan tidak menghasilkan masalah tertentu.

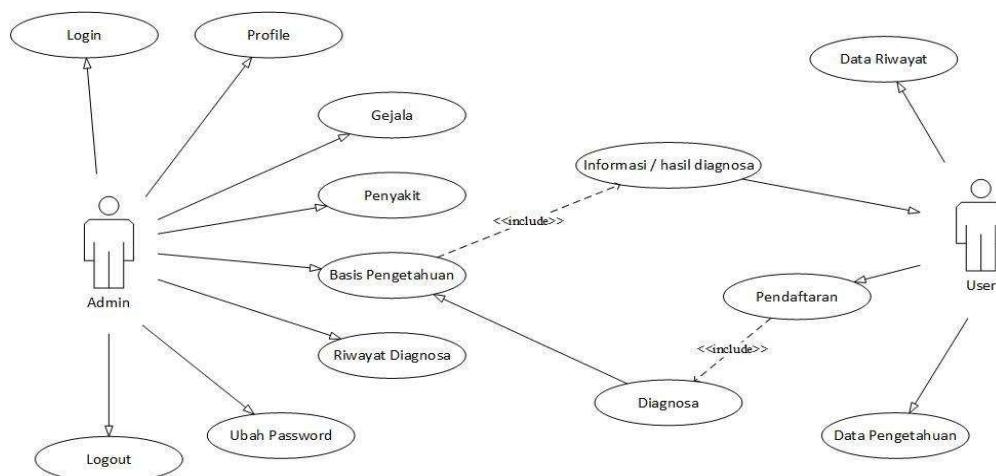
### 3.4.2 UML (Unified Modeling Language)

*Unified Modeling Language* (UML) adalah keluarga notasi grafis yang didukung oleh meta-model tunggal, yang membantu pendeskripsian dan desain sistem perangkat lunak, khususnya sistem yang dibangun menggunakan pemrograman berorientasi objek. ((Isa & Hartawan, 2017).

Struktur diagram dalam UML terdiri atas :

#### 1. *Use Case Diagram*

*Use Case* diagram digunakan untuk menggambarkan sistem dari sudut pandang pengguna sistem tersebut (*user*). sehingga pembuatan *use case* diagram lebih dititik beratkan pada fungsionalitas yang ada pada sistem, bukan berdasarkan alur atau urutan kejadian. Sebuah *use case* diagram mempresentasikan sebuah interaksi antara aktor dengan sistem. ((Isa & Hartawan, 2017).



**Gambar 3.3** *Use case Diagram*  
(Sumber: Data Penelitian 2018)

Berdasarkan gambar 3.3 maka dapat dijelaskan bahwa terdapat dua level user pada sistem ini yaitu seorang *admin*, dan juga seorang *user* yaitu masyarakat publik. Untuk mengakses sistem, *admin* perlu melakukan *login* dalam sistem. Kemudian *admin* dapat mengelola data penyakit tanaman jagung, faktor penyebab, dan juga solusi mengatasi penyakit tanaman jagung. *Admin* juga dapat mengubah dan menghapus data. Sedangkan masyarakat umum sebagai *user* melakukan pendaftaran, lalu mengelola Diagnosa untuk mengetahui penyakit tanaman jagung yang dialami.

## 2. *Activity Diagram*

*Activity* diagram menggambarkan rangkaian aliran dari aktivitas, digunakan untuk mendeskripsikan aktifitas yang dibentuk dalam suatu operasi sehingga dapat juga digunakan untuk aktifitas lainnya. Diagram ini sangat mirip dengan *flowchart* karena memodelkan *workflow* dari suatu aktifitas ke aktifitas yang lainnya, atau dari aktifitas ke status. Pembuatan *activity* diagram pada awal pemodelan proses dapat membantu memahami keseluruhan proses. *Activity* diagram juga digunakan untuk menggambarkan interaksi antara beberapa *use case*. ((Isa & Hartawan, 2017).

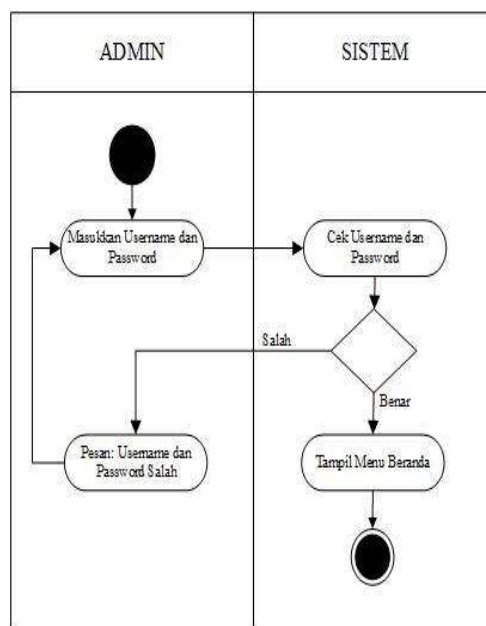
### 1. *Activity Diagram* dengan admin

*Activity* diagram admin adalah alur dari aktifitas yang dapat diakses dan dilakukan oleh admin seperti form login, profile, data gejala, data penyakit,

data pengetahuan, riwayat diagnosa, ubah *password* dan *logout*, untuk lebih jelasnya bisa dilihat pada *activity* diagram dibawah ini:

a. *Activity Diagram Data Login*

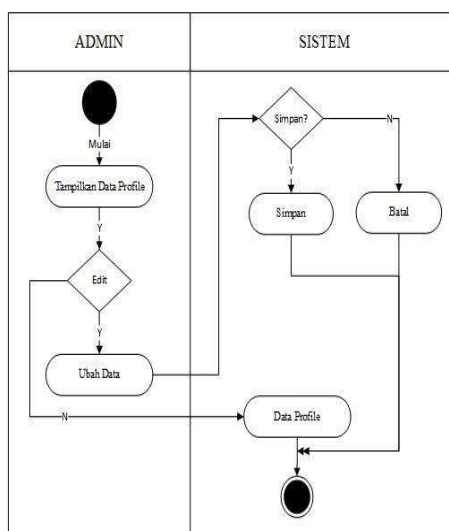
*Activity form login* digunakan untuk mengakses dan mengelola aplikasi sistem pakar. Dapat dijelaskan dari aktifitas *form login* yaitu pertama admin memasukkan *username* dan *passwordnya* lalu sistem melakukan pengecekan apakah *username* dan *passwordnya* terdaftar, jika dinyatakan salah atau tidak terdaftar maka akan menampilkan pesan *username* dan *password* salah, jika benar maka sistem akan menampilkan menu beranda admin, untuk lebih jelasnya dapat dilihat pada gambar berikut:



**Gambar 3.4** *Activity Diagram Data Login*  
(Sumber: Data Penelitian 2018)

b. *Activity Diagram Profile*

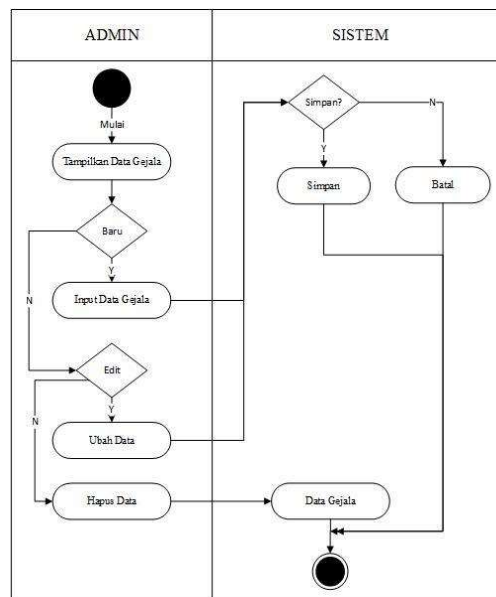
*Activity diagram profile* digunakan untuk mengubah data *username* dan nama lengkap. Dapat dijelaskan dari aktifitas *profile* yaitu pertama admin menampilkan data *profile* lalu kemudian jika admin tidak melakukan edit maka tetap menampilkan data *profile*, jika admin melakukan edit maka data dapat diubah dan sistem menampilkan pilihan apakah mau disimpan atau batal jika disimpan maka data yang diubah akan diperbaharui lalu menampilkan kembali data *profile*, jika batal maka akan kembali tampil ke data *profile*, untuk lebih jelasnya dapat dilihat pada gambar berikut:



**Gambar 3.5** *Activity Diagram Data Profile*  
(Sumber: Data Penelitian 2018)

c. *Activity Diagram* Data Gejala

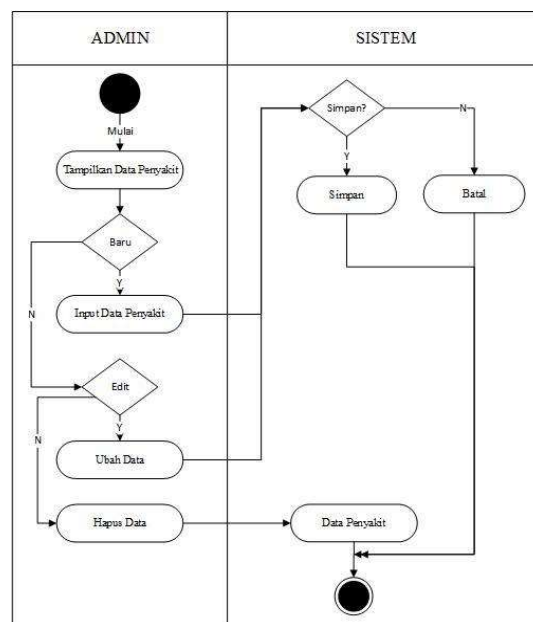
*Activity* diagram data gejala digunakan untuk menampilkan pernyataan saat mendiagnosa dan di halaman ini admin dapat mengelola data seperti menampilkan, menambah, mengubah dan menghapus data gejala. Dapat dijelaskan dari aktifitas data gejala yaitu pertama admin menampilkan data gejala dan jika admin ingin menambah gejala baru maka input data gejala baru jika tidak maka admin melakukan edit lalu ubah data dan jika tidak admin bisa menghapus data gejala. Pada penambahan data gejala yang baru dan edit data gejala, sistem akan tersimpan lalu kemudian sistem menampilkan kembali data gejala, jika batal maka sistem langsung menampilkan data gejala, untuk lebih jelasnya dapat dilihat pada gambar berikut:



**Gambar 3.6** *Activity Diagram* Data Gejala  
(Sumber: Data Penelitian 2018)

d. *Activity Diagram* Data Penyakit

*Activity diagram* data penyakit digunakan untuk hasil dari diagnosa yang berdasarkan gejala-gejala tertentu dan dihalaman ini admin dapat mengelola data seperti menampilkan, menambah, mengubah dan menghapus data penyakit. Dapat dijelaskan dari aktifitas data penyakit yaitu pertama admin menampilkan data penyakit dan jika admin ingin menambah penyakit baru maka input data penyakit baru, jika tidak maka admin melakukan edit lalu ubah data dan jika tidak admin bisa menghapus data penyakit. Pada penambahan data penyakit baru dan edit data penyakit, sistem akan menentukan simpan dan batal, jika disimpan maka data tersebut akan tersimpan kemudian kembali menampilkan data penyakit, jika batal maka langsung menampilkan data penyakit, untuk lebih jelasnya dapat dilihat pada gambar berikut.

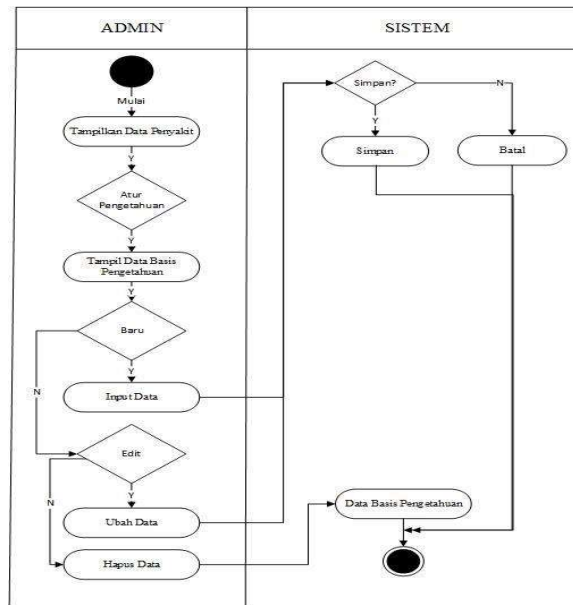


**Gambar 3.7** *Activity Diagram* Data Penyakit  
(Sumber: Data Penelitian 2018)

e. *Activity Diagram* Data Pengetahuan

*Activity diagram* data pengetahuan digunakan untuk mengolah, mengatur dan menghubungkan data gejala dengan data penyakit. Pada halaman ini admin dapat mengelola data seperti menampilkan, menambah, mengubah dan menghapus data basis pengetahuan. Dapat dijelaskan dari aktifitas data pengetahuan yaitu pertama admin menampilkan data penyakit, jika kemudian admin mengatur pengetahuan dengan memilih salah satu penyakit, jika admin mengatur data pengetahuan maka akan menampilkan data pengetahuan yang berisi pertanyaan dari data gejala, di halaman data pengetahuan admin bisa menambah, mengubah dan menghapus data pengetahuan, jika admin menambah data pengetahuan maka input data pengetahuan baru, jika tidak maka admin melakukan edit lalu ubah data dan jika tidak admin bisa menghapus data pengetahuan. Pada penambahan data pengetahuan baru dan edit data pengetahuan, sistem akan menentukan simpan dan batal, jika disimpan maka data tersebut akan tersimpan kemudian kembali menampilkan data basis pengetahuan, jika batal maka langsung menampilkan data basis pengetahuan, untuk lebih jelasnya dapat dilihat pada gambar dibawah ini:

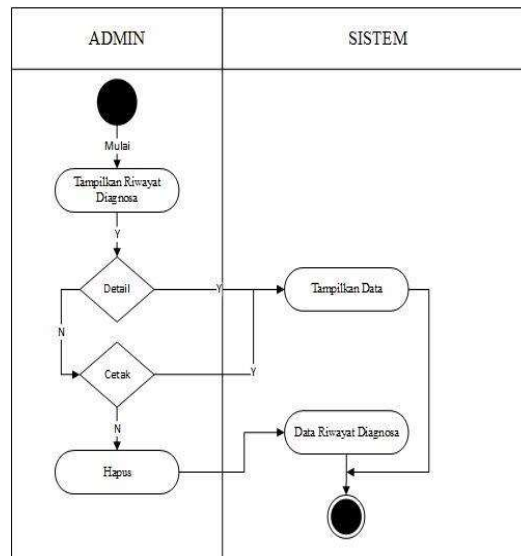




**Gambar 3.8** *Activity Diagram* Data Pengetahuan  
(Sumber: Data Penelitian 2018)

f. *Activity Diagram* Data Riwayat Diagnosa

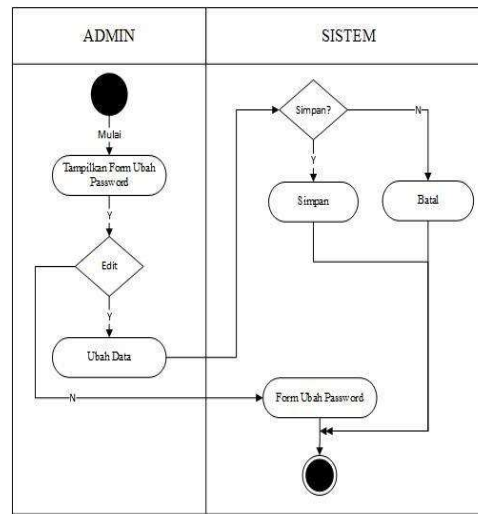
*Activity diagram* data riwayat diagnosa digunakan untuk melihat catatan terakhir user saat melakukan diagnosa. Dapat dijelaskan dari aktifitas data riwayat diagnosa yaitu pertama admin menampilkan data riwayat, jika admin mengklik menu detail dan cetak maka sistem akan menampilkan data riwayat diagnosa dan jika admin mengklik menu hapus maka data tersebut akan dihapus dan menampilkan kembali data diagnosa, untuk lebih jelasnya dapat dilihat pada gambar berikut:



**Gambar 3.9** *Activity Diagram* Data Riwayat Diagnosa  
(Sumber: Data Penelitian 2018)

g. *Activity Diagram* Form Ubah Password

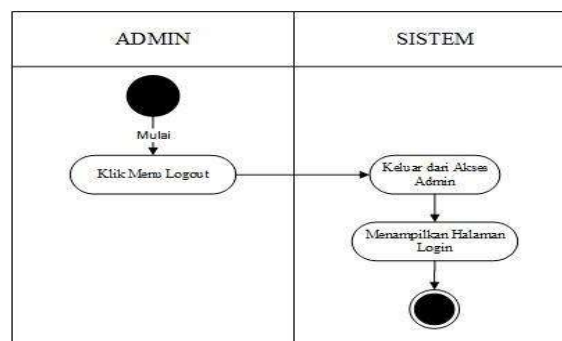
*Activity diagram form* ubah password digunakan untuk mengubah password admin. Dapat dijelaskan dari aktifitas ubah password yaitu pertama admin menampilkan form password kemudian jika admin tidak melakukan edit maka tetap menampilkan form password, jika admin melakukan edit maka data dapat diubah dan sistem menampilkan pilihan apakah mau disimpan atau batal jika disimpan maka data yang diubah akan diperbaharui lalu menampilkan kembali form password, jika batal maka akan kembali tampil ke data form password, penjelasannya dapat dilihat pada gambar dibawah ini:



**Gambar 3.10** *Activity Diagram Ubah Password*  
(Sumber: Data Penelitian 2018)

h. *Activity Diagram Logout*

*Activity diagram logout* digunakan oleh admin untuk keluar dari halaman admin setelah selesai mengelola data-data yang ada pada sistem, jika admin mengklik menu logout maka sistem akan mengeluarkan akses admin dan mengarahkan admin kembali ke halaman login.



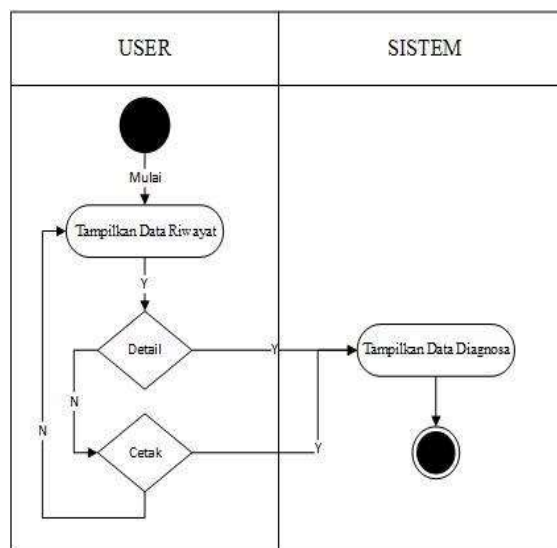
**Gambar 3.11** *Activity Diagram Logout*  
(Sumber: Data Penelitian 2018)

## 2. *Activity Diagram* dengan user

*Activity diagram user* adalah alur dari aktifitas yang dapat diakses dan dilakukan oleh user seperti diagnosa, data riwayat dan data pengetahuan, untuk lebih jelasnya bisa dilihat *activity diagram* user sebagai berikut:

### a. *Activity Diagram* Data Riwayat

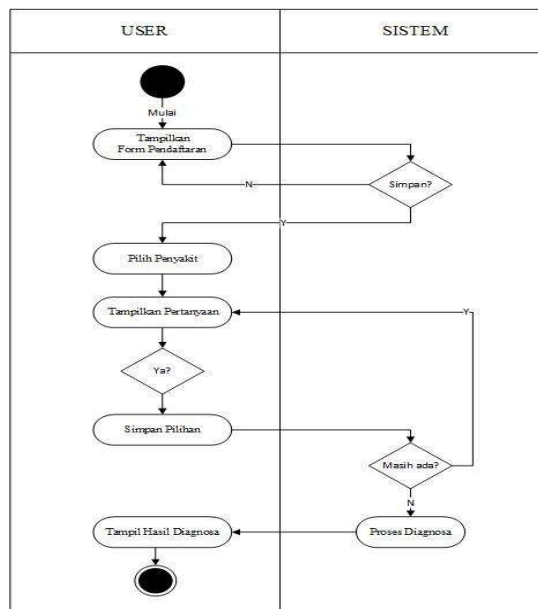
*Activity diagram* data riwayat digunakan untuk melihat catatan terakhir user saat melakukan diagnosa. Dapat dijelaskan dari aktifitas data riwayat yaitu pertama user menampilkan data riwayat jika user mengklik tombol detail dan cetak maka sistem akan menampilkan data riwayat diagnosa, untuk lebih jelasnya dapat dilihat pada gambar berikut.



**Gambar 3.12** *Activity Diagram* Data Riwayat  
(Sumber: Data Penelitian 2018)

b. *Activity Diagram Data Diagnosa*

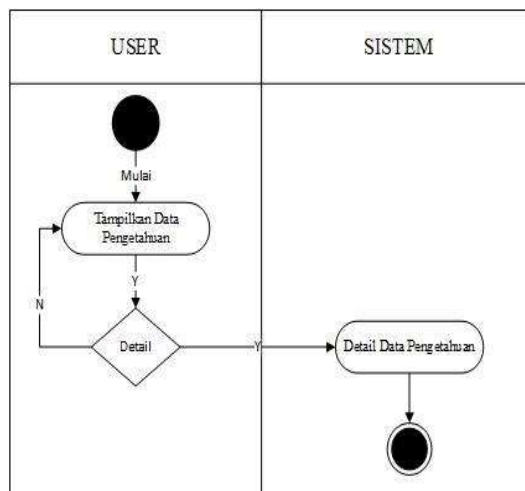
*Activity diagram* diagnosa digunakan untuk dapat mendiagnosa penyakit pada tanaman jagung. Dapat dijelaskan dari aktifitas diagnosa yaitu pertama user menampilkan form pendaftaran kemudian jika user lanjut untuk mendiagnosa maka sistem akan menyimpannya dan menampilkan daftar penyakit kemudian user memilih saah satu penyakit jika user sudah memilih penyakit makan akan menampilkan beberapa pertanyaan atau gejala yang dirasakan oleh user, disetiap jawaban yang dilakukan user tersebut akan disimpan dan sistem akan menanyakan kembali apakah masih ada pertanyaan? Jika tidak ada lagi pertanyaan maka sistem akan memproses diagnosa user dan akan menampilkan data hasil diagnosa ke user, untuk lebih jelasnya dapat dilihat pada gambar berikut:



**Gambar 3.13** *Activity Diagram Data Diagnosa*  
(Sumber: Data Penelitian 2018)

c. *Activity Diagram* Data Pengetahuan

*Activity diagram* data pengetahuan pada user digunakan untuk melihat informasi data pengetahuan yang tersedia di aplikasi sistem pakar ini. Dapat dijelaskan bahwa user menampilkan data pengetahuan dan jika user mengklik menu detail maka sistem akan menampilkan detail lengkap data pengetahuan, penjelasannya di buat pada gambar berikut:



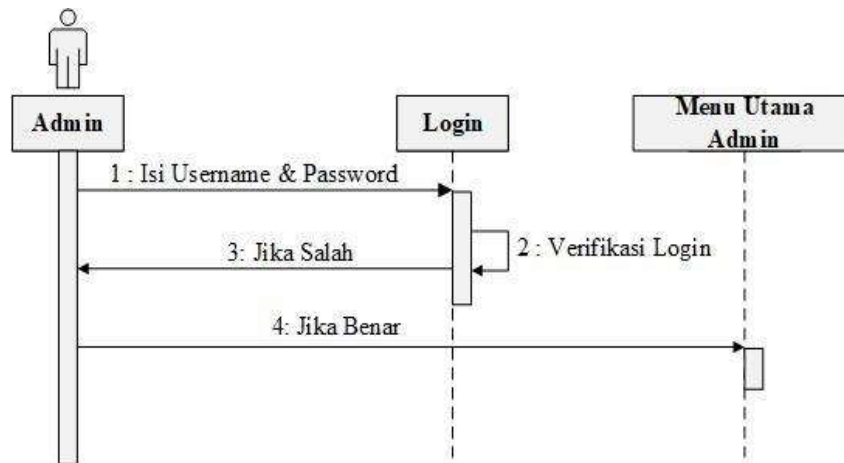
**Gambar 3.14** *Activity Diagram* Data Pengetahuan  
(Sumber: Data Penelitian 2018)

### 3. *Sequence Diagram*

*Sequence diagram* menggambarkan interaksi antara sejumlah objek dalam urutan waktu. Kegunaannya untuk menunjukkan rangkaian pesan yang dikirim antara objek juga interaksi antar objek yang terjadi pada titik tertentu dalam eksekusi sistem. (Isa & Hartawan, 2017).

1. *Sequence Diagram Data Login*

Adapun *Sequence Diagram* form data *login* dapat dilihat pada gambar berikut.

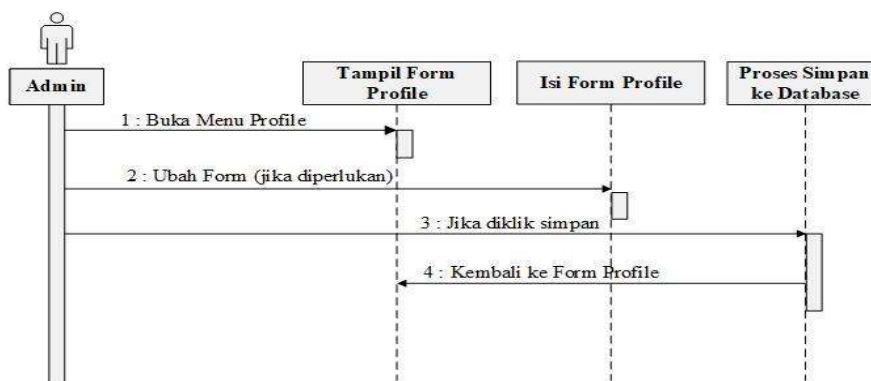


**Gambar 3.15** *Sequence Diagram Data Login*  
(Sumber: Data Penelitian 2018)

- Admin mengisi *username & password* pada halaman *login*,
- Kemudian melakukan verifikasi *login*,
- Jika dinyatakan salah maka kembali ke halaman *login*
- Jika dinyatakan benar maka admin masuk ke menu utama dan bisa melakukan pengolahan data pada aplikasi sistem pakar ini.

## 2. *Sequence Diagram Profile*

Adapun *Sequence Diagram* form *profile* dapat dilihat pada gambar berikut.

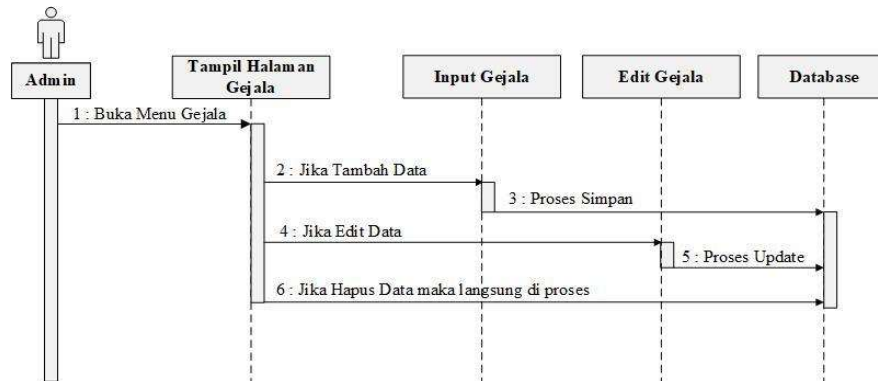


**Gambar 3.16** *Sequence Diagram Data Profile*  
(Sumber: Data Penelitian 2018)

- a. Admin membuka menu *profile* dan menampilkan *form profile*
  - b. Admin mengubah data *profile* jika diperlukan
  - c. Jika diklik simpan maka data *profile* akan diproes dan menyimpannya ke *database*
  - d. Jika proses simpan selesai dilakukan maka akan menampilkan kembali *form profile*
3. *Sequence Diagram Data Gejala*

Adapun *Sequence Diagram* data gejala dapat dilihat pada gambar berikut.





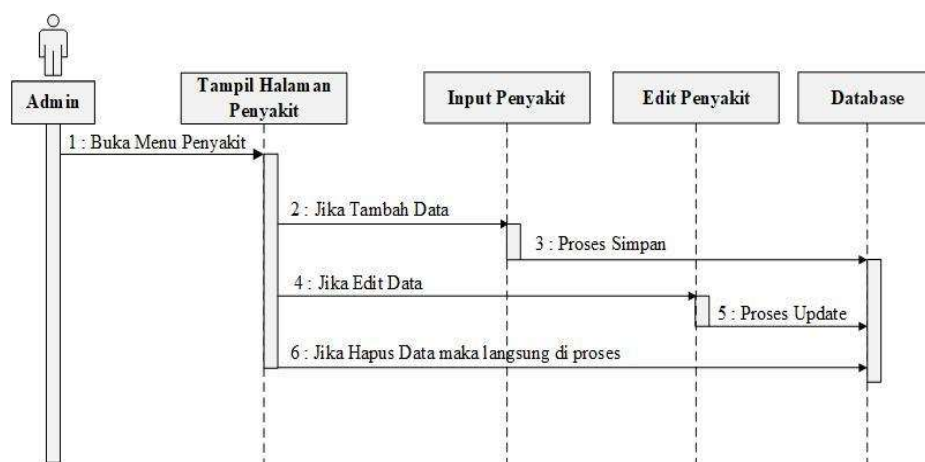
**Gambar 3.17** *Sequence Diagram* Data Gejala  
(Sumber : Data Penelitian 2018)

Pada gambar 3.17 *sequence diagram* data gejala dapat dijelaskan bahwa:

- a. Admin membuka menu gejala kemudian menampilkan halaman gejala
- b. Jika admin ingin menambahkan gejala baru maka input gejala baru
- c. Jika sudah di input gejalanya maka otomatis sistem melakukan proses simpan data ke *database*
- d. Jika admin ingin mengedit atau mengubah data maka edit data gejala.
- e. Jika sudah diedit gejalanya maka otomatis sistem melakukan proses *update* data ke *database*
- f. Jika admin menghapus data gejala maka otomatis sistem menghapus data dari *database*

#### 4. *Sequence Diagram* Data Penyakit

Adapun *Sequence Diagram* data penyakit dapat dilihat pada gambar berikut.



**Gambar 3.18** *Sequence Diagram* Data Penyakit  
(Sumber: Data Penelitian 2018)

Pada gambar 3.18 sequence diagram data penyakit dapat dijelaskan bahwa:

- a. *Admin* membuka menu penyakit kemudian menampilkan halaman penyakit
- b. Jika admin ingin menambahkan penyakit baru maka input penyakit baru
- c. Jika sudah di input penyakitnya maka otomatis sistem melakukan proses simpan data ke *database*,
- d. Jika admin ingin mengedit atau mengubah data maka edit data penyakit
- e. Jika sudah di edit penyakitnya maka otomatis sistem melakukan proses *update* data ke *database*
- f. Jika admin menghapus data penyakit maka otomatis sistem menghapus data dari *database*

## 5. *Sequence Diagram* Data Pengetahuan

Adapun *Sequence Diagram* data pengetahuan dapat dilihat pada gambar berikut.



**Gambar 3.19** *Sequence Diagram* Data Pengetahuan  
(Sumber: Data Penelitian 2018)

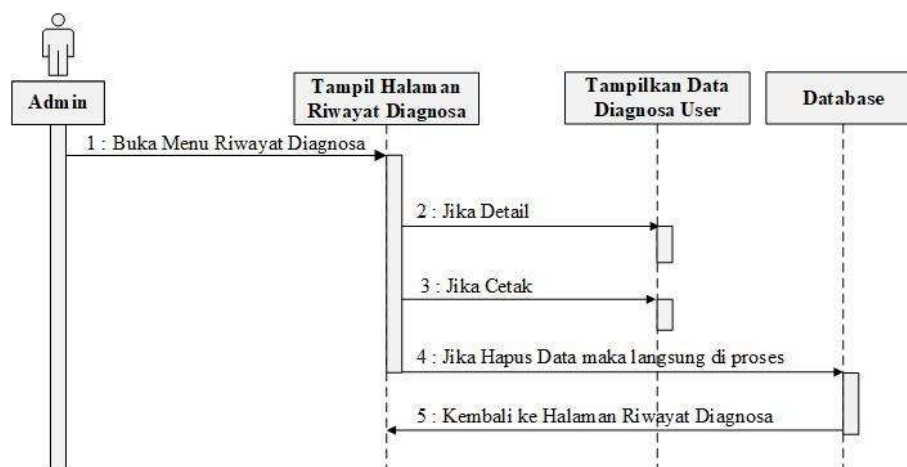
Pada gambar 3.19 *sequence diagram* data pengetahuan dapat dijelaskan bahwa:

- a. Admin membuka menu basis pengetahuan kemudian menampilkan halaman basis pengetahuan
- b. Jika di klik menu atur maka akan menampilkan halaman atur pengetahuan
- c. Jika admin ingin menambahkan pengetahuan baru maka input pengetahuan baru
- d. Jika sudah di input pengetahuannya maka otomatis sistem melakukan proses simpan data ke *database*
- e. Jika admin ingin mengedit atau mengubah data maka edit data pengetahuan
- f. Jika sudah di edit pengetahuannya maka otomatis sistem melakukan proses *update* data ke *database*

- g. Jika admin menghapus data pengetahuan maka otomatis sistem menghapus data dari *database*

#### 6. *Sequence Diagram* Data Riwayat Diagnosa

Adapun *Sequence Diagram* data riwayat diagnosa dapat dilihat pada gambar berikut.



**Gambar 3.20** *Sequence Diagram* Data Riwayat Diagnosa  
(Sumber : Data penelitian 2018)

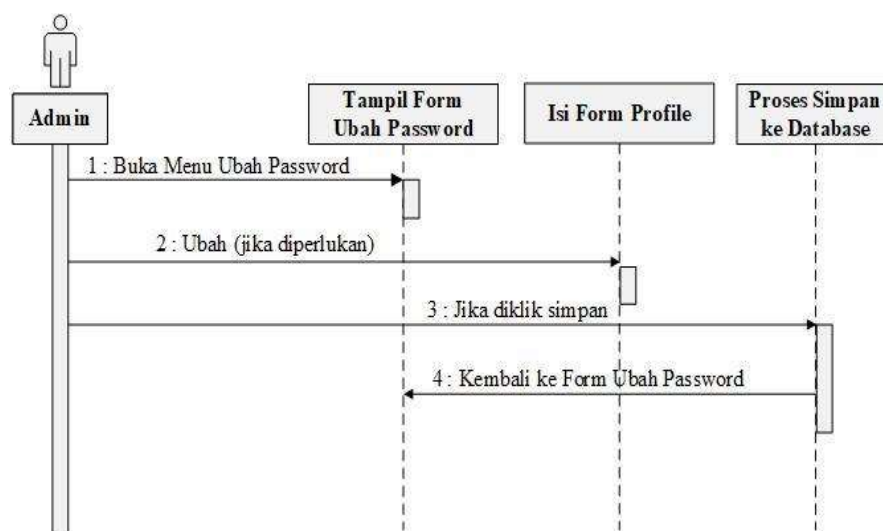
Pada gambar 3.20 *sequence diagram* data riwayat diagnosa dapat dijelaskan bahwa:

- a. Admin membuka menu riwayat diagnosa maka akan menampilkan halaman riwayat diagnosa
- b. Jika admin ingin melihat detail diagnosa maka akan menampilkan data diagnosa *user*.

- c. Jika admin ingin melakukan cetak diagnosa *user* maka sistem akan menampilkan diagnosa *user* yang ingin dicetak.
- d. Jika admin ingin menghapus catatan diagnosa *user* maka otomatis sistem akan menghapus data dari *database*.
- e. Jika data sudah terhapus maka akan menampilkan kembali halaman riwayat diagnosa.

#### 7. *Sequence Diagram Form Ubah Password*

Adapun *Sequence Diagram form ubah password* dapat dilihat pada gambar berikut.



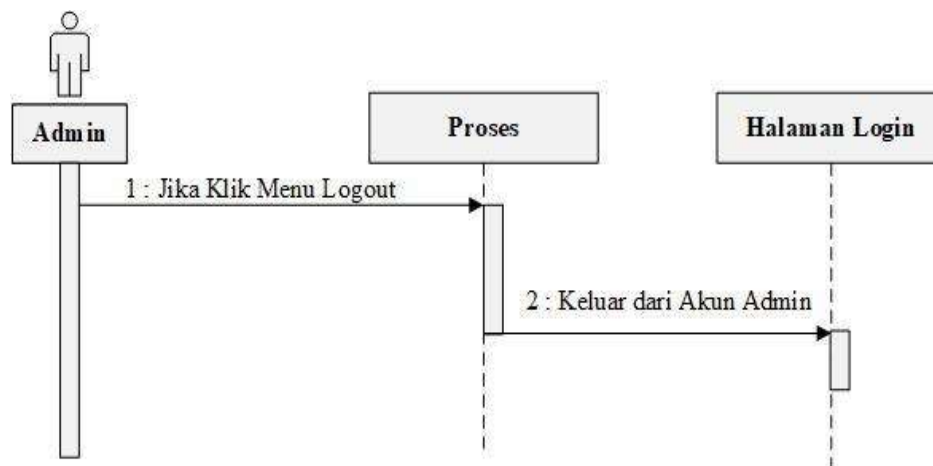
**Gambar 3.21** *Sequence Diagram Ubah Password*  
(Sumber: Data Penelitian 2018)

Pada gambar 3.21 *sequence* diagram form ubah *password* dapat dijelaskan bahwa:

- a. Admin membuka menu ubah *password* dan menampilkan form ubah *password*
- b. Kemudian admin mengubah *password* jika diperlukan
- c. Jika diklik simpan maka data *password* akan diproses dan menyimpannya ke *database*
- d. Jika proses simpan selesai dilakukan maka akan menampilkan kembali *form* ubah *password*

#### 8. *Sequence Diagram Logout*

Adapun *Sequence Diagram logout* dapat dilihat pada gambar berikut.



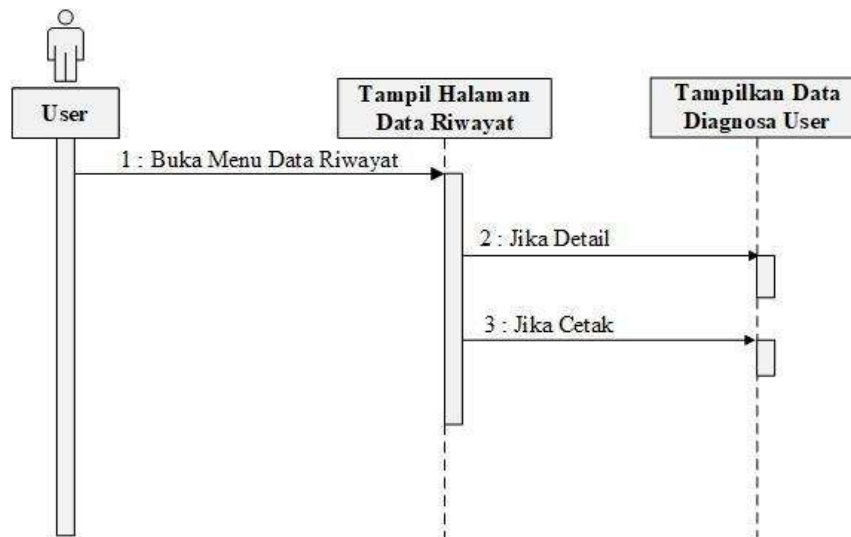
**Gambar 3.22** *Sequence Diagram Logout*  
(Sumber: Data Penelitian 2018)

Pada gambar 3.22 *sequence diagram logout* dapat dijelaskan bahwa:

- a. Jika admin klik menu *logout* maka sistem akan memprosesnya
- b. Jika selesai diproses maka otomatis keluar dari akun admin dan dialihkan ke halaman login

#### 9. *Sequence Diagram Data Riwayat*

Adapun *Sequence Diagram* data riwayat dapat dilihat pada gambar berikut.



**Gambar 3.23** *Sequence Diagram Data Riwayat*  
(Sumber: Data Penelitian 2018)

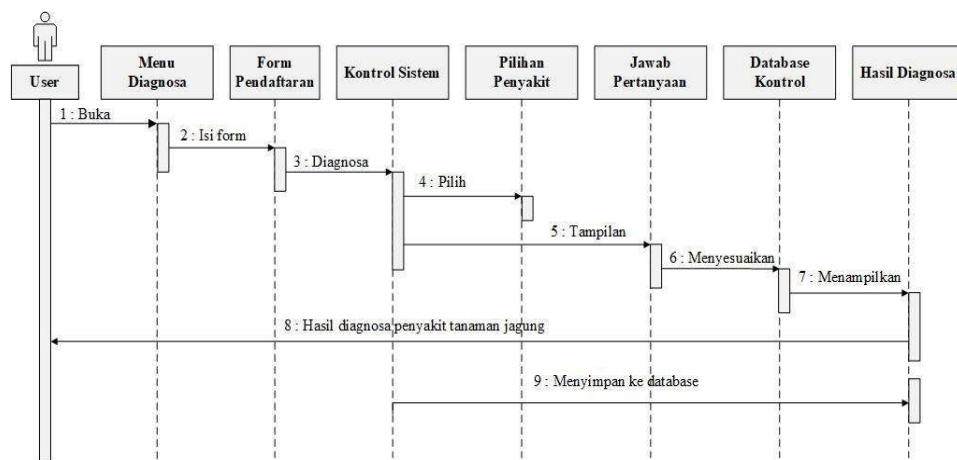
Pada gambar 3.23 *sequence diagram data riwayat* dapat dijelaskan bahwa:

- a. User membuka menu data riwayat maka akan menampilkan halaman data riwayat.
- b. Jika *user* ingin melihat detail diagnosa maka akan menampilkan data diagnosa *user*.

- c. Jika user ingin melakukan cetak diagnosa maka sistem akan menampilkan diagnosa *user* yang di cetak.

#### 10. *Sequence Diagram* Data Diagnosa

Adapun *Sequence Diagram* data riwayat dapat dilihat pada gambar berikut.



**Gambar 3.24** *Sequence Diagram* Data Diagnosa  
(Sumber: Data Penelitian 2018)

Pada gambar 3.24 *sequence* diagram data diagnosa dapat dijelaskan bahwa:

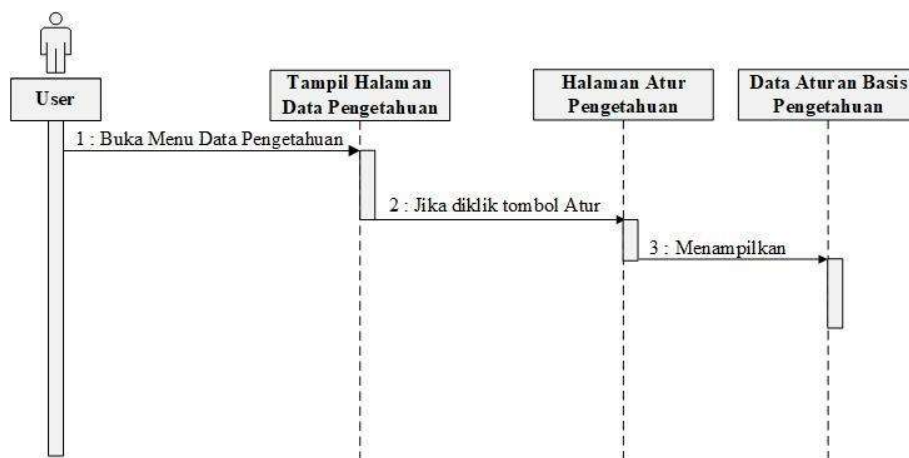
- User membuka menu diagnosa.
- User mengisi data pada *form* pendaftaran.
- user melakukan diagnosa dan sistem akan menampilkan daftar penyakit.
- User wajib memilih salah satu penyakit untuk melanjutkan diagnosa.
- Jika user sudah memilih penyakitnya maka selanjutnya sistem akan menampilkan pertanyaan dan *user* wajib menjawab pertanyaan tersebut.



- f. Jawaban yang dilakukan *user* akan disesuaikan dengan basis data pengetahuan yang tersimpan di *database*
- g. Sistem menampilkan hasil diagnosa berdasarkan basis pengetahuan dan pertanyaan dari *user*
- h. User bisa melihat hasil diagnosa.
- i. Selama diagnosa interaksi yang dilakukan *user* akan selalu disimpan ke *database* untuk menentukan dan menyesuaikan basis data pengetahuan dengan jawaban dari *user*.

#### 11. *Sequence Diagram* Data Pengetahuan

Adapun *Sequence Diagram* data pengetahuan dapat dilihat pada gambar berikut.



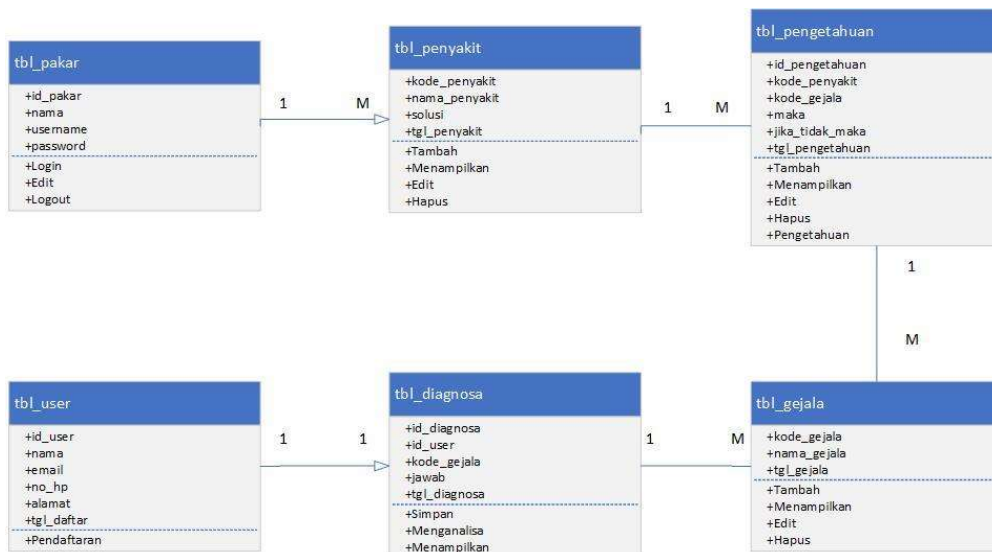
**Gambar 3.25** *Sequence Diagram* Data Pengetahuan  
(Sumber: Data Penelitian 2018)

Pada gambar 3.25 *sequence* diagram data pengetahuan dapat dijelaskan bahwa:

- User* membuka menu data pengetahuan maka akan menampilkan halaman data pengetahuan
- Jika *user* ingin melihat aturan basis pengetahuan maka *user* harus mengklik tombol atur dan menampilkan halaman atur pengetahuan
- Sistem menampilkan data aturan basis pengetahuan

#### 4. *Class Diagram*

*Class* adalah spesifikasi yang akan menghasilkan objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut atau properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metode atau fungsi), (Isa & Hartawan, 2017).



**Gambar 3.26** *Class Diagram*  
(Sumber: Data Penelitian 2018)

### 3.4.3 Desain Database

*Database* adalah sistem terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah diolah atau informasi dan membuat informasi tersedia saat dibutuhkan. *Database* merupakan media untuk menyimpan data agar dapat diakses dengan mudah dan cepat. Kebutuhan *database* dalam sistem informasi yaitu meliputi memasukkan, menyimpan, dan mengambil data, serta membuat laporan berdasarkan data yang telah disimpan. Salah satu bentuk yang dibutuhkan dalam basis data dalam bentuk sistem yaitu *database management system* (DBMS) adalah suatu sistem aplikasi yang digunakan untuk menyimpan, mengelola, dan menampilkan data (Rosa A.S, 2013:43-44).

#### 1. Tabel Pakar

Tabel Pakar berguna untuk menyimpan data nama, *username* dan *password* agar *admin* pakar dapat masuk ke Menu Utama *Admin* dan dapat melakukan manipulasi data.

**Tabel 3.7** Tabel Pakar

Field	Tipe	Panjang	Kunci
id_pakar	int	11	PK
nama	varchar	100	
username	varchar	100	
password	text		

(Sumber: Data Penelitian 2018)

## 2. Tabel Gejala

Tabel ini berguna untuk menyimpan semua daftar gejala.

**Tabel 3.8** Tabel Gejala

Field	Tipe	Panjang	Kunci
kode_gejala	int	10	PK
nama_gejala	text		
tgl_gejala	datetime		

(Sumber: Data Penelitian 2018)

## 3. Tabel Penyakit

Tabel ini berguna untuk menyimpan semua daftar kriteria penyakit dan solusinya.

**Tabel 3.9** Tabel Penyakit

Field	Tipe	Panjang	Kunci
kode_penyakit	int	10	PK
nama_penyakit	text		
solusi	text		
tgl_penyakit	datetime		

(Sumber: Data Penelitian 2018)

## 4. Tabel Pengetahuan

Tabel Pengetahuan berguna untuk data kecerdasan. Tujuan dibuat tabel ini adalah untuk menyimpan daftar kemungkinan potensi kecerdasan pada saat menjawab pertanyaan yang diajukan.

**Tabel 3.10** Tabel Pengetahuan

<b>Field</b>	<b>Tipe</b>	<b>Panjang</b>	<b>Kunci</b>
id_relasi	int	10	PK
kode_penyakit	varchar	10	FK
kode_gejala	varchar	10	FK
maka	text		
jika_tidak_maka	text		
tgl_relasi	datetime		

(Sumber: Data Penelitian 2018)

5. Tabel *User*

Tabel ini berguna untuk menyimpan data *user* dari form pendaftaran.

**Tabel 3.11** Tabel *User*

<b>Field</b>	<b>Tipe</b>	<b>Panjang</b>	<b>Kunci</b>
id_user	varchar	225	PK
nama	varchar	100	
no_hp	varchar	14	
alamat	text		
tgl_daftar	datetime		

(Sumber: Data Penelitian 2018)

## 6. Tabel Diagnosa

Tabel ini berguna untuk menyimpan data hasil analisa diagnosa *user* yang telah selesai menjawab semua pertanyaan yang diajukan sehingga mendapatkan hasil berdasarkan pertanyaan yang telah dijawab.

**Tabel 3. 12** Tabel Diagnosa

Field	Tipe	Panjang	Kunci
id_diagnosa	int	10	PK
id_user	varchar	225	
kode_gejala	varchar	10	
jawab	enum('Ya','Tidak')		
tgl_diagnosa	datetime		

(Sumber: Data Penelitian 2018)

### 3.4.4 Desain Antarmuka

Desain antarmuka merupakan rancangan antarmuka yang akan digunakan untuk mendeskripsikan rencana tampilan dari setiap *form* yang akan digunakan pada tampilan aplikasi sistem pakar yang sebenarnya. Berikut adalah tampilan antarmuka pada sistem pakar penyakit tanaman jagung:

#### 1. Tampilan Halaman Utama *Web*

Halaman utama *web* adalah halaman utama saat mengakses *web* sistem pakar diagnosa penyakit tanaman jagung. Berikut adalah tampilan Halaman utama *web*:

	Beranda	Diagnosa	Data Riwayat	Basis Pengetahuan	Login Pakar
<div style="border: 1px solid black; width: 60px; margin: 0 auto; padding: 2px;">Logo</div> <p style="text-align: center;"> <b>SISTEM PAKAR</b>  <b>MENDIAGNOSA PENYAKIT PADA TANAMAN JAGUNG</b>          METODE BACKWARD CHAINING       </p> <div style="border: 1px solid black; width: 150px; margin: 0 auto; padding: 5px; text-align: center;">         Mulai Diagnosa       </div>					
FOOTER					

**Gambar 3.27** Halaman Utama *Web*  
(Sumber: Data Penelitian 2018)

## 2. Tampilan Halaman Diagnosa

Halaman Diagnosa akan muncul ketika *user* telah selesai melakukan pengisian form pendaftaran. Halaman ini berguna bagi *user* untuk melakukan konsultasi dengan sistem pakar. *User* akan diberikan beberapa pertanyaan yang harus dijawab dengan pilihan ‘Ya’ atau ‘Tidak’. Berikut adalah tampilan halaman diagnosa:

	Beranda	Diagnosa	Data Riwayat	Basis Pengetahuan	Login
<div style="border: 1px solid black; padding: 5px; margin: 0 auto; width: 80%;">         SISTEM PAKAR UNTUK MENDIAGNOS PENYAKIT PADA TANAMAN JAGUNG MENGGUNAKAN METODE BACKWARD CHAINING       </div> <div style="display: flex; flex-direction: column; align-items: flex-start;"> <div style="margin-bottom: 10px;">       Nama Lengkap    : <input style="width: 300px;" type="text"/> </div> <div style="margin-bottom: 10px;">       NO. Hp           : <input style="width: 300px;" type="text"/> </div> <div style="margin-bottom: 10px;">       Alamat           : <input style="width: 300px;" type="text"/> </div> </div> <div style="text-align: right; margin-top: 20px;"> <div style="border: 1px solid black; padding: 2px 10px;">Lanjut Diagnosa</div> </div>					
FOOTER					

**Gambar 3.28** Tampilan Halaman Diagnosa Pendaftaran  
(Sumber: Data Penelitian 2018)

	BERANDA	DIAGNOSA	DATA RIWAYAT	BASIS PENGETAHUAN	LOGIN					
<p>Silahkan pilih salah satu penyakit di bawah ini :</p> <table border="1"> <tr><td>1. P001</td></tr> <tr><td>2. P002</td></tr> <tr><td>3. P003</td></tr> <tr><td>4. P004</td></tr> <tr><td>5. P005</td></tr> </table>						1. P001	2. P002	3. P003	4. P004	5. P005
1. P001										
2. P002										
3. P003										
4. P004										
5. P005										
FOOTER										

**Gambar 3.29** Tampilan Halaman Diagnosa Pilihan Penyakit  
(Sumber: Data Penelitian 2018)

	Beranda	Diagnosa	Data Riwayat	Basis Pengetahuan	Login							
<p>Jawablah pertanyaan berikut sesuai dengan gejala yang ada</p> <table border="1" style="margin: 20px auto;"> <tr> <td colspan="2" style="text-align: center;">GEJALA / PERTANYAAN</td> </tr> <tr> <td style="text-align: center;">Ya</td> <td style="text-align: center;">Tidak</td> </tr> <tr> <td colspan="2" style="text-align: center;"> <table border="1" style="margin: 10px auto;"> <tr> <td>Batal</td> </tr> </table> </td> </tr> </table>						GEJALA / PERTANYAAN		Ya	Tidak	<table border="1" style="margin: 10px auto;"> <tr> <td>Batal</td> </tr> </table>		Batal
GEJALA / PERTANYAAN												
Ya	Tidak											
<table border="1" style="margin: 10px auto;"> <tr> <td>Batal</td> </tr> </table>		Batal										
Batal												
FOOTER												

**Gambar 3.30** Tampilan Halaman Diagnosa Pertanyaan  
(Sumber: Data Penelitian 2018)



	BERANDA	DIAGNOSA	DATA RIWAYAT	BASIS PENGETAHUAN	Login
--	---------	----------	--------------	-------------------	-------

HASIL MENDIAGNOSA PENYAKIT PADA TANAMAN JAGUNG METODE BACKWARD CHAINING

Nama Lengkap :  
NO. Hp :  
Alamat :  
Hasil Backward Chaining :

RIWAYAT PERTANYAAN

1.	----	
2.	----	

FOOTER

**Gambar 3.31** Tampilan Halaman Diagnosa Hasil  
(Sumber: Data Penelitian 2018)

HASIL DIAGNOSA

Nama Lengkap :  
NO. Hp :  
Alamat :  
Hasil Backward Chaining :

RIWAYAT PERTANYAAN

1.	----	
2.	----	

**Gambar 3.32** Tampilan Halaman Diagnosa Cetak  
(Sumber: Data Penelitian 2018)

### 3. Tampilan Halaman Data Riwayat

Halaman riwayat berfungsi untuk melihat riwayat diagnosa *user*. Berikut adalah tampilan Halaman data riwayat:

	Beranda	Diagnosa	Data Riwayat	Basis Pengetahuan	Login
DATA RIWAYAT DIAGNOSA					
FOOTER					

**Gambar 3.33** Halaman Data Riwayat  
(Sumber: Data Penelitian 2018)

### 4. Tampilan Halaman Basis Pengetahuan

Halaman basis pengetahuan berfungsi untuk melihat data informasi masalah seperti data gejala, data penyakit dan data basis pengetahuan. Berikut adalah tampilan Halaman basis pengetahuan:

Logo	Beranda	Diagnosa	Data Riwayat	Data Pengetahuan	Login Pakar
DATA GEJALA					▽
DATA ALTERNATIF					▽
BASIS PENGETAHUAN					▽
FOOTER					

**Gambar 3.34** Halaman Basis Pengetahuan  
(Sumber: Data Penelitian 2018)

### 5. Tampilan Halaman *Login Pakar*

Halaman *login Pakar* berfungsi untuk dapat masuk ke Halaman Utama *Admin* agar dapat memelihara dan memanipulasi sistem. Berikut adalah tampilan Halaman *login pakar*:

SISTEM PAKAR  
MENDIAGNOSA PENYAKIT PADA  
TANAMAN JAGUNG  
METODE BACKWARD CHAINING

**Gambar 3.35** Tampilan Halaman *Login Pakar*  
(Sumber: Data Penelitian 2018)

## 6. Tampilan Halaman Menu Utama *Admin*

Halaman Menu utama *Admin* adalah halaman yang pertama kali muncul pada saat *admin* mengakses sistem. Halaman Menu Utama *admin* akan menampilkan semua konten yang akan digunakan baik sebagai pakar atau sebagai *admin*. Berikut adalah tampilan menu utama *admin*:

HEADER	
Beranda	<div style="border: 1px solid black; padding: 10px; text-align: center;"> <p>SISTEM PAKAR MENDIAGNOSA PENYAKIT PADA TANAMAN JAGUNG METODE BACKWARD CHAINING</p> <hr/> <p>Data Pengetahuan</p> </div>
Profile	
Data Gejala	
Data Penyakit	
Basis Pengetahuan	
Riwayat Diagnosa	
Ubah Password	
Log Out	
FOOTER	

**Gambar 3.36** Tampilan Menu Utama *Admin*  
(Sumber: Data Penelitian 2018)

## 5. Tampilan Halaman Menu *Profile Admin*

Halaman Menu *Profile Admin* adalah halaman untuk mengubah nama dan *username admin*. Berikut adalah tampilan menu *profile admin*:

HEADER	
Beranda	<div style="border: 1px solid black; padding: 10px; text-align: center;"> <div style="border: 1px solid black; padding: 5px; margin: 0 auto; width: 80%;"> <input type="text" value="Username"/> </div> <div style="border: 1px solid black; padding: 5px; margin: 5px auto; width: 80%;"> <input type="text" value="Nama Lengkap"/> </div> <div style="border: 1px solid black; padding: 5px; margin: 5px auto; width: 20%;"> <input type="button" value="Simpan"/> </div> </div>
Profile	
Data Gejala	
Data Penyakit	
Basis Pengetahuan	
Riwayat Diagnosa	
Ubah Password	
Log Out	
FOOTER	

**Gambar 3.37** Tampilan Menu *Profile Admin*  
(Sumber: Data Penelitian 2018)

## 6. Tampilan Halaman Menu Gejala *Admin*

Halaman Menu Gejala *Admin* adalah halaman untuk menambah, mengubah, menghapus dan menampilkan data gejala. Berikut adalah tampilan menu gejala *admin*:

HEADER	
<ul style="list-style-type: none"> <li>Beranda</li> <li>Profile</li> <li>Data Gejala</li> <li>Data Penyakit</li> <li>Basis Pengetahuan</li> <li>Riwayat Diagnosa</li> <li>Ubah Password</li> <li>Log Out</li> </ul>	<div style="border: 1px solid black; padding: 5px; margin-bottom: 5px;">Tambah Gejala</div> <div style="border: 1px solid black; padding: 10px; display: flex; justify-content: space-between;"> <div style="flex-grow: 1;">Data Gejala</div> <div style="text-align: center;"> <div style="border: 1px solid black; padding: 2px 5px;">Edit</div> <div style="border: 1px solid black; padding: 2px 5px;">Hapus</div> </div> </div>
FOOTER	

**Gambar 3.38** Tampilan Menu Gejala *Admin*  
(Sumber: Data Penelitian 2018)

HEADER	
<ul style="list-style-type: none"> <li>Beranda</li> <li>Profile</li> <li>Data Gejala</li> <li>Data Penyakit</li> <li>Basis Pengetahuan</li> <li>Riwayat Diagnosa</li> <li>Ubah Password</li> <li>Log Out</li> </ul>	<div style="border: 1px solid black; padding: 10px; margin: 10px auto; width: 80%;"> <div style="border: 1px solid black; padding: 5px; margin-bottom: 5px; width: 80%; margin-left: auto; margin-right: auto;">Kode Gejala</div> <div style="border: 1px solid black; padding: 5px; margin-bottom: 5px; width: 80%; margin-left: auto; margin-right: auto;">Nama Gejala</div> <div style="display: flex; justify-content: space-around; width: 80%; margin-left: auto; margin-right: auto;"> <div style="border: 1px solid black; padding: 2px 5px;">Data Gejala</div> <div style="border: 1px solid black; padding: 2px 5px;">Simpan</div> </div> </div>
FOOTER	

**Gambar 3.39** Tampilan Menu Gejala *Admin* Tambah dan Edit  
(Sumber: Data Penelitian 2018)

## 7. Tampilan Halaman Menu Penyakit *Admin*

Halaman Menu Penyakit *Admin* adalah halaman untuk menambah, mengubah, menghapus dan menampilkan data kriteria penyakit. Berikut adalah tampilan menu penyakit *admin*:

HEADER				
<ul style="list-style-type: none"> <li>Beranda</li> <li>Profile</li> <li>Data Gejala</li> <li>Data Penyakit</li> <li>Basis Pengetahuan</li> <li>Riwayat Diagnosa</li> <li>Ubah Password</li> <li>Log Out</li> </ul>	<div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"> <p style="text-align: center;">Tambah Penyakit</p> </div> <div style="border: 1px solid black; padding: 10px;"> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 70%; padding: 5px;">Data Penyakit</td> <td style="width: 10%; padding: 5px; text-align: center;">Edit</td> <td style="width: 20%; padding: 5px; text-align: center;">Hapus</td> </tr> </table> </div>	Data Penyakit	Edit	Hapus
Data Penyakit	Edit	Hapus		
FOOTER				

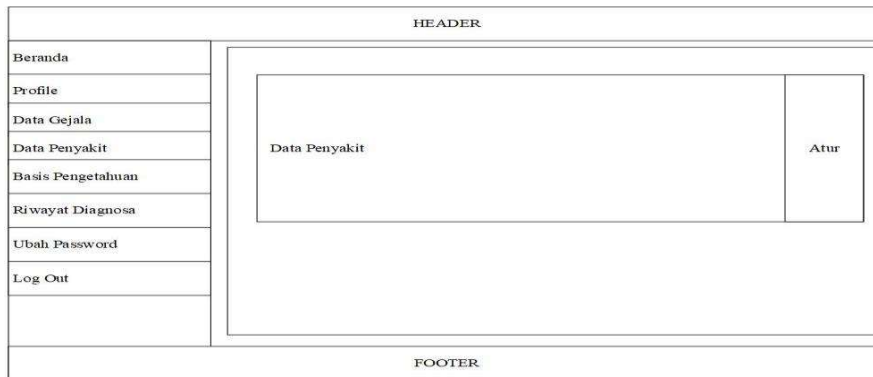
**Gambar 3.40** Tampilan Menu Penyakit *Admin*  
(Sumber: Data Penelitian 2018)

HEADER	
<ul style="list-style-type: none"> <li>Beranda</li> <li>Profile</li> <li>Data Gejala</li> <li>Data Penyakit</li> <li>Basis Pengetahuan</li> <li>Riwayat Diagnosa</li> <li>Ubah Password</li> <li>Log Out</li> </ul>	<div style="border: 1px solid black; padding: 10px; margin: 10px auto; width: 80%;"> <div style="border: 1px solid black; padding: 5px; margin-bottom: 5px; text-align: center;">Kode Penyakit</div> <div style="border: 1px solid black; padding: 5px; margin-bottom: 5px; text-align: center;">Nama Penyakit</div> <div style="border: 1px solid black; padding: 5px; margin-bottom: 5px; text-align: center;">Solusi</div> <div style="display: flex; justify-content: space-between; margin-top: 10px;"> <span>Data Penyakit</span> <span>Simpan</span> </div> </div>
FOOTER	

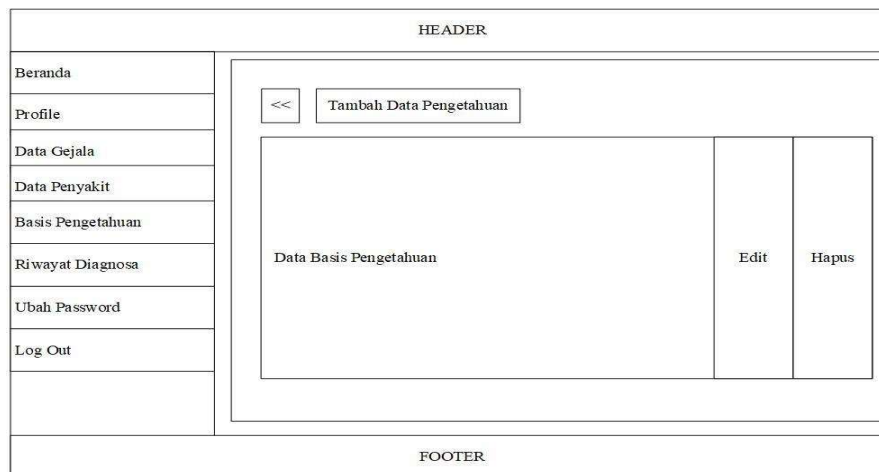
**Gambar 3.41** Tampilan Menu Penyakit *Admin* Tambah dan Edit  
(Sumber: Data Penelitian 2018)

## 8. Tampilan Halaman Menu Basis Pengetahuan *Admin*

Halaman Menu Basis Pengetahuan *Admin* adalah halaman untuk menambah, mengubah, menghapus, menampilkan dan mengatur data basis pengetahuan. Berikut adalah tampilan menu basis pengetahuan *admin*:



**Gambar 3.42** Tampilan Menu Basis Pengetahuan *Admin*  
(Sumber: Data Penelitian 2018)



**Gambar 3.43** Tampilan Menu Atur Basis Pengetahuan *Admin*  
(Sumber: Data Penelitian 2018)

HEADER	
Beranda	<div style="text-align: center;"> <input type="text" value="Nama Penyakit"/>  <input type="text" value="Jika"/> ▾  <input type="text" value="Maka"/> ▾  <input type="text" value="Jika tidak maka"/> ▾  <input type="button" value=" &lt;&lt;"/> <input type="button" value=" Simpan"/> </div>
Profile	
Data Gejala	
Data Penyakit	
Basis Pengetahuan	
Riwayat Diagnosa	
Ubah Password	
Log Out	
FOOTER	

**Gambar 3.44** Tampilan Menu Basis Pengetahuan *Admin* Tambah & Edit  
(Sumber: Data Penelitian 2018)

### 9. Tampilan Halaman Menu Riwayat Diagnosa *Admin*

Halaman Menu Riwayat Diagnosa *Admin* adalah halaman untuk menghapus dan menampilkan data riwayat terakhir *user* yang sudah melakukan diagnosa. Berikut adalah tampilan menu riwayat diagnosa *admin*:

HEADER	
Beranda	<div style="text-align: center;"> <p>Data Riwayat Diagnosa</p> </div>
Profile	
Data Gejala	
Data Penyakit	
Basis Pengetahuan	
Riwayat Diagnosa	
Ubah Password	
Log Out	
FOOTER	

**Gambar 3.45** Tampilan Menu Riwayat Diagnosa *Admin*  
(Sumber: Data Penelitian 2018)

### 10. Tampilan Halaman Menu Ubah *Password Admin*



Halaman Menu Ubah *Password Admin* adalah halaman untuk mengubah *password* admin. Berikut adalah tampilan menu *password admin*:

HEADER	
<ul style="list-style-type: none"> <li>Beranda</li> <li>Profile</li> <li>Data Gejala</li> <li>Data Penyakit</li> <li>Basis Pengetahuan</li> <li>Riwayat Diagnosa</li> <li>Ubah Password</li> <li>Log Out</li> </ul>	<div style="border: 1px solid black; padding: 10px; width: fit-content; margin: 0 auto;"> <input type="text" value="Password Lama"/>  <input type="text" value="Password Baru"/>  <input type="text" value="Ulangi Password Baru"/>  <input type="button" value="Simpan"/> </div>
FOOTER	

**Gambar 3.46** Tampilan Menu Ubah *Password Admin*  
(Sumber: Data Penelitian 2018)

### 3.5 Lokasi dan Jadwal Penelitian

#### 3.5.1 Lokasi Penelitian

Lokasi penelitian yang digunakan oleh penulis dalam penelitian ini adalah Kantor dinas ketahanan pangan dan pertanian kota Matam bersama Bapak Devi Januardi Sartely, S.P, di Jln. Raja Haji No. 03 Batam Kepulauan Riau.

#### 3.5.2 Jadwal Penelitian

Penelitian mengambil waktu selama 1 semester terhitung sejak bulan September 2018 sampai dengan Januari 2019. Sedangkan jadwal penelitian disesuaikan dengan kondisi jadwal yang telah ditetapkan sesuai Tabel berikut ini.

**Tabel 3. 13** Jadwal Kegiatan

No	Kegiatan	Jadwal Kegiatan																												
		Sep 2018				Okt 2018				Nov 2018				Des 2018				Jan 2018				Feb 2018								
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4					
1	Pengajuan judul																													
2	Penyusunan Bab I																													
3	Penyusunan Bab II																													
4	Penyusunan Bab III																													
5	Penyusunan Bab IV																													
6	Penyusunan Bab V, daftar pustaka, lampiran																													
7	Pengumpulan Skripsi																													

(Sumber: Data Penelitian 2018)