

**SISTEM PAKAR DIAGNOSA PENYAKIT PADA AYAM  
MENGGUNAKAN METODE CERTAINTY FACTOR  
BERBASIS WEBSITE RESPONSIVE**

**SKRIPSI**



disusun oleh

**Fajar Januriawan**

**14.11.7858**

**PROGRAM SARJANA  
PROGRAM STUDI INFORMATIKA  
UNIVERSITAS AMIKOM YOGYAKARTA  
YOGYAKARTA  
2018**

**SISTEM PAKAR DIAGNOSA PENYAKIT PADA AYAM  
MENGGUNAKAN METODE CERTAINTY FACTOR  
BERBASIS WEBSITE RESPONSIVE**

**SKRIPSI**

untuk memenuhi sebagian persyaratan  
mencapai gelar Sarjana  
pada Program Studi Informatika



disusun oleh  
**Fajar Januriawan**  
**14.11.7858**

**PROGRAM SARJANA**  
**PROGRAM STUDI INFORMATIKA**  
**UNIVERSITAS AMIKOM YOGYAKARTA**  
**YOGYAKARTA**  
**2018**

## **PERSETUJUAN**

### **SKRIPSI**

#### **SISTEM PAKAR DIAGNOSA PENYAKIT PADA AYAM MENGGUNAKAN METODE CERTAINTY FACTOR BERBASIS WEBSITE RESPONSIVE**

yang dipersiapkan dan disusun oleh

**Fajar Januriawan**

**14.11.7858**

telah disetujui oleh Dosen Pembimbing Skripsi  
pada tanggal 10 April 2018

Dosen Pembimbing,

**Kusrini, Dr., M.Kom**  
**NIK. 190302106**

**PENGESAHAN**  
**SKRIPSI**  
**SISTEM PAKAR DIAGNOSA PENYAKIT PADA AYAM**  
**MENGGUNAKAN METODE CERTAINTY FACTOR**  
**BERBASIS WEBSITE RESPONSIVE**

yang dipersiapkan dan disusun oleh

**Fajar Januriawan**

**14.11.7858**

telah dipertahankan di depan Dewan Pengaji  
pada tanggal 15 Maret 2018

**Susunan Dewan Pengaji**

**Nama Pengaji**

Dina Maulina, M.Kom  
NIK. 190302250

**Tanda Tangan**



Ike Verawati, M.Kom  
NIK. 190302237

Kusrini, Dr., M.Kom  
NIK. 190302106

Skripsi ini telah diterima sebagai salah satu persyaratan  
untuk memperoleh gelar Sarjana Komputer  
Tanggal 10 April 2018

**DEKAN FAKULTAS ILMU KOMPUTER**



Krisnawati, S.Si, M.T  
NIK. 190302038

## **MOTTO**

*“Nasib memang diserahkan kepada manusia untuk digarap, tetapi takdir harus ditandatangi di atas materai dan tidak boleh digugat kalau nanti terjadi apa-apa, baik atau buruk. Kata yang ada di Langit sana, kalau baik ya alhamdulillah, kalau buruk ya disyukuri saja. ” - Sapardi Djoko Damono*

## DAFTAR ISI

<b>JUDUL .....</b>	i
<b>PERSETUJUAN.....</b>	ii
<b>PENGESAHAN .....</b>	iii
<b>PERNYATAAN.....</b>	iv
<b>MOTTO .....</b>	v
<b>PERSEMBAHAN.....</b>	vi
<b>KATA PENGANTAR.....</b>	vii
<b>DAFTAR ISI.....</b>	ix
<b>DAFTAR TABEL .....</b>	xii
<b>DAFTAR GAMBAR.....</b>	xiii
<b>INTISARI .....</b>	xvi
<b>ABSTRACT .....</b>	xvii
<b>BAB I PENDAHULUAN.....</b>	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah .....	4
1.3 Batasan Masalah.....	4
1.4 Maksud dan Tujuan Penelitian .....	5
1.5 Metode Penelitian .....	5
1.5.1 Metode Pengumpulan Data.....	5
1.5.2 Metode Analisis .....	6
1.5.3 Metode Perancangan.....	6
1.5.4 Metode <i>Testing</i> .....	7
1.5.5 Metode Implementasi .....	7
1.6 Sistematika Penulisan .....	7
<b>BAB II LANDASAN TEORI .....</b>	9
2.1 Tinjauan Pustaka .....	9
2.2 Teori Dasar Ayam .....	12
2.3 Sistem Pakar .....	14
2.5.1 Pengertian Sistem Pakar .....	14

2.5.2	Pemakaian Sistem Pakar .....	16
2.5.3	Ciri-ciri Sistem Pakar .....	17
2.5.4	Keuntungan Pemakaian Sistem Pakar .....	17
2.5.5	Arsitektur Sistem Pakar .....	18
2.5.6	Orang Yang Terlibat Dalam Sistem Pakar .....	20
2.5.7	Kategori Masalah Sistem Pakar .....	21
2.4	Representasi Pengetahuan.....	22
2.5	Mesin Inferensi .....	23
2.5.1	Runut maju ( <i>forward chaining</i> ) .....	24
2.5.2	Runut balik ( <i>backward chaining</i> ) .....	24
2.6	Certainty Factor (Faktor Kepastian) .....	25
2.6.1	Bentuk <i>Certainty Factor</i> .....	26
2.6.2	Kombinasi Aturan.....	27
2.7	Konsep Basis Data.....	28
2.8	ERD (Entity Relationship Diagram) .....	28
2.9	Unified Modeling Language (UML) .....	31
2.9.1	<i>Use Case Diagram</i> .....	31
2.9.2	<i>Class Diagram</i> .....	33
2.9.3	<i>Sequence Diagram</i> .....	34
2.9.4	<i>Activity Diagram</i> .....	35
2.10	Konsep OOP ( <i>Object Oriented Programming</i> ).....	35
2.11	Perangkat Lunak yang Digunakan.....	37
2.12	Bahasa Pemrograman .....	37
2.13	PHP (Hypertext Preprocessor).....	37
2.13.1	Pengertian PHP .....	37
2.13.2	Skrip PHP .....	38
2.14	MySQL .....	38
	<b>BAB III ANALISIS DAN PERANCANGAN .....</b>	<b>40</b>
3.1	Gambaran Umum Aplikasi .....	40
3.2	Analisis Masalah .....	41
3.2.1	Identifikasi Masalah.....	41
3.3	Arsitektur Sistem .....	43
3.4	Representasi Pengetahuan .....	44
3.4.1	Analisis Pengetahuan .....	45
3.4.2	Daftar Penyakit .....	45
3.4.3	Daftar Gejala.....	46
3.4.4	Rekomendasi Pakar .....	49
3.4.5	Aturan Dan Aturan Produksi ( <i>Production Rules</i> ).....	60
3.4.6	Inferensi .....	62
3.4.7	Teknik Inferensi.....	63
3.4.8	Proses Perhitungan Pada Aplikasi Sistem Pakar .....	66
3.4.9	Perhitungan Manual Certainty Factor .....	69

3.4.10 Inferensi Diagnosa .....	76
3.5 Struktur Program & Perancangan sistem .....	78
3.5.1 Solusi Yang Dapat Diterapkan .....	78
3.5.2 Solusi Yang Dipilih .....	79
3.6 Perancangan UML .....	79
3.6.1 <i>Use Case Diagram</i> .....	79
3.6.2 <i>Activity Diagram</i> .....	82
3.6.3 <i>Class Diagram</i> .....	99
3.6.4 <i>Sequence Diagram</i> .....	100
3.7 Perancangan Basis Data.....	108
3.8 Perancangan <i>User Interface</i> .....	113
3.8.1 Halaman Beranda.....	113
3.8.2 Halaman Diagnosa.....	114
3.8.3 Halaman Hasil Diagnosis .....	115
3.8.4 Halaman Keterangan Penyakit .....	116
3.8.5 Halaman Tentang .....	117
3.8.6 Halaman Login .....	117
3.8.7 Halaman Pakar .....	118
<b>BAB IV IMPLEMENTASI DAN PEMBAHASAN .....</b>	<b>121</b>
4.1 Implementasi .....	121
4.1.1 Implementasi Perangkat Lunak .....	121
4.1.2 Implementasi Antar Muka .....	122
4.1.3 Implementasi Instalasi Program .....	133
4.2 Uji Coba Sistem dan Program .....	136
4.2.1 <i>Black-box Testing</i> .....	137
4.2.2 <i>White-box Testing</i> .....	140
4.2.3 Kesalahan Kode Program ( <i>Syntax Error</i> ) .....	141
4.2.4 Kesalahan Proses ( <i>Run Time Error</i> ) .....	142
4.2.5 Kesalahan Logika .....	142
4.2.6 Kesalahan Koneksi Database ( <i>Database Connection</i> ) .	142
4.3 Pengujian Hasil Diagnosa.....	142
4.4 Implementasi Kode Program .....	147
<b>BAB V PENUTUP .....</b>	<b>148</b>
5.1 Kesimpulan.....	148
5.2 Saran .....	149
<b>DAFTAR PUSTAKA .....</b>	<b>150</b>
<b>LAMPIRAN .....</b>	<b>153</b>

## DAFTAR TABEL

Tabel 2.1	Perbandingan Jurnal Terkait .....	11
Tabel 2.2	Kombinasi Evidence Anteseden .....	27
Tabel 2.3	Simbol <i>Use Case Diagram</i> .....	31
Tabel 2.4	Simbol <i>Class Diagram</i> .....	33
Tabel 2.5	Simbol <i>Sequence Diagram</i> .....	34
Tabel 2.6	Simbol <i>Activity Diagram</i> .....	35
Tabel 3.1	Jenis-Jenis Penyakit .....	45
Tabel 3.2	Gejala-Gejala Penyakit.....	46
Tabel 3.3	Tabel Basis Pengetahuan Diagnosa Penyakit .....	47
Tabel 3.4	Rekomendasi Pakar.....	49
Tabel 3.5	Daftar Aturan Diagnosa .....	60
Tabel 3.6	Tabel Interpretasi Certainty Factor.....	63
Tabel 3.7	Representasi Gejala Penyakit Ayam .....	64
Tabel 4.1	<i>Black-box Testing</i> Menu Utama.....	137
Tabel 4.2	<i>Black Box Testing</i> Menu Mulai Diagnosa.....	137
Tabel 4.3	<i>Black Box Testing</i> Menu Pakar .....	137
Tabel 4.4	<i>Black Box Testing</i> Menu Admin .....	138
Tabel 4.5	<i>Black Box Testing</i> Menu Penyakit .....	138
Tabel 4.6	<i>Black Box Testing</i> Menu Gejala.....	139
Tabel 4.7	<i>Black Box Testing</i> Menu Pengetahuan.....	139
Tabel 4.8	<i>Black Box Testing</i> Ubah Password.....	140
Tabel 4.8	Pengujian Hasil Diagnosa .....	143

## DAFTAR GAMBAR

Gambar 2.1	Struktur Sistem Pakar.....	15
Gambar 2.2	Arsitektur Sistem Pakar.....	20
Gambar 2.3	Contoh Penggambaran ERD .....	29
Gambar 2.4	Simbol Dasar Notasi Diagram E-R Dasar.....	30
Gambar 2.5	Derajat Relasi Notasi Diagram E-R Dasar.....	30
Gambar 2.6	Kardinalitas Relasi Notasi Diagram E-R Dasar.....	30
Gambar 3.1	Arsitektur Sistem.....	44
Gambar 3.2	Algoritma Diagnosa Penyakit Ayam .....	77
Gambar 3.3	Struktur Program Sistem Pakar Ayam .....	78
Gambar 3.4	Use Case Diagram User Aplikasi Sistem Pakar Ayam.....	80
Gambar 3.5	Use Case Diagram Pakar Aplikasi Sistem Pakar Ayam .....	81
Gambar 3.6	Activity Diagram Menu Beranda .....	82
Gambar 3.7	Activity Diagram Menu Diagnosa .....	83
Gambar 3.8	<i>Activity Diagram</i> Menu Keterangan .....	85
Gambar 3.9	<i>Activity Diagram</i> Menu Tentang.....	86
Gambar 3.10	<i>Activity Diagram</i> Menu <i>Login</i> .....	87
Gambar 3.11	<i>Activity Diagram</i> Menu Admin.....	88
Gambar 3.12	<i>Activity Diagram</i> Menu Penyakit.....	90
Gambar 3.13	<i>Activity Diagram</i> Menu Gejala .....	92
Gambar 3.14	<i>Activity Diagram</i> Menu Pengetahuan .....	94
Gambar 3.15	<i>Activity Diagram</i> Menu Ubah <i>Password</i> .....	96
Gambar 3.16	<i>Activity Diagram</i> Menu <i>Logout</i> .....	98
Gambar 3.17	<i>Class Diagram</i> Sistem Pakar Ayam.....	99
Gambar 3.18	<i>Sequence Diagram</i> Beranda / <i>Main Activity</i> .....	100
Gambar 3.19	<i>Sequence Diagram</i> Diagnosa Penyakit .....	101
Gambar 3.20	<i>Sequence Diagram</i> Keterangan.....	101
Gambar 3.21	<i>Sequence Diagram</i> Tentang .....	102
Gambar 3.22	<i>Sequence Diagram</i> <i>Login</i> .....	102
Gambar 3.23	<i>Sequence Diagram</i> Admin .....	103

Gambar 3.24	<i>Sequence Diagram</i> Penyakit .....	104
Gambar 3.25	<i>Sequence Diagram</i> Gejala.....	105
Gambar 3.26	<i>Sequence Diagram</i> Pengetahuan.....	106
Gambar 3.27	<i>Sequence Diagram</i> Ubah Password.....	107
Gambar 3.28	<i>Sequence Diagram</i> Logout.....	107
Gambar 3.29	Rancangan Relasi Antar Tabel.....	110
Gambar 3.30	Entity Relationship Diagram (ERD) .....	110
Gambar 3.31	Relasi Antartabel .....	111
Gambar 3.32	Menu Beranda .....	114
Gambar 3.33	Menu Diagnosa .....	115
Gambar 3.34	Menu Hasil Diagnosis .....	116
Gambar 3.35	Menu Keterangan Penyakit .....	116
Gambar 3.36	Menu Tentang .....	117
Gambar 3.37	Menu Login.....	118
Gambar 3.38	Menu Admin .....	119
Gambar 3.39	Menu Penyakit .....	119
Gambar 3.40	Menu Gejala.....	120
Gambar 3.41	Menu Basis Pengetahuan .....	120
Gambar 4.1	Halaman Beranda Sistem Pakar .....	122
Gambar 4.2	Halaman Diagnosa Sistem Pakar .....	123
Gambar 4.3	Halaman Hasil Diagnosis .....	124
Gambar 4.4	Halaman Riwayat .....	124
Gambar 4.5	Halaman Keterangan .....	125
Gambar 4.6	Halaman Harga.....	126
Gambar 4.7	Halaman Login Pakar .....	126
Gambar 4.8	Halaman Gagal Login .....	127
Gambar 4.9	Halaman Tentang .....	128
Gambar 4.10	Halaman Admin Pakar .....	128
Gambar 4.11	Halaman Penyakit .....	129
Gambar 4.12	Halaman Gejala.....	130
Gambar 4.13	Halaman Pengetahuan.....	130

Gambar 4.14	Halaman Post Keterangan .....	131
Gambar 4.15	Halaman Ubah Password .....	132
Gambar 4.16	Halaman LogOut.....	132
Gambar 4.17	Bahan yang diperlukan untuk instalasi .....	133
Gambar 4.18	Tampilan <i>Setup</i> pilih komponen XAMPP .....	134
Gambar 4.19	Memulai Proses Instalasi XAMPP.....	135
Gambar 4.20	XAMPP Control Panel.....	136
Gambar 4.21	Mengakses Aplikasi Melalui peramban .....	136
Gambar 4.22	Proses Pengecekan error log Apache .....	141
Gambar 4.23	White-box Testing log Apache .....	141

## INTISARI

Sistem pakar yang mampu mendiagnosa penyakit pada ayam berdasarkan pengetahuan yang diberikan langsung dari pakar/ahlinya dan melalui studi literatur. Penelitian ini menggunakan metode perhitungan Certainty Factor (CF) dalam menghitung tingkat kepakaran. Data penelitian ini terdiri dari data gejala dan data penyakit ayam, serta data aturan. Sistem pakar pada organisasi ditujukan untuk penambahan value, peningkatan produktivitas serta area manajerial yang dapat mengambil kesimpulan dengan cepat.

Begitu pula dengan organisasi yang melakukan bisnis peternakan, yang sangat menjanjikan, namun perlu kewaspadaan yang tinggi terhadap adanya serangan penyakit, seperti halnya unggas (ayam) yang sangat rentan terhadap berbagai jenis penyakit yang disebabkan virus ataupun bakteri. Manfaat yang diperoleh dari sistem pakar yang mampu melakukan diagnosis dengan cepat, tepat dan akurat terhadap gejala penyakit yang ditimbulkan diharapkan mampu membantu para peternak dalam mengantisipasi kerugian yang diakibatkan serangan penyakit.

Diperlukan keakuratan dan ketepatan penghitungan dalam mendiagnosis gejala penyakit guna menyimpulkan hasil dengan menggunakan metode certainty factors (CFs).

**Kata Kunci:** *Certainty Factor, Sistem Pakar*

## **ABSTRACT**

*Expert systems capable of diagnosing disease in chickens based on the knowledge given directly from experts / experts and through literature studies. This research uses Certainty Factor (CF) calculation method to calculate the level of expertise. This research data consist of data of symptoms and data of chicken disease, and data of rule. Expert systems in the organization aimed at adding value, increased productivity and managerial areas that can draw conclusions quickly.*

*Similarly, organizations that do business farms, which is very promising, but the need for high vigilance against the presence of disease, such as poultry (chicken) are very vulnerable to various types of diseases caused by viruses or bacteria. Benefits derived from expert systems that are able to make a diagnosis quickly, accurately and accurately against the symptoms of the disease is expected to help farmers in anticipation of losses caused by disease.*

*Required accuracy and precision calculation in diagnosing symptoms of disease to conclude results using certainty factors (CFs) method.*

**Keyword:** *Certainty Factor, Expert System*

## **BAB I**

### **PENDAHULUAN**

#### **1.1 Latar Belakang**

Kesehatan merupakan hal yang paling penting bagi makhluk hidup. Tidak hanya manusia yang membutuhkan kesehatan tetapi hewan juga membutuhkannya. Dalam suatu peternakan ayam, dapat terjadi banyak sekali variasi penyakit yang sudah sangat dipahami atau familiar bagi peternak terutama peternak skala menengah dan besar. Berbicara keberhasilan mengenai peternakan (tanpa tergantung skala bisnisnya) oleh seorang peternak ditentukan dari pengetahuan dan pemahaman dengan pengenalan sumber hambatan dan ancaman dari penyakit yang mungkin dapat menjadikan ledakan penyakit menular dan berakibat sangat merugikan. Oleh sebab itu, pengamanan dan menjauhkan ternak ayam dari sumber wabah dan hambatan potensial tersebut menjadi prioritas dan perhatian khusus.

Menurut Badan Pusat Statistik di tahun 2014 ayam merupakan unggas yang paling banyak dipelihara dan di ternakan oleh masyarakat, karena populasinya yang banyak dan kebutuhan masyarakat yang besar pula. Dari data tersebut, jumlah ayam yang diternakkan di Indonesia, baik ayam buras, ayam ras pedaging, atau ayam ras petelur, totalnya mencapai 1.934.065.744 ekor pada tahun 2015 dan angka ini terus bertambah setiap tahunnya. Pemilihan indukan yang unggul, pengelolaan yang baik, sanitasi, peningkatan daya tahan ayam dengan vaksinasi dan usaha menjauhkan ternak ayam dari sumber penyakit adalah

kunci sukses dalam beternak ayam. Tetapi kurangnya informasi pengetahuan dan pemahaman dalam pengenalan suatu penyakit dapat mengakibatkan kesalahan diagnosis dan pengobatan suatu penyakit pada ayam.

Penyebaran ayam secara keseluruhan juga paling banyak terdapat pada Provinsi Jawa Barat dengan jumlah 720.279.514 ekor ayam pada tahun 2015. Walaupun jumlah ayam yang diternakkan semakin bertambah setiap tahunnya, akan tetapi jumlah peternak terutama peternak perorangan semakin berkurang diambil Badan Pusat Statistik, Jumlah Perusahaan Peternakan Ternak Besar dan Kecil. Pada tahun 2000, Badan Pusat Statistik mencatat bahwa terdapat 387 peternak perorangan. Sedangkan mulai dari tahun 2008 sampai tahun 2014, jumlah peternak perorangan itu tidak tercatat lagi di Badan Pusat Statistik. Hal ini dapat disebabkan karena banyaknya ayam ternak yang terserang penyakit sehingga membuat peternak mengalami kerugian karena gagal panen. Peternak yang gagal dalam beternak sebagian besar adalah peternak perorangan, karena minimnya pengetahuan tentang penyakit dan penyebaran dokter hewan yang tidak merata di semua daerah. Oleh karena itu dibutuhkan sebuah sistem yang dapat memberikan rekomendasi tentang penyakit ayam dan salah satu caranya yaitu dengan membuat aplikasi yang menerapkan sistem pakar.

Jenis penyakit ayam yang tidak diketahui sebagai virus alami atau parasit yang menyebabkan timbulnya penyakit. Infeksi yang terdapat pada ayam dapat menularkan ke ayam lainnya. Sulitnya peternak mendapatkan informasi tentang jenis-jenis penyakit pada ayam yang akan menghambat dan merupakan ancaman besar bagi para peternak.

Informasi yang menampilkan penyakit ayam dianggap sangat sedikit. Sulitnya pasien dalam mendiagnosa penyakit ayam secara dini, akan menyebabkan keterlambatan atas penangulangan penyakit dan kematian ayam. Hal ini, merugikan bagi peternak dengan berkurangnya populasi jumlah ternaknya. Begitu juga dengan ayam lainnya bisa tertular penyakit yang disebabkan oleh ayam yang sudah terinfeksi sebelumnya.

Gejala penyakit yang muncul tiba-tiba harus ditangani dengan cepat dan tepat karena jika tidak penyakit pada ayam dapat menular ke ayam lainnya maupun ke manusia itu sendiri dengan virus seperti *Avian influenza* yang sering menjangkit pada unggas. Tingginya tingkat kematian dipengaruhi oleh keterlambatan penanganan pasien, sehingga kondisi pasien memburuk. Penangan yang tepat dan cepat dapat ditangani melalui alat bantu yang dapat mengetahui penyakit dari gejala-gejala yang diderita atau dirasakan dan dapat memberikan solusi penanganan awal dalam mengatasi penyakit yang di derita.

Perkembangan teknologi saat ini dapat dimanfaatkan untuk mendekteksi gejala-gejala secara dini penyakit pada ayam dengan memanfaatkan konsep sistem pakar. Pemanfaatan sistem pakar dapat digunakan salah satunya untuk mengdiagnosis penyakit ayam pada lingkungan peternakan dengan lebih cepat, mudah dan terjangkau dengan berbasis *website* dimana bisa di akses di perangkat *desktop* maupun *mobile*.

Berdasarkan permasalahan yang muncul, dalam penelitian ini dibuat suatu sistem pakar dengan judul “**SISTEM PAKAR DIAGNOSA PENYAKIT PADA AYAM MENGGUNAKAN METODE CERTAINTY FACTOR**” yang

dapat membantu pasien atau *user* dalam mendiagnosa gejala penyakit ayam yang diderita. Adapun *output* yang akan dihasilkan dari sistem ini adalah jenis penyakit, detail penyakit dan saran menangani penyakit yang diderita.

## 1.2 Rumusan Masalah

Berdasarkan latar belakang yang diangkat, maka dibuat rumusan masalah yang akan diselesaikan dalam penelitian ini adalah “Bagaimana merancang suatu aplikasi sistem pakar yang dapat mendiagnosa penyakit ayam dengan menggunakan metode *certainty factor* yang di aplikasikan pada laman website?”

## 1.3 Batasan Masalah

Untuk membatasi pembatasan agar terfokuskan pada aspek yang diangkat sebagai pembahasan utama dalam penelitian ini, maka perlu dibuat adanya batasan-batasan masalah, antara lain:

1. Pembuatan sistem pakar ini hanya mendiagnosa penyakit Pada Ayam dengan mencocokan gejala yang ada dan mendapatkan hasil dari pencocokan yang sudah dilakukan.
2. Pembuatan sistem pakar ini ditujukan untuk membantu peternak dan penghobi ayam agar tahu gejala dan penyakit yang diderita, serta untuk pengetahuan tambahan untuk pencegahan lebih dini.
3. Menggunakan metode *Certainty Factor* untuk perhitungan dan kepastiannya.
4. Referensi tentang penyakit ayam diperoleh dari literatur/buku medis, internet, dan pakar/dokter.

5. Hasil akhir diagnosa menampilkan definisi penyakit, gejala penyakit, penyebab, pencegahan, tindakan umum, pengobatan.
6. Hasil diagnosa bersifat deteksi awal dan masih dibutuhkan pemeriksaan lebih lanjut oleh dokter dan tindakan lab untuk memastikan penyakit secara tepat.
7. Sistem pakar yang dibuat berbasis website responsive yang kompatibel dengan semua perangkat desktop dan mobile.
8. Sumber pakar dari Drh. Rizki Yuni Utami yang merupakan lulusan Fakultas Kedokteran Hewan Universitas Gajah mada.

#### **1.4 Maksud dan Tujuan Penelitian**

Tujuan dari pembuatan aplikasi sistem pakar ini adalah sebagai berikut:

1. Merancang dan membangun aplikasi sistem pakar yang dapat membantu peran masyarakat dalam mengetahui penyakit ayam berdasarkan gejala-gejala yang ditimbulkan dengan cepat dan tepat berbasis website dengan menggunakan metode *certainty factor*.
2. Memudahkan peternak dalam mendiagnosa penyakit ayam yang diderita.

#### **1.5 Metodologi Penelitian**

Dalam penelitian ini data-data yang digunakan didapat dari beberapa metode, antara lain:

##### **1.5.1 Metode Pengumpulan Data**

Dalam penelitian ini data-data yang digunakan di dapat dari beberapa metode, antara lain:

### 1. Metode Wawancara

Dalam metode ini, pengumpulan data yang dilakukan adalah mewawancarai dokter hewan.

### 2. Metode Observasi

Dalam metode ini, dilakukan observasi pada ayam yang terkena penyakit bakterial, mikal maupun viral untuk pengumpulan data-data yang akan dibutuhkan oleh sistem.

### 3. Metode Studi Literatur

Pengumpulan data dalam metode ini dilakukan dengan cara mempelajari dan memahami berbagai literatur seperti buku, jurnal ilmiah, situs-situs internet dan berbagai bahan lain yang berkaitan dengan topik penelitian.

## **1.5.2 Metode Analisis**

Merupakan tahapan dalam menganalisis atau mendefinisikan permasalahan yang akan dibangun. Adapun metode analisis yang digunakan adalah sebagai berikut:

1. Analisis Masalah dengan menggunakan metode respresentasi pengetahuan, aturan produksi, dan inferensi.
2. Analisis kebutuhan sistem yang terdiri dari analisis kebutuhan fungsional dan analisis kebutuhan non fungsional.

## **1.5.3 Metode Perancangan**

Merupakan tahapan dalam merancang proses yang terjadi pada sistem, serta relasi yang terdapat dalam *Database*. Perancangan UML (*Unified*

*Modeling Language*) untuk memvisualisasikan proses yang terjadi pada sistem dan merancang *interface*, untuk membuat tampilan sistem bagi *user*.

#### **1.5.4 Metode Testing**

Merupakan tahapan untuk menguji coba sistem aplikasi. Ada dua jenis pengujian yang dilakukan yaitu sebagai berikut:

1. Pengujian *White box testing*, yaitu pengujian per modul.
2. Pengujian *Black box testing*, yaitu pengujian secara terintegrasi.

#### **1.5.5 Metode Implementasi**

Tahap implementasi merupakan tahap dimana sistem telah melewati proses pengujian dan dinyatakan bekerja sesuai fungsinya dan layak digunakan oleh pengguna.

### **1.6 Sistematika Penulisan**

Sistematika penulisan bertujuan untuk mempermudah dalam penulisan laporan skripsi. Adapun sistematika penulisan pada laporan perancangan aplikasi sistem pakar ini adalah sebagai berikut:

## **BAB I PENDAHULUAN**

Bab ini berisi pengantar pokok permasalahan dan gambaran penelitian secara keseluruhan, adapun hal-hal yang dibahas adalah latar belakang, rumusan masalah, batasan masalah, maksud dan tujuan penelitian, metode penelitian, dan sistematika penulisan.

**BAB II      LANDASAN TEORI**

Bab ini berisi tentang konsep dasar serta teori-teori yang berkaitan dengan topik penelitian dari sumber pustaka dan refensi yang menjadi landasan dasar dalam perancangan, analisis kebutuhan sampai implementasi dan pengujian sistem.

**BAB III    ANALISIS DAN PERANCANGAN SISTEM**

Bab ini berisi tentang analisis kebutuhan dalam membangun aplikasi ini, analisis sistem yang sedang berjalan pada aplikasi sesuai dengan metode pembangunan perangkat lunak yang digunakan. Selain itu terdapat juga perancangan antar muka yang akan dibangun sesuai dengan analisis yang telah dibuat.

**BAB IV     IMPLEMENTASI DAN PERANCANGAN**

Bab ini berisi tahapan implementasi dan pengujian yang merupakan tahap mengimplementasikan dari hasil penelitian, analisis dan perancangan untuk mengimplementasikan dan menguji aplikasi.

**BAB V       PENUTUP**

Bab ini berisi tentang kesimpulan yang diambil dari hasil penulisan dan saran yang dapat menjadi masukan bagi peternak dan penelitian selanjutnya.

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Tinjauan Pustaka**

Mella Risna Effendi (2013), adalah mahasiswi jurusan Sistem Informasi, Universitas Komputer Indonesia dalam jurnalnya yang berjudul “Sistem Pakar Untuk Mendiagnosa Penyakit Ayam”. Dalam jurnalnya yang ditulis oleh Mella Risna Effendi tersebut membahas tentang sistem pakar untuk diagnosa penyakit ayam yang diterapkan di Koperasi Serba Usaha (KSU) Jati Mekar Desa Caracas Kecamatan Kalijati Kabupaten Subang. [2]

Aplikasi sistem pakar ini akan digunakan pada Koperasi Serba Usaha (KSU) yang memiliki jenis usaha peternakan ayam. Dalam jurnal tersebut berisi tentang perancangan sistem pakar dengan menggunakan metode *forward chaining* dengan metode penelusuran *Best First Search*, tetapi dalam aplikasi sistem pakar ini hanya bisa digunakan pada android. Hasil akhir dari jurnal tersebut berupa aplikasi sistem pakar untuk mendiagnosis penyakit ayam berbasis android dengan tidak adanya menu pakar untuk menambahkan dan mengedit data pakar.

Penelitian lain yang menggunakan perangkat desktop dengan visual basic 6.0 salah satunya adalah “Perancangan Sistem Pakar Diagnosis Penyakit Ayam dengan Menggunakan Metode Forward Chaining” yang dibuat oleh Taofhik Aris Yanto tahun 2013 di Universitas Widyatama. [3]

Aplikasi milik Taofhik mempunyai kesamaan yaitu mendeteksi penyakit pada ayam melalui gejala-gejala yang diketahui. Aplikasi yang dibuat oleh

Taofhik menggunakan metode *Forward Chaining* atau dengan kata lain menarik kesimpulan dari ide-ide awal. Basis pengetahuan dikumpulkan dari pakar dan dari buku. Aplikasi yang dibuat memiliki fitur untuk mencari penyakit pada ayam sesuai dengan gejala – gejala pada ayam tersebut. Untuk mengetahui penyakit tersebut, aplikasi akan menanyakan beberapa pertanyaan sehingga dapat menyimpulkan penyakit apa yang terdapat pada ayam, atau mungkin aplikasi tidak mengetahui penyakit apa yang terdapat pada ayam tersebut. aplikasi akan menampilkan cara pengobatan dan cara pencegahan untuk penyakit tersebut. [3]

Salah satu kelemahan dari aplikasi yang telah dibuat oleh Taofhik adalah tidak dapatnya menambah pengetahuan baik gejala maupun penyakit. Aplikasi yang dibuat terbatas pada beberapa penyakit sehingga jika ada penyakit baru atau gejala baru maka aplikasi tersebut sudah tidak sesuai sehingga perlu adanya pembaharuan aplikasi untuk menambah gejala atau penyakit baru dan metode pemilihan yang bersifat pertanyaan akan memakan waktu lebih jika gejalanya sangat banyak.

Penelitian lain yang dibuat oleh Siti dan Rina tahun 2010 di Universitas Pakuan Bogor adalah “Sistem Pakar: Diagnosis Penyakit Unggas dengan Metode Certainty Factor”. Aplikasi yang dibuat oleh Siti dan Rina ini juga memiliki kesamaan yaitu mendiagnosa penyakit melalui gejala . Aplikasi ini menerapkan metode *forward chaining* untuk pelacakan datanya, tetapi untuk menentukan penyakitnya menggunakan metode *certainty factors*. Setiap gejala yang dipilih oleh pengguna akan dihitung berdasarkan rumus dan bobot yang telah ditentukan. Nilai yang tertinggi merupakan jenis penyakit yang paling mungkin terjadi.

Dalam penelitian aplikasi berbasis *website* ini, tidak disebutkan apakah bisa menambahkan gejala atau penyakit baru atau tidak ada fasilitas untuk pakar menambah, mengubah, dan menghapus gejala, penyakit dan pengetahuan dalam sistem itu sendiri. Pada Tabel 2.1 akan dijelaskan perbandingan jurnal sistem pakar yang membahas diagnosa ayam. [4]

**Tabel 2.1 Perbandingan Jurnal Terkait**

No	Judul	Peneliti, Tahun	Hasil	Pembeda
1.	Sistem Pakar Untuk Mendiagnosa Penyakit Ayam Menggunakan Pemrograman Android Studio	Mella Risna Effendi (2013)	Sistem pakar untuk mendiagnosa penyakit ayam berbasis android yang bisa digunakan di koperasi serba usaha.	Memberikan informasi tentang penyakit ayam menggunakan metode <i>forward chaining</i> dan menggunakan pemrograman Android studio
2.	Perancangan Sistem Pakar Diagnosis Penyakit Ayam dengan Menggunakan Metode Forward Chaining	Taofhik Aris Yanto (2013)	Sistem pakar untuk mendiagnosa penyakit ayam dengan pertanyaan setiap diagnosa	Memberikan hasil informasi dan diagnosis penyakit ayam menggunakan metode <i>forward chaining</i> dan berbasis <i>visual basic 6.0</i>

No	Judul	Peneliti, Tahun	Hasil	Pembeda
3.	Sistem Pakar: Diagnosis Penyakit Unggas dengan Metode Certainty Factor	Siti Rohajawati, Rina Supriyati (2010)	Sistem Pakar identifikasi penyakit ayam petelur berbasis <i>website</i> .	Memberikan hasil infomasi penyakit ayam dengan <i>forward chaining</i> untuk pelacakan datanya dan dengan pembanding di setiap hasil diagnosa, menggunakan metode <i>certainty factor</i> .

## 2.2 Teori Dasar Ayam

Ayam adalah salah satu jenis burung atau unggas. Salah satu fungsi yang membedakannya dari kebanyakan burung lainnya adalah ia memiliki sisir/jengger dan dua gelambir/pial. Jengger tambahan merah di atas kepalanya, dan dua gelambir pelengkap di bawah dagu yang merupakan karakteristik seksual sekunder dan lebih menonjol dalam laki-laki. Ayam yaitu jenis burung yang termasuk *phylum chordata*, subphilum dari vertebrata kelas aves (Burung), sub kelas neormithes, super ordernya *carinatae*, dan genus atau berspesies *Gallus Domesticus* adalah suatu unggas yang berasal dari daerah Indian, Dalam bahasa latin, *gallus* berarti sisir, dan ayam lokal adalah *Gallus Domesticus*. Burung-burung hutan merah, atau nenek moyang ayam, adalah *Gallus Bankiva*. Ada

delapan jenis jengger yaitu: *single, rose, pea, cushion, buttercup, strawberry, V-shaped*, dan *silkie*. [5]

Ayam memiliki dua kaki dan sayap. Ayam domestik pada dasarnya kehilangan kemampuan untuk terbang. Pada ayam pedaging atau peternakan dengan berat yang digunakan untuk produksi daging, maka ayam tidak bisa melakukan lebih dari mengepakkan sayap mereka dan melompat lebih tinggi sedikit atau bergerak lebih cepat. Ayam yang bertubuh ringan bisa terbang dengan jarak pendek, dan beberapa dapat terbang di atas pagar yang relatif tinggi. Kaki dan betisnya memiliki sisik. Tingkat respirasi lebih tinggi di ayam dari hewan yang lebih besar. Secara umum, dapat dikatakan bahwa semakin kecil jenis burung maka semakin cepat bernafas. Sebagai contoh, hummingbird bernafas lebih per menit daripada ayam. Ayam jantan bernafas sekitar 18 sampai 21 kali satu menit dan sekitar 31-37 kali ketika mereka tidak di bawah tekanan.

Detak jantung ayam agak cepat, menjadi sekitar 286 kali per menit pada jantan dan 312 pada betina dalam keadaan istirahat. Tingkat pencernaan agak cepat dalam ayam. Ini bervariasi dari 2 1/2 untuk 25 jam untuk bagian makanan tergantung pada apakah pada saluran pencernaannya sedang penuh, sebagian penuh atau kosong ketika pakan tertelan. Suhu ayam sekitar 107 derajat ke 107.5 derajat fahrenheit. Ayam sejatinya menetas, tidak dilahirkan dalam arti sebenarnya. Ayam ditutupi dengan bulu tetapi memiliki beberapa rambut vestigial yang tersebar tubuh. Konsumen rata-rata tidak melihat rambut ini, karena mereka biasanya hangus di pabrik pengolahan. Ayam memiliki paruh dan tidak memiliki gigi. Penguncian apapun terjadi di ampela. Banyak produsen unggas tidak

menyediakan grit/tambahan ramsum untuk ayam mereka, dan kebanyakan hanya di biarkan makan sendiri mencari-cari di tanah. Ayam memiliki tulang *pneumatik*, yang membuat tubuhnya lebih ringan untuk terbang, dan jika itu tidak kehilangan kemampuan untuk melakukannya. [5]

Ada 13 kantung udara dalam tubuh ayam, sekaligus untuk membuat tubuhnya lebih ringan, dan sebagai bagian yang berfungsi pada sistem pernafasan. Ayam memiliki rentang hidup yang relatif pendek, beberapa hidup 10 sampai 15 tahun. Dalam produksi telur komersial, hanya sekitar 18 bulan sebelum ayam diganti dengan yang lebih muda. Dibutuhkan enam bulan untuk ayam betina berumur seksual dan mulai bertelur, kemudian mereka bertelur sampai 12-14 bulan. Setelah itu nilai jual mereka akan menurun dengan cepat, sehingga kebanyakan mereka yang dibuang ketika mereka berumur sekitar 18 bulan. [5]

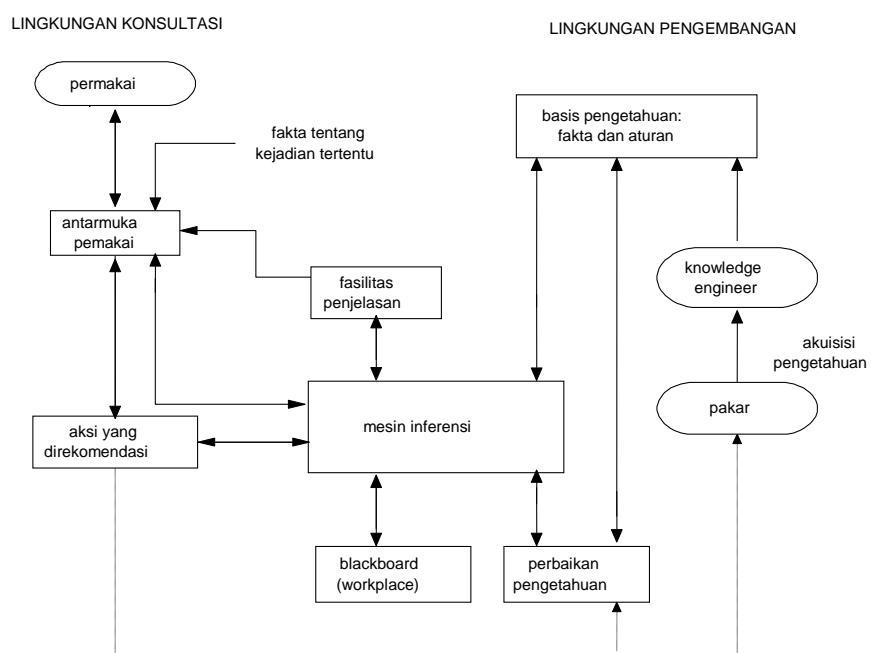
## 2.3 Sistem Pakar

### 2.3.1 Pengertian Sistem Pakar

Salah satu bagian dari Artificial Intelligence (AI) yang akhir-akhir ini mengalami perkembangan pesat yaitu Sistem Pakar (expert system). Pengertian dari Sistem pakar adalah sistem berbasis komputer yang menggunakan pengetahuan, fakta dan teknik penalaran dalam memecahkan masalah yang biasanya hanya dapat di selesaikan oleh seorang pakar dalam bidang tertentu. Sistem pakar yang baik dirancang agar dapat menyelesaikan suatu permasalahan tertentu dengan meniru kerja dari para ahli. Dengan adanya sistem pakar ini, orang awam pun dapat menyelesaikan masalah yang cukup rumit yang sebenarnya hanya dapat diselesaikan dengan bantuan para ahli. Seorang pakar adalah orang

yang mempunyai keahlian dalam bidang tertentu, yaitu pakar yang mempunyai *knowledge* atau kemampuan khusus yang orang lain tidak mengetahui atau mampu dalam bidang yang dimilikinya. [6]

Pada dasarnya sistem pakar diterapkan untuk mendukung aktivitas pemecahan masalah. Beberapa aktivitas pemecah yang diamaksud antara lain: pembuatan keputusan (*decicion making*), oemaduan pengetahuan (*knowledge fusing*), pembuatan desain (*designing*), perencanaan (*planning*), prakiraan (*forecasting*), pengaturan (*regulating*), pengendalian (*controlling*), diagnosis (*diagnosing*), perumusan (*prescribing*), penjelasan (*explaining*), pemberian nasihat (*advising*) dan pelatihan (*tutoring*). Selain itu sistem pakar juga dapat berfungsi sebagai asisten yang pandai dari seorang pakar. Struktur sistem pakar dapat dilihat pada Gambar 2.1 [7]



**Gambar 2.1 Struktur Sistem Pakar (Turban,1995)**

Sistem pakar disusun oleh dua bagian utama yaitu lingkungan konsultasi dan lingkungan pengembangan. Lingkungan konsultasi digunakan oleh pengguna yang bukan pakar untuk guna memperoleh pengetahuan pakar sedangkan lingkungan pengembangan digunakan untuk memasukkan pengetahuan pakar ke dalam lingkungan sistem pakar.

### **2.3.2 Pemakaian Sistem Pakar**

Sistem pakar dapat digunakan oleh: [8]

1. Orang awam yang bukan pakar untuk meningkatkan kemampuan mereka dalam memecahkan masalah.
2. Pakar sebagai asisten yang berpengetahuan.
3. Memperbanyak atau menyebarluaskan sumber pengetahuan yang semakin langka.

Sistem pakar merupakan program yang dapat mengantikan keberadaan seorang pakar. Alasan mendasar mengapa *Expert System* dikembangkan untuk mengantikan seorang pakar: [8]

1. Dapat menyediakan kepakaran setiap waktu dan diberbagai lokasi.
2. Secara otomatis mengerjakan tugas-tugas rutin yang membutuhkan seorang pakar.
3. Seorang pakar akan pensiun atau pergi.
4. Menghadirkan atau menggunakan jasa seorang pakar memerlukan biaya mahal.
5. Kepakaran dibutuhkan juga pada lingkungan yang tidak bersahabat (*hostile environment*)

### 2.3.3 Ciri-ciri Sistem Pakar

Ciri-ciri dari sistem pakar adalah sebagai berikut: [9]

1. Terbatas pada bidang yang spesifik.
2. Dapat memberikan penalaran untuk data-data yang tidak lengkap atau tidak pasti.
3. Dapat mengemukakan rangkaian alasan yang diberikan dengan cara yang dipahami.
4. Berdasarkan pada *rule* atau kaidah tertentu.
5. Dirancang untuk dapat dikembangkan secara bertahap.
6. Outputnya bersifat nasihat atau anjuran.
7. Output tergantung dari dialog dengan *user*.
8. *Knowledge base* dan *inference engine* terpisah.

### 2.3.4 Keuntungan Pemakaian Sistem Pakar

Banyak keuntungan yang diberikan oleh sistem pakar, diantaranya: [9]

1. Membuat seorang yang awam dapat bekerja seperti layaknya seorang pakar.
2. Dapat bekerja dengan informasi yang tidak lengkap atau tidak pasti.
3. Meningkatkan *output* dan produktifitas. *Expert System* dapat bekerja lebih cepat dari manusia. Keuntungan ini berarti mengurangi jumlah pekerja yang dibutuhkan, dan akhirnya akan mereduksi biaya.
4. Meningkatkan kualitas.
5. *Expert System* menyediakan nasihat yang konsisten dan dapat mengurangi tingkat kesalahan.

6. Membuat peralatan yang kompleks lebih mudah dioperasikan karena ES dapat melatih pekerja yang tidak berpengalaman.
7. Handal (*reability*).
8. ES tidak dapat lelah atau bosan. Juga konsisten dalam memberi jawaban dan atau selalu memberikan perhatian penuh.
9. Memiliki kemampuan untuk memecahkan masalah yang kompleks.
10. Memungkinkan pemindahan pengetahuan ke lokasi yang jauh serta memperluas jangkauan seorang pakar, dapat diperoleh dan dipakai dimana saja. Merupakan arsip yang terpercaya dari sebuah keahlian sehingga user seolah-olah berkonsultasi langsung dengan sang pakar meskipun mungkin sang pakar sudah pensiun.

### **2.3.5 Arsitektur Sistem Pakar**

Menurut Martin dan Oxman (1988), sistem pakar memiliki beberapa komponen utama, yaitu antarmuka pengguna (*user interface*), basis data sistem pakar (*expert system database*), fasilitas akuisisi pengetahuan (*knowledge acquisition facility*), dan mekanisme inferensi (*inference mechanism*). Selain itu ada satu komponen yang hanya ada pada beberapa sistem pakar, yaitu fasilitas penjelasan (*explanation facility*).[7]

Antarmuka pengguna adalah perangkat lunak yang menyediakan media komunikasi antara pengguna dengan sistem. Basis data sistem pakar berisi pengetahuan setingkat pakar pada subjek tertentu. Berisi pengetahuan yang dibutuhkan untuk memahami, merumuskan, dan menyelesaikan masalah. Basis data ini terdiri atas 2 elemen dasar: [7]

1. Fakta, sesuai masalah dalam teori terkait.
2. Heuristik khusus atau *rules*, yang langsung menggunakan pengetahuan untuk menyelesaikan masalah khusus.

Pengetahuan ini dapat bersalah dari pakar, jurnal, majalah, dan sumber pengetahuan lain. Fasilitas akuisisi pengetahuan merupakan perangkat lunak yang menyediakan fasilitas dialog antara pakar dengan sistem. Fasilitas akuisisi ini digunakan untuk memasukkan fakta-fakta dan kaidah-kaidah sesuai dengan perkembangan ilmu. Meliputi proses pengumpulan, pemindahan, dan perubahan dari kemampuan pemecahan masalah seorang pakar atau sumber terdokumentasi ke program computer, yang bertujuan untuk memperbaiki dan atau mengembangkan basis pengetahuan (*knowledge base*).

Mekanisme inferensi merupakan perangkat lunak yang melakukan penalaran dengan menggunakan pengetahuan yang ada untuk menghasilkan suatu kesimpulan atau hasil akhir. Dalam komponen ini dilakukan permodelan proses berfikir manusia.

Fasilitas penjelasan berguna dalam memberikan penjelasan kepada pengguna mengapa computer meminta suatu informasi tertentu dari pengguna dan dasar apa yang digunakan komputer sehingga dapat menyimpulkan suatu kondisi.

Menurut Schnupp (1989) ada 4 tipe penjelasan yang digunakan dalam sistem pakar, yaitu: [8]

1. Penjelasan mengenai jejak aturan yang menunjukan status konsultasi.
2. Penjelasan mengenai bagaimana sebuah keputusan diperoleh.

3. Penjelasan mengapa sistem menanyakan suatu pertanyaan.
4. Penjelasan mengapa sistem tidak memberikan keputusan seperti yang dikehendaki pengguna.

Arsitektur dasar dari sistem pakar dapat dilihat pada Gambar 2.2 (Giarratano dan Riley, 1994)



**Gambar 2.2 Arsitektur Sistem Pakar**

Memori kerja dalam arsitektur sistem pakar merupakan bagian dari sistem pakar yang berisi fakta-fakta masalah yang ditemukan dalam suatu sesi, berisi fakta-fakta tentang suatu masalah yang ditemukan dalam proses konsultasi. [10]

### 2.3.6 Orang Yang Terlibat Dalam Sistem Pakar

Untuk memahami perancangan sistem pakar, perlu dipahami mengenai siapa saja yang berinteraksi dengan sistem. Mereka adalah: [8]

1. Pakar (*domain expert*): seorang ahli yang dapat menyelesaikan masalah yang sedang diusahakan untuk dipecahkan oleh sistem.

2. Pembangunan pengetahuan (*knowledge engineer*): seseorang yang menerjemahkan pengetahuan seorang pakar dalam bentuk deklaratif sehingga dapat digunakan oleh sistem pakar.
3. Pengguna (*user*): seseorang yang berkonsultasi dengan sistem untuk mendapatkan saran yang disediakan oleh pakar.
4. Pembangunan sistem (*system engineer*): seorang yang membuat antarmuka pengguna, merancang bentuk basis pengetahuan secara deklaratif dan mengimplementasikan mesin inferensi.

### 2.3.7 Kategori Masalah Sistem Pakar

Masalah-masalah yang dapat diselesaikan dengan sistem pakar, diantaranya: [8]

1. Interpretasi: membuat kesimpulan atau deskripsi dari sekumpulan data mentah.
2. Prediksi: memproyeksikan akibat-akibat yang dimungkinkan dari situasi-situasi tertentu.
3. Diagnosis: menentukan sebab malfungsi dalam situasi kompleks yang didasarkan pada gejala-gejala yang teramati.
4. Desain: menentukan konfigurasi komponen-komponen sistem yang
5. Pengendalian: mengatur tingkah laku suatu environment yang kompleks.
6. *Selection*: mengidentifikasi pilihan terbaik dari sekumpulan (list) kemungkinan.

7. *Simulation:* permodelan interaksi antara komponen-komponen sistem.
8. *Monitoring:* membandingkan hasil pengamatan dengan kondisi yang diharapkan.

## **2.4 Respresentasi Pengetahuan**

Representasi pengetahuan adalah suatu teknik untuk merepresentasikan basis pengetahuan yang diperoleh ke dalam suatu skema atau diagram tertentu sehingga dapat diketahui relasi atau keterhubungan antara suatu data dengan data yang lain. Teknik ini membantu knowledge engineer dalam memahami struktur pengetahuan yang akan dibuat sistem pakarnya. Beberapa model representasi pengetahuan adalah logika, jaringan semantik (Semantic Nets), Bingkai (frame), pohon, dan kaidah produksi (Production Rules).[11]

### 1. Logika

Logika adalah bentuk representasi pengetahuan yang paling tua. Pada dasarnya proses logika adalah proses membentuk kesimpulan atau menarik suatu inferensi berdasarkan fakta yang telah ada..

### 2. Pohon

Pohon merupakan struktur penggambaran pohon secara hierarkis. Struktur pohon terdiri dari node-node yang menunjukkan objek, dan arc busur yang menunjukkan hubungan antar objek

### 3. Kaidah Produksi (Production Rules)

Metode kaidah produksi biasanya dituliskan dalam bentuk jika-maka (if-then). Kaidah ini dapat dikatakan sebagai hubungan implikasi dua bagian

yaitu bagian premise (jika) dan bagian konklusi (maka). Apabila bagian premise dipenuhi maka bagian konklusi juga akan bernilai benar.

IF premis THEN konklusi

IF masukan THEN keluaran

IF gejala THEN diagnosa

Premis mengacu pada fakta yang harus benar sebelum konklusi tertentu dapat diperoleh. Anteseden mengacu situasi yang terjadi sebelum konsekuensi dapat diamati.

Sebelum sampai pada bentuk kaidah produksi terdapat langkah – langkah yang harus ditempuh dari pengetahuan yang didapatkan dalam domain tertentu. Langkah– langkah tersebut adalah menyajikan pengetahuan yang berhasil didapatkan dalam bentuk tabel keputusan (decision table) kemudian dari tabel keputusan dibuat pohon keputusan (decision tree).

## 2.5 Mesin Inferensi

Inferensi adalah proses menghasilkan kesimpulan berdasarkan fakta atau pengetahuan yang diketahui atau diasumsikan, mesin inferensi memulai pelacakannya dengan mencocokan kaidah-kaidah dalam basis pengetahuan dengan fakta-fakta yang ada dalam basis data, secara deduktif mesin inferensi memilih pengetahuan yang relevan dalam rangka mencapai kesimpulan. Dengan demikian, sistem ini dapat menjawab pertanyaan pemakai meskipun jawaban tersebut tidak tersimpan secara eksplisit didalam basis pengetahuan dan fakta-fakta yang ada dalam basis data. Metode inferensi ada dua jenis yaitu runut maju (forward chaining) dan runut balik (backward chaining). [12]

### 2.5.1 Runut maju (*forward chaining*)

*Forward chaining* merupakan proses peruntutan yang dimulai dengan menampilkan kumpulan data atau fakta yang meyakinkan menuju konklusi akhir atau pelacakan yang dimotori data (data-driven). Dalam pendekatan ini pelacakan dimulai dari informasi masukan, dan selanjutnya mencoba menggambarkan kesimpulan. Pencocokan fakta atau pernyataan dimulai dari sebelah kiri (IF dulu). Pelacakan kedepan mencari fakta yang sesuai dengan bagian IF dari aturan IF-THEN.

Contoh :

IF Nafsu makan berkurang

AND Tampak lesu

AND Sayap menggantung

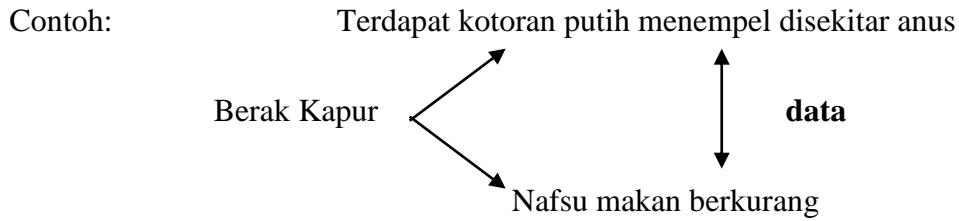
AND Mencret keputih-putihan

AND Terdapat kotoran putih menempel disekitar anus

THEN Berak Kapur.

### 2.5.2 Runut balik (*backward chaining*)

Backward chaining adalah pendekatan yang dimotori tujuan (goal-driven). Dalam pendekatan ini pelacakan dimulai dari tujuan, selanjutnya dicari aturan yang memiliki tujuan tersebut untuk kesimpulannya. Selanjutnya proses pelacakan menggunakan premis untuk aturan tersebut sebagai tujuan baru dan mencari aturan lain dengan tujuan baru sebagai kesimpulannya. Pencocokan fakta atau pernyataan di mulai dari bagian sebelah kanan (THEN dulu). Proses berlanjut sampai semua kemungkinan ditemukan.



## 2.6 Certainty Factor (Faktor Kepastian)

Faktor kepastian (*certainty factor*) diperkenalkan oleh Shorthliffe Buchanam dalam pembuatan MYCIN. *Certainty factor* merupakan nilai parameter klinis yang diberikan MYCIN untuk menunjukkan besarnya kepercayaan. *Certainty factor* didefinisikan sebagai berikut. [8]

$$CF(H,E) = MB(H,E) - MD(H,E)$$

Keterangan

$CF(H,E)$  : *Certainty factor* dari hipotesis H yang dipengaruhi oleh gejala (evidence) E. Besarnya CF berkisar antara -1 sampai dengan 1. Nilai -1 menunjukkan ketidakpercayaan mutlak sedangkan nilai 1 menunjukkan kepercayaan mutlak.

$MB(H,E)$  : Ukuran kenaikan kepercayaan (*measure of increased belief*) terhadap hipotesis H yang dipengaruhi oleh gejala E.

$MD(H,E)$  : Ukuran kenaikan ketidakpercayaan (*measure of increased disbelief*) terhadap hipotesis H yang dipengaruhi oleh gejala E.

### 2.6.1 Bentuk *Certainty Factor*

#### 1. CF Sequensial

Bentuk dasar rumus *certainty factor* sebuah aturan jika E maka H.

[12]

$$\mathbf{CF}(H, e) = \mathbf{CF}(E, e) * \mathbf{CF}(H, E)$$

Keterangan

$\mathbf{CF}(E, e)$  : *Certainty factor evidence* E yang dipengaruhi oleh *evidence* e

$\mathbf{CF}(H, E)$  : *Certainty factor hipotesis* dengan asumsi *evidence* diketahui dengan pasti, yaitu ketika  $\mathbf{CF}(E, e) = 1$

$\mathbf{CF}(H, e)$  : *Certainty factor hipotesis* yang dipengaruhi oleh *evidence* e

Jika semua *evidence* pada *antecedent* diketahui dengan pasti, maka rumusnya ditunjukkan dibawah ini. [13]

$$\mathbf{CF}(H, e) = \mathbf{CF}(H, E)$$

CF Sequensial diperoleh dari hasil perhitungan CF paralel dari semua premis dalam satu aturan dengan CF aturan yang diberikan oleh pakar. Adapun rumus untuk melakukan perhitungan untuk CF Sequensial. [12]

$$\mathbf{CF}(x, y) = \mathbf{CF}(x) * (\mathbf{CF}y)$$

Keterangan

$\mathbf{CF}(x, y)$  : CF Paralel     $\mathbf{CF}(x)$  : CF Sequensial dari semua premis

$\mathbf{CF}(y)$  : CF Pakar

## 2. CF Gabungan

CF Gabungan merupakan CF akhir dari sebuah calon konsklusi.

CF ini dipengaruhi oleh CF Paralel dari aturan yang menghasilkan konsklusi tersebut. CF gabungan diperlukan jika suatu konsklusi diperoleh dari beberapa aturan sekaligus. CF akhir dari satu aturan dengan aturan yang lain digabungkan untuk mendapatkan nilai CF akhir bagi calon konsklusi tersebut. [13]

$$\mathbf{CF}(x, y)$$

$$= \begin{cases} \mathbf{CF}(x) + \mathbf{CF}(y) * (1 - \mathbf{CF}(x)), \mathbf{CF}(x) > 0 \text{ dan } \mathbf{CF}(y) > 0 \\ \frac{\mathbf{CF}(x) + \mathbf{CF}(y)}{(1 - (\mathbf{Min}(|\mathbf{CF}(x)|, |\mathbf{CF}(y)|)))}, \text{ salah_satu}(\mathbf{CF}(x), \mathbf{CF}(y)) < 0 \\ \mathbf{CF}(x) + (\mathbf{CF}(y) * (1 + \mathbf{CF}(x))), \mathbf{CF}(x) < 0 \text{ dan } \mathbf{CF}(y) < 0 \end{cases}$$

### 2.6.2 Kombinasi Aturan

Untuk menggabungkan evidence anteseden yang terdapat dalam sebuah kaidah. Hal ini dapat dilihat pada Tabel 2.2 dibawah ini:

**Tabel 2.2 Kombinasi Evidence Anteseden**

EVIDENCE E	NILAI KETIDAKPASTIAN
E1 and E2	Min [ CF (H,E1), CF (H,E2) ]
E1 or E2	Max [ CF (H,E1), CF (H,E2) ]
Not E	-CF (H,E)

## 2.7 Konsep Basis Data

Menurut kusrini, basis data adalah kumpulan data yang saling berelasi. Data sendiri merupakan fakta mengenai objek, orang, dan lain-lain. Data dinyatakan dengan nilai (angka, deretan karakter, atau simbol). [14]

Menurut Fatahansyah, basis data dapat didefinisikan dalam sejumlah sudut pandang seperti: [15]

1. Himpunan kelompok data (arsip) yang saling berhubungan yang diorganisasikan sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah.
2. Kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa dan tanpa pengulangan (redundansi) yang tidak perlu, untuk memenuhi berbagai kebutuhan.
3. Kumpulan file/tabel/arsip yang saling berhubungan yang disimpan dalam media penyimpanan.

## 2.8 ERD (*Entity Relationship Diagram*)

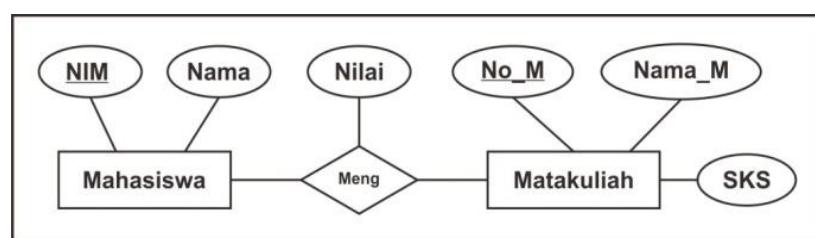
Model data diagram hubungan entitas (*Entity Relationship Diagram/ERD*) dibuat berdasarkan anggapan bahwa dunia nyata terdiri atas koleksi objek-objek dasar yang dinamakan entitas (*entity*) serta hubungan (*relationship*) antara entitas-entitas itu. Entitas adalah “sesuatu” atau “objek” pada dunia nyata yang dapat dibedakan satu dengan yang lainnya, yang bermanfaat bagi aplikasi yang sedang kita kembangkan.

Entitas dalam basis data dideskripsikan berdasarkan atributnya. Sebagai contoh, nomor rekening membedakan suatu rekening seseorang yang menyimpan

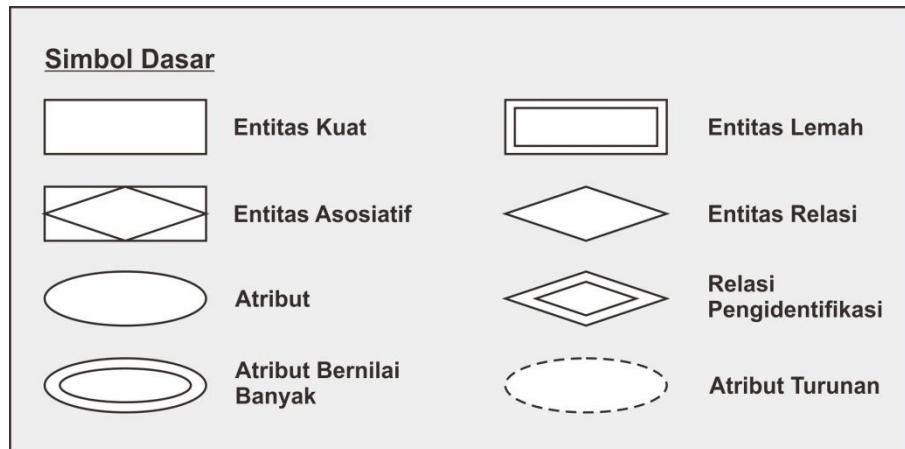
uangnya di suatu bank tertentu, dan nomor-nomor rekening tersebut merupakan atribut dari entitas rekening yang bersangkutan. Dalam hal ini, nomor rekening secara unik membedakan sebuah rekening dengan rekening yang lainnya.

Hubungan menjelaskan kaitan antara beberapa entitas. Sebagai contoh, mahasiswa memiliki orangtua, memiliki menjelaskan hubungan tertentu antara mahasiswa dengan orangtuanya. Dalam hal ini, himpunan semua entitas dengan tipe yang sama dan himpunan semua hubungan antar entitas dirujuk sebagai himpunan entitas (entity set), dan himpunan relasi (relationship set).

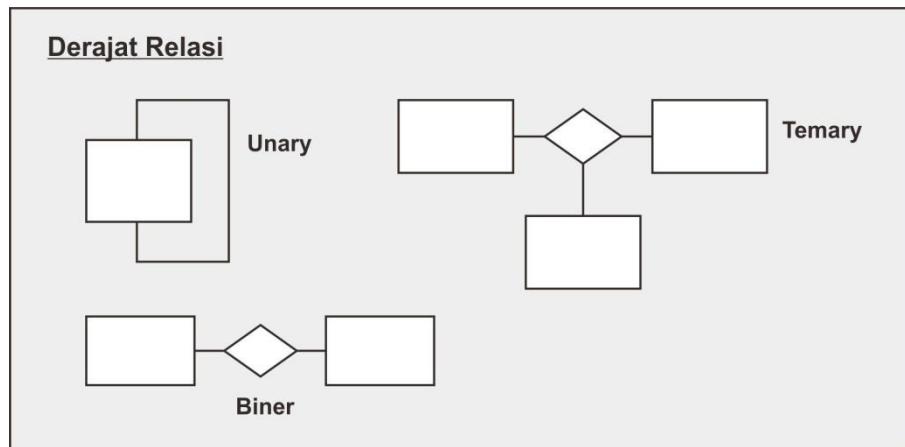
Secara skematis, basis data dapat dideskripsikan secara grafis dengan ERD yang memiliki komponen-komponen utama sebagai berikut, empat persegi panjang yang menggambarkan himpunan entitas, elips yang menggambarkan atribut, jajar genjang yang menggambarkan relasi/hubungan antar entitas, dan garis yang menyatukan atribut-atribut pada entitas tertentu serta menyatukan entitas-entitas dalam suatu relasi tertentu. Contoh penggambaran ERD dapat dilihat pada Gambar 2.3. Untuk simbol dasar, derajat dan kardinalitas relasi dapat dilihat di Gambar 2.4 sampai 2.6 [15]



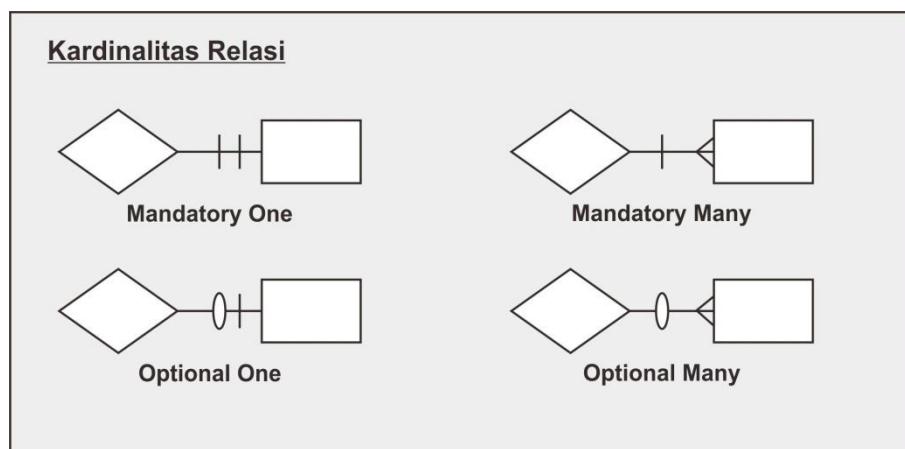
**Gambar 2.3 Contoh Penggambaran ERD**



Gambar 2.4 Simbol Dasar Notasi Diagram E-R Dasar



Gambar 2.5 Derajat Relasi Notasi Diagram E-R Dasar



Gambar 2.6 Kardinalitas Relasi Notasi Diagram E-R Dasar

## 2.9 Unified Modeling Language (UML)

UML (*Unified Modeling Language*) adalah salah standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan requirement, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek. [16]

### 2.9.1 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* merepresentasikan sebuah interaksi antara aktor dengan sistem. Penggunaan simbol pada *use case* dapat dilihat pada Tabel 2.3 [16]

**Tabel 2.3 Simbol Use Case Diagram**

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.
1		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .

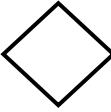
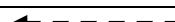
2		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri ( <i>independent</i> ) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri ( <i>independent</i> ).
NO	GAMBAR	NAMA	KETERANGAN
3		<i>Generalization</i>	Hubungan dimana objek anak ( <i>descendent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

10		Note	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi
----	---	------	---

### 2.9.2 Class Diagram

*Class diagram* adalah sebuah spesifikasi yang akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metode/fungsi). *Class Diagram* menggambarkan struktur dan deskripsi *Classs*, *package* dan objek beserta hubungan satu sama lain seperti pewarisan, asosiasi, dan lain-lain. Penggunaan simbol pada *class diagram* dapat dilihat pada Tabel 2.4 [17]

**Tabel 2.4 Simbol Class Diagram**

Gambar	Nama	Keterangan
	<i>Generalization</i>	Hubungan dimana objek anak ( <i>descendent</i> ) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk ( <i>ancestor</i> ).
	<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari dua objek.
	<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
	<i>Collaboration</i>	Deskripsi dari aturan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.
	<i>Realization</i>	Operasi yang benar-benar

		dilakukan oleh suatu objek.
- - - - - →	<i>Depedency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri ( <i>independent</i> ) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri.
—————	<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya

### 2.9.3 *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.

Penggunaan simbol pada *sequence diagram* dapat dilihat pada Tabel 2.5 [17]

**Tabel 2.5 Simbol *Sequence Diagram***

NO	GAMBAR	NAMA	KETERANGAN
1		<i>LifeLine</i>	Objek entity, antarmuka yang saling berinteraksi.
2		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi
3		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi

#### 2.9.4 *Activity Diagram*

Menggambarkan rangkaian aliran dari aktivitas, digunakan untuk mendeskripsikan aktivitas yang dibentuk dalam suatu operasi seingga dapat juga digunakan untuk aktifitas lainnya. Pembuatan *Activity Diagram* pada awal pemodelan proses dapat membantu memahami keseluruhan proses. *Activity Diagram* juga digunakan untuk menggambarkan interaksi antara beberapa *use case*. Penggunaan simbol pada *activity diagram* dapat dilihat pada Tabel 2.6 [17]

**Tabel 2.6 Simbol *Activity Diagram***

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

#### 2.10 Konsep OOP (*Object Oriented Programming*)

Pemrograman berorientasi objek berarti sebuah teknik pemrograman yang dalam proses pengembangannya menggunakan terminologi objek, dimana setiap objek memiliki atribut beserta dengan fungsi yang dapat saling berinteraksi

satu dengan yang lain seperti halnya objek. Objek sendiri adalah bentuk nyata dari sebuah class dan class merupakan kumpulan dari atribut dan *method/fungsi*.

Menurut Sakur (2010) Pemrograman Berorientasi Objek memiliki tiga pilar utama, yaitu: [18]

1. *Encapsulation* (Pengkapsulan)

*Encapsulation* atau pengkapsulan merupakan sebuah teknik yang digunakan untuk melakukan penyembunyian informasi yang tidak diperlukan oleh *user*. Data yang disembunyikan berupa atribut dan fungsi yang saling berhubungan membentuk serangkaian struktur data.

2. *Inheritance* (Turunan)

*Inheritance* atau turunan merupakan salah satu pilar dari OOP yang merepresentasikan permodelan turunan dari dunia nyata, dimana setiap tipe data baru dapat merupakan turunan dari tipe data induknya atau *superclass* sehingga kelas turunan akan mewarisi seluruh sifat-sifat induknya.

3. *Polymorphism*

Jika ditinjau dari arti kata *poly*, berarti banyak dan *morphism* merupakan bentuk atau model (*form*). Sehingga *polymorphism* diartikan sebagai model atau *form* yang dapat digunakan pada objek yang beragam. Objek-objek yang berbeda yang berasal dari induk yang sama dan dapat memiliki fungsi yang beragama.

## **2.11 Perangkat Lunak yang Digunakan**

Perangkat lunak adalah istilah khusus untuk data yang diformat, dan disimpan secara digital, termasuk program komputer, dokumentasinya, dan berbagai informasi yang bisa dibaca, dan ditulis oleh komputer. Dengan kata lain, bagian sistem komputer yang tidak berwujud.[19]

## **2.12 Bahasa Pemrograman**

Bahasa pemrograman, atau sering diistilahkan juga dengan bahasa komputer atau bahasa pemrograman komputer, adalah instruksi standar untuk memerintah komputer. Bahasa pemrograman ini merupakan suatu himpunan dari aturan sintaks dan semantik yang dipakai untuk mendefinisikan program komputer.

## **2.13 PHP (Hypertext Preprocessor)**

### **2.13.1 Pengertian PHP**

PHP adalah salah satu bahasa pemrograman yang berjalan pada sebuah web server dan berfungsi sebagai pengolah data pada sebuah server. Sintak PHP mirip dengan bahasa Perl dan C. PHP biasanya sering digunakan bersama web server Apache di beragam sistem operasi. PHP juga men-support ISAPI dan dapat digunakan bersama dengan Microsoft IIS di Windows (Sunyoto, 2007:119). Secara khusus PHP dirancang untuk web dinamis. Artinya PHP dapat membentuk suatu tampilan berdasarkan permintaan terkini. Misalnya dapat menampilkan isi database ke halaman web. Pada prinsipnya PHP memiliki fungsi yang sama dengan skrip-skrip seperti ASP (Active Server Page), Cold Fusion ataupun Perl. [20]

### 2.13.2 Skrip PHP

Skrip PHP berkedudukan sebagai tag dalam bahasa HTML (Hypertext Markup Language) adalah bahasa standar membuat halaman-halaman web (Kadir, 2001:2). Adapun kode berikut adalah contoh kode php yang berada di dalam kode HTML:

```
<html>
<head>
<title> menyapa dunia </title>
</head>
<body>
    hello duniaku.<br>
    <?php
        printf("tanggal, sekarang: %s ", date("d f Y"));
    ?>
</body>
</html>
```

Kode diatas disimpan dengan ekstensi .php.

```
<?php
    printf("tanggal, sekarang: %s ", Date("d F Y"));
?>
```

Kode inilah yang merupakan kode PHP. Kode PHP diawali dengan <?php dan diakhiri dengan ?>. pasangan kedua kode inilah yang berfungsi sebagai tag kode PHP. [20]

### 2.14 MySQL

MySQL adalah suatu database server merupakan open source SQL database (Sunyoto, 2007:145). My SQL merupakan database server dimana pemrosesan data terjadi di server dan client hanya mengirim data dan memindah data. Pengaksesan dapat dilakukan dimana saja dan oleh siapa saja dengan catatan komputer telah terhubung ke server. Lain halnya dengan database dekstop dimana segala pemrosesan data seperti penambahan data ataupun penghapusan data harus dilakukan pada komputer yang bersangkutan. MySQL termasuk database yang

terstruktur dalam pengolahan dan penampilan data. MySQL merupakan Relational Database Management System (RDBMS) yaitu hubungan antar tabel yang berisi data-data pada suatu database (Kadir, 2001:353). Tabel-tabel tersebut di-link oleh suatu relasi yang memungkinkan untuk mengkombinasikan data dari beberapa tabel ketika seorang user menginginkan menampilkan informasi dari suatu database. [21]

## **BAB III**

### **ANALISIS DAN PERANCANGAN SISTEM**

#### **3.1 Gambaran Umum Aplikasi**

Pada bagian ini akan membahas tentang analisis dan perancangan sistem pakar pendiagnosa penyakit ayam berbasis website. Agar menghasilkan sistem yang sesuai dengan kebutuhan maka harus dilakukan analisis dan perancangan yang baik dan benar. Untuk dapat melakukan diagnosa dengan menggunakan sistem ini, user mencentang atau memilih gejala yang dapat dilihat di lapangan. Jika user mencentang atau memilih suatu gejala, maka gejala yang akan dihitung oleh sistem hanya gejala yang dipilih oleh user. Dan jika user tidak memilih atau mencentang satu gejala pun, maka sistem akan menampilkan sebuah peringatan agar user memilih satu atau lebih gejala.

Setelah itu sistem akan melakukan perhitungan dan hasil diagnosa menampilkan kemungkinan virus atau penyakit yang menyerang, sesuai dengan gejala yang telah dipilih oleh user dengan nilai presentase terbesar kemungkinannya. Sistem pakar ini menggunakan metode certainty factor untuk memberikan tahapan proses, yang akan memberikan hasil kemungkinan virus atau penyakit yang menyerang.

Analisis dan perancangan sistem ini dilakukan dengan metode UML (*Unified Modelling Language*) sebagai alat bantu pemodelan sistem. Pada metode UML, *Use Case* digunakan untuk mendeskripsikan kebutuhan sistem dan bagaimana user mampu berinteraksi dengan sistem. Sedangkan *Activity diagram*

menggambarkan rangkaian aliran dari aktivitas *user* (pengguna) dan sistem. *Class diagram* digunakan untuk mendeskripsikan jenis-jenis objek dalam sistem dan berbagai macam hubungan statis yang terdapat diantara mereka. *Diagram sequence* digunakan untuk menggambarkan kelakuan dinamis sistem yang terjadi diantara objek atau entitas.

Aplikasi sistem pakar ini menggunakan metode *certainty factor*. Metode *certainty factor* digunakan untuk memberikan tahapan proses dengan memberikan alternatif gejala, maka akan diberikan hasil kemungkinan penyakit yang diderita.

### **3.2 Analisis Masalah**

#### **3.2.1 Identifikasi Masalah**

Budidaya ayam sudah sangat popular di masyarakat kita, baik ayam ras pedaging, petelur, maupun ayam buras. Namun jika di amati dengan seksama, kendala utama pengembangan usaha ternak ayam adalah adanya berbagai macam penyakit. Tidak jarang usaha peternakan hancur karena adanya serangan berbagai macam penyakit.

Penyakit ayam adalah kendala utama pada lingkungan tropis di Indonesia, para peternak ayam menghadapi beberapa kendala dalam beternak ayam. Kendalanya antara lain adalah banyaknya serangan dari berbagai penyakit, mulai dari penyakit yang ringan sampai yang mematikan bahkan penyakit yang dapat menular kepada manusia. Kerugian yang disebabkan oleh penyakit tersebut dapat berupa kematian atau penurunan produksi. Penanganan terhadap penyakit ayam terdiri dari upaya pencegahan dan pembasmian penyakit.

Penanganan terhadap penyakit ayam merupakan prioritas utama dan harus mendapatkan perhatian khusus. Tujuan pengendalian penyakit adalah mengurangi terjangkitnya suatu penyakit seminimal mungkin sehingga kerugian yang ditimbulkan dapat ditekan sekecil mungkin. Tujuan pembasmian penyakit adalah menghilangkan penyakit tertentu secara tuntas sehingga sumber penyakit tersebut dapat dimusnahkan. Penyakit yang menyerang ternak ayam dapat ditimbulkan oleh 2 penyebab, yaitu : [21]

1. Penyebab hidup (*Living Agent*)

Penyebab hidup dapat berupa bakteri, virus, kapang, riketsia, protozoa binatang bersel satu, cacing, insekta, dan lain-lain.

2. Penyebab tidak hidup (*Non-living Agent*)

Penyebab tidak hidup dapat berupa temperatur yang tinggi atau rendah, keracunan zat kimia atau nabati, defisiensi makanan.

Menurut pakar ayam Drh. Nur Rokhman, ayam seperti makhluk hidup lain yang dapat terserang penyakit mulai dari penyakit ringan sampai penyakit yang mematikan, bahkan menular kepada manusia. Terdapat 14 jenis penyakit utama yang sering menyerang pada ayam ternak. Beberapa contohnya adalah Avian influenza, Newcastle disease, dan Gumboro (data penyakit keseluruhan dapat dilihat pada BAB 3 yaitu Analisis Dan Perancangan Sistem). [22]

Dari 14 penyakit tersebut, gejala yang paling sering muncul adalah nafsu makan berkurang, produksi telur menurun, dan diare. Nafsu makan berkurang dapat dilihat dari pola makan ayam yang berubah. Biasanya ayam dewasa makan 125 gram per satu ekor ayam setiap harinya, jika terjadi penumpukan makanan

maka dapat dilihat bahwa nafsu makan ayam berkurang. Produksi telur menurun dapat dilihat dari menurunnya persentase telur ayam setiap harinya. Produksi telur ayam yang optimal biasanya sekitar 85% dari jumlah ayam dan dalam 1 tahun ayam dapat menghasilkan kurang lebih 330 butir telur. Diare dapat dilihat dari banyaknya kotoran ayam pada kandang serta biasanya disertai dengan mencret hijau atau mencret putih atau mencret darah.

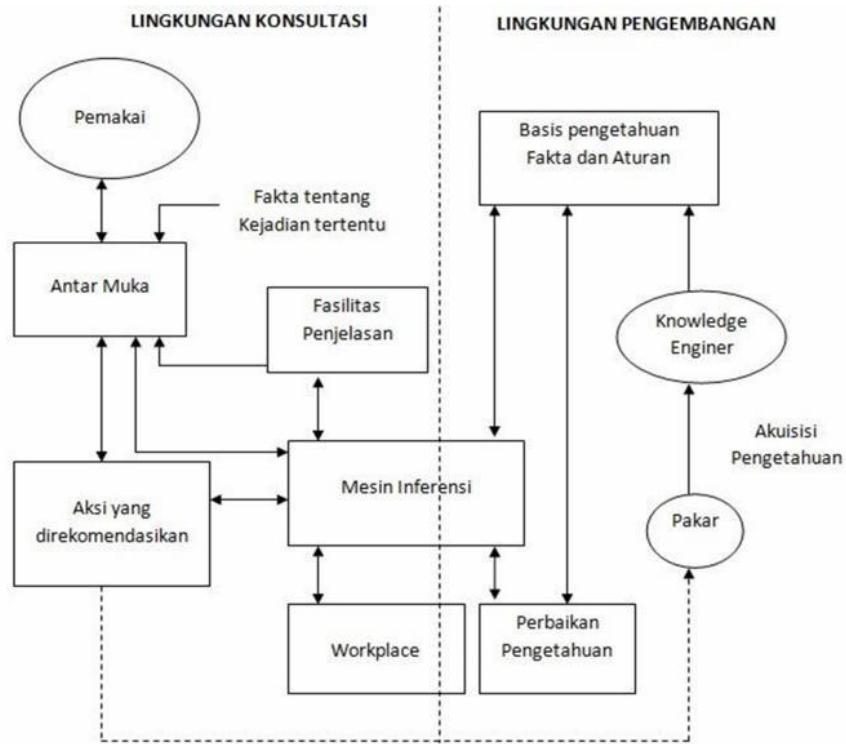
Untuk mengatasi permasalahan di atas penulis memutuskan untuk membuat sebuah sistem untuk memudahkan para peternak dalam mendiagnosa virus dan penyakit yang menyerang ayam secara dini berdasarkan gejala yang ada di lapangan. Dengan memanfaatkan teknologi *website* yang berbasis PHP disertai dengan kecerdasan buatan, maka peneliti memutuskan membuat sebuah sistem pakar berbasis *website responsive* ditujukan untuk membantu peternak dan penghobi ayam agar tahu gejala dan penyakit yang diderita, serta untuk pengetahuan tambahan untuk pencegahan lebih dini tanpa harus berkonsultasi langsung dengan pakar. Sistem ini juga membantu pakar, jika pakar sedang tidak ada di tempat. Sistem pakar ini akan memberikan hasil diagnosa penyakit beserta solusi yang diberikan.

### 3.3 Arsitektur Sistem

Arsitektur sistem diberikan dalam Gambar 3.1. Dalam arsitektur ini digambarkan bahwa sistem akan melayani dua orang pemakai (*user*) dan pakar dengan dua lingkungan yang berbeda untuk memanfaatkan fasilitas diagnosa dan penyembuhan virus dan penyakit yang menyerang ayam. Akuisisi pengetahuan

merupakan pengetahuan pakar yang dimasukkan ke dalam basis pengetahuan.

Terdapat juga perbaikan pengetahuan dimana pakar bisa mengedit nilai baru.



**Gambar 3.1 Arsitektur Sistem**

### 3.4 Representasi Pengetahuan

Dalam sistem pakar, pengetahuan harus direpresentasikan kedalam bentuk yang dapat diproses oleh komputer. Menurut Firebaugh (1989), terdapat empat teknik untuk representasi pengetahuan yaitu jaringan semantik, frame, script, dan aturan produksi. Di dalam sistem ini, menggunakan model representasi pengetahuan aturan produksi atau kaidah produksi. Kaidah produksi dituliskan dalam bentuk jika-maka (*if-then*). Penggunaan kaidah produksi di dalam sistem ini, dikarenakan salah satunya adalah kaidah mudah dimengerti dan mudah disampaikan. Guna mendukung penalaran dalam mendiagnosa dan menentukan saran yang tepat untuk mencegah dan mengendalikan virus dan penyakit ayam,

maka berikut dijelaskan metode yang digunakan untuk mengkodekan pengetahuan yang diperoleh dari pakar.

### **3.4.1    Analisis Pengetahuan**

Keberhasilan suatu sistem pakar terletak pada pengetahuan dan bagaimana mengolah pengetahuan tersebut agar dapat ditarik suatu kesimpulan. Pengetahuan yang diperoleh dari hasil wawancara dan analisa lewat buku dikonversi kedalam sebuah tabel penyakit dan gejala guna mempermudah proses pencarian solusi. Tabel jenis penyakit dan gejala ini digunakan sebagai pola pencocokan informasi yang dimasukan oleh pengguna (*user*) dan basis pengetahuan.

### **3.4.2    Daftar Penyakit**

Daftar jenis-jenis penyakit ayam dapat dilihat pada tabel 3.1.

**Tabel 3.1 Jenis-Jenis Penyakit**

<b>Kode</b>	<b>Nama Penyakit</b>	<b>Nama Latin</b>
<b>P01</b>	Berak Kapur	Pullorum Disease
<b>P02</b>	Kolera Ayam	Fowl Cholera
<b>P03</b>	Flu Burung	Avian Influenza
<b>P04</b>	Tetelo	Newcastle Disease
<b>P05</b>	Tipus Ayam	Fowl Typhoid
<b>P06</b>	Berak Darah	Coccidiosis
<b>P07</b>	Gumboro	Gumboro Disease
<b>P08</b>	Salesma Ayam	Infectious Coryza
<b>P09</b>	Batuk Ayam Menahun	Infectious Bronchitis
<b>P10</b>	Busung Ayam	Lymphoid Leukosis
<b>P11</b>	Batuk Darah	Infectious Laryngotrac

<b>P12</b>	Mareks	Mareks Disease
<b>P13</b>	Produksi Telur	Egg Drop Syndrome 76/EDS 76

### 3.4.3 Daftar Gejala

Daftar gejala pada penyakit ayam dapat dilihat pada tabel 3.2.

**Tabel 3.2 Gejala-Gejala Penyakit**

<b>Kode</b>	<b>Gejala</b>
<b>G01</b>	Nafsu makan berkurang
<b>G02</b>	Nafas sesak / megap-megap
<b>G03</b>	Nafas ngorok basah
<b>G04</b>	Bersin-bersin
<b>G05</b>	Batuk
<b>G06</b>	Bulu kusam dan berkerut
<b>G07</b>	Diare
<b>G08</b>	Produksi telur menurun
<b>G09</b>	Kedinginan
<b>G10</b>	Tampak lesu
<b>G11</b>	Mencret kehijau-hijauan
<b>G12</b>	Mencret keputih-putihan
<b>G13</b>	Muka pucat
<b>G14</b>	Nampak membiru
<b>G15</b>	Pembengkakan pial
<b>G16</b>	Jengger pucat
<b>G17</b>	Kaki dan sayap lumpuh
<b>G18</b>	Keluar cairan dari mata dan hidung
<b>G19</b>	Kepala bengkak
<b>G20</b>	Kepala terputar
<b>G21</b>	Pembengkakan dari sinus dan mata
<b>G22</b>	Perut membesar
<b>G23</b>	Sayap menggantung
<b>G24</b>	Terdapat kotoran putih menempel disekitar anus
<b>G25</b>	Mati secara mendadak
<b>G26</b>	Kerabang telur kasar
<b>G27</b>	Putih Telur Encer

<b>G28</b>	Kotoran kuning kehijauan
<b>G29</b>	Pembengkakan daerah fasial dan sekitar mata
<b>G30</b>	Kotoran atau feses berdarah
<b>G31</b>	Bergerombol di sudut kandang
<b>G32</b>	Mematuk daerah kloaka
<b>G33</b>	Kerabang telur pucat
<b>G34</b>	Telur lebih kecil
<b>G35</b>	Kelumpuhan pada tembolok
<b>G36</b>	Bernafas dengan mulut sambil menjulurkan leher
<b>G37</b>	Batuk berdarah
<b>G38</b>	Tidur paruhnya diletakkan dilantai
<b>G39</b>	Duduk dengan sikap membungkuk
<b>G40</b>	Kelihatan mengantuk dengan bulu berdiri
<b>G41</b>	Badan kurus
<b>G42</b>	Terdapat lendir bercampur darah pada rongga mulut
<b>G43</b>	Kaki pincang

Gejala merupakan basis pengetahuan untuk membuat suatu kesimpulan yang menjadi *goal*. Pada tabel 3.3 menjelaskan keterhubungan/relasi kode gejala dan kode penyakit.

**Tabel 3.3 Tabel Basis Pengetahuan Diagnosa Penyakit**

<b>Kode</b>	<b>Penyakit</b>
<b>G01</b>	P01, P02, P04, P05, P07, P09, P10, P12
<b>G02</b>	P04, P08, P10, P11
<b>G03</b>	P02
<b>G04</b>	P04
<b>G05</b>	P09
<b>G06</b>	P06
<b>G07</b>	P13
<b>G08</b>	P13
<b>G09</b>	P06, P09
<b>G10</b>	P04, P06
<b>G11</b>	P04
<b>G12</b>	P01
<b>G13</b>	P03

<b>G14</b>	P03
<b>G15</b>	P16
<b>G16</b>	P05
<b>G17</b>	P12
<b>G18</b>	P08
<b>G19</b>	P03
<b>G20</b>	P04
<b>G21</b>	P02
<b>G22</b>	P10
<b>G23</b>	P01, P05
<b>G24</b>	P01
<b>G25</b>	P03
<b>G26</b>	P13
<b>G27</b>	P13
<b>G28</b>	P05
<b>G29</b>	P08
<b>G30</b>	P06
<b>G31</b>	P06
<b>G32</b>	P07
<b>G33</b>	P13
<b>G34</b>	P13
<b>G35</b>	P12
<b>G36</b>	P11
<b>G37</b>	P11
<b>G38</b>	P07
<b>G39</b>	P07
<b>G40</b>	P09
<b>G41</b>	P10
<b>G42</b>	P11
<b>G43</b>	P12

Keterangan:

G01, G02, ...., G43 : Kode Gejala

P01, P02, ....., P13 : Kode Penyakit

Solusi atau saran cara pencegahan dan pengendalian penyakit yang menyerang ayam ini bersumber dari pakar dan beberapa studi literatur lainnya.

Solusi atau saran pencegahan dan tindakan dapat dilihat pada Tabel 3.4. Saat ayam terkena virus atau penyakit, hanya bisa dilakukan upaya pencegahan atau tindakan terhadap ayam yang terkena virus atau penyakit.

#### **3.4.4 Rekomendasi Pakar**

Rekomendasi merupakan solusi atau saran yang diberikan kepada sistem atas hasil yang telah dilakukan.

**Tabel 3.4 Rekomendasi Pakar**

<b>1. Berak Kapur (Pullorum Disease)</b>	
<p>Pullorum Disease disebut juga Bacillary White Diarrhea dan yang lebih popular disebut penyakit berak kapur atau berak putih.</p>	
Pencegahan	<ol style="list-style-type: none"> <li>1. Menjaga sanitasi mulai dari mesin penetasan hingga sanitasi kandang</li> <li>2. Melakukan desinfeksi kandang dengan formaldehyde sebanyak 40%.</li> <li>3. Ayam yang terkena penyakit sebaiknya dipisahkan dari kelompoknya, sedangkan ayam yang parah dimusnahkan.</li> </ol>
Tindakan	<ol style="list-style-type: none"> <li>1. Berikan Master Coliprim dosis: 1 gr/1 ltr air selama 3-4 hari (1/2 hari) berturut-turut. setelah itu berikan Master Vit-Stress selama 3-4 hari untuk membantu proses penyembuhan.</li> </ol>

	<p>2. Menyuntikkan antibiotik seperti furozolidon, coccilin, neo terramycin, tetra atau mycomas di dada ayam. pengobatan dapat dilakukan dengan menggunakan preparat sulfonamide.</p>
<b>2. Kolera Ayam (Fowl Cholera)</b>	<p>Merupakan penyakit ayam yang dapat menyerang secara pelan-pelan dan juga dapat menyerang secara mendadak.</p>
Pencegahan	<p>1. Menjaga agar litter tetap kering, mengurangi kepadatan kandang, menjaga kebersihan peralatan kandang dan memberikan vitamin dan pakan yang cukup agar stamina ayam tetap terjaga</p>
Tindakan	<p>1. Berikan Master Kolericid dosis: 1 gr/1 ltr air selama 3-4 hari berturut-turut. berikan Master Vit-Stress dosis: 1 gr/3 ltr air untuk membantu proses penyembuhan.  2. Dapat dilakukan dengan menggunakan preparat sulfat atau antibiotik seperti noxal, ampisol atau inequil.</p>
<b>3. Flu Burung (Avian Influenza)</b>	<p>Penyakit Avian Influenza, disebut juga penyakit Fowl Plaque. Pertama kali terjadi di Italia sekitar tahun 1800. Selanjutnya menyebar luas sampai tahun 1930, setelah itu menjadi sporadis dan terlokalisasi terutama di timur tengah.</p>
Pencegahan	<p>1. Melakukan desinfeksi kandang dengan formaldehyde sebanyak 40%.</p>

	<ol style="list-style-type: none"> <li>2. Menjaga sanitasi mulai dari mesin penetasan hingga sanitasi kandang.</li> <li>3. Ayam yang terkena penyakit sebaiknya dipisahkan dari kelompoknya, sedangkan ayam yang parah dimusnahkan.</li> </ol>
Tindakan	<ol style="list-style-type: none"> <li>1. Tidak ada obat. Dianjurkan untuk disingkirkan dan dimusnakan dengan cara dibakar dan bangkainya dikubur.</li> </ol>
<b>4. Tetelo (Newcastle Disease</b>	
	<p>Penyakit Newcastle Disease disebut juga Pseudovogel pest Rhaniket, Pheumoencephalitis, Tortor Furrens, dan di Indonesia popular dengan sebutan tetelo. Penyakit ini pertama kali ditemukan oleh Doyle pada tahun 1927, didaerah Newcastle on Tyne, Inggris</p>
Pencegahan	<ol style="list-style-type: none"> <li>1. memelihara kebersihan kandang dan sekitarnya. Kandang harus mendapat sinar matahari yang cukup dan ventilasi yang baik.</li> <li>2. Memisahkan ayam lain yang dicurigai dapat menularkan penyakit ini.</li> <li>3. Memberikan ransum jamu yang baik.</li> </ol>
Tindakan	<ol style="list-style-type: none"> <li>1. Ayam yang tertular harus dimusnahkan.</li> <li>2. Vaksinasi harus dilakukan untuk memperoleh kekebalan. Jenis vaksin yang</li> </ol>

	<p>kami gunakan adalah ND Lasota yang kami beli dari PT. SHS. Vaksinasi ND yang pertama, kami lakukan dengan cara pemberian melalui tetes mata pada hari ke 2. Untuk berikutnya pemberian vaksin kami lakukan dengan cara suntikan di intramuskuler otot dada.</p> <p>3. Untuk memudahkan untuk mengingat mengenai waktu pemberian vaksin, seorang pakar menyarankan agar memberikan vaksin ini dengan pola 444. maksudnya vaksin ND diberikan pada ayam yang berumur 4 hari, 4 minggu, 4 bulan dan seterusnya dilakukan 4 bulan sekali. Namun kami mempunyai sedikit perbedaan dengan jadwal pola 444.</p>
	<p><b>5. Tipus Ayam (Fowl Typhoid)</b></p> <p>Penyakit Fowl Typhoid dikenal sebagai penyakit tipus ayam, tergolong penyakit menular.</p>
Pencegahan	<ol style="list-style-type: none"> <li>1. Ayam yang masih kelihatan sehat segera dipindah ke kandang baru, bersih dan telah disemprot dengan cetamium.</li> <li>2. Diberi vaksinasi secara periodik.</li> <li>3. Sinar matahari diusahakan dapat masuk ke</li> </ol>

	<p>dalam kandang minimal pada pagi hari.</p> <ol style="list-style-type: none"> <li>4. Kandang lama bila mempergunakan lantai litter atau pasir diganti baru yang bersih, kering dan bebas debu dan kemudian disemprot dengan jodophor.</li> <li>5. Pada air minum diberi campuran coxalin.</li> <li>6. Ayam yang sudah terserang penyakit cepat-cepat disingkirkan ke kandang isolasi untuk mendapatkan perawatan intensif.</li> </ol>
Tindakan	<ol style="list-style-type: none"> <li>1. Berikan Neo Terramycin dosis: 2 sendok teh/3,8 ltr air selama 3-4 hari berturut-turut menurut aturan sesuai pada kemasannya.</li> </ol>
<b>6. Berak Darah (Coccidosis)</b>	<p>Coccidosis merupakan penyakit menular yang ganas, dikalangan para peternak ayam disebut juga penyakit berak darah. Penyakit ini ditemukan pada tahun 1674.</p>
Pencegahan	<ol style="list-style-type: none"> <li>1. Memberikan vaksinasi pada ayam pada usia 4 hari. Biasanya kami akan memberikan vaksinasi ini dengan melakukan penyemprotan pada pakan. Selain itu harus dilakukan sanitasi yang baik pada kandang.</li> <li>2. Pilihlah pakan yang sudah mengandung koksidiostat ( preparat pembunuh protozoa Eimeria)</li> </ol>

Tindakan	<p>1. Berikan Master Coliprim dosis: 1gr/1 ltr air selama 3-4 hari (1/2 hari) berturut-turut. setelah pengobatan berikan Vitamin Master Vit-Stress dosis: 1gr/3 ltr selama 3-4 hari berturut-turut.</p>
<b>7. Gumboro (Gumboro Disease)</b>	<p>Penyakit Gumboro, disebut juga Infectious Bursal Disease. Pertama kali ditemukan dan dilaporkan pada tahun 1975 oleh Dr. Csgrove di daerah Gumboro, Deleware, Amerika Serikat.</p>
Pencegahan	<p>1. Gumoro menyebar melalui kontak langsung, air minum, pakan, alat-alat yang sudah tercemar virus dan udara. Yang sangat menarik adalah gumoro tidak menular dengan perantaraan telur dan ayam sudah sembuh tidak menjadi “carrier”.</p> <p>2. Upaya penanggulangan gumoro ini dapat dilakukan dengan beberapa cara yaitu vaksinasi, menjaga kebersihan lingkungan kandang.</p>
Tindakan	<p>1. Tidak ada obat. Air gula 30-50 gr/ltr air dan ditambah Master Vit-Stress dosis: 1 gr/2 ltr air untuk meningkatkan kondisi tubuh.</p>
<b>8. Salesma Ayam (Infectious Coryza)</b>	<p>Penyakit Infectious Coryza disebut juga Infectious Cold, Snot,</p>

Rhinitis, Roup atau yang populer disebut salesma ayam.	
Pencegahan	<ol style="list-style-type: none"> <li>Menjaga kebersihan kandang dan lingkungan dengan baik. Kandang sebaiknya terkena sinar matahari langsung sehingga mengurangi kelembaban.</li> <li>Kandang yang lembab dan basah memudahkan timbulnya penyakit ini.</li> </ol>
Tindakan	<ol style="list-style-type: none"> <li>Diberikan adalah preparat sulfat seperti sulfadimethoxine atau sulfathiazole, menurut beberapa penulis penyakit ini dapat diobati dengan antibiotika seperti Ultramycin, imequil atau corivit. Kami menggunakan preparat enrofloksacyn atau lebih dikenal dengan Enflox produksi SHS dan saat ini kami sedang mencoba menggantinya dengan preparat amphycillin dan colistin atau lebih dikenal dengan Amphyvitacol produksi Vaksindo.</li> <li>Dapat juga dengan pengobatan tradisional dengan memberikan susu bubuk yang dicampur dengan air dan dibentuk sebesar kelereng sesuai dengan bukaan mulut ayam dan diberikan 3 kali sehari.</li> </ol>

	<p>3. Bisa juga dengan memberikan perasan tumbukan jahe, kunir, kencur dan lempuyang. Air perasan ini dicampurkan pada air minum. Sedangkan ampasnya kami campurkan pada sedikit pakan. Selain ramuan ini menghangatkan tubuh ayam, ramuan ini juga berkhasiat untuk menambah napsu makan ayam.</p> <p>4. Berikan obat Sulfamix dengan dosis 0.4 cc/kg BB ayam. Hal lain yang perlu dilakukan karena penyakit ini mempunyai penularan yang sangat cepat dan luas, ayam yang terkena Snot harus sesegera mungkin dipisahkan dari kelompoknya</p>
<b>9. Batuk Ayam Menahun (Infectious Bronchitis)</b>	<p>Penyakit Infectious Bronchitis pertama kali ditemukan pada tahun 1930 dan penyakit ini mulai menjadi wabah sejak tahun 1940. Pada tahun 1950 penyakit Infectious Bronchitis sudah dapat dikendalikan dengan efektif.</p>
Pencegahan	<p>1. Sanitasi merupakan factor pemutus rantai penularan penyakit karena virus tersebut sangat rentan terhadap desinfektan dan panas.</p> <p>2. Pencegahan lain yang sangat umum</p>

	dilakukan adalah dengan memberikan vaksinasi secara teratur.
Tindakan	1. Berikan vitamin Master Vit-Stress dosis: 1 gr/1 ltr air memperbaiki kondisi tubuh.
<b>10. Busung Ayam (Lymphoid Leukosis)</b>	
Penyakit Lymphoid Leukosis termasuk kelompok Leukosis Komplex Disease. Penyakit ini banyak menyerang ayam di Indonesia.	
Pencegahan	<ol style="list-style-type: none"> <li>1. Memelihara kebersihan kandang dan sekitarnya. Kandang harus mendapat sinar matahari yang cukup dan ventilasi yang baik.</li> <li>2. Memisahkan ayam lain yang dicurigai dapat menularkan penyakit ini.</li> </ol>
Tindakan	1. Tidak ada obat. Dianjurkan untuk disingkirkan dan dimusnakan, gejala umum ayam yang terkena virus ini adalah pialnya kering dan pada perutnya membesar keras.
<b>11. Batuk Darah (Infectious Laryngotracheitis)</b>	
Penyakit Infectious Laryngotracheitis disebut juga Infectious Tracheitis. Jenis penyakit ini ditemukan pada tahun 1925, dan secara resmi diakui oleh Committee on Poultry Disease of the American Veterinary Medical Association, pada tahun 1931.	
Pencegahan	1. Menjaga sanitasi mulai dari mesin penetasan hingga sanitasi kandang.

	<p>2. Memisahkan ayam lain yang dicurigai dapat menularkan penyakit ini.</p>
Tindakan	<p>1. Vaksinasi LT dengan cara meneteskan pada mata pada unggas seperti anak ayam sebelum menginjak usia 14 hari</p> <p>2. Berikan vitamin Master Vit-Stress dosis: 1 gr/1 ltr air untuk membantu memperbaiki kondisi tubuh.</p>
<b>12. Mareks (Mareks Disease)</b>	
<p>Penyakit Mareks Disease pada awalnya dimasukan dalam kelompok Leukosis Complex Disease. Namun setelah ditemukan penyebabnya dan penanggulangannya, penyakit ini dipisahkan dari kelompok Leukosis Complex Disease.</p>	
Pencegahan	<p>1. Menjaga sanitasi mulai dari mesin penetasan hingga sanitasi kandang.</p> <p>2. Ayam yang terkena penyakit sebaiknya dipisahkan dari kelompoknya, sedangkan ayam yang parah dimusnahkan.</p>
Tindakan	<p>1. Memberikan vaksinasi pada DOC berumur 1 hari dengan vaksin Cryomarex HVT atau Cryomarex Rispens.</p>
<b>13. Produksi Telur (Egg Drop Syndrome 76/EDS 76)</b>	
<p>Merupakan penyakit ayam yang pada tahun 1976, dilaporkan Van Eck di Nederland. Dikalangan pakar kesehatan ternak, penyakit itu</p>	

disebut Egg Drop Syndrome '76.	
Pencegahan	<ol style="list-style-type: none"> <li>1. Memberikan untuk menurunkan shedding virus tetapi tidak mencegah infeksi. Vaksinasi EDS'76 pada umur 16-18 minggu.</li> <li>2. Menjaga sanitasi mulai dari mesin penetasan hingga sanitasi kandang.</li> </ol>
Tindakan	<ol style="list-style-type: none"> <li>1. Tidak ada obat yang dapat menyembuhkan penyakit ini, usaha yang dapat dilakukan adalah menjaga kondisi badan tetap baik dan meningkatkan nafsu makan dengan vitamin. Infeksi sekunder dicegah dengan memberikan antibiotik.</li> </ol>

Pengetahuan disimpan dalam basis data relasional yang terdiri dari 3 tabel.

Adapun tabel yang dimaksud adalah:

1. Tabel Gejala (kode\_gejala, nama\_gejala)
2. Tabel Penyakit (kode\_penyakit, nama\_penyakit, det\_penyakit, srn\_penyakit)
3. Tabel Pengetahuan (kode\_pengetahuan, kode\_penyakit, kode\_gejala, mb, md)

### **3.4.5 Aturan Dan Aturan Produksi (*Production Rules*)**

Aturan-aturan yang digunakan untuk melakukan penalaran atau penelusuran basis pengetahuan awal sehingga menghasilkan *knowladge* baru untuk mencapai tujuan disebut dengan *Production Rules*. Secara default sistem ini memiliki 13 aturan diagnosa. Daftar aturan diagnosa dalam sistem ini ditunjukkan pada tabel 3.5.

**Tabel 3.5 Daftar Aturan Diagnosa**

No	Aturan
1	JIKA Nafsu makan berkurang DAN Mencret keputih-putihan DAN Sayap menggantung DAN Terdapat kotoran putih menempel disekitar anus MAKA Berak Kapur, CF = 0.52
2	JIKA Nafsu makan berkurang DAN Nafas ngorok basah DAN Pembengkakan pial DAN Pembengkakan dari sinus dan mata MAKA Kolera Ayam, CF = 0.64
3	JIKA Muka Pucat 13 DAN Nampak membiru 14 DAN Kepala bengkak 19 DAN Mati secara mendadak 25 MAKA Flu Burung, CF = 0.64
4	JIKA Nafsu makan berkurang DAN Bersin-Bersin DAN Mencret kehijau-hijauan DAN Kepala terputar DAN Nafas sesak / megap-megap

	DAN Tampak lesu MAKA Tetelo, CF = 0.40
5	JIKA Nafsu makan berkurang DAN Jengger pucat DAN Sayap menggantung DAN Kotoran kuning kehijauan MAKA Tipus Ayam, CF = 0.51
6	JIKA Bulu kusam dan berkerut DAN Kedinginan DAN Tampak lesu DAN Kotoran atau feses berdarah DAN Bergerombol di sudut kandang MAKA Berak Darah, CF = 0.40
No	<b>Aturan</b>
7	JIKA Duduk dengan sikap membungkuk DAN Mematuk daerah kloaka DAN Nafsu makan berkurang DAN Tidur paruhnya diletakkan dilantai MAKA Gumboro, CF = 0.64
8	JIKA Nafas sesak / megap-megap DAN Keluar cairan dari mata dan hidung DAN Pembengkakan daerah fasial dan sekitar mata MAKA Salesma Ayam, CF = 0.56
9	JIKA Nafsu makan berkurang DAN Batuk DAN Kedinginan DAN Kelihatan mengantuk dengan bulu berdiri MAKA Batuk Ayam Menahun, CF = 0.64
10	JIKA Nafas sesak / megap-megap

	DAN Nafsu makan berkurang DAN Perut membesar DAN Badan kurus MAKA Busung Ayam, CF = 0.65
11	JIKA Nafas sesak / megap-megap DAN Bernafas dengan mulut sambil menjulurkan leher DAN Batuk berdarah DAN Terdapat lendir bercampur darah pada rongga mulut MAKA Batuk Darah, CF = 0.64
12	JIKA Nafsu makan berkurang DAN Kaki dan sayap lumpuh DAN Kaki pincang DAN Kelumpuhan pada tembolok MAKA Mareks, CF = 0.64
13	JIKA Diare DAN Produksi telur menurun DAN Kerabang telur pucat DAN Kerabang telur kasar DAN Putih Telur Encer DAN Telur lebih kecil MAKA Produksi Telur (Egg Drop Syndrome 76) , CF = 0.32

### 3.4.6 Inferensi

Pada sistem ini, untuk melakukan diagnosa menggunakan metode inferensi, yaitu runut maju. Data yang digunakan dalam inferensi diperoleh dari jawaban yang diberikan oleh pengguna. Sistem menampilkan seluruh data gejala, sehingga nantinya pengguna akan mencentang gejala sesuai dengan gejala yang

dilihat di lapangan. Tabel yang digunakan dalam proses inferensi tersebut adalah:

Tabel Pengetahuan.

### 3.4.7 Teknik Inferensi

Teknik penalaran (*inference*) pada sistem ini menggunakan teknik *forward chaining* (pelacakan ke depan). Dalam sistem ini, data yang didapat berasal dari penggalian hasil wawancara dengan pakar yang bersangkutan, maupun literatur. Nilai CF (*Rule*) didapat dari interpretasi “term” dari pakar menjadi nilai sebuah MD/MB tertentu seperti terlihat pada Tabel 3.6.

**Tabel 3.6 Tabel Interpretasi Certainty Factor**

No	Certainty Term	CF <sub>akhir</sub>
1	Pasti Tidak	-1,0
2	Hampir Pasti Tidak	-0,8
3	Kemungkinan Besar Tidak	-0,6
4	Mungkin Tidak	-0,4
5	Tidak Tahu atau Tidak Yakin	-0,2 – 0,2
6	Mungkin	0,4
7	Kemungkinan Besar	0,6
8	Hampir Pasti	0,8
9	Pasti	1,0

Pada tabel 3.7 merupakan tabel representasi pengetahuan, dimana nilai CF Rule untuk gejala penyakit dicantumkan. Nilai CF Rule untuk gejala penyakit adalah nilai hipotesis dengan asumsi *evidence* diketahui. Adapun nilai CF Rule tersebut di dapat dari pakar yang terkait dengan sistem pakar yang dibuat.

**Tabel 3.7 Representasi Gejala Penyakit Ayam**

<b>Rule</b>	<b>Kaidah Produksi</b>	<b>MB</b>	<b>MD</b>	<b>CF Rule [MB - MD]</b>
R1	IF G01 THEN P01	0.4	0.2	0.2
R2	IF G12 THEN P01	0.8	0.2	0.6
R3	IF G23 THEN P01	0.6	0.2	0.4
R4	IF G24 THEN P01	1.0	0.0	1.0
R5	IF G01 THEN P02	0.4	0.2	0.2
R6	IF G03 THEN P02	1.0	0.2	0.8
R7	IF G15 THEN P02	1.0	0.0	1.0
R8	IF G21 THEN P02	1.0	0.0	1.0
R9	IF G13 THEN P03	0.6	0.2	0.4
R10	IF G14 THEN P03	1.0	0.0	1.0
R11	IF G19 THEN P03	0.6	0.2	0.4
R12	IF G25 THEN P03	1.0	0.0	1.0
R13	IF G01 THEN P04	0.4	0.2	0.2
R14	IF G02 THEN P04	0.4	0.2	0.2
R15	IF G04 THEN P04	0.8	0.2	0.6
R16	IF G10 THEN P04	0.4	0.2	0.2
R17	IF G11 THEN P04	0.8	0.0	0.8
R18	IF G20 THEN P04	0.8	0.2	0.6
R19	IF G01 THEN P05	0.6	0.2	0.2
R20	IF G16 THEN P05	1.0	0.2	0.8
R21	IF G23 THEN P05	0.8	0.2	0.6
R22	IF G28 THEN P05	1.0	0.0	1.0
R23	IF G06 THEN P06	1.0	0.2	0.8
R24	IF G09 THEN P06	0.8	0.2	0.6
R25	IF G10 THEN P06	0.8	0.2	0.6
R26	IF G30 THEN P06	1.0	0.0	1.0

R27	IF G31 THEN P06	0.8	0.2	0.6
R28	IF G01 THEN P07	0.4	0.2	0.2
R29	IF G32 THEN P07	1.0	0.0	1.0
R30	IF G38 THEN P07	1.0	0.0	1.0
R31	IF G39 THEN P07	1.0	0.2	0.8
R32	IF G02 THEN P08	0.8	0.2	0.6
R33	IF G18 THEN P08	0.8	0.2	0.6
R34	IF G29 THEN P08	1.0	0.0	1.0
R35	IF G01 THEN P09	0.4	0.2	0.2
R36	IF G05 THEN P09	1.0	0.0	1.0
R37	IF G09 THEN P09	0.8	0.2	0.6
R38	IF G40 THEN P09	1.0	0.0	1.0
R39	IF G01 THEN P10	0.4	0.2	0.2
R40	IF G02 THEN P10	0.8	0.2	0.6
R41	IF G22 THEN P10	1.0	0.0	1.0
R42	IF G41 THEN P10	1.0	0.0	1.0
R43	IF G02 THEN P11	0.8	0.2	0.6
R44	IF G36 THEN P11	1.0	0.2	0.8
R45	IF G37 THEN P11	1.0	0.0	1.0
R46	IF G42 THEN P11	1.0	0.0	1.0
R47	IF G01 THEN P12	0.4	0.2	0.2
R48	IF G17 THEN P12	0.8	0.2	0.6
R49	IF G35 THEN P12	0.8	0.0	0.8
R50	IF G43 THEN P12	1.0	0.0	1.0
R51	IF G07 THEN P13	0.8	0.2	0.6
R52	IF G08 THEN P13	1.0	0.0	1.0
R53	IF G26 THEN P13	0.8	0.2	0.6
R54	IF G27 THEN P13	0.8	0.2	0.6
R55	IF G33 THEN P13	0.4	0.2	0.2
R56	IF G34 THEN P13	0.6	0.2	0.4

### 3.4.8 Proses Perhitungan Pada Aplikasi Sistem Pakar

Pada sesi diagnosa penyakit, user diberikan pilihan interpretasi yang masing-masing memiliki nilai CF sebagai berikut:

- (Tidak) Pasti Tidak = - 1.0
- (Tidak) Hampir Pasti Tidak = - 0.8
- (Tidak) Kemungkinan Besar Tidak = - 0.6
- (Tidak) Mungkin Tidak = - 0.4
- (Tidak) Tidak Tahu = 0,0
- (Ya) Mungkin ya = 0.4
- (Ya) Kemungkinan besar ya = 0.6
- (Ya) Hampir pasti ya = 0.8
- (Ya) Pasti ya = 1.0

Proses penghitungan prosentase keyakinan diawali dengan pemecahan sebuah kaidah (rule) yang memiliki gejala majemuk, menjadi kaidah-kaidah (rules) yang memiliki gejala tunggal. Kemudian masing-masing rule baru dihitung CF nya dengan menggunakan persamaan 2.2:

$$CF_{gejala} = CF(user) * CF(pakar)$$

Diantara kondisi yang terjadi adalah terdapat beberapa antensenden (dalam rule yang berbeda) dengan satu konsekuensi yang sama. Dalam kasus ini, kita harus mengagregasikan nilai CF keseluruhan dari setiap kondisi yang ada. Berikut formula yang digunakan:

$$CF_c (CF_1, CF_2) = CF_1 + CF_2 (1 - CF_1) ; \text{ jika } CF_1 \text{ dan } CF_2 \text{ keduanya positif}$$

$$CF_c (CF_1, CF_2) = CF_1 + CF_2 (1 + CF_1) ; \text{ jika } CF_1 \text{ dan } CF_2 \text{ keduanya negatif}$$

$$CF_c (CF_1, CF_2) = \{CF_1 + CF_2\} / (1 - \min\{|CF_1|, |CF_2|\}) ; \text{ jika salah satu negatif}$$

## Contoh :

[R1] : IF Demam THEN thypus {cf : -0.40}

[R2] : IF Jumlah trombosit rendah THEN thypus {cf : -0.50}

[R3] : IF Tubuh lemah THEN thypus {cf : 0.75}

[R4] :IF Diare THEN thypus {cf : 0.60}

Tentukan Nilai dari CF gabungannya:

Jawab:

1.R1 dan R2 :: CFc ( $CF_1, CF_2$ ) =  $CF_1 + CF_2(1+CF_1)$

$$= -0,40 + (-0,50)(1+(-0,40))$$

$$= -0,40 + (-0,50)(0,60)$$

= -0,40 – 0,30

2. R3 dan R4 :: CFc ( $CF_1, CF_2$ ) =  $CF_1 + CF_2(1 - CF_1)$

$$= 0,75 + 0,6(1-0,75)$$

$$= 0,75 + 0,6 \cdot 0,25$$

$$= 0,75 + 0,15$$

3. Gabungkan (a) dan (b) :: CFc ( $CF_1, CF_2$ ) =  $\{CF_1 + CF_2\} / (1 - \min\{|CF_1|, |CF_2|\})$

$$= \{-0,70+0,9\} / (1 - \min\{|-0,70|, |0,90|\})$$

$$= 0,20 / ( 1-\{0,70\})$$

= 0,20 / 0,30

= 0,67

Keterangan: CF terakhir merupakan CF penyakit, berdasarkan hasil perhitungan CF di atas, maka CF penyakit adalah 0.67. Selanjutnya hitung persentase keyakinan terhadap penyakit dengan persamaan

$$\begin{aligned}
 \text{Persentase} &= \text{CF}_{\text{penyakit}} * 100 \\
 &= 0.67 * 100 \\
 &\equiv 67\%
 \end{aligned}$$

Berdasarkan hasil perhitungan, maka keterangan tingkat keyakinan berdasarkan tabel interpretasi adalah **KEMUNGKINAN BESAR**.

### **3.4.9 Contoh Perhitungan Manual Metode *Certainty Factor***

Untuk lebih memahami sistem ini berjalan, berikut 3 contoh kasus dengan melakukan perhitungan manual dengan menggunakan metode *certainty factor*.

#### **Contoh Kasus I :**

Seorang peternak mendapati ayam miliknya terkena penyakit, dia melihat bahwa ayam miliknya terdapat gejala nafsu makan berkurang, kaki dan sayap lumpuh, kepala terputar, dan kelumpuhan pada tembolok.

Pemecahan Kasus:

Diketahui:

1. Nafsu makan berkurang (G01), merupakan gejala yang dimiliki oleh Berak Kapur (P01), Kolera Ayam (P02), Tetelo (P04), Tipus Ayam (P05), Gumboro (P07), dan Mareks (P12).
2. Kaki dan sayap lumpuh (G17), merupakan gejala yang dimiliki oleh Tetelo (P04) dan Mareks (P12).
3. Kepala terputar (G25), merupakan gejala yang dimiliki oleh Tetelo (P04) dan Mareks (P12).
4. Kelumpuhan pada tembolok (G35), merupakan gejala yang dimiliki oleh Mareks (P12).

Perhitungan Manual:

1. Tipus Ayam (P05) =

Nafsu makan berkurang (G01) = 0.4 = Nilai Paling tinggi untuk (G01)

$$CF(H, E) = CF(E) * CF(RULE)$$

$$\text{CF 1} = 1 * 0.4$$

$$(G11) = 0.4$$

2. Tetelo (P04) =

Nafsu makan berkurang (G01) = 0.2

Kaki dan sayap lumpuh (G17) = 0.6

Kepala terputar (G20) = 0.4

$$CFkombinasi(CFlama, CFbaru) = CFlama + CFbaru * (1 - Cflama)$$

$$\text{CFkombinasi G01,G17} = 0.2 + 0.6 * (1 - 0.2)$$

$$= 0.68$$

$$\text{CFkombinasi G17, G20} = 0.68 + 0.4 * (1 - 0.68)$$

$$= 0.808$$

3. Mareks (P12) =

Nafsu makan berkurang (G01) = 0.2

Kaki dan sayap lumpuh (G17) = 0.6

Kepala terputar (G20) = 0.6

Kelumpuhan pada tembolok (G35) = 0.8

$$CFkombinasi(CFlama, CFbaru) = CFlama + CFbaru * (1 - Cflama)$$

$$\text{CFkombinasi G01,G17} = 0.2 + 0.6 * (1 - 0.2)$$

$$= 0.68$$

$$\text{CFkombinasi G17, G20} = 0.68 + 0.6 * (1 - 0.68)$$

$$\begin{aligned}
 &= 0.872 \\
 \text{CFkombinasi G20, G35} &= 0.872 + 0.8 * (1 - 0.872) \\
 &= 0.974
 \end{aligned}$$

Hasil Perhitungan :

Dari perhitungan di atas maka dapat diambil kesimpulan bahwa hasil diagnosa dengan kemungkinan penyakit terbesar yang menyerang ayam milik peternak tersebut adalah penyakit Mareks (P12) dengan nilai CF sebesar  $0.974 * 100\% = 97.4\%$ .

### **Contoh Kasus II :**

Salah seorang mahasiswa jurusan kedokteran hewan di salah satu perguruan tinggi di Yogyakarta, memelihara seekor ayam. Namun, ayam miliknya terjangkit penyakit dengan gejala nafas sesak atau megap-megap, keluar cairan dari mata dan hidung, dan pembengkakan daerah fasial dan sekitar mata.

Pemecahan Kasus :

Diketahui:

1. Nafas sesak atau megap-megap (G02) dimiliki oleh penyakit Salesma Ayam (P08).
2. Keluar cairan dari mata dan hidung (G18) dimiliki oleh penyakit Salesma Ayam (P08).
3. Pembengkakan daerah fasial dan sekitar mata (G29) dimiliki oleh penyakit Salesma Ayam (P08).

Perhitungan Manual :

1. Salesma Ayam (P08):

Nafas sesak atau megap-megap (G02) = 0.8

Keluar cairan dari mata dan hidung (G18) = 0.1

Pembengkakan daerah fasial dan sekitar mata (G29) = 1.0

$$CF_{kombinasi}(CF_{flama}, CF_{baru}) = CF_{flama} + CF_{baru} * (1 - CF_{flama})$$

$$\begin{aligned} CF_{kombinasi} \text{ G08, G18} &= 0.8 + 1.0 * (1 - 0.8) \\ &= 1.0 \end{aligned}$$

$$\begin{aligned} CF_{kombinasi} \text{ G18, G29} &= 1.0 + 1.0 * (1 - 1.0) \\ &= 1.0 \end{aligned}$$

Hasil Perhitungan :

Dari perhitungan di atas, dapat disimpulkan bahwa hasil diagnosa penyakit yang menyerang ayam milik mahasiswa tersebut adalah “Salesma Ayam” dengan nilai kemungkinan terbesar  $1.0 * 100\% = 100\%$ .

**Contoh Kasus III :**

Misalkan pada penyakit Osteoarthritis ditunjukkan oleh gejala nyeri pada sendi, Bengkak pada sendi, nyeri pada sendi dipagi hari, turun berat badan. Seandainya diketahui dari pakar penyakit tulang bahwa probabilitas berpenyakit Osteoarthritis adalah 0.03

(nyeri pada sendi /tulang) = 0.03

(bengkak pada sendi )= 0.5

(nyeri sendi pada pagi hari) = 0.4

(rasa hangat pada sendi) = 0.4

(merah pada sendi yang sakit) = 0.1	E1 : nyeri sendi pada pagi hari
(Berat badan menurun ) = 0.1	E2 : bengkak pada sendi
	E3 : rasa hangat pada sendi
dengan menganggap :	E4 : merah pada sendi yang sakit
H : nyeri pada sendi /tulang	E5 : Berat badan menurun

$$\text{MB (H,E1)} = (0.4 - 0.03) / (1-0.03)$$

$$= 0.37 / 0.97$$

$$= 0.381$$

$$\text{MD (H, E1)} = (0.03 - 0.03) / (0 - 0.03) = 0$$

$$\text{CF (H, E1)} = \text{MB (H, E1)} - \text{MD (H, E1)}$$

$$= 0.381 - 0$$

$$= 0.381$$

$$\text{CF1} = 0.381$$

Dengan cara yang sama sistem menghitung tingkat kepastian penyakit Osteoarthritis berdasarkan gejala bengkak pada sendi

$$\text{MB (H, E2)} = (0.5 - 0.03) / (1-0.03)$$

$$= 0.47 / 0.97$$

$$= 0.484$$

$$\text{MD (H,E2)} = 0.03 - 0.03) / (0-0.03) = 0$$

$$\text{CF (H, E2)} \text{ MB (H,E2)} = 0.484 - 0$$

$$= 0.484$$

$$\text{CF2} = 0.484$$

Tingkat kepastian penyakit Osteoarthritis berdasarkan gejala Rasa hangat pada sendi

$$\text{MB (H, E3)} = (0.4 - 0.03) / (1 - 0.03)$$

$$= 0.37 / 0.97$$

$$= 0.381$$

$$\text{MD (H,E3)} = (0.03 - 0.03) / (0 - 0.03) = 0$$

$$\text{CF (H, E3)} = \text{MB (H,E3)} - \text{MD (H,E3)}$$

$$= 0.381 - 0$$

$$= 0.381$$

$$\text{CF3} = 0.381$$

Tingkat kepastian penyakit Osteoarthritis berdasarkan gejala merah pada sendi yang sakit

$$\text{MB (H, E4)} = (0.1 - 0.03) / (1 - 0.03)$$

$$= 0.07 / 0.97$$

$$= 0.072$$

$$\text{MD (H,E4)} = (0.03 - 0.03) / (0 - 0.03) = 0$$

$$\text{CF (H, E4)} = \text{MB (H,E4)} - \text{MD (H,E4)}$$

$$= 0.072 - 0$$

$$= 0.072$$

$$\text{CF4} = 0.072$$

Tingkat kepastian penyakit Osteoarthritis berdasarkan gejala berat badan menurun

$$\text{MB (H, E5)} = (0.1 - 0.03) / (1 - 0.03)$$

$$= 0.07 / 0.97$$

$$= 0.072$$

$$MD (H,E5) = (0.03 - 0.03) / (0-0.03) = 0$$

$$CF (H, E5) = MB (H,E5) - MD (H,E5)$$

$$= 0.072-0$$

$$= 0.072$$

$$CF5 = 0.072$$

Dari kelima perhitungan di atas, ketika sistem menyimpulkan bahwa penyakit yang diderita pasien adalah Osteoarthritis maka tingkat kepastiannya adalah hasil perhitungan berikut ini:

$$CF_{kombinasi} (CF1, CF2, CF3, CF4, CF5) = CF$$

$$(H,E1) + CF (H,E2) + CF (H,E3) + CF (H,E4) (1-CF(H,E1))$$

$$CF_k = CF(H,E1) + CF(H,E2) (1-CF(H,E1))$$

$$= 0.381 + 0.484 (1-0.381)$$

$$= 0.619 * 0.484 + 0.381$$

$$= 0.3 + 0.381$$

$$CF_{k1} = 0.681$$

$$CF_{k2} = CF_{k1} + CF(H,E3) (1-CF_{k1})$$

$$= 0.681 + 0.381 (1-0.681)$$

$$= 0.319 * 0.381 + 0.68$$

$$= 0.121539 + 0.681$$

$$CF_{k2} = 0.803$$

$$CF_{k3} = CF_{k2} + CF(H,E4) (1-CF_{k2})$$

$$= 0.803 + 0.1 (1-0.803)$$

$$= 0.197 * 0.1 + 0.803$$

$$= 0.0197 + 0.803$$

$$CFk3 = 0.823$$

$$CFk4 = CFk3 \cdot CF(H,E4) \cdot (1-CFk3)$$

$$= 0.823 + 0.1(1-0.823)$$

$$= 0.177 * 0.1 + 0.823$$

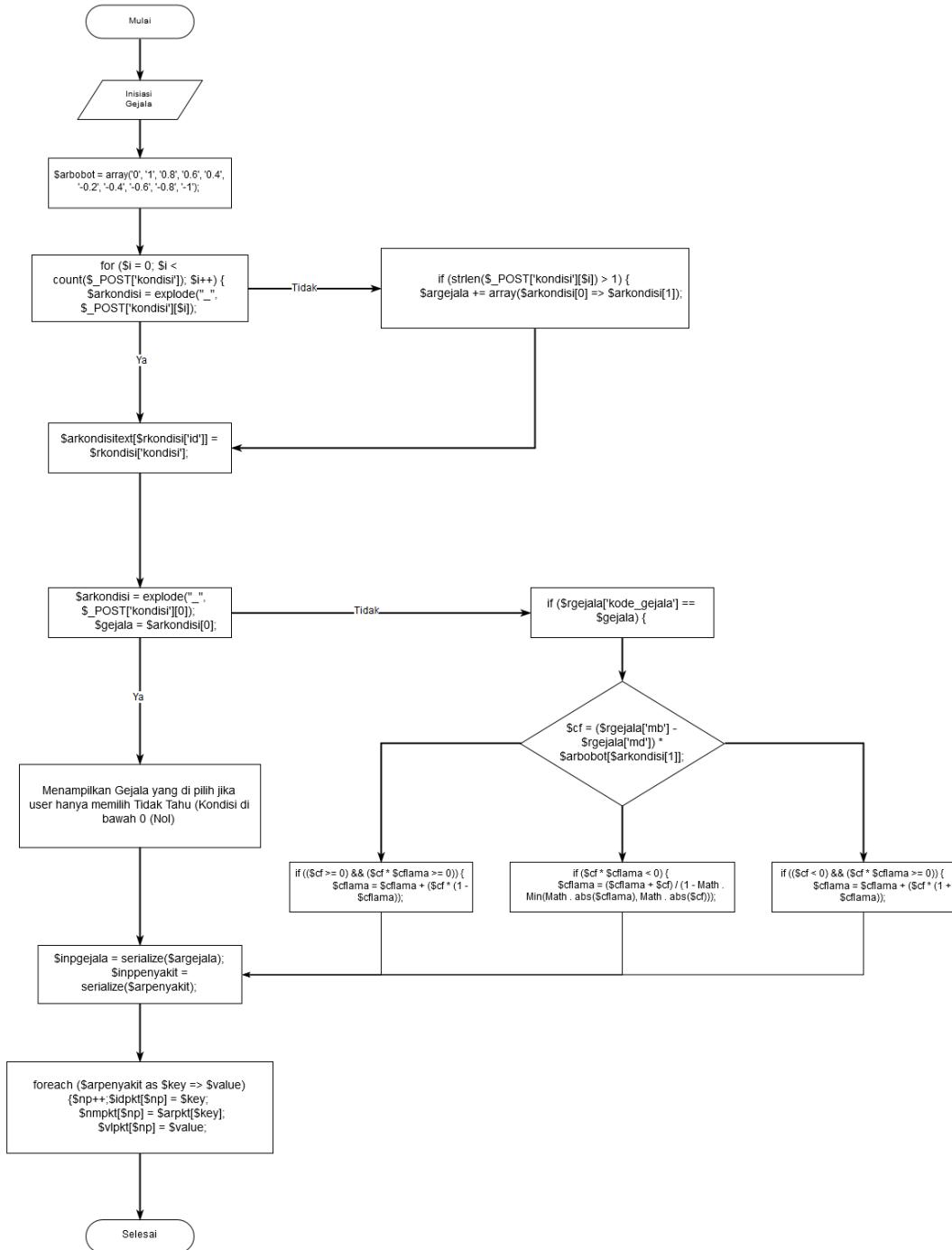
$$= 0.0177 + 0.823$$

$$CFk4 = 0.8407$$

Hasil dari perhitungan menunjukkan bahwa nilai kepastian pasien menderita penyakit Osteoarthritis dengan tingkat kepastian 0.8407

### **3.4.10 Inferensi Diagnosa**

Diagnosa adalah penentuan penyakit yang menyerang ayam berdasarkan data-data yang diberikan kepada user. Dalam melakukan diagnosa, sistem ini menggunakan algoritma runut maju (*forward chaining*). Runut maju digunakan untuk menentukan penyakit yang menyerang ayam berdasarkan gejala yang dimasukkan oleh user untuk menentukan calon konklusi.



**Gambar 3.2 Algoritma Diagnosa Penyakit Ayam**

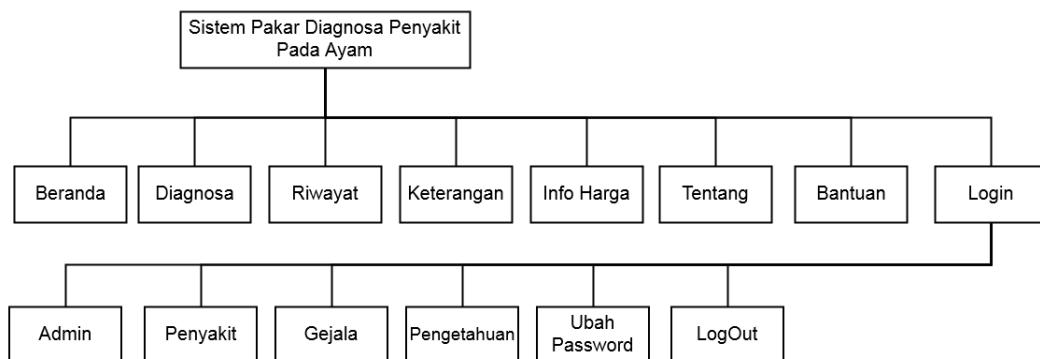
Penentuan penyakit yang menyerang ayam dilakukan dengan memilih di antara calon konklusi yang memiliki CF tertinggi. Setelah mendapatkan nilai CF tertinggi, sistem akan secara otomatis akan menampilkan penyakit yang terpilih

dengan CF terbesar serta tindakannya, sesuai dengan penyakit yang terpilih.

Gambar 3.2 menunjukkan penelusuran secara runut maju setelah diperoleh calon-calon konklusi.

### 3.5 Struktur Program & Perancangan sistem

Gambar 3.3 merupakan struktur program dalam perancangan sistem ini.



**Gambar 3.3 Struktur Program Sistem Pakar Ayam**

#### 3.5.1 Solusi Yang Dapat Diterapkan

1. Mempermudah mengkodekan pengetahuan ke dalam sebuah sistem pakar.
2. Menangkap sifat-sifat penting masalah dan membuat informasi dapat diakses oleh prosedur pemecahan masalah.
3. Memberikan informasi gejala yang diderita dan memberikan nilai kepastian.
4. Memberikan hasil yang lebih akurat dalam memecahkan masalah.
5. Membantu pakar menambah, mengedit, menghapus data system.
6. Memberikan hak akses terhadap pakar yang bersangkutan.

### **3.5.2 Solusi Yang Dipilih**

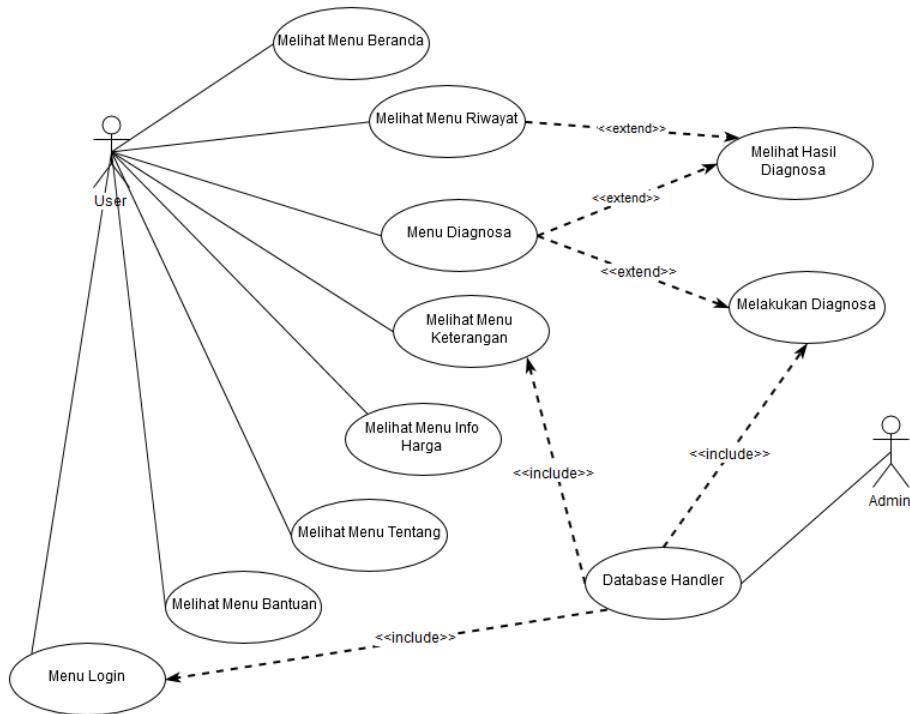
Berdasarkan analisis masalah yang telah dibuat dengan menggunakan metode representasi pengetahuan dan metode inferensi, dipilih solusi dimana representasi dimaksudkan untuk menangkap sifat-sifat penting masalah dan membuat informasi yang bisa diakses oleh prosedur pemecahan masalah. Basis pengetahuan dan mesin inferensi merupakan komponen utama yang harus dimiliki oleh sebuah sistem pakar.

## **3.6 Perancangan UML**

UML (*Unified Modeling Language*) adalah sebuah bahasa yang menggunakan grafik atau gambar untuk memvisualisasi, menspesifikasi, membangun, dan mendokumentasikan dari sebuah sistem pengembangan *software* berbasis *object-oriented*.

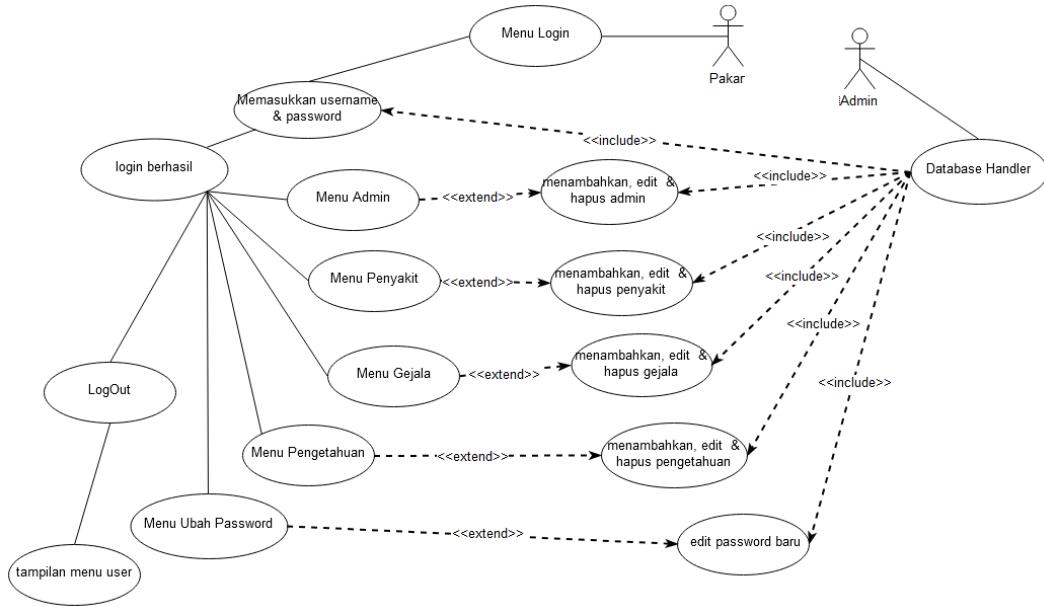
### **3.6.1 Use Case Diagram**

Berikut ini adalah bentuk dari *use case diagram user* dan pakar yang digambarkan dalam perancangan sistem. Di Gambar 3.4 akan dijelaskan siapa dan apa saja yang dilakukan ketika aplikasi berjalan dengan *user* yaitu tanpa hak akses *login*.



**Gambar 3.4 Use Case Diagram User Aplikasi Sistem Pakar Ayam**

Pada gambar *use case* diagram *user* terdapat dua *actor* yaitu “Admin” dan “User”. Admin di aplikasi sistem pakar ini berperan dalam pembuatan database handler yang berisi basis pengetahuan, dimana basis pengetahuan ini nantinya akan disertakan untuk *user* bisa melakukan diagnosa, melihat penyakit, dan saran yang diberikan setelah diagnosa penyakit Ayam. Aktor *user* pada *use case* diagram di atas, merupakan pengguna aplikasi ini nantinya. Dimana *User* mampu melihat beranda, menu diagnosa dan melakukan diagnosa, serta melihat hasil diagnosa. Dan *user* mampu melihat menu “keterangan” dan menu “tentang”, serta *user* bisa melihat menu *login*, jika *user* memiliki hak akses pakar maka *user* akan di bawa ke *use case* pada Gambar 3.5.

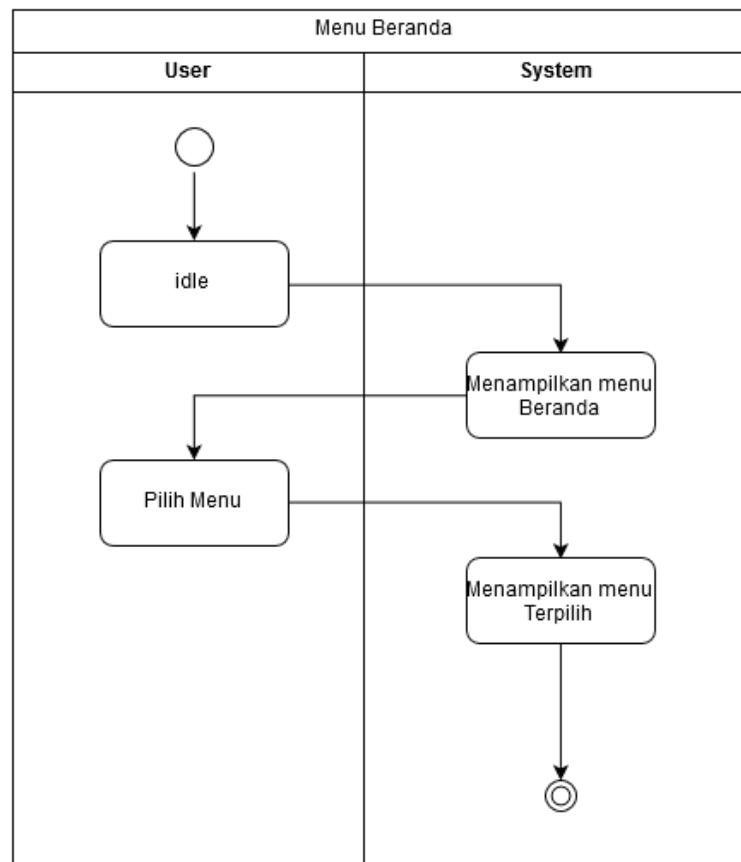


**Gambar 3.5 Use Case Diagram Pakar Aplikasi Sistem Pakar Ayam**

Pada gambar *use case* diagram Pakar terdapat dua *actor* yaitu “Admin” dan “Pakar”. Admin di aplikasi sistem pakar ini berperan dalam pembuatan database handler yang berisi gejala, penyakit, pengetahuan, admin dan password, dimana gejala, penyakit, pengetahuan ini nantinya akan di isi oleh pakar untuk diagnosis *user*. Pakar di beri hak akses berupa *username* dan *password* dari admin untuk melakukan proses login, jika *login* berhasil maka pakar dapat menginputkan, mengedit atau menghapus daftar gejala, penyakit, pengetahuan. Pakar juga bisa mengundang admin baru untuk membantu dalam pengisian, juga dapat mengubah password lama ke password yang baru di dalam sistem, pakar telah dimudahkan dengan bantuan tombol-tombol dan informasi yang di sesuaikan di setiap menu. Hasil tindakan dari pakar langsung terhubung dengan database dengan konsep *CRUD*, istilah tersebut merupakan singkatan dari *Create*, *Read*, *Update* dan *Delete*.

### 3.6.2 Activity Diagram

*Activity Diagram* menggambarkan aktivitas yang akan dilakukan pada sistem pakar ini aktivitas berdasarkan user dan pakar dengan hak akses ke menu login. Pada sistem ini terdapat sebelas proses *activity diagram* yang disajikan pada Gambar 3.6 sampai Gambar 3.16.

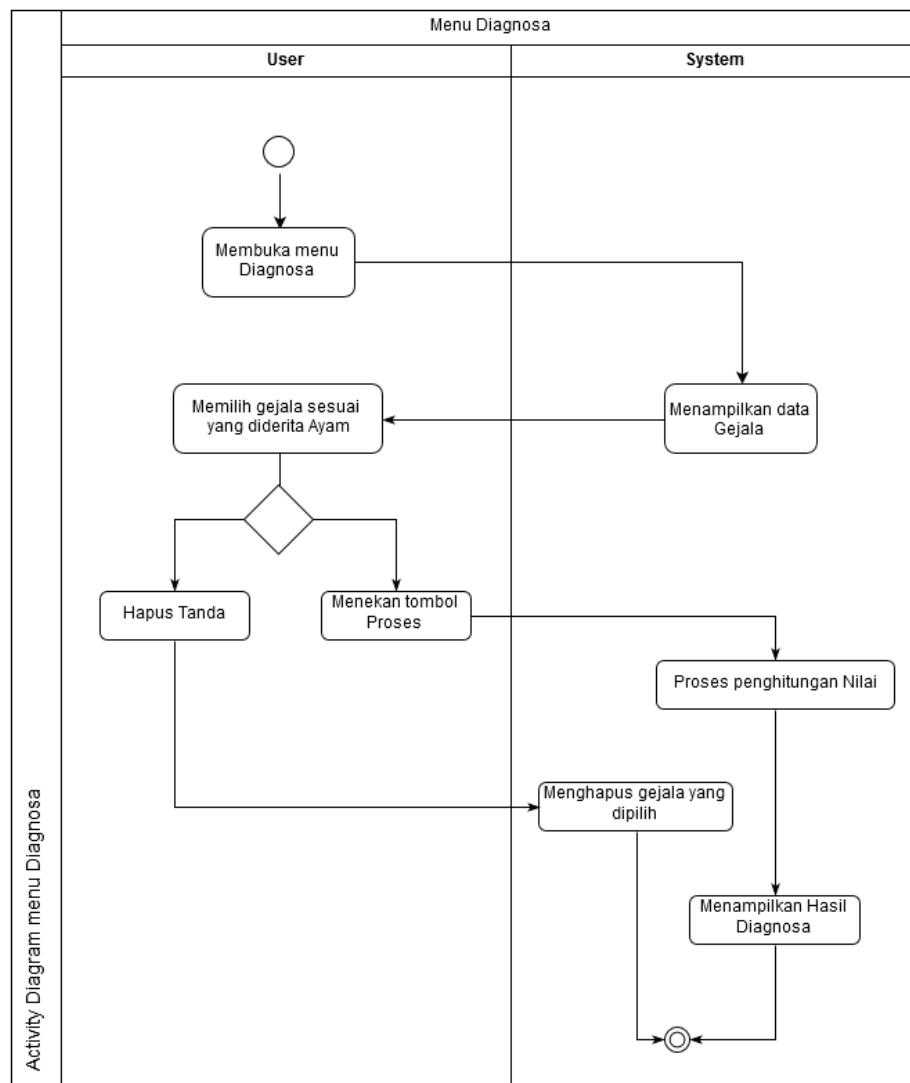


**Gambar 3.6 Activity Diagram Menu Beranda**

*Activity diagram* pada Gambar 3.6 terdiri dari *user* dan *system*. Proses pada Gambar 3.6 dijelaskan sebagai berikut.

1. *User* mengawali aktifitas masuk di *system* dengan loading atau idle, dan saat *user* membuka aplikasi *system* akan menampilkan menu Beranda.

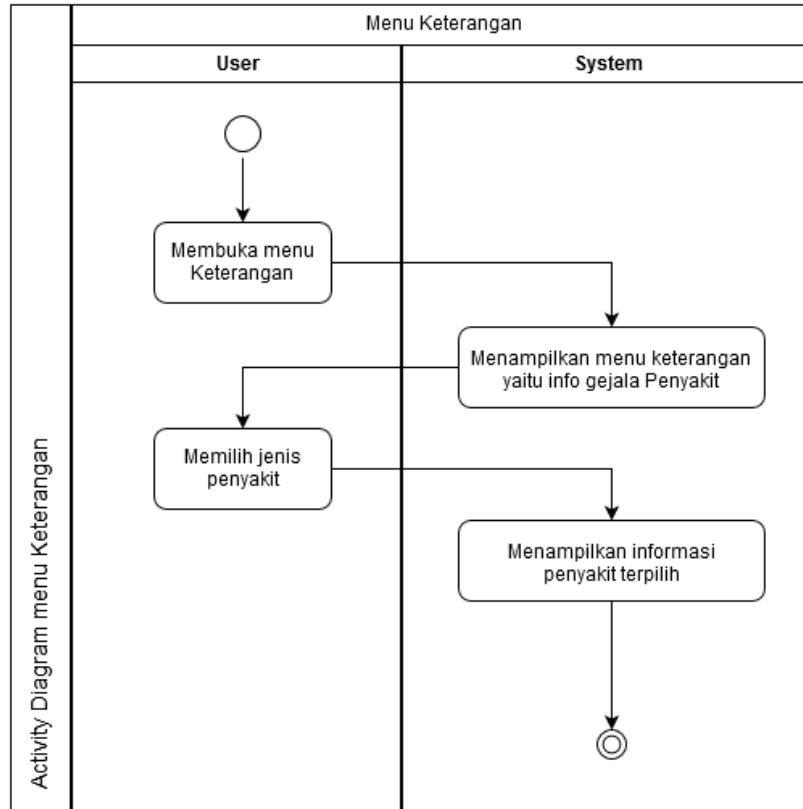
2. Kemudian *user* akan memilih salah satu menu yang dikehendaki yaitu menu diagnosa, keterangan, tentang dan login.
3. Selanjutnya, *system* akan menampilkan menu terpilih di menu utama.



**Gambar 3.7 Activity Diagram Menu Diagnosa**

Activity Diagram pada Gambar 3.7 terdiri dari *user* dan sistem. Proses pada Gambar 3.7 dapat dijelaskan sebagai berikut.

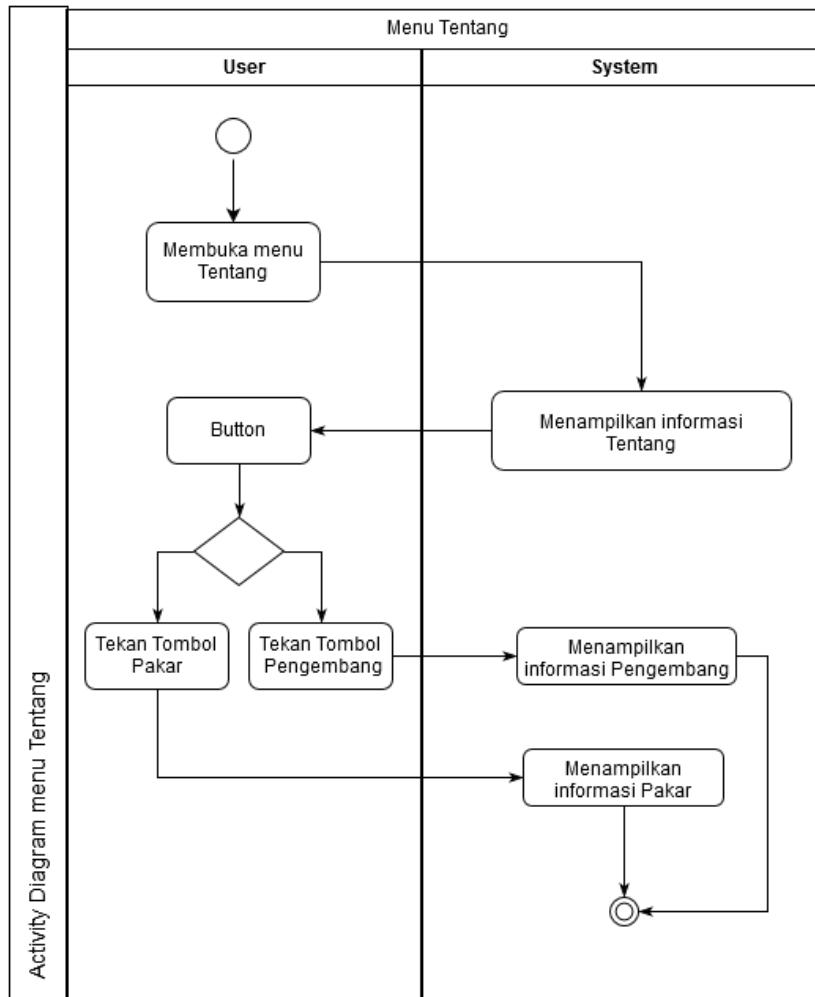
1. *User* mengawali aktifitas, memilih menu diagnosa dan sistem akan menampilkan *list* gejala.
2. Kemudian *user* mencari dan memilih gejala yang sesuai dengan yang di derita ayam. Jika *user* membatalkan tanda centang di *list* diagnosa, maka gejala yang dipilih *user* sebelumnya akan terhapus. Namun, jika *user* selesai memilih kemudian menekan *button* “Proses”. Maka sistem akan memproses penghitungan nilai berdasarkan gejala yang telah dipilih *user*. Setelah sistem menyelesaikan proses perhitungan, maka sistem menampilkan hama atau penyakit dengan nilai terbesar dan calon konklusi yang telah menyerang ayam berdasarkan gejala yang telah dipilih *user*.
3. Selanjutnya *user* akan memperoleh informasi data penyakit yang telah menyerang ayam sesuai dengan pilihan yang di derita.



**Gambar 3.8 Activity Diagram Menu Keterangan**

Gambar 3.8 merupakan *activity diagram* menu keterangan. Proses pada gambar 3.8 dapat dijelaskan sebagai berikut.

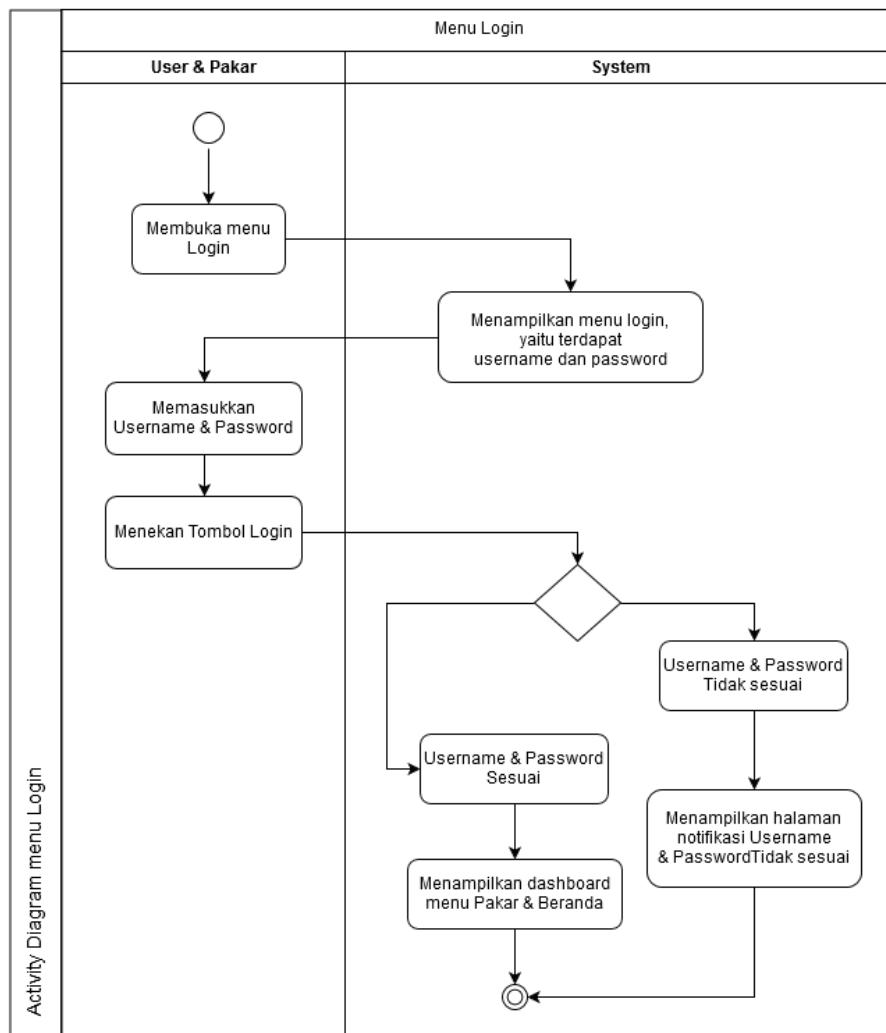
1. *User* mengawali aktifitas dengan memilih menu keterangan, dimana sistem akan menampilkan menu data informasi gejala dan penyakit pada ayam, termasuk tindakan yang harus di lakukan serta pencegahan.
2. Kemudian *user* memilih salah satu penyakit yang ingin dilihat oleh *user*, dan sistem akan menampilkan informasi tentang gejala dan penyakit pada ayam yang telah dipilih *user*.



**Gambar 3.9 Activity Diagram Menu Tentang**

Gambar 3.9 merupakan *activity diagram* menu tentang. Proses pada gambar 3.9 dapat dijelaskan berikut ini.

1. *User* mengawali aktifitas dengan memilih menu tentang. Kemudian sistem akan menampilkan informasi tentang aplikasi.
2. *User* menekan salah satu tombol yaitu pakar atau pengembang dan akan muncul informasi sesuai tombol yang di pilih.

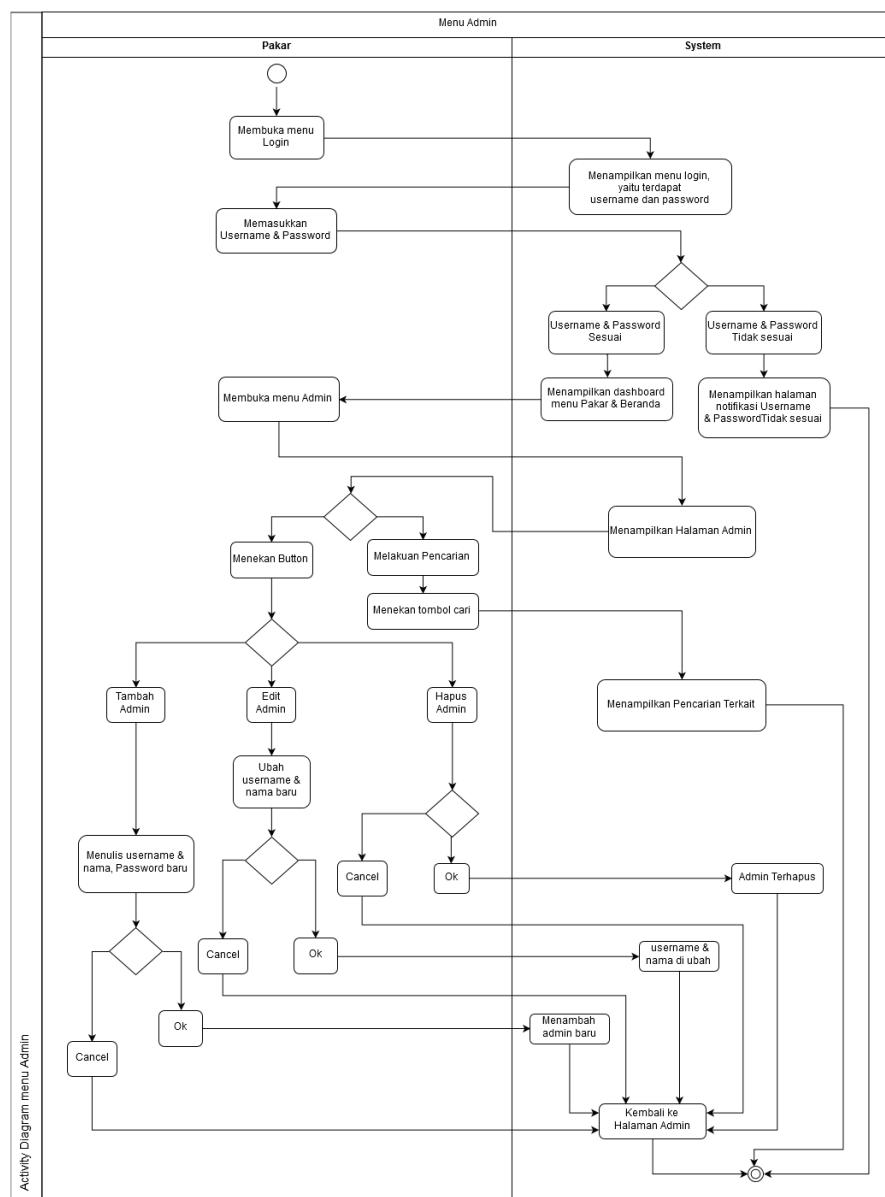


**Gambar 3.10 Activity Diagram Menu Login**

Gambar 3.12 merupakan *activity diagram* menu login. Proses pada gambar 3.10 dapat dijelaskan berikut ini.

1. *User* mengawali aktifitas dengan memilih menu *login*. Kemudian sistem akan menampilkan data login yaitu username dan password.
2. *User* akan bisa login jika sudah memiliki akun, atau sebagai pakar yang sudah di buatkan akun oleh admin, selanjutnya user akan menginputkan *username* dan *password* dan menekan tombol *login*.

3. Sistem akan mengidentifikasi kecocokan databasenya jika *username* dan *password* sesuai maka sistem akan menampilkan menu pakar.
4. Jika jika *username* dan *password* tidak sesuai dengan database maka sistem akan menampilkan halaman *username* dan *password* tidak sesuai dan ulangi silahkan kembali.



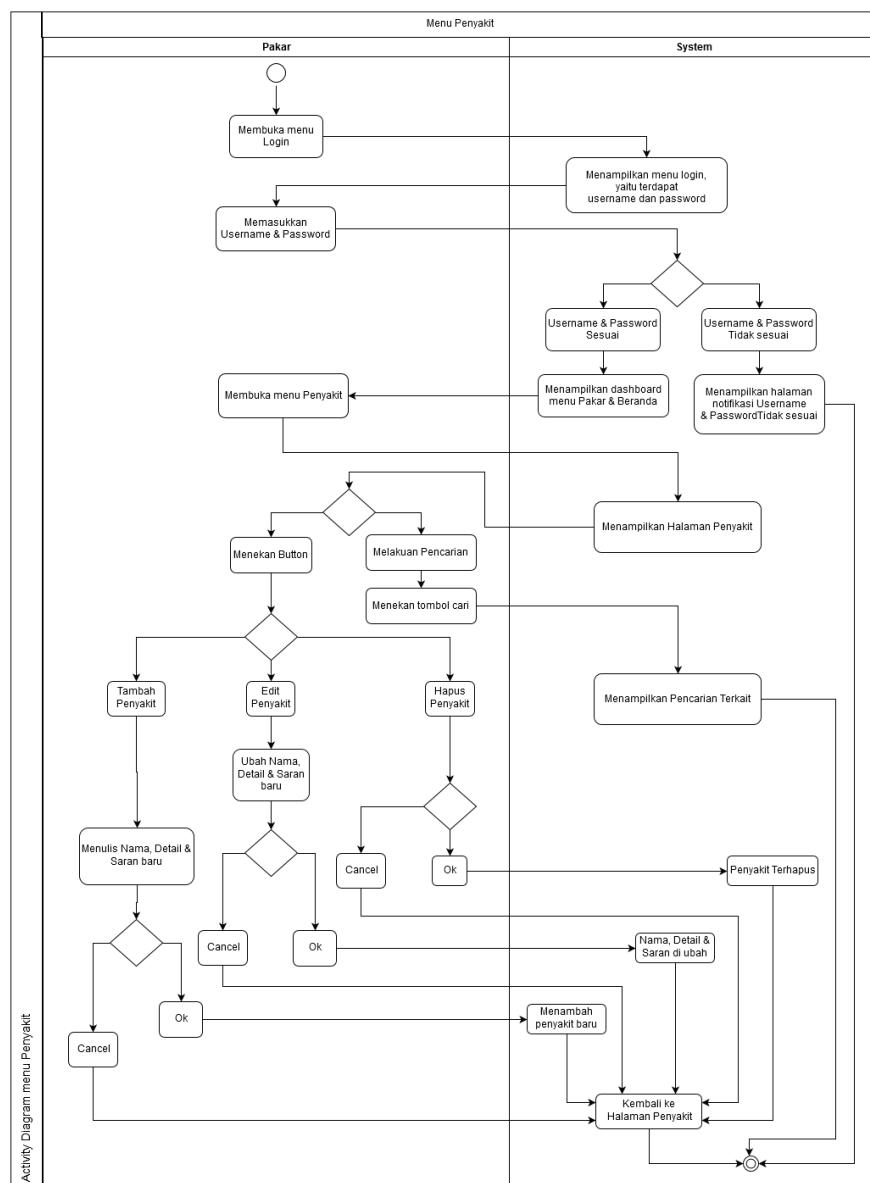
**Gambar 3.11 Activity Diagram Menu Admin**

Gambar 3.11 merupakan *activity diagram* menu admin dan sudah melalui proses *login* pakar, keterangan proses yang di tampilkan dibawah di mulai dari menu Admin. Proses pada gambar 3.11 dapat dijelaskan berikut ini.

1. Setelah *login* pakar, pakar memilih menu admin. Kemudian sistem akan menampilkan halaman admin.
2. Pakar menekan salah satu tombol yaitu tambah, edit atau hapus dan akan muncul pilihan sesuai tombol yang di pilih.
3. Jika pakar memilih tambah admin maka akan muncul form pengisian *username*, nama dan *password* baru, setelah selesai input pakar bisa menyimpannya dengan menekan tombol Ok dan sistem akan menambah admin baru, pakar juga dapat membatalkannya dengan menekan tombol *Cancel*, setelah kedua proses tersebut akan diarahkan ke menu admin kembali oleh sistem.
4. Jika pakar memilih edit admin maka akan muncul form edit *username*, dan nama baru dari username yang sedang login, setelah selesai mengganti pakar bisa menyimpannya dengan menekan tombol Ok dan sistem akan mengubah admin baru, pakar juga dapat membatalkannya dengan menekan tombol *Cancel*, setelah kedua proses tersebut akan diarahkan ke menu admin kembali oleh sistem.
5. Jika pakar memilih hapus admin maka akan muncul *popup* aksi, pakar bisa menghapus dengan menekan tombol Ok dan sistem akan menghapus admin yang di pilih, pakar juga dapat membatalkannya

dengan menekan tombol *Cancel*, setelah kedua proses tersebut akan diarahkan ke menu admin kembali oleh sistem.

6. Pakar juga bisa mencari admin dengan fitur pencarian, dengan menginput admin yang dicari dan menekan tombol cari, jika pencarian valid maka admin akan muncul, jika tidak akan ada pemberitauan tidak di temukan.



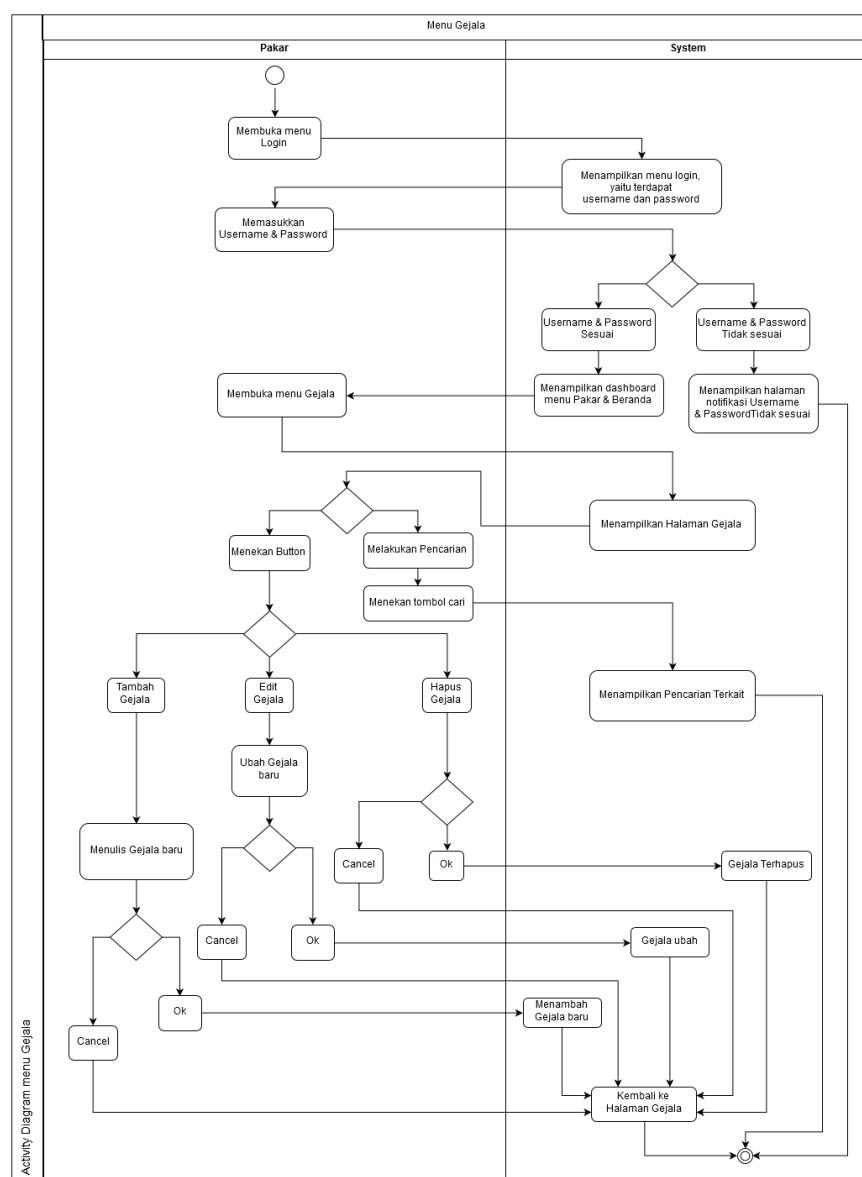
Gambar 3.12 *Activity Diagram* Menu Penyakit

Gambar 3.12 merupakan *activity diagram* menu penyakit dan sudah melalui proses *login* pakar, keterangan proses yang di tampilkan dibawah di mulai dari menu Penyakit. Proses pada gambar 3.12 dapat dijelaskan berikut ini.

1. Setelah *login* pakar, pakar memilih menu penyakit. Kemudian sistem akan menampilkan halaman penyakit.
2. Pakar menekan salah satu tombol yaitu tambah, edit atau hapus dan akan muncul pilihan sesuai tombol yang di pilih.
3. Jika pakar memilih tambah penyakit maka akan muncul form pengisian nama, detail dan saran baru, setelah selesai input pakar bisa menyimpannya dengan menekan tombol Ok dan sistem akan menambah penyakit baru, pakar juga dapat membatalkannya dengan menekan tombol *Cancel*, setelah kedua proses tersebut akan diarahkan ke menu penyakit kembali oleh sistem.
4. Jika pakar memilih edit penyakit maka akan muncul form edit nama penyakit, detail dan saran, setelah selesai mengganti pakar bisa menyimpannya dengan menekan tombol Ok dan sistem akan mengubah penyakit baru, pakar juga dapat membatalkannya dengan menekan tombol *Cancel*, setelah kedua proses tersebut akan diarahkan ke menu penyakit kembali oleh sistem.
5. Jika pakar memilih hapus penyakit maka akan muncul *popup* aksi, pakar bisa menghapus dengan menekan tombol Ok dan sistem akan menghapus penyakit yang di pilih, pakar juga dapat membatalkannya

dengan menekan tombol *Cancel*, setelah kedua proses tersebut akan diarahkan ke menu penyakit kembali oleh sistem.

6. Pakar juga bisa mencari penyakit dengan fitur pencarian, dengan menginput penyakit yang dicari dan menekan tombol cari, jika pencarian valid maka penyakit akan muncul, jika tidak akan ada pemberitauan tidak di temukan.



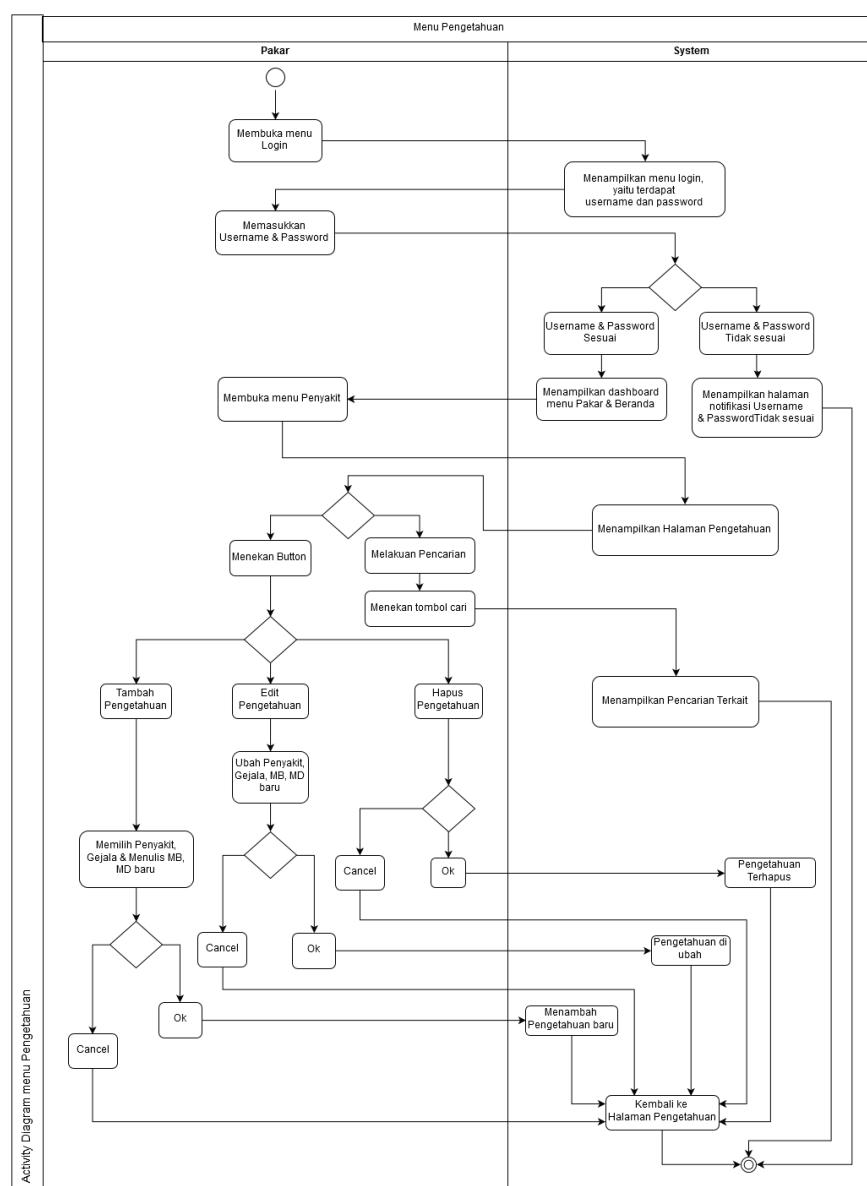
Gambar 3.13 *Activity Diagram* Menu Gejala

Gambar 3.13 merupakan *activity diagram* menu gejala dan sudah melalui proses *login* pakar, keterangan proses yang di tampilkan dibawah di mulai dari menu Gejala. Proses pada gambar 3.13 dapat dijelaskan berikut ini.

1. Setelah *login* pakar, pakar memilih menu gejala. Kemudian sistem akan menampilkan halaman gejala.
2. Pakar menekan salah satu tombol yaitu tambah, edit atau hapus dan akan muncul pilihan sesuai tombol yang di pilih.
3. Jika pakar memilih tambah gejala maka akan muncul form pengisian nama gejala baru, setelah selesai input pakar bisa menyimpannya dengan menekan tombol Ok dan sistem akan menambah gejala baru, pakar juga dapat membatalkannya dengan menekan tombol *Cancel*, setelah kedua proses tersebut akan diarahkan ke menu gejala kembali oleh sistem.
4. Jika pakar memilih edit gejala maka akan muncul form edit nama gejala, setelah selesai mengganti pakar bisa menyimpannya dengan menekan tombol Ok dan sistem akan mengubah gejala baru, pakar juga dapat membatalkannya dengan menekan tombol *Cancel*, setelah kedua proses tersebut akan diarahkan ke menu gejala kembali oleh sistem.
5. Jika pakar memilih hapus gejala maka akan muncul *popup* aksi, pakar bisa menghapus dengan menekan tombol Ok dan sistem akan menghapus gejala yang di pilih, pakar juga dapat membatalkannya

dengan menekan tombol *Cancel*, setelah kedua proses tersebut akan diarahkan ke menu gejala kembali oleh sistem.

6. Pakar juga bisa mencari gejala dengan fitur pencarian, dengan menginput gejala yang dicari dan menekan tombol cari, jika pencarian valid maka gejala akan muncul, jika tidak akan ada pemberitauan tidak di temukan.



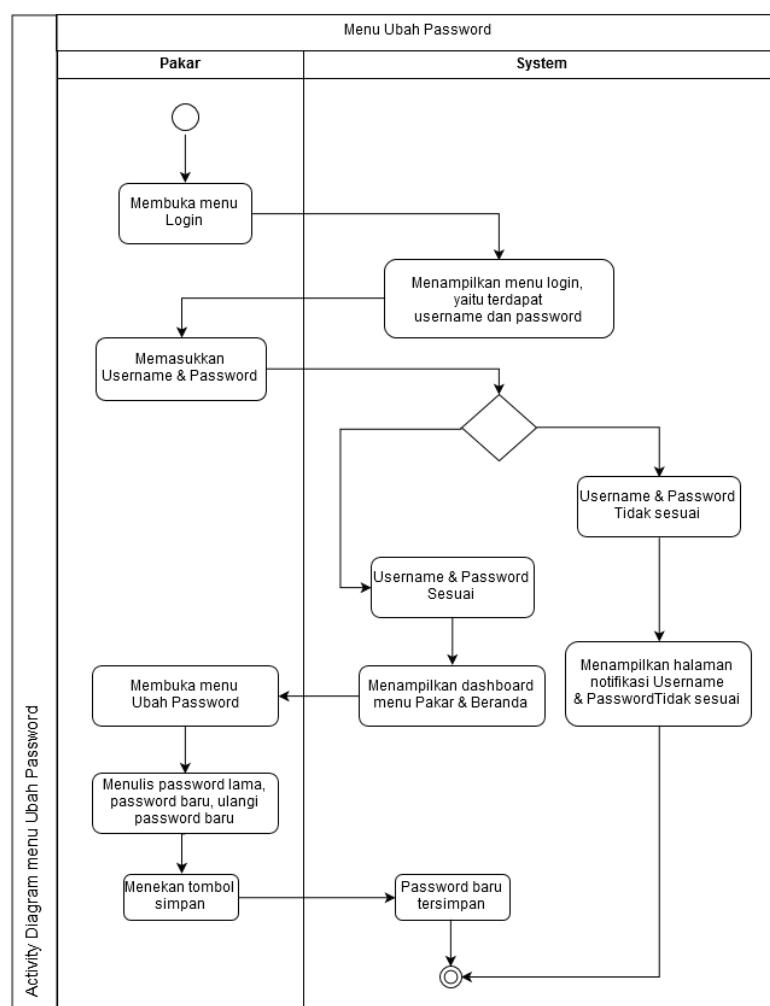
Gambar 3.14 *Activity Diagram* Menu Pengetahuan

Gambar 3.14 merupakan *activity diagram* menu pengetahuan dan sudah melalui proses *login* pakar, keterangan proses yang di tampilkan dibawah di mulai dari menu Pengetahuan. Proses pada gambar 3.14 dapat dijelaskan berikut ini.

1. Setelah *login* pakar, pakar memilih menu pengetahuan. Kemudian sistem akan menampilkan halaman pengetahuan.
2. Pakar menekan salah satu tombol yaitu tambah, edit atau hapus dan akan muncul pilihan sesuai tombol yang di pilih.
3. Jika pakar memilih tambah pengetahuan maka akan muncul form pemilihan penyakit, gejala dan pengisian mb, mb yang baru, setelah selesai input pakar bisa menyimpannya dengan menekan tombol Ok dan sistem akan menambah pengetahuan baru, pakar juga dapat membatalkannya dengan menekan tombol *Cancel*, setelah kedua proses tersebut akan diarahkan ke menu pengetahuan kembali oleh sistem.
4. Jika pakar memilih edit pengetahuan maka akan muncul form edit pemilihan penyakit, gejala, dan pengisian mb, mb setelah selesai mengganti pakar bisa menyimpannya dengan menekan tombol Ok dan sistem akan mengubah pengetahuan baru, pakar juga dapat membatalkannya dengan menekan tombol *Cancel*, setelah kedua proses tersebut akan diarahkan ke menu pengetahuan kembali oleh sistem.
5. Jika pakar memilih hapus pengetahuan maka akan muncul *popup* aksi, pakar bisa menghapus dengan menekan tombol Ok dan sistem

akan menghapus pengetahuan yang di pilih, pakar juga dapat membatalkannya dengan menekan tombol *Cancel*, setelah kedua proses tersebut akan di arahkan ke menu pengetahuan kembali oleh sistem.

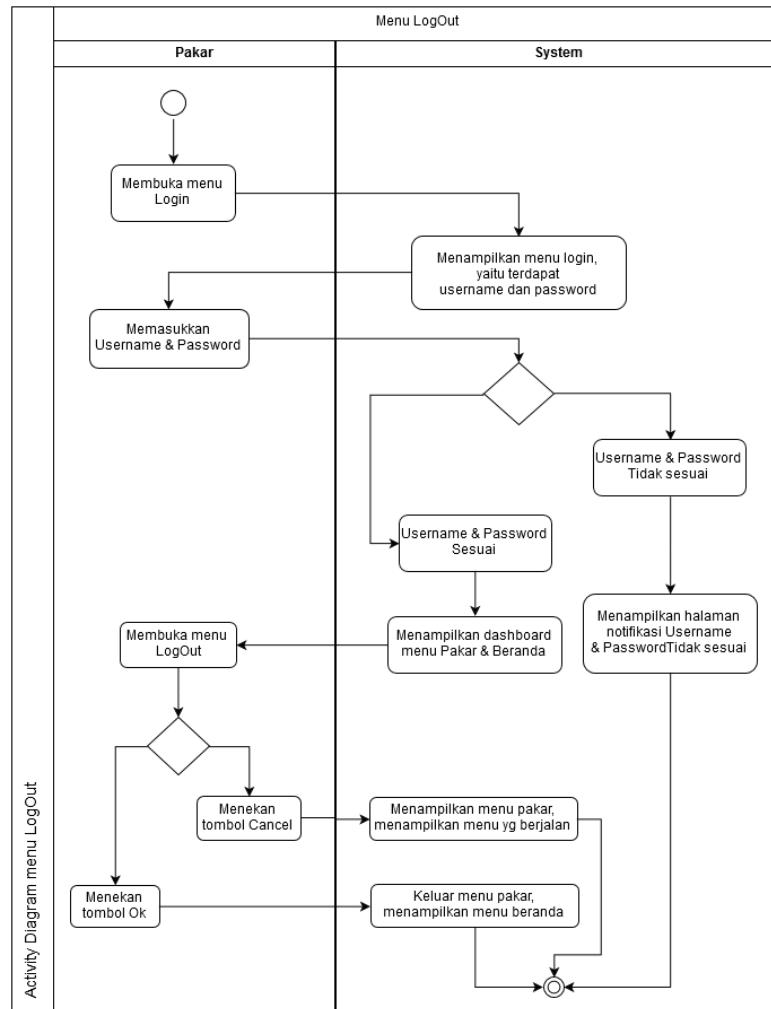
6. Pakar juga bisa mencari pengetahuan dengan fitur pencarian, dengan menginput pengetahuan yang dicari dan menekan tombol cari, jika pencarian valid maka pengetahuan akan muncul, jika tidak akan ada pemberitauan tidak di temukan.



**Gambar 3.15 Activity Diagram Menu Ubah Password**

Gambar 3.15 merupakan *activity diagram* menu ubah *password* dan sudah melalui proses *login* pakar, keterangan proses yang di tampilkan dibawah di mulai dari menu Ubah Password. Proses pada gambar 3.15 dapat dijelaskan berikut ini.

1. Setelah *login* pakar, pakar memilih menu Ubah *Password*. Kemudian sistem akan menampilkan halaman ubah password.
2. Pakar harus mengisi form ubah *password* dengan *password* lama, konfirmasi *password* lama dan *password* baru, jika *password* lama tidak sesuai database sistem, maka pakar tidak bisa mengubah *password*.
3. Setelah semua *password* sesuai maka pakar harus menekan tombol simpan untuk menyimpan *password* baru ke database sistem .



**Gambar 3.16 Activity Diagram Menu LogOut**

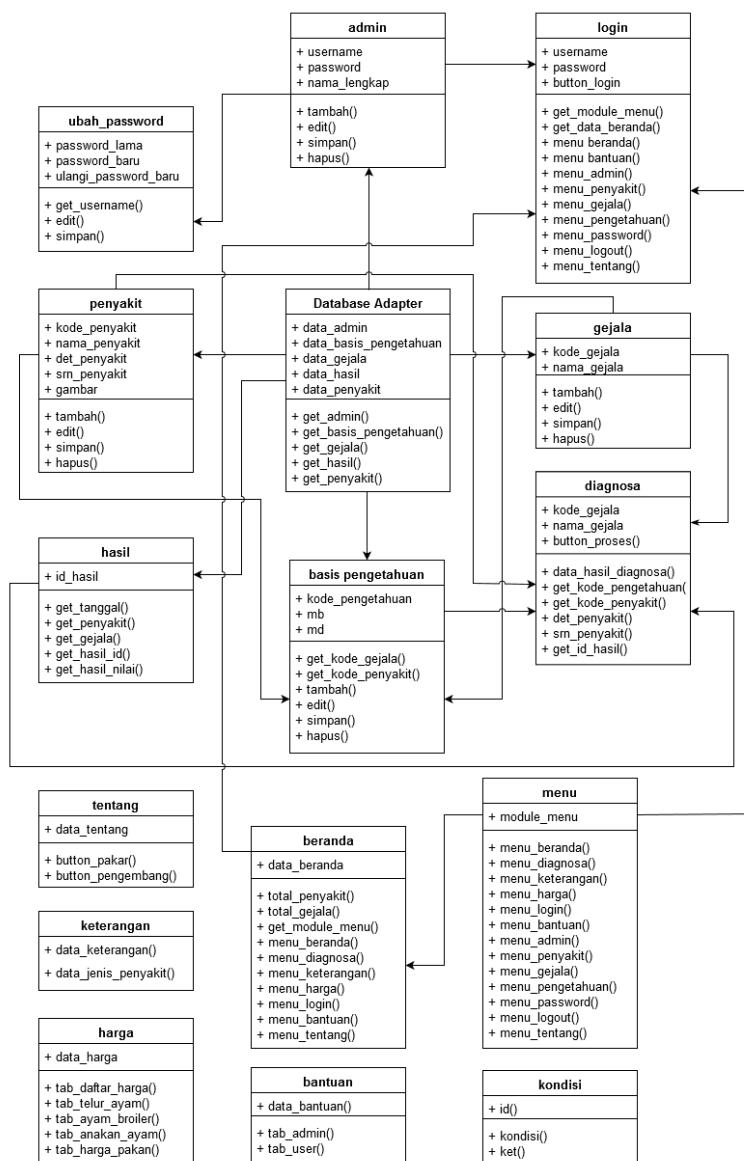
Gambar 3.16 merupakan *activity diagram* menu *logOut* dan sudah melalui proses *login* pakar, keterangan proses yang di tampilkan dibawah di mulai dari menu *Logout*. Proses pada gambar 3.18 dapat dijelaskan berikut ini.

1. Setelah *login* pakar, pakar memilih menu *Logout*. Kemudian sistem akan menampilkan *popup Logout*.
2. Pakar bisa keluar menu pakar dengan menekan tombol *Ok* dan sistem akan keluar menu pakar dan mengalihkan ke menu beranda, pakar juga dapat membatalkannya dengan menekan tombol *Cancel*, maka

pakar masih berada di menu pakar dan halaman sebelumnya yang sedang berjalan.

### 3.6.3 Class Diagram

*Class diagram* digunakan untuk menampilkan beberapa kelas serta paket-paket yang ada pada sistem aplikasi diagnosa penyakit ayam ini. Model *class diagram* pada sistem ini, bisa dilihat pada Gambar 3.17.

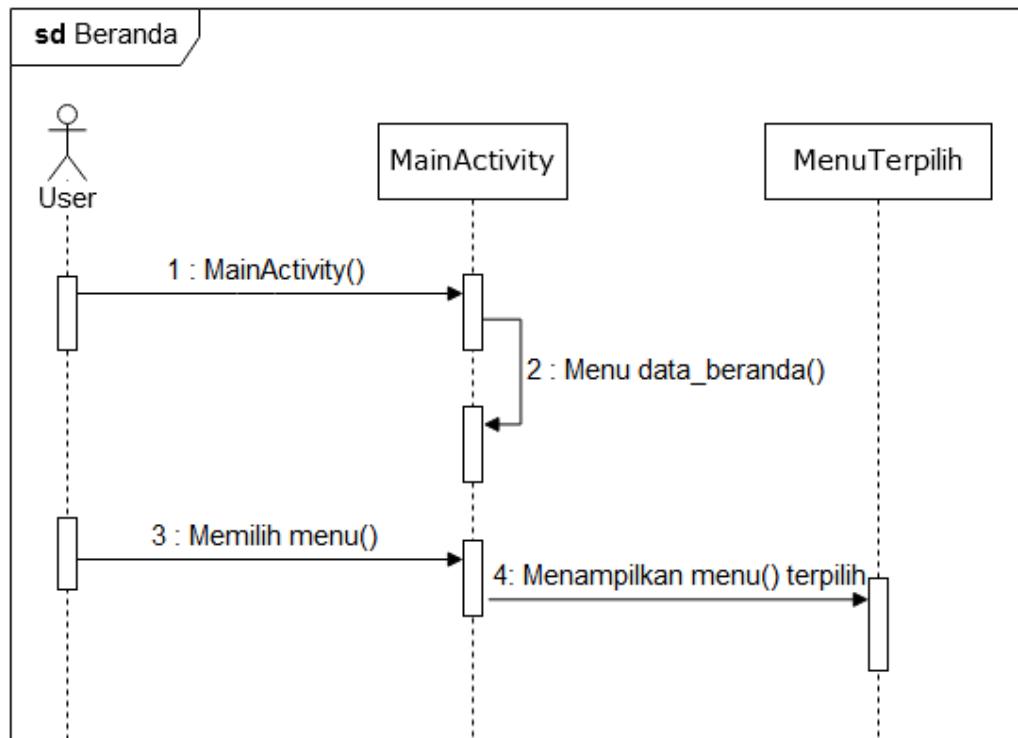


Gambar 3.17 Class Diagram Sistem Pakar Ayam

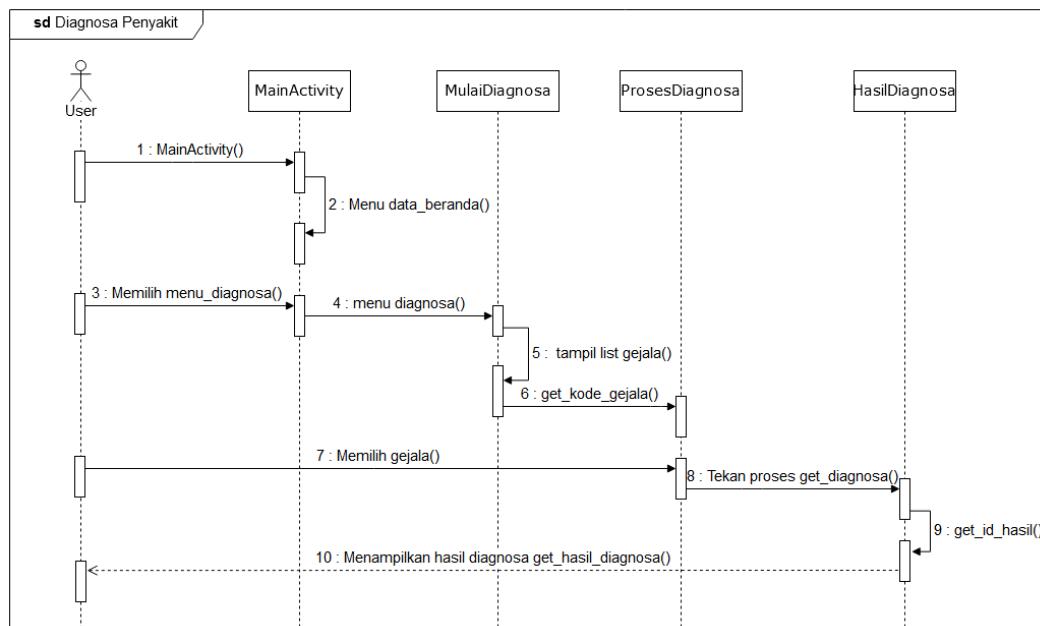
Gambar 3.17 menjelaskan bahwa *class diagram* dari aplikasi diagnosa penyakit ayam memiliki 13 kelas yang terdiri dari kelas beranda, kelas menu, kelas diagnosa, kelas hasil, kelas keterangan, kelas login , kelas database adapter, kelas admin, kelas penyakit, kelas gejala, kelas pengetahuan, kelas ubah\_password dan kelas tentang.

### 3.6.4 Sequence Diagram

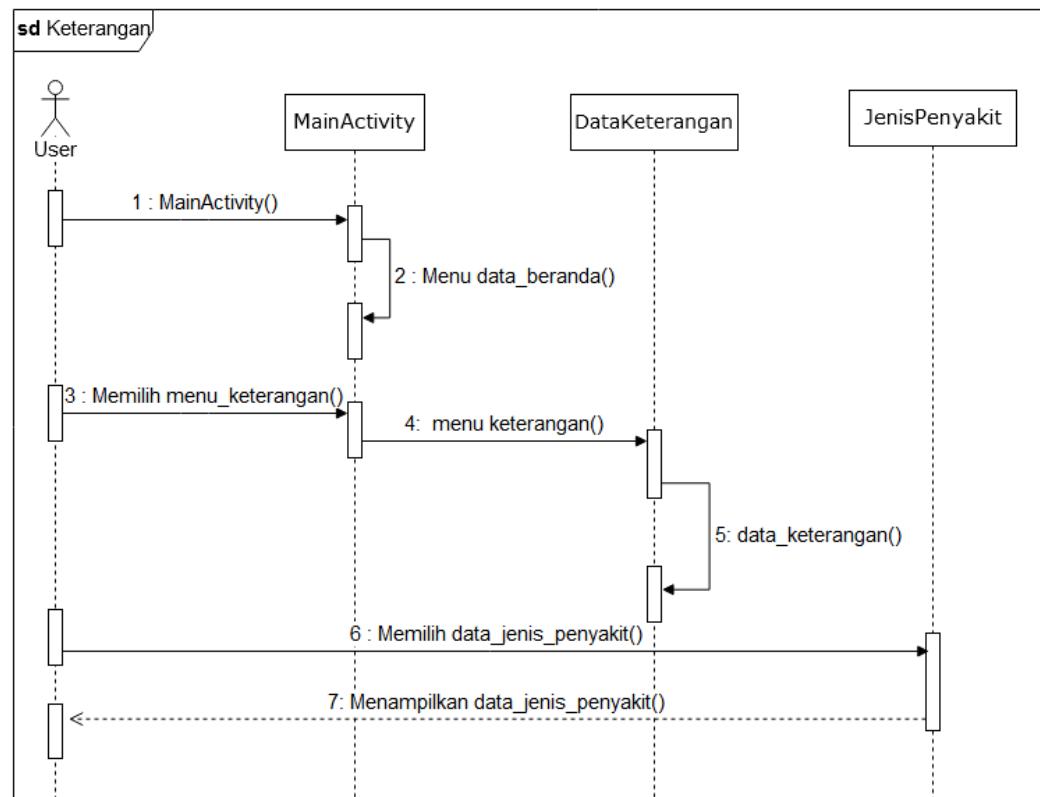
Diagram ini berguna untuk melihat spesifikasi dari sebuah pilihan *button* sehingga *user* dapat memilih *button* tersebut dan akan ditampilkan *sub-menu* dari masing-masing *button* tersebut. Pada sistem ini terdapat enam proses *sequence diagram* yang disajikan pada Gambar 3.18 sampai Gambar 3.28.



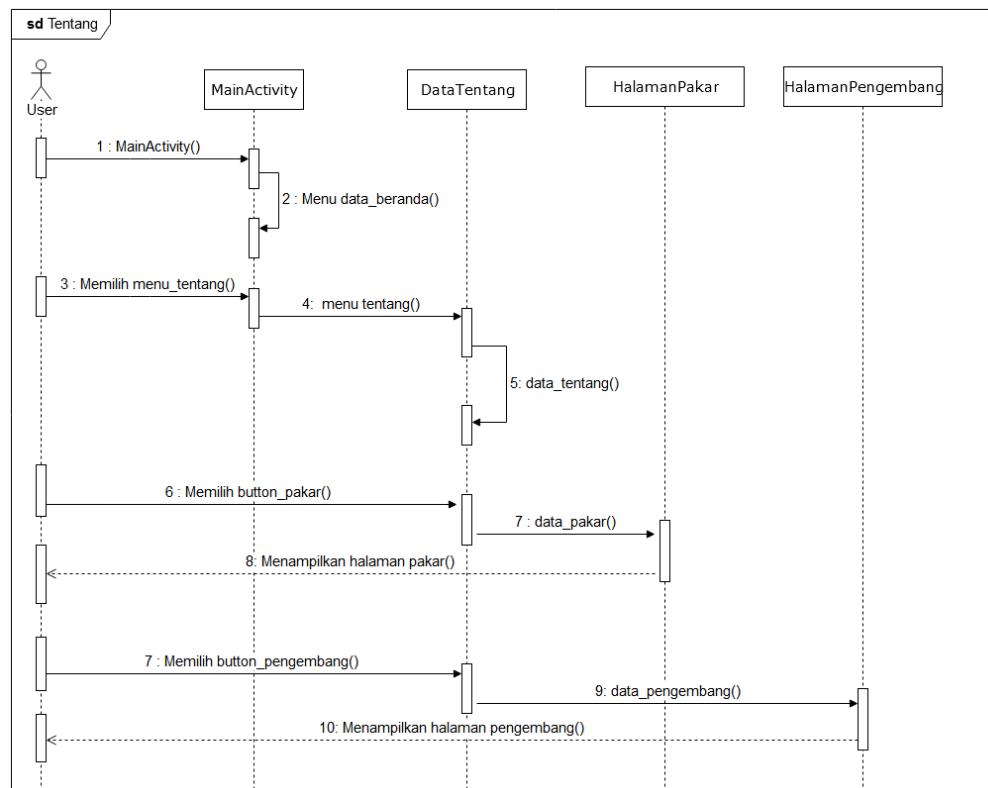
**Gambar 3.18 Sequence Diagram Beranda / Main Activity**



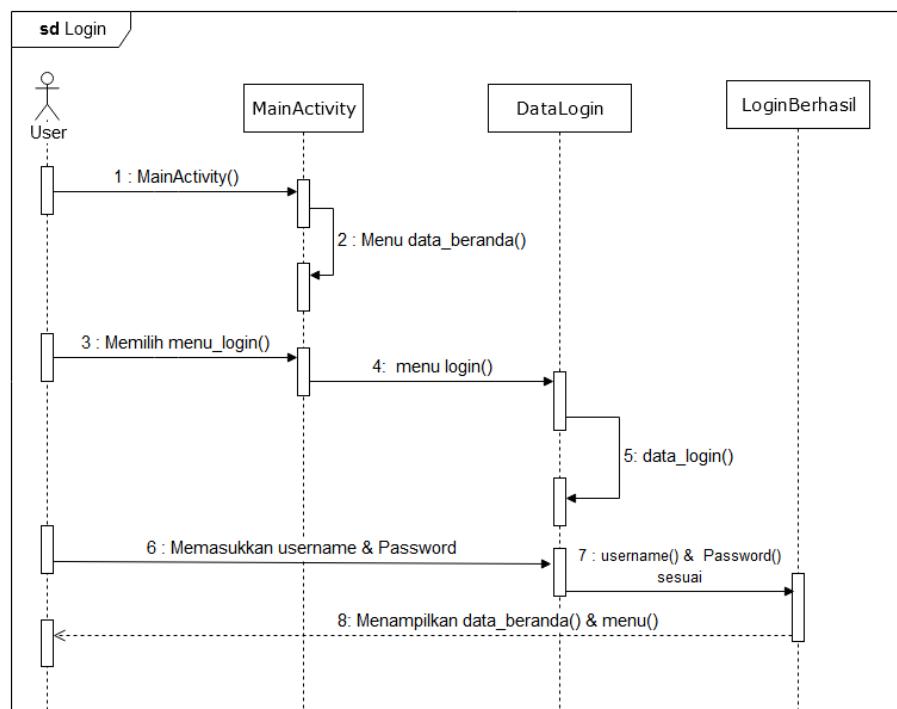
Gambar 3.19 Sequence Diagram Diagnosa Penyakit



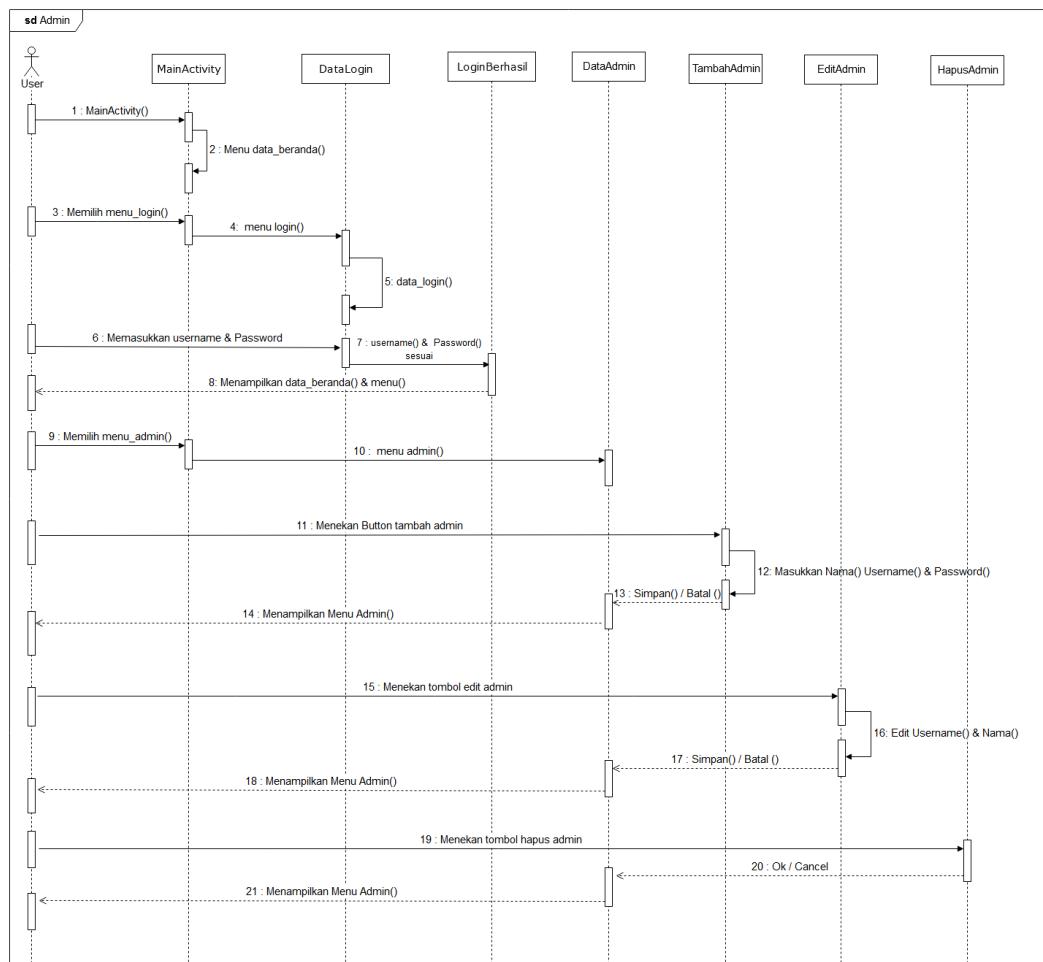
Gambar 3.20 Sequence Diagram Keterangan



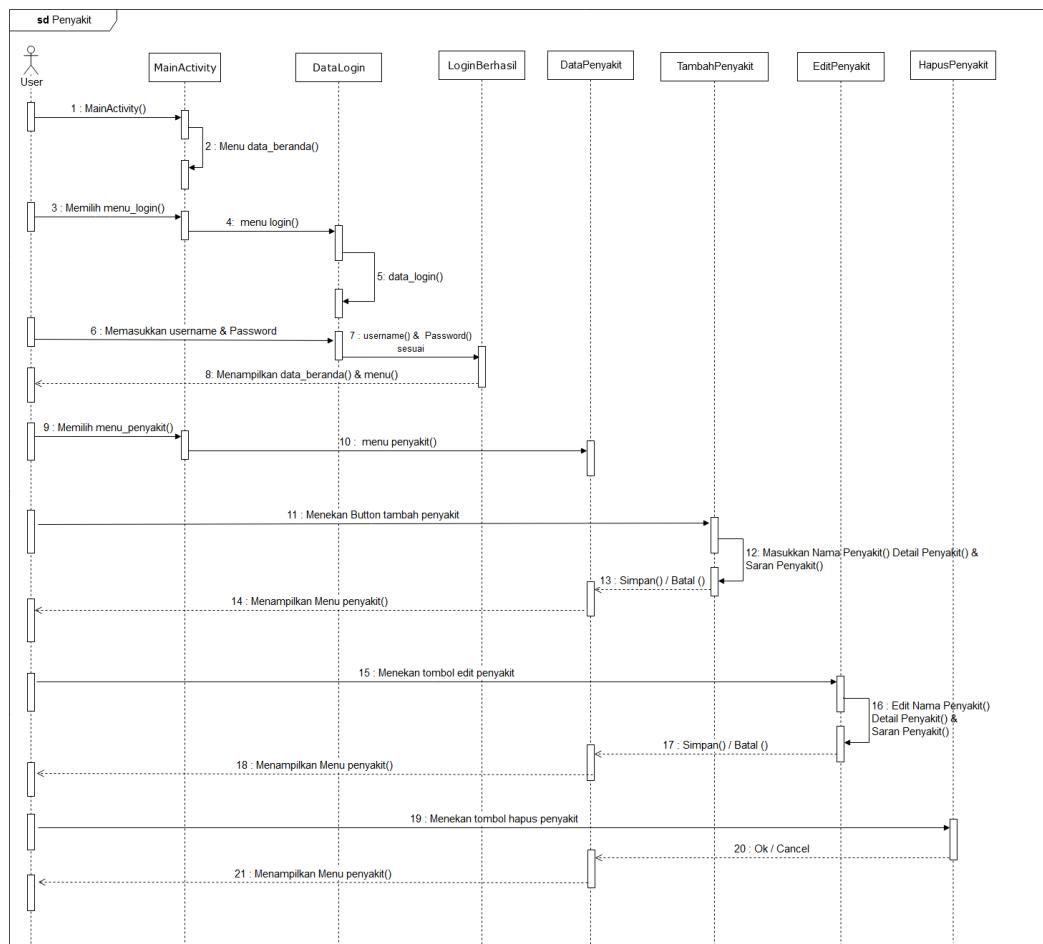
**Gambar 3.21 Sequence Diagram Tentang**



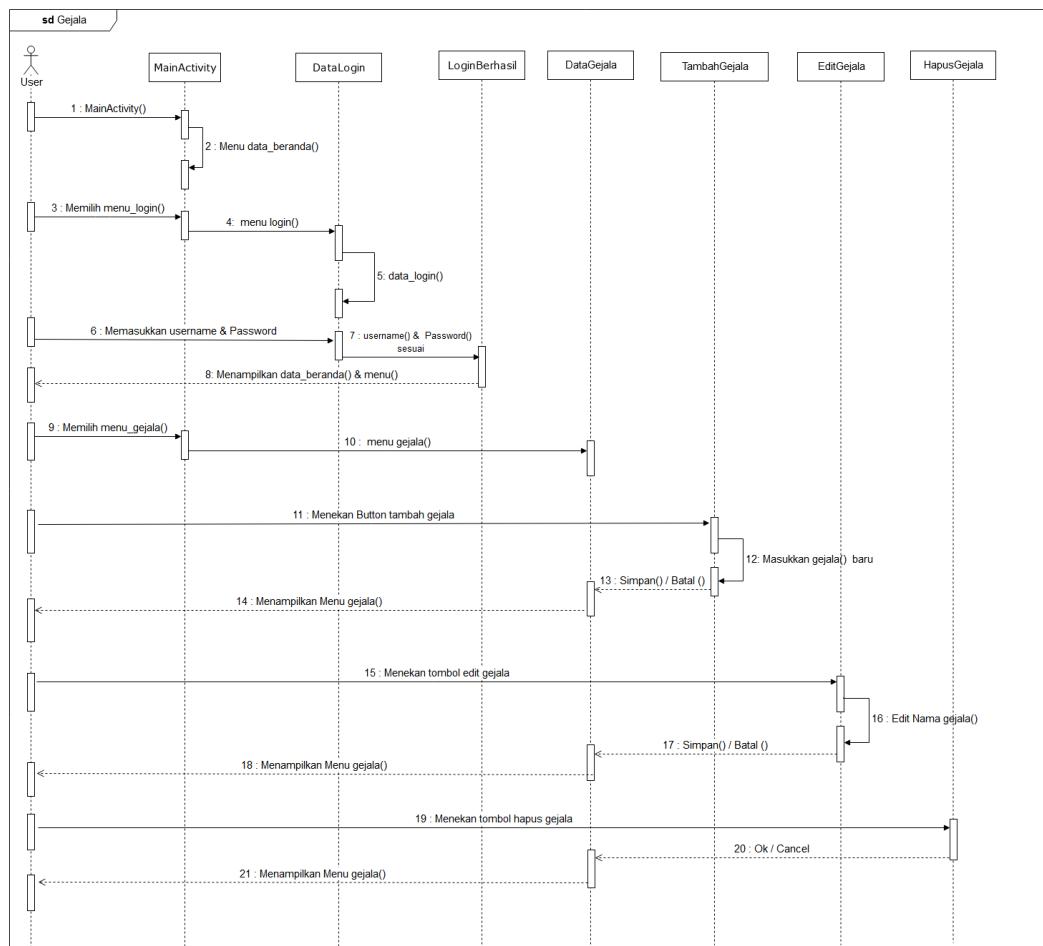
**Gambar 3.22 Sequence Diagram Login**



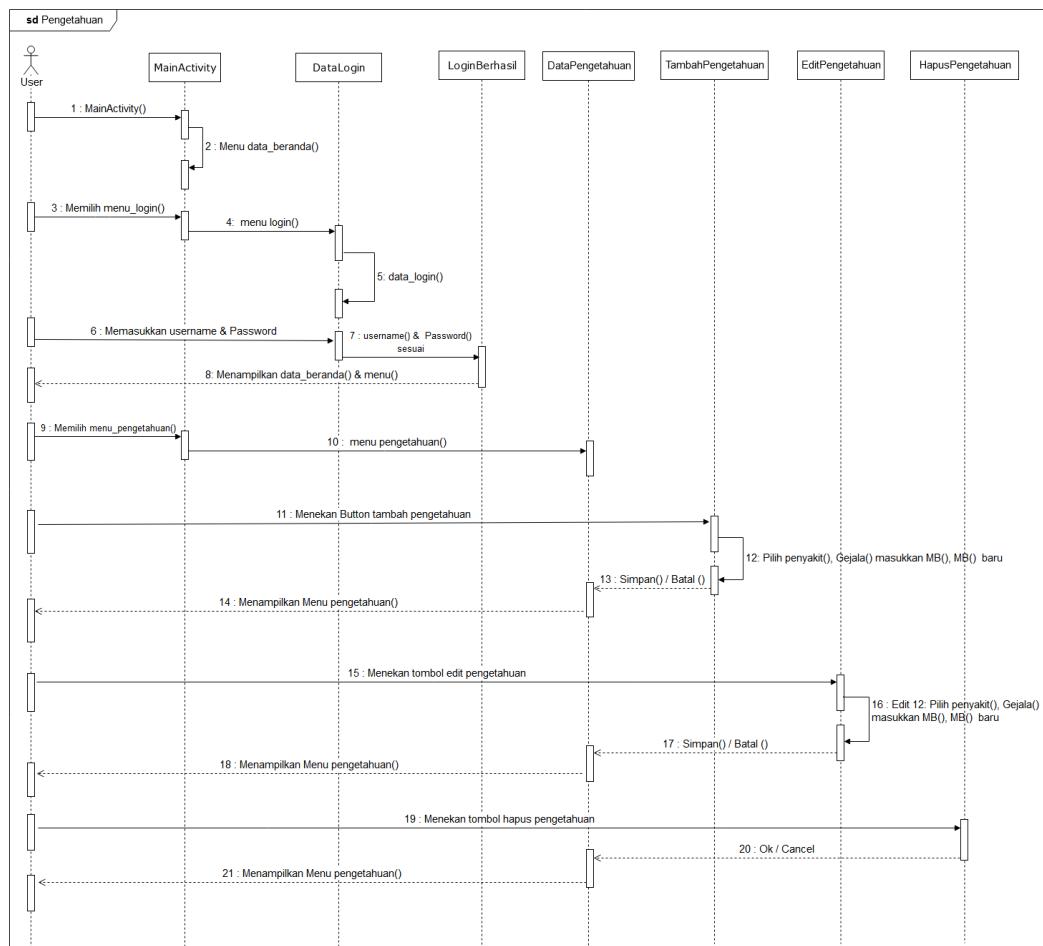
**Gambar 3.23 Sequence Diagram Admin**



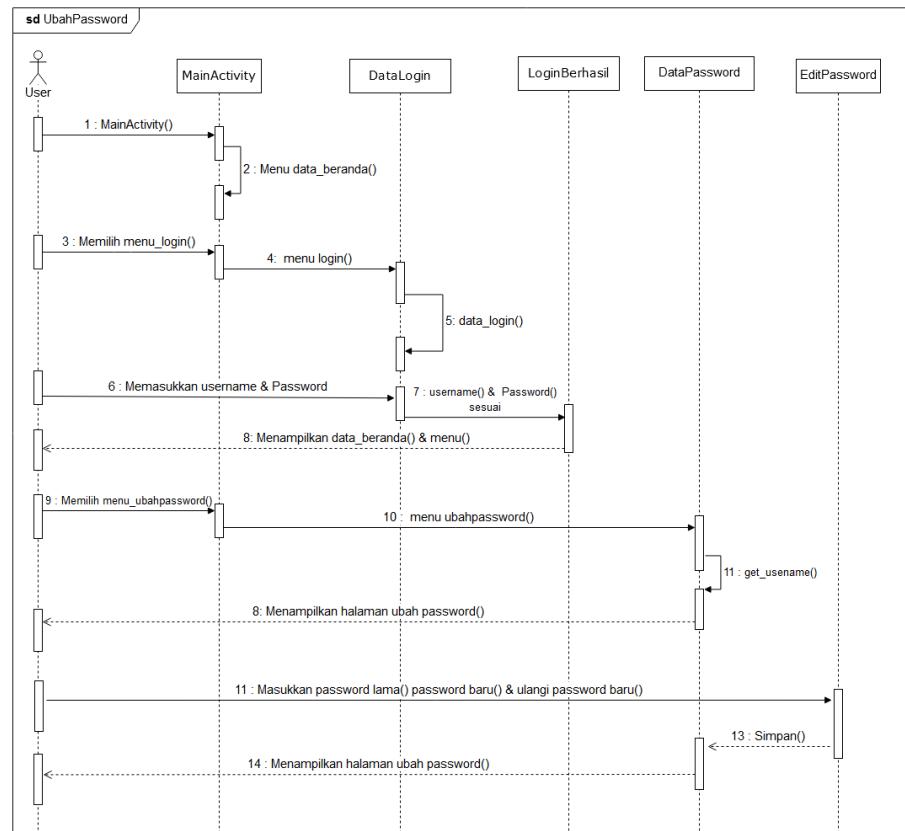
**Gambar 3.24 Sequence Diagram Penyakit**



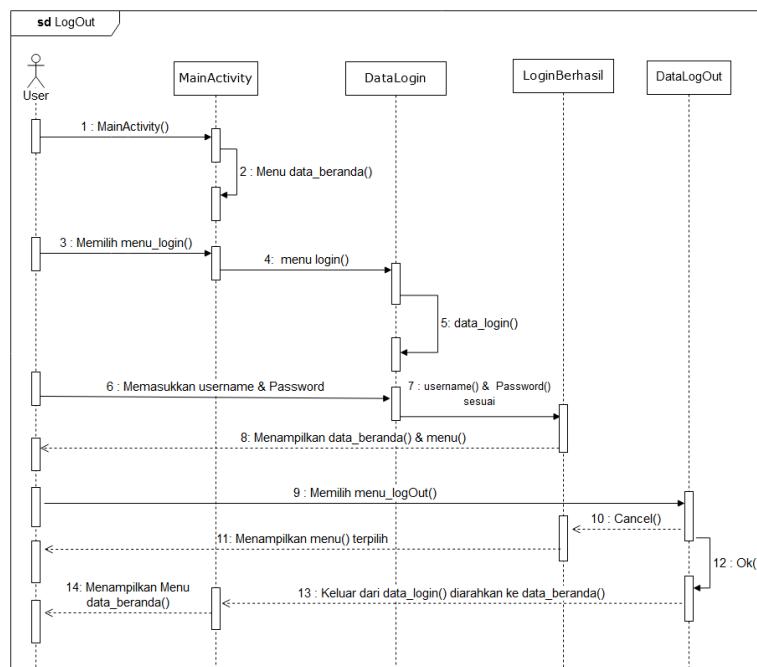
**Gambar 3.25 Sequence Diagram Gejala**



**Gambar 3.26 Sequence Diagram Pengetahuan**



**Gambar 3.27 Sequence Diagram Ubah Password**



**Gambar 3.28 Sequence Diagram LogOut**

### 3.7 Perancangan Basis Data

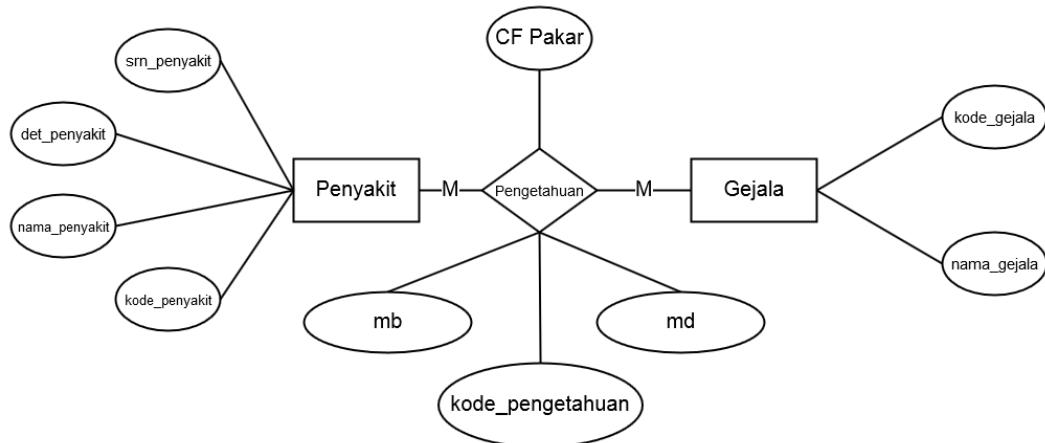
Sebelum melakukan perancangan basis data, dilakukan pencarian entitas dan atribut entitas yang berhubungan. Entitas-entitas yang ditemukan adalah:

1. DataGejala. Entitas ini berupa gejala penyakit yang menyerang ayam. Atribut entitas ini adalah: kode\_gejala dan nama\_gejala.
2. DataPenyakit. Entitas ini detail penyakit dan cara pencegahan dan tindakan penyakit. Atribut entitas ini adalah: kode\_penyakit, nama\_penyakit, det\_penyakit, srn\_penyakit, gambar.
3. DataBase\_Pengetahuan. Entitas ini berupa nilai yang berisi CF pakar. Atribut entitas ini adalah: kode\_pengetahuan, kode\_penyakit, kode\_gejala, md, mb.
4. DataHasil. Entitas ini berupa hasil diagnosa penyakit. Atribut entitas ini adalah: Id\_hasil, tanggal, penyakit, gejala, hasil\_id, hasil\_nilai.
5. DataAdmin. Entitas ini berupa data pengguna administator dimana sebagai pakar itu sendiri. Atribut entitas ini adalah: username, password, nama\_lengkap.
6. DataKondisi. Entitas ini berupa masukan CF *user* pada proses diagnosa. Atribut entitas ini adalah: id, kondisi, ket.
7. DataPost. Entitas ini untuk menambah, mengedit dan menghapus data pada menu keterangan, fungsinya hampir sama pada posting blog. Atribut entitas ini adalah: kode\_post, nama\_post, det\_post, srn\_post, gambar.

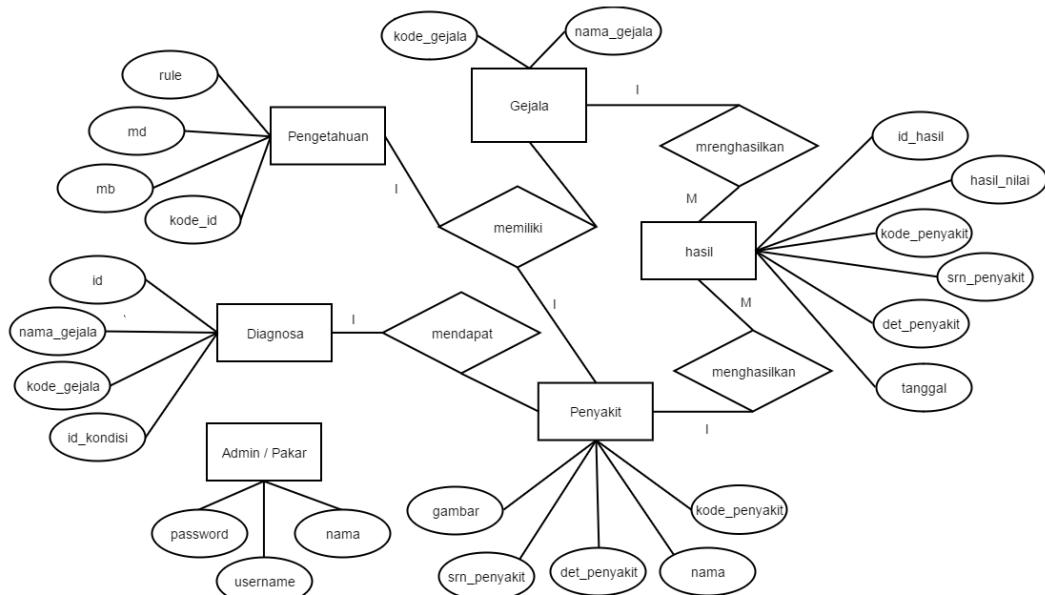
Dari entitas di atas dapat di jelaskan bahwa tabel data gejala berupa kode gejala dan nama gejala di gunakan untuk basis pengetahuan dan pemilihan diagnosa, sedangkan tabel data penyakit berupa kode penyakit, nama penyakit, detail dan saran penyakit, kode dan nama penyakit di gunakan pada tabel basis pengetahuan dan ditampilkan bersama saran dan detail penyakit pada halaman hasil diagnosa, untuk basis pengetahuan sendiri berupa kode pengetahuan untuk pencocokan diagnosa dari data yang diambil dari kode gejala dan penyakit dengan di isi mb dan mb di setiap pengetahuan dan menghasilkan CF *rule*, dan di setiap hasil diagnosa akan di rekam hasilnya oleh tabel hasil dan di bandingkan hasilnya per penyakit yang ada dan mana yang memiliki nilai CF terbesarlah yang tampil di hasil diagnosa berdasarkan gejala yang dipilih.

Untuk tabel admin sendiri berisi hak akses pakar untuk masuk *dashboard* pakar dan melakukan CRUD terhadap data gejala, penyakit dan basis pengetahuan. Dan berikutnya adalah rancangan relasi dari tabel yang telah di buat, dimana tabel penyakit dan gejala saling terhubung ke basis pengetahuan untuk menghasilkan sebuah CF *rule*, dan di gunakan untuk pembanding, penilaian dan aturan dengan CF *user* saat proses diagnosa.

Gambar 3.29 merupakan gambar rancangan relasi antar tabel dalam pembuatan aplikasi ini, dan Gambar 3.30 merupakan gambar Entity Relationship Diagram (ERD), yang merupakan tahap pembuatan diagram relasi antara entitas yang satu dengan yang lain. Menerangkan hubungan antara entitas yang satu dengan lainnya.

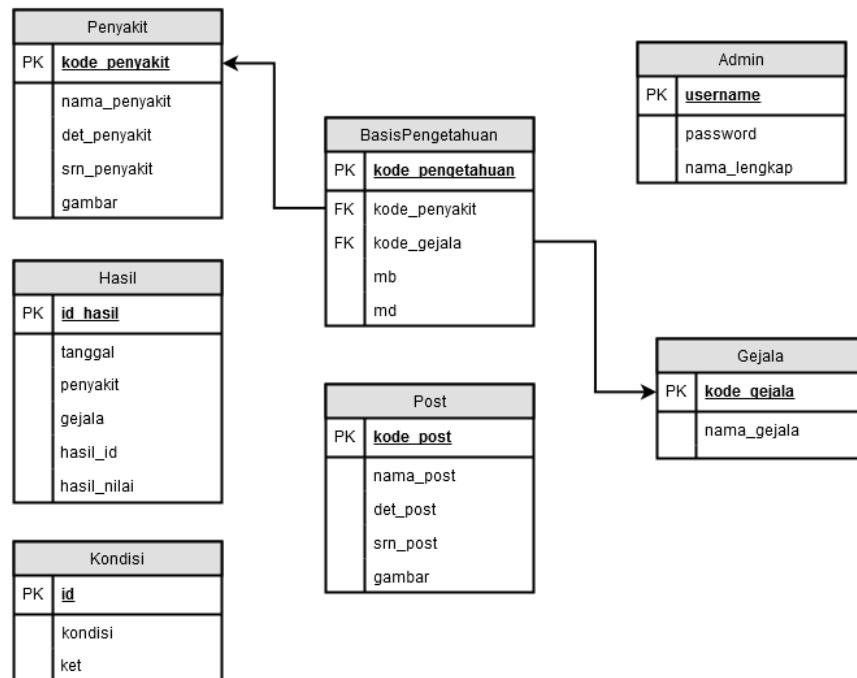


Gambar 3.29 Rancangan Relasi Antar Tabel



Gambar 3.30 Entity Relationship Diagram (ERD)

ERD yang ada di implementasikan dalam tabel beserta relasinya yang dapat dilihat pada Gambar 3.31.



**Gambar 3.31 Relasi Antartabel**

Adapun struktur data dari table-table tersebut, berikut:

1. Table Gejala

Nama Field	Tipe & Nilai Data	Keterangan
kode_gejala	Integer (11)	PrimaryKey, Not Null, Auto increment
nama_gejala	Varchar (50)	Not Null

2. Table Penyakit

Nama Field	Tipe & Nilai Data	Keterangan
kode_penyakit	Integer (11)	PrimaryKey, Not Null, Auto increment
nama_penyakit	Varchar (50)	Not Null
det_penyakit	Varchar (500)	Not Null
srn_penyakit	Varchar (500)	Not Null
gambar	Varchar (500)	Not Null

3. Table Basis\_Pengetahuan

<b>Nama Field</b>	<b>Tipe &amp; Nilai Data</b>	<b>Keterangan</b>
kode_pengetahuan	Integer (11)	PrimaryKey, Not Null, Auto increment
kode penyakit	Integer (11)	ForeignKey dari Table DataPenyakit
kode_gejala	Integer (11)	ForeignKey dari Table DataGejala
md	Double (11,1)	Not Null
mb	Double (11,1)	Not Null

4. Table Hasil

<b>Nama Field</b>	<b>Tipe &amp; Nilai Data</b>	<b>Keterangan</b>
Id_hasil	Integer (11)	PrimaryKey, Not Null, Auto increment
tanggal	Varchar (50)	Not Null, Default 0
penyakit	Text	Not Null
gejala	Text	Not Null
hasil_id	Integer (11)	Not Null
hasil_nilai	Varchar (16)	Not Null

5. Table Admin

<b>Nama Field</b>	<b>Tipe &amp; Nilai Data</b>	<b>Keterangan</b>
username	Varchar (20)	PrimaryKey, Not Null
password	Varchar (32)	Not Null
nama_lengkap	Varchar (30)	Not Null

## 6. Table Kondisi

Nama Field	Tipe & Nilai Data	Keterangan
id	Integer (11)	PrimaryKey, Not Null, Auto increment
kondisi	Varchar (64)	Not Null
ket	Varchar (256)	Not Null

## 7. Table Post

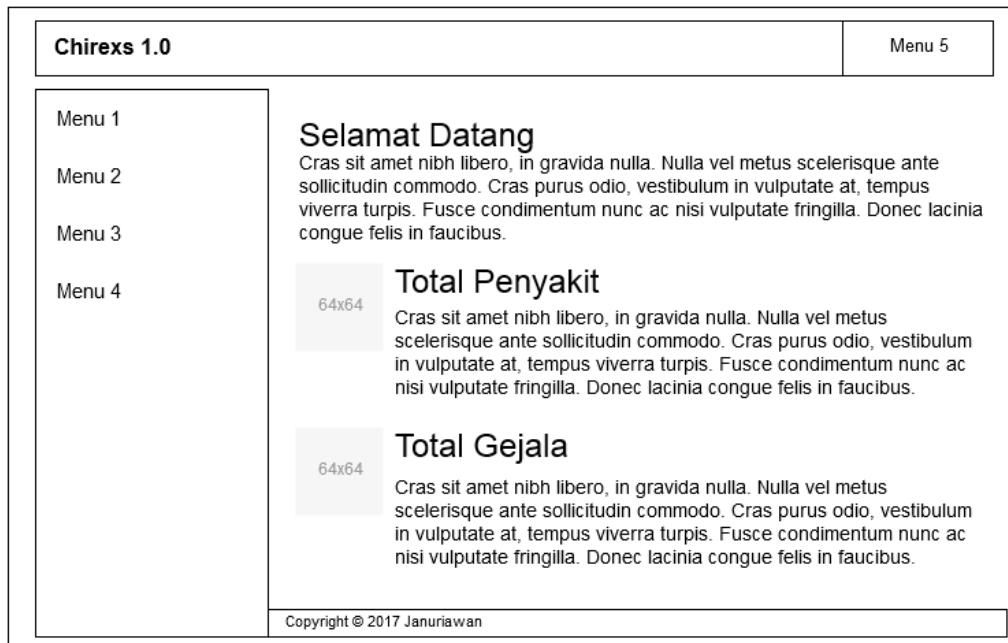
Nama Field	Tipe & Nilai Data	Keterangan
kode_post	Integer (11)	PrimaryKey, Not Null, Auto increment
nama_post	Varchar (50)	Not Null
det_post	Varchar (15000)	Not Null
srn_post	Varchar (15000)	Not Null
gambar	Varchar (500)	Not Null

## 3.8 Perancangan *User Interface*

Perancangan *interface* atau antar muka program berperan untuk menghubungkan antara pengguna dengan aplikasi. *Interface* yang baik akan membuat pengguna program menjadi lebih mudah digunakan serta tidak membingungkan, untuk itu peranan *interface* sangat penting. Berikut adalah *interface* yang ada pada aplikasi “Sistem Pakar Diagnosa Penyakit Ayam Dengan Menerapkan Certainty Factor” akan di jelaskan dari Gambar 3.32 sampai Gambar 3.41 .

### 3.8.1 Halaman Beranda

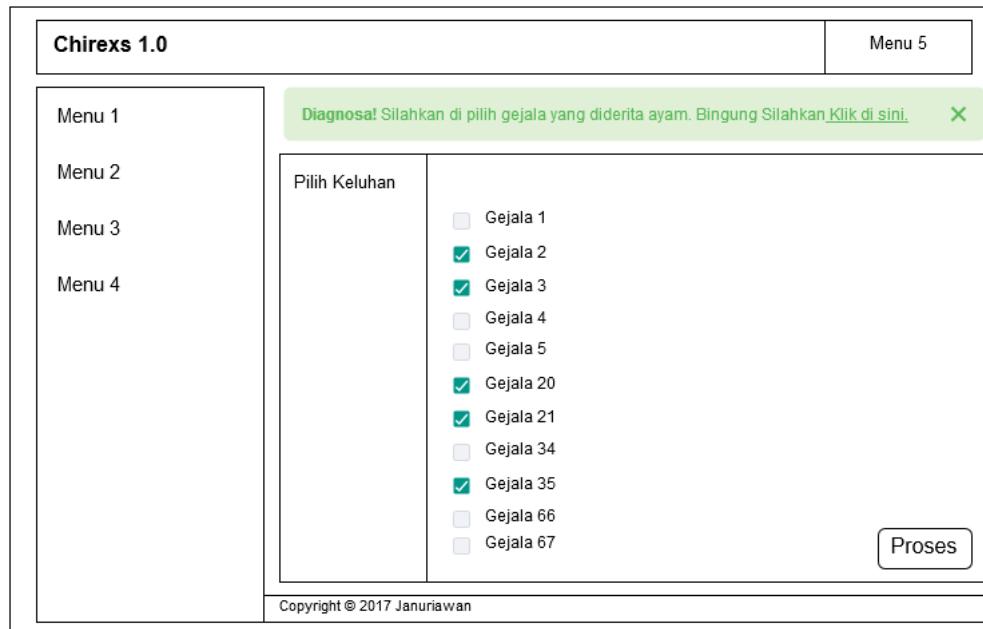
Tampilan pada Gambar 3.32 adalah tampilan utama, tampilan ini akan muncul ketika aplikasi pertama kali dibuka dan berisi informasi singkat deskripsi sistem pakar yang di buat dan di samping menu beranda terdapat menu utama *non login*.



**Gambar 3.32 Menu Beranda**

### 3.8.2 Halaman Diagnosa

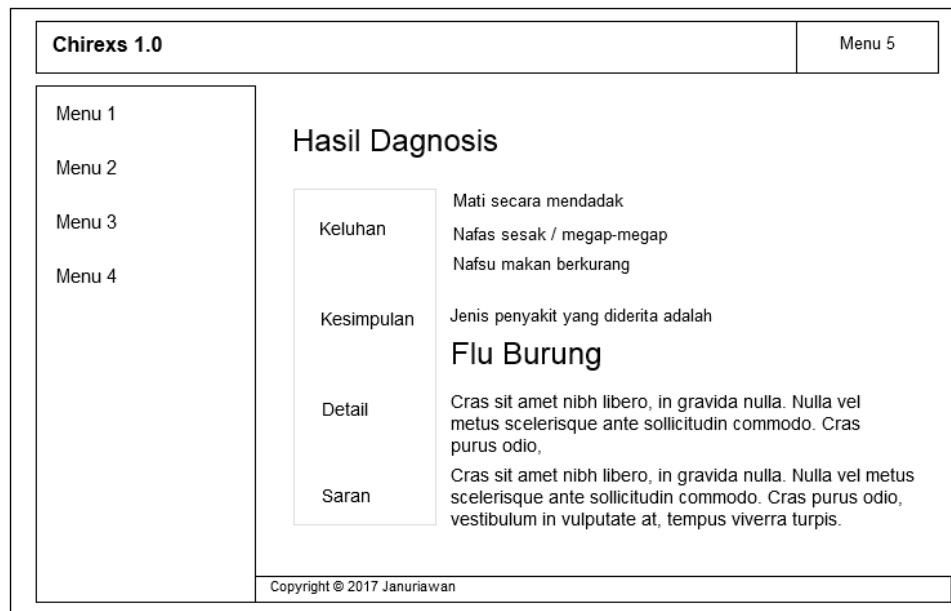
Tampilan pada Gambar 3.33 merupakan proses pemilihan gejala yang diderita ayam. Pada halaman ini akan menampilkan daftar gejala yang telah di inputkan pakar sebelumnya.



**Gambar 3.33 Menu Diagnosa**

### 3.8.3 Halaman Hasil Diagnosis

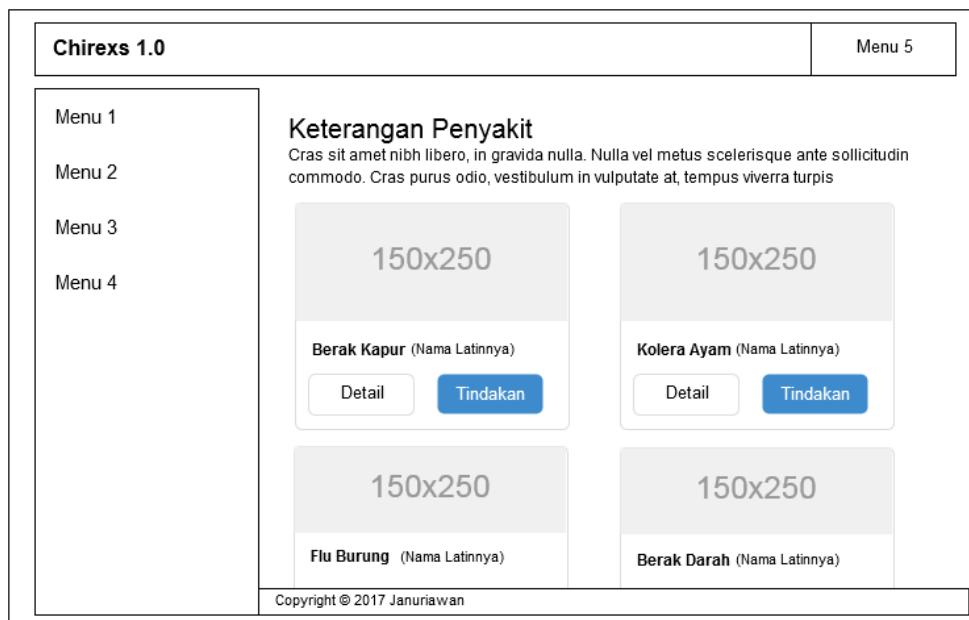
Gambar 3.34 merupakan desain halaman hasil diagnosa yang merupakan sebuah halaman yang memunculkan kemungkinan penyakit yang menyerang ayam berdasarkan hasil perhitungan. Dan menampilkan penyakit ayam yang memiliki persentase kemungkinan dengan nilai tertinggi. Dan saran berserta tindakan singkat ada di bawah hasil diagnosa.



**Gambar 3.34 Menu Hasil Diagnosis**

#### 3.8.4 Halaman Keterangan Penyakit

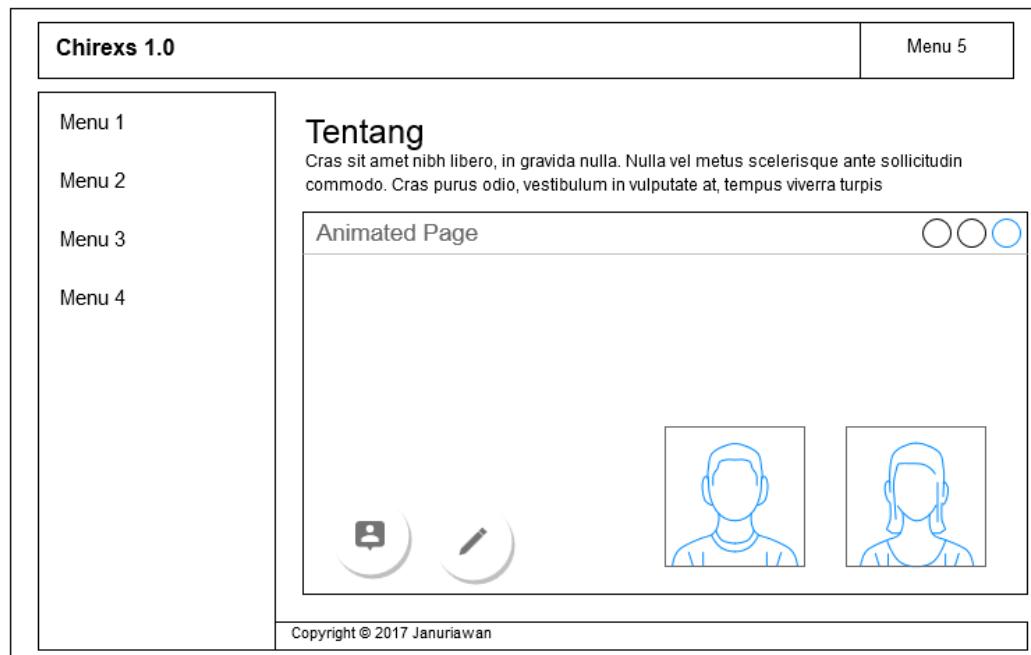
Pada Gambar 3.35 merupakan desain halaman dari menu Keterangan Penyakit berupa daftar informasi lengkap untuk pencegahan dan penanganan.



**Gambar 3.35 Menu Keterangan Penyakit**

### 3.8.5 Halaman Tentang

Halaman ini menyediakan informasi tentang pengembang aplikasi dan pakar yang berkontribusi di halaman ini juga di tampilkan animasi css. Gambar 3.36 merupakan desain dari halaman tentang.



**Gambar 3.36 Menu Tentang**

### 3.8.6 Halaman Login

Pada Gambar 3.37 merupakan desain halaman login. Dimana pakar atau admin yang memiliki hak akses bisa masuk ke menu gejala, penyakit dan pengetahuan. Hak akses sendiri di buatkan admin terlebih dahulu di database handler.



**Gambar 3.37 Menu Login**

### 3.8.7 Halaman Pakar

Pada Gambar 3.38 sampai 3.41 merupakan isi dari menu pakar yaitu halaman admin, penyakit, gejala, pengetahuan, dan password. Di halaman admin, penyakit, gejala, pengetahuan pakar bisa melakuan CRUD atau *Create, Read, Update* dan *Delete*.

Chirexs 1.0		Menu 5																	
Menu 1  Menu 2  Menu 3  Menu 4	<b>Tambah Admin</b> <input type="text" value="Ketik dan Cari....."/> <input type="button" value="Cari"/>																		
	<table border="1"> <thead> <tr> <th>#</th> <th>Username</th> <th>Nama Lengkap</th> <th>Aksi</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>John</td> <td>Boo</td> <td><input type="button" value="Ubah"/> <input type="button" value="Hapus"/></td> </tr> <tr> <td>2</td> <td>Mary</td> <td>Brown</td> <td><input type="button" value="Ubah"/> <input type="button" value="Hapus"/></td> </tr> <tr> <td>3</td> <td>James</td> <td>Mooray</td> <td><input type="button" value="Ubah"/> <input type="button" value="Hapus"/></td> </tr> </tbody> </table>			#	Username	Nama Lengkap	Aksi	1	John	Boo	<input type="button" value="Ubah"/> <input type="button" value="Hapus"/>	2	Mary	Brown	<input type="button" value="Ubah"/> <input type="button" value="Hapus"/>	3	James	Mooray	<input type="button" value="Ubah"/> <input type="button" value="Hapus"/>
	#	Username	Nama Lengkap	Aksi															
	1	John	Boo	<input type="button" value="Ubah"/> <input type="button" value="Hapus"/>															
	2	Mary	Brown	<input type="button" value="Ubah"/> <input type="button" value="Hapus"/>															
3	James	Mooray	<input type="button" value="Ubah"/> <input type="button" value="Hapus"/>																
1 <input type="button" value="2"/> <input type="button" value="3"/> <input type="button" value="4"/> <input type="button" value="Next"/>																			
Copyright © 2017 Januriawan																			

**Gambar 3.38 Menu Admin**

Chirexs 1.0		Menu 5																										
Menu 1  Menu 2  Menu 3  Menu 4	<b>Tambah Penyakit</b> <input type="text" value="Ketik dan Cari....."/> <input type="button" value="Cari"/>																											
	<table border="1"> <thead> <tr> <th>No</th> <th>Nama Penyakit</th> <th>Detail Penyakit</th> <th>Saran penyakit</th> <th>Aksi</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Berak Kapur</td> <td>Hello! At vero eos et accusamus et iusto odio dignissimos ducimus qui blanditiis praesentium deleniti atque corrupti quos doloreset quas molestias excepturi sint occaecati cupiditate non provident, similiq</td> <td>Hello! At vero eos et accusamus et iusto odio dignissimos ducimus qui blanditiis praesentium deleniti atque corrupti quos doloreset quas molestias excepturi sint occaecati cupiditate non provident, similiq</td> <td><input type="button" value="Ubah"/> <input type="button" value="Hapus"/></td> </tr> <tr> <td>2</td> <td>Kolera Ayam</td> <td>Hello, deleniti atque corrupti quos dolores et quas molestias excepturi sint occaecati cupiditate non provident, similiq sunt in culpa</td> <td>Hello, deleniti atque corrupti quos dolores et quas molestias excepturi sint occaecati cupiditate non provident, similiq sunt in culpa</td> <td><input type="button" value="Ubah"/> <input type="button" value="Hapus"/></td> </tr> <tr> <td>3</td> <td>Berak Darah</td> <td>Hello! Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab il</td> <td>Hello! Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab il</td> <td><input type="button" value="Ubah"/> <input type="button" value="Hapus"/></td> </tr> <tr> <td colspan="3">           1 <input type="button" value="2"/> <input type="button" value="3"/> <input type="button" value="4"/> <input type="button" value="Next"/> </td> <td colspan="2">Copyright © 2017 Januriawan</td> </tr> </tbody> </table>			No	Nama Penyakit	Detail Penyakit	Saran penyakit	Aksi	1	Berak Kapur	Hello! At vero eos et accusamus et iusto odio dignissimos ducimus qui blanditiis praesentium deleniti atque corrupti quos doloreset quas molestias excepturi sint occaecati cupiditate non provident, similiq	Hello! At vero eos et accusamus et iusto odio dignissimos ducimus qui blanditiis praesentium deleniti atque corrupti quos doloreset quas molestias excepturi sint occaecati cupiditate non provident, similiq	<input type="button" value="Ubah"/> <input type="button" value="Hapus"/>	2	Kolera Ayam	Hello, deleniti atque corrupti quos dolores et quas molestias excepturi sint occaecati cupiditate non provident, similiq sunt in culpa	Hello, deleniti atque corrupti quos dolores et quas molestias excepturi sint occaecati cupiditate non provident, similiq sunt in culpa	<input type="button" value="Ubah"/> <input type="button" value="Hapus"/>	3	Berak Darah	Hello! Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab il	Hello! Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab il	<input type="button" value="Ubah"/> <input type="button" value="Hapus"/>	1 <input type="button" value="2"/> <input type="button" value="3"/> <input type="button" value="4"/> <input type="button" value="Next"/>			Copyright © 2017 Januriawan	
	No	Nama Penyakit	Detail Penyakit	Saran penyakit	Aksi																							
	1	Berak Kapur	Hello! At vero eos et accusamus et iusto odio dignissimos ducimus qui blanditiis praesentium deleniti atque corrupti quos doloreset quas molestias excepturi sint occaecati cupiditate non provident, similiq	Hello! At vero eos et accusamus et iusto odio dignissimos ducimus qui blanditiis praesentium deleniti atque corrupti quos doloreset quas molestias excepturi sint occaecati cupiditate non provident, similiq	<input type="button" value="Ubah"/> <input type="button" value="Hapus"/>																							
	2	Kolera Ayam	Hello, deleniti atque corrupti quos dolores et quas molestias excepturi sint occaecati cupiditate non provident, similiq sunt in culpa	Hello, deleniti atque corrupti quos dolores et quas molestias excepturi sint occaecati cupiditate non provident, similiq sunt in culpa	<input type="button" value="Ubah"/> <input type="button" value="Hapus"/>																							
3	Berak Darah	Hello! Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab il	Hello! Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab il	<input type="button" value="Ubah"/> <input type="button" value="Hapus"/>																								
1 <input type="button" value="2"/> <input type="button" value="3"/> <input type="button" value="4"/> <input type="button" value="Next"/>			Copyright © 2017 Januriawan																									

**Gambar 3.39 Menu Penyakit**

<b>Chirexs 1.0</b>		Menu 5												
Menu 1  Menu 2  Menu 3  Menu 4	<div style="display: flex; justify-content: space-between; align-items: center;"> <span><b>Tambah gejala</b></span> <input type="text" value="Ketik dan Cari....."/> <span><b>Cari</b></span> </div> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>#</th> <th>Nama Gejala</th> <th>Aksi</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Nafsu makan berkurang</td> <td style="text-align: right;"><b>Ubah</b> <b>Hapus</b></td> </tr> <tr> <td>2</td> <td>Bulu kusam dan berkerut</td> <td style="text-align: right;"><b>Ubah</b> <b>Hapus</b></td> </tr> <tr> <td>3</td> <td>Mencret kehijau-hijauan</td> <td style="text-align: right;"><b>Ubah</b> <b>Hapus</b></td> </tr> </tbody> </table> <div style="text-align: center; margin-top: 10px;"> <a href="#">1</a> <a href="#">2</a> <a href="#">3</a> <a href="#">4</a> <a href="#">Next</a> </div>	#	Nama Gejala	Aksi	1	Nafsu makan berkurang	<b>Ubah</b> <b>Hapus</b>	2	Bulu kusam dan berkerut	<b>Ubah</b> <b>Hapus</b>	3	Mencret kehijau-hijauan	<b>Ubah</b> <b>Hapus</b>	Copyright © 2017 Januriawan
#	Nama Gejala	Aksi												
1	Nafsu makan berkurang	<b>Ubah</b> <b>Hapus</b>												
2	Bulu kusam dan berkerut	<b>Ubah</b> <b>Hapus</b>												
3	Mencret kehijau-hijauan	<b>Ubah</b> <b>Hapus</b>												

**Gambar 3.40 Menu Gejala**

<b>Chirexs 1.0</b>		Menu 5																								
Menu 1  Menu 2  Menu 3  Menu 4	<div style="display: flex; justify-content: space-between; align-items: center;"> <span><b>Tambah Pengetahuan</b></span> <input type="text" value="Ketik dan Cari....."/> <span><b>Cari</b></span> </div> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>#</th> <th>Penyakit</th> <th>Gejala</th> <th>MB</th> <th>MD</th> <th>Aksi</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Berak Kapur</td> <td>Nampak membiru</td> <td>0.8</td> <td>0.2</td> <td style="text-align: right;"><b>Ubah</b> <b>Hapus</b></td> </tr> <tr> <td>2</td> <td>Flu Burung</td> <td>Tampak lesu</td> <td>0.6</td> <td>0.4</td> <td style="text-align: right;"><b>Ubah</b> <b>Hapus</b></td> </tr> <tr> <td>3</td> <td>Tetelo</td> <td>Kepala terputar</td> <td>1.0</td> <td>0.0</td> <td style="text-align: right;"><b>Ubah</b> <b>Hapus</b></td> </tr> </tbody> </table> <div style="text-align: center; margin-top: 10px;"> <a href="#">1</a> <a href="#">2</a> <a href="#">3</a> <a href="#">4</a> <a href="#">Next</a> </div>	#	Penyakit	Gejala	MB	MD	Aksi	1	Berak Kapur	Nampak membiru	0.8	0.2	<b>Ubah</b> <b>Hapus</b>	2	Flu Burung	Tampak lesu	0.6	0.4	<b>Ubah</b> <b>Hapus</b>	3	Tetelo	Kepala terputar	1.0	0.0	<b>Ubah</b> <b>Hapus</b>	Copyright © 2017 Januriawan
#	Penyakit	Gejala	MB	MD	Aksi																					
1	Berak Kapur	Nampak membiru	0.8	0.2	<b>Ubah</b> <b>Hapus</b>																					
2	Flu Burung	Tampak lesu	0.6	0.4	<b>Ubah</b> <b>Hapus</b>																					
3	Tetelo	Kepala terputar	1.0	0.0	<b>Ubah</b> <b>Hapus</b>																					

**Gambar 3.41 Menu Basis Pengetahuan**

## **BAB IV**

### **HASIL DAN PEMBAHASAN**

Bab ini menunjukkan hasil penelitian dan pembahasan tentang implementasi sistem pakar diagnosa ayam dengan metode Certainty Factor berbasis website. Pembahasan Implementasi dan pengujian sistem meliputi perangkat lunak dan perangkat keras.

#### **4.1      Implementasi**

Implementasi adalah suatu proses penerapan rancangan program yang telah dibuat ke dalam sebuah aplikasi pemrograman sesuai dengan tujuan yang diharapkan dari program aplikasi tersebut. Kegiatan implementasi dilakukan dengan dasar yang telah direncanakan dalam tahap perancangan.

##### **4.1.1    Implementasi Perangkat Lunak**

Adapun beberapa perangkat lunak yang harus disiapkan dalam pembangunan sistem ini antara lain :

1. Apache digunakan untuk mengolah kode PHP yang akan dikirim ke client.
2. Database Server yang digunakan adalah menggunakan MySQL untuk menyimpan data.
3. Pemrograman server yang digunakan menggunakan bahasa pemrograman server PHP.
4. Mozilla Firefox digunakan sebagai browser untuk menampilkan website.

#### 4.1.2 Implementasi Antar Muka

Implementasi antar muka merupakan tahapan dalam memenuhi kebutuhan pengguna dalam berinteraksi dengan sistem yang dibuat. Fasilitas antar muka yang baik akan sangat membantu pemakai dalam memahami proses yang sedang dilakukan oleh sistem sehingga dapat meningkatkan kinerja sistem.

##### 1. Halaman Beranda Sistem Pakar

Halaman ini sebagai halaman awal atau halaman utama berupa tampilan utama sistem pakar, terdapat slider gambar, statistik sistem pakar berupa total penyakit, gejala, pengetahuan dan admin, tak hanya itu terdapat juga deskripsi singkat tentang sistem pakar, untuk detail antar muka Halaman Beranda Sistem Pakar dapat dilihat pada Gambar 4.1.



**Gambar 4.1 Halaman Beranda Sistem Pakar**

##### 2. Halaman Diagnosa Sistem Pakar

Halaman ini berisi pemilihan diagnosa terdapat pilihan gejala dan pengguna harus memilih gejala yang didapati di lapangan dan

kemungkinan yang ada jika sudah maka pengguna menekan tombol proses, untuk detail antar muka Halaman Diagnosa Sistem Pakar dapat di lihat pada Gambar 4.2.

No	Kode	Gejala	Pilih Kondisi
1	G001	Nafsu makan berkurang	<input type="checkbox"/> Pilih jika sesuai
2	G002	Nafas sesak / megap-megap	<input type="checkbox"/> Pilih jika sesuai
3	G003	Nafas nggorok basah	<input type="checkbox"/> Pilih jika sesuai
4	G004	Bersin-bersin	<input type="checkbox"/> Pilih jika sesuai
5	G005	Batuk	<input type="checkbox"/> Pilih jika sesuai
6	G006	Bulu kusam dan berkerut	<input type="checkbox"/> Pilih jika sesuai
7	G007	Diare	<input type="checkbox"/> Pilih jika sesuai

**Gambar 4.2 Halaman Diagnosa Sistem Pakar**

### 3. Halaman Hasil Diagnosis

Halaman ini adalah lanjutan dari halaman diagnosa diamana dari hasil gejala yang telah di pilih akan di tampilkan penyakit yang menyerang ayam dalam bentuk nilai CF dan % (persen), terdapat juga saran dan deskripsi singkat dan nilai kemungkinan penyakit yang lainnya jika terdapat gejala yang sama, untuk detail antar muka 3. Halaman Hasil Diagnosis dapat di lihat pada Gambar 4.3.

The screenshot shows the 'Hasil Diagnosis' (Diagnosis Results) page of the hirexs 1.0 software. At the top, there is a navigation bar with links for 'Bantuan' (Help) and 'Login'. On the left, a sidebar menu includes 'Beranda', 'Diagnosa', 'Riwayat', 'Keterangan', 'Info Harga', and 'Tentang'. The main content area has a title 'Hasil Diagnosis' with a 'Cetak' (Print) button. Below this is a table with columns 'No', 'Kode', 'Gejala yang dialami (keluhan)', and 'Pilihan'. The table lists four symptoms: 'Nafsu makan berkurang' (Likelihood: Hampir pasti ya), 'Nafas sesak / megap-megap' (Likelihood: Kemungkinan besar ya), 'Nafas ngorok basah' (Likelihood: Pasti ya), and 'Bersin-bersin' (Likelihood: Kemungkinan besar tidak). Below the table is a section titled 'Hasil Diagnosa' which states 'Jenis penyakit yang diderita adalah Kolera Ayam (Fowl Cholera) / 0.83 % (0.8320)'. To the right of this text is a photograph of a chicken's head.

**Gambar 4.3 Halaman Hasil Diagnosis**

#### 4. Halaman Riwayat

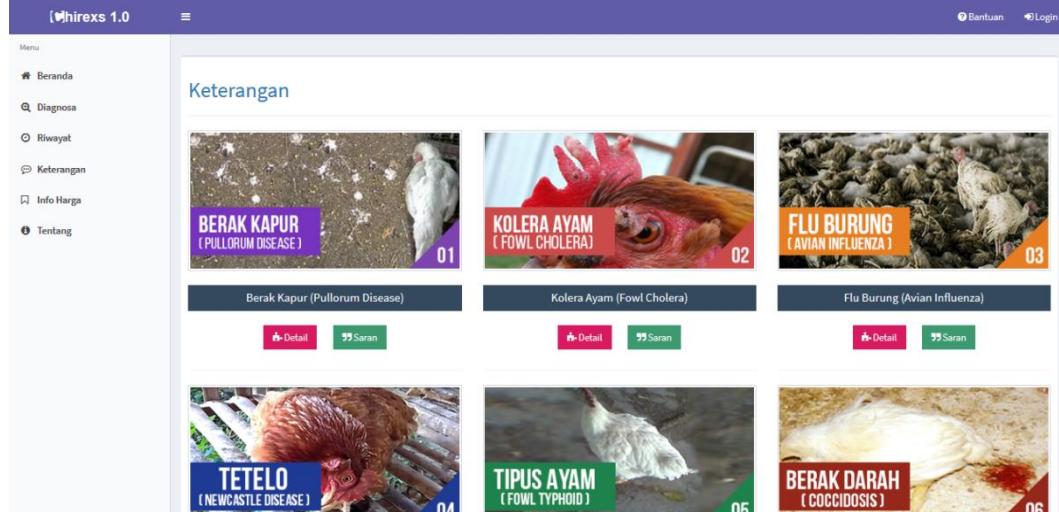
Halaman ini berisi tentang riwayat diagnosa penyakit ayam, terdapat kapan waktu konsultasi dan berapa nilai CF nya, Disampingnya ada grafik yang diambil dari total gejala yang pernah muncul dan nilai % dari total riwayat yang pernah di lakukan, Halaman Riwayat dapat di lihat pada Gambar 4.4.

The screenshot shows the 'Riwayat Konsultasi' (Consultation History) page of the hirexs 1.0 software. At the top, there is a navigation bar with links for 'Bantuan' (Help) and 'Login'. On the left, a sidebar menu includes 'Beranda', 'Diagnosa', 'Riwayat', 'Keterangan', 'Info Harga', and 'Tentang'. The main content area has a title 'Riwayat Konsultasi'. Below this is a table with columns 'No', 'Tanggal', 'Penyakit', 'Nilai CF', and 'Aksi'. The table lists 12 consultations with various diagnoses and their corresponding CF values. To the right of the table is a circular pie chart titled 'Grafik' showing the distribution of diseases. The chart is divided into several segments with percentages: Tipus Ayam (Fowl Typhoid) 18%, Kolera Ayam (Fowl Cholera) 16%, Batuk Ayam Menahun (Infectious Bronchitis) 16%, Busung Ayam (Lymphoid Leukosis) 11%, Produksi Telur (Egg Drop Syndrome 76/EDS 76) 8%, Mareks (Mareks Disease) 8%, Berak Darah (Coccidiosis) 5%, and Flu Burung (Avian Influenza) 5%.

**Gambar 4.4 Halaman Riwayat**

## 5. Halaman Keterangan

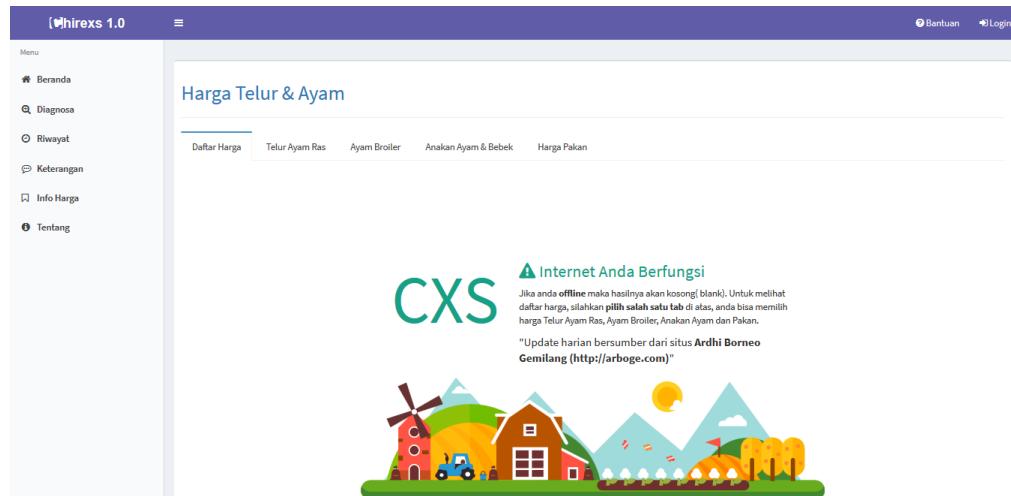
Halaman ini berisi tentang detail lengkap penyakit ayam, dapat berisi sejarah, penanggulangan dan tindakan maupun obat yang di sarankan, untuk detail antar muka Halaman Keterangan dapat di lihat pada Gambar 4.5.



**Gambar 4.5 Halaman Keterangan**

## 6. Halaman Harga

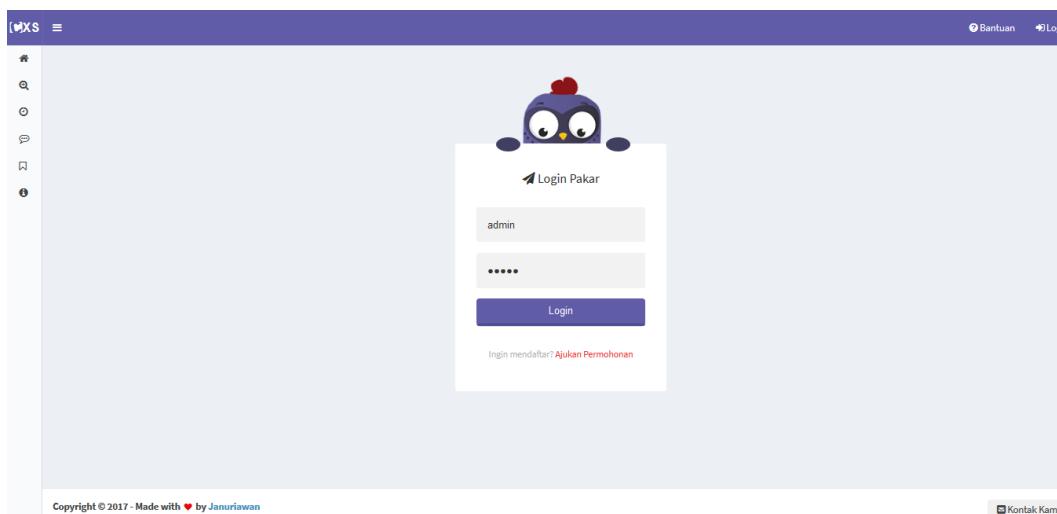
Halaman ini berisi tentang informasi harga harian, terdapat fitur untuk mengetahui harga telur ayam, harga ayam broiler, harga anakan, dan harga pakan, Halaman harga dapat di lihat pada Gambar 4.6.



**Gambar 4.6 Halaman Harga**

## 7. Halaman Login Pakar

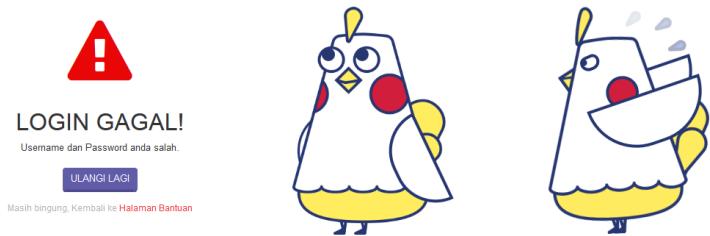
Halaman ini ditujukan kepada pakar untuk masuk ke dalam menu pakar, sebelumnya pakar sudah di daftarkan oleh admin, dan masuk dengan menginputkan *username* dan *password*, untuk detail antar muka Halaman Login Pakar dapat di lihat pada Gambar 4.7.



**Gambar 4.7 Halaman Login Pakar**

## 8. Halaman Gagal Login

Halaman ini ditujukan kepada pakar untuk masuk ke dalam menu pakar, dimana username dan password tidak sesuai dengan yang ada di database maka muncul halaman gagal login, untuk detail antar muka Halaman Gagal Login dapat di lihat pada Gambar 4.8.



**Gambar 4.8 Halaman Gagal Login**

## 9. Halaman Tentang

Halaman ini berisi tentang informasi kontributor terhadap sistem pakar ini, berisi tentang info pengembang aplikasi dan dokter pakar. Tak hanya itu halaman ini berupa animasi css terkait dengan peternakan, untuk detail antar muka Halaman Tentang dapat di lihat pada Gambar 4.9.



**Gambar 4.9 Halaman Tentang**

#### 10. Halaman Admin Pakar

Halaman ini hanya bisa di akses setelah login pakar, berisi tentang informasi admin yang terkait di sistem, tak hanya itu pakar dapat membuat akun admin baru atau mengedit admin yang sudah ada maupun menghapus admin yang telah dibuat, untuk detail antar muka Halaman Admin Pakar dapat di lihat pada Gambar 4.10.

No	Username	Nama Lengkap	Aksi
1	admin	Administrator	<button>Ubah</button> <button>Hapus</button>
2	januriawan	Fajar Januriawan	<button>Ubah</button> <button>Hapus</button>
3	segerawisuda	segerawisuda	<button>Ubah</button> <button>Hapus</button>

**Gambar 4.10 Halaman Admin Pakar**

## 11. Halaman Penyakit

Halaman ini hanya bisa di akses setelah login pakar, berisi tentang informasi penyakit yang terkait di sistem, tak hanya itu pakar dapat membuat penyakit baru atau mengedit penyakit yang sudah ada maupun menghapus penyakit yang telah di buat, nantinya penyakit di gunakan di menu pengetahuan dan hasil diagnosis, untuk detail antar muka Halaman Penyakit dapat di lihat pada Gambar 4.11.

No	Nama Penyakit	Detail Penyakit	Saran Penyakit	Aksi
1	Berak Kapur (Pullorum Disease)	Pullorum Disease disebut juga Bacillary White Diarrhea dan yang lebih popular disebut penyakit berak kapur atau berak putih.	Berikan Master Colprim dosis: 1 gr/1 ltr air selama 3-4 hari (1/2 hari) berturut-turut. setelah itu berikan Master Vit-Stress selama 3-4 hari untuk membantu proses penyembuhan.	<button>Ubah</button> <button>Hapus</button>
2	Kolera Ayam (Fowl Cholera)	Penyakit Fowl Cholera merupakan penyakit ayam yang dapat menyerang secara pelan-pelan dan juga dapat menyerang secara mendadak.	Berikan Master Kolerid dosis: 1 gr/1 ltr air selama 3-4 hari berturut-turut. berikan Master Vit-Stress dosis: 1 gr/3 ltr air untuk membantu proses penyembuhan.	<button>Ubah</button> <button>Hapus</button>
3	Flu Burung (Avian Influenza)	Penyakit Avian Influenza, disebut juga penyakit Fowl Plaque. Pertama kali terjadi di Italia sekitar tahun 1800. Selanjutnya menyebar luas sampai tahun 1930, setelah itu menjadi sporadis dan terlokalisasi terutama di timur tengah.	Tidak ada obat. Dianjurkan untuk disingkirkan dan dimusnakan dengan cara dibakar dan bangkainya dikubur.	<button>Ubah</button> <button>Hapus</button>
4	Tetelo (Newcastle Disease)	Penyakit Newcastle Disease disebut juga Pseudovogel pest Rhaniket, Pneumonecephalitis, Tortor Furrens, dan di Indonesia populer dengan sebutan tetelo. Penyakit ini pertama kali ditemukan oleh Doyle pada tahun 1927, didaerah Newcastle on Tyne, Inggris	Vaksinasi harus dilakukan untuk memperoleh kekebalan. Jenis vaksin yang kami gunakan adalah ND Lasota yang kami beli dari PT. SHS. Vaksinasi ND yang pertama, kami lakukan dengan cara pemberian melalui tetes mata pada hari ke 2. Untuk berikutnya pemberian vaksin kami lakukan dengan cara suntikan di intramuskuuler otot dada.	<button>Ubah</button> <button>Hapus</button>

**Gambar 4.11 Halaman Penyakit**

## 12. Halaman Gejala

Halaman ini hanya bisa di akses setelah login pakar, berisi tentang informasi gejala yang terkait di sistem, tak hanya itu pakar dapat membuat gejala baru atau mengedit gejala yang sudah ada maupun menghapus gejala yang telah di buat, nantinya gejala di gunakan di menu pengetahuan dan diagnosa, untuk detail antar muka Halaman Gejala dapat di lihat pada Gambar 4.12.

No	Nama Gejala	Aksi
1	Nafsu makan berkurang	[Ubah] [Hapus]
2	Nafas sesak / megap-megap	[Ubah] [Hapus]
3	Nafas ngorok basah	[Ubah] [Hapus]
4	Bersin-bersin	[Ubah] [Hapus]
5	Batuk	[Ubah] [Hapus]
6	Bulu kusam dan berkerut	[Ubah] [Hapus]
7	Diare	[Ubah] [Hapus]

**Gambar 4.12 Halaman Gejala**

### 13. Halaman Pengetahuan

Halaman ini hanya bisa di akses setelah login pakar, merupakan halaman untuk mengolah gejala, penyakit dan nilai mb, mb untuk di proses di tabel nilai untuk menghasilkan nilai CF, pakar dapat membuat pengetahuan baru atau mengedit pengetahuan yang sudah ada maupun menghapus pengetahuan yang telah di buat, untuk detail antar muka Halaman Pengetahuan dapat di lihat pada Gambar 4.13.

No	Penyakit	Gejala	MB	MD	Aksi
1	Berak Kapur (Pullorum Disease)	Mencoret keputih-putihan	0.8	0.0	[Ubah] [Hapus]
2	Berak Kapur (Pullorum Disease)	Sayap mengantung	0.6	0.2	[Ubah] [Hapus]
3	Berak Kapur (Pullorum Disease)	Terdapat kotoran putih menempel disekitar anus	1.0	0.0	[Ubah] [Hapus]
4	Kolera Ayam (Fowl Cholera)	Nafsu makan berkurang	0.4	0.2	[Ubah] [Hapus]
5	Kolera Ayam (Fowl Cholera)	Nafas ngorok basah	1.0	0.2	[Ubah] [Hapus]
6	Kolera Ayam (Fowl Cholera)	Pembengkakan pial	1.0	0.0	[Ubah] [Hapus]
7	Kolera Ayam (Fowl Cholera)	Pembengkakan dari sinus dan mata	1.0	0.0	[Ubah] [Hapus]

**Gambar 4.13 Halaman Pengetahuan**

#### 14. Halaman Post Keterangan

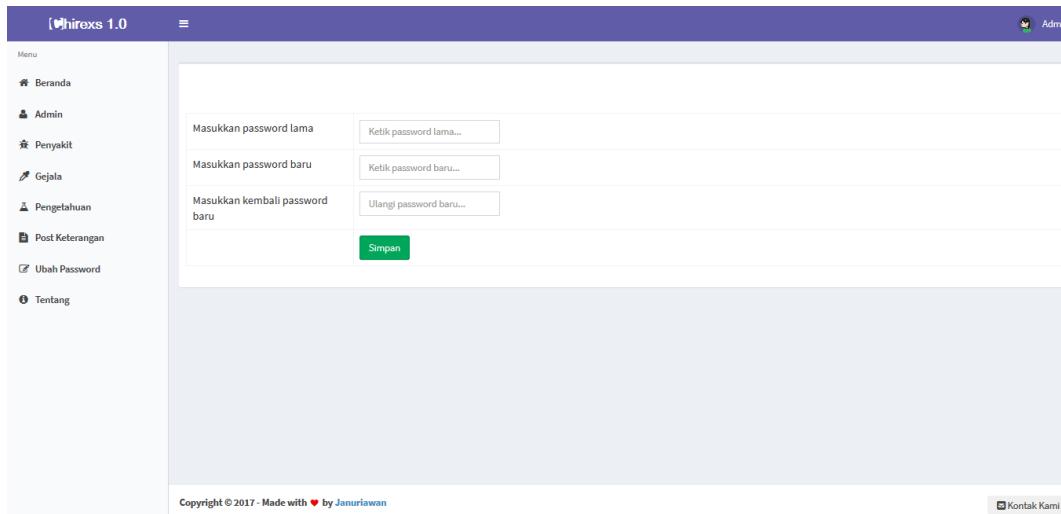
Halaman ini berfungsi untuk mengubah data pada halaman keterangan, fitur CRUD dan menggunakan Ckeditor sehingga seperti layaknya posting di website blog untuk detail antar muka Halaman Ubah Password dapat di lihat pada Gambar 4.14.

No	Nama Post	Detail Post	Saran Post	Aksi
1	Berak Kapur (Pulorum Disease)	Salah satu penyakit pada unggas adalah penyakit Berak Kapur (Pulorum Disease). Pulorum merupakan suatu penyakit infeksius pada unggas,	<b>Pencegahan:</b> 1. Menjaga sanitasi mulai dari mesin penetasan hingga sanitasi kandang 2. Melakukan	<button>Ubah</button> <button>Hapus</button>
2	Kolera Ayam (Fowl Cholera)	Kolera unggas ( <i>fowl cholera</i> ) adalah penyakit menular pada unggas dengan angka kesakitan dan kematian yang tinggi (Akoso, 1993).	<b>Pencegahan :</b> Prinsip pencegahan penyakit tersebut adalah : 1. Mengurangi populasi bibit penyakit di sekitar	<button>Ubah</button> <button>Hapus</button>
3	Flu Burung (Avian Influenza)	Flu burung atau dalam bahasa Inggris dikenal dengan avian flu atau avian influenza (AI) adalah penyakit menular yang disebabkan virus influenza A.	<b>Pencegahan:</b> Kebiasaan pola hidup sehat tetap memegang peranan penting dalam pencegahan.Untuk flu adalah tetap	<button>Ubah</button> <button>Hapus</button>
4	Tetelo (Newcastle Disease)	Newcastle Disease (ND) adalah penyakit yang sangat menular, dengan angka kematian yang tinggi, disebabkan oleh virus genus paramyxovi	<b>Pencegahan:</b> Tindakan vaksinasi merupakan langkah yang tepat sebagai upaya pencegahan terhadap penyakit ND. Pro	<button>Ubah</button> <button>Hapus</button>

**Gambar 4.14 Halaman Post Keterangan**

#### 15. Halaman Ubah Password

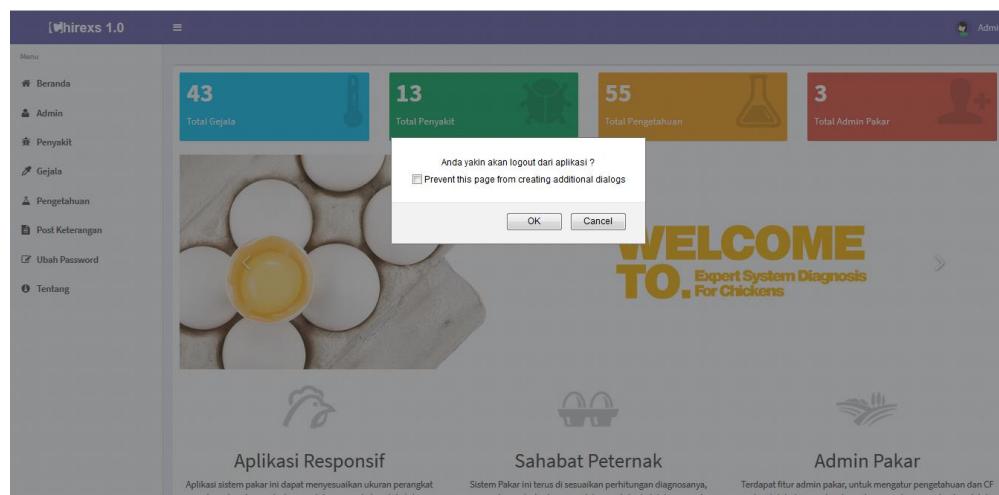
Halaman ini berfungsi untuk mengubah password pakar yang sedang login, pakar harus memasukkan password lama dan password baru untuk konfirmasi, nantinya password akan di simpan di database dan di enkripsi menggunakan algoritma MD5 *hash*, untuk detail antar muka Halaman Ubah Password dapat di lihat pada Gambar 4.15.



**Gambar 4.15 Halaman Ubah Password**

#### 16. Halaman LogOut

Berfungsi untuk keluar dari menu admin, nantinya akan muncul popup konfirmasi keluar dari aplikasi atau tidak, untuk detail antar muka Halaman LogOut dapat di lihat pada Gambar 4.16.



**Gambar 4.16 Halaman LogOut**

### 4.1.3 Implementasi Instalasi Program

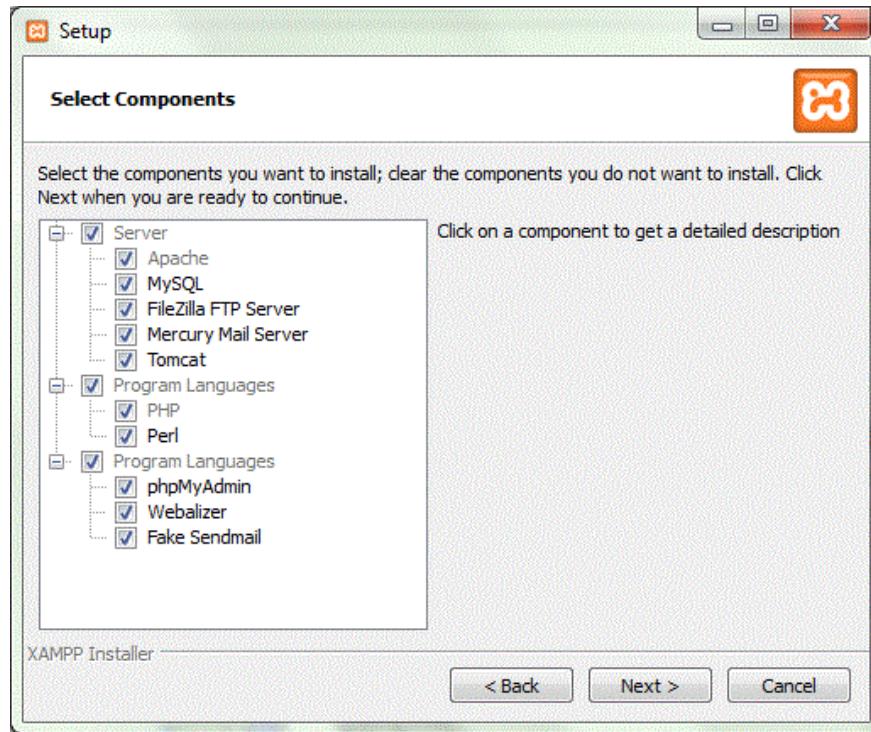
Untuk menjalankan Sistem Pakar Diagnosa Ayam dengan Metode Certainty Factor ini memerlukan web server. Web server masih berbasis *offline* dan di jalankan melalui *localhost* dengan software pembantu XAMPP, XAMPP adalah satu paket komplit web server yang mudah dan dapat digunakan dibergai sistem operasi. Dalam paketnya sudah terkandung Apache (web server), MySQL (database). Langkah-langkah untuk instalasi XAMP dan Program ialah sebagai berikut :

1. Siapkan bahan yang diperlukan yakni file instalasi XAMPP dan juga folder yang berisi sistem pakar ayam, atau dapat dilihat di Gambar 4.17.



**Gambar 4.17 Bahan yang diperlukan untuk instalasi**

2. Klik dua kali file instalasi pilih komponen XAMPP yang mau di install atau dapat di lihat di Gambar 4.18 , dan klik *Next*.



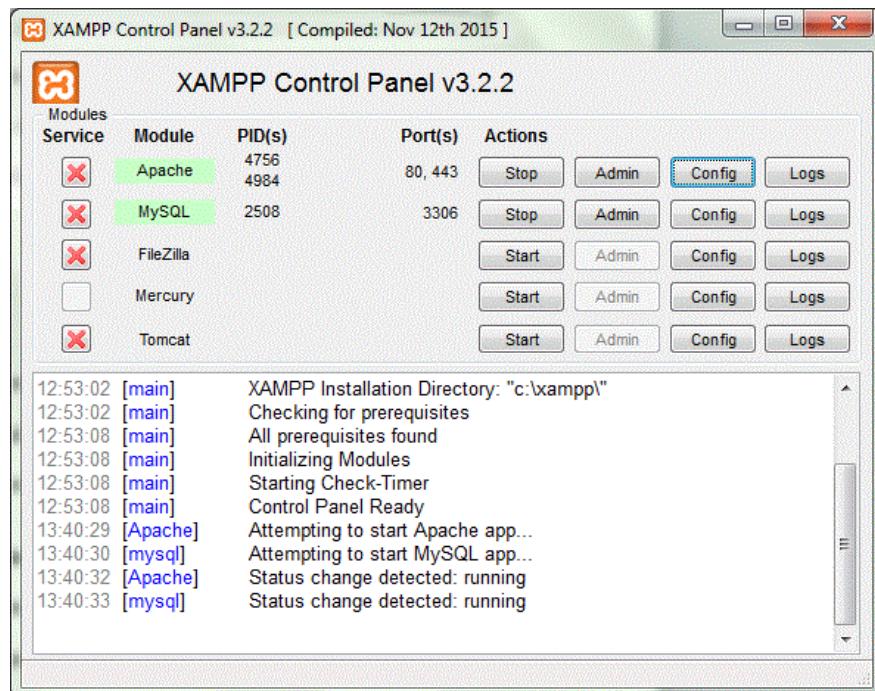
**Gambar 4.18 Tampilan *Setup* pilih komponen XAMPP**

3. Kemudian tekan *browse* untuk memilih lokasi penginstalan dan tekan *Next* untuk melanjutkan.
4. Tunggu proses instalasi selesai lihat pada Gambar 4.19.



**Gambar 4.19 Memulai Proses Instalasi XAMPP**

5. Setelah selesai, copy-kan folder sistem ttd digital tadi ke dalam folder htdocs di Direktori\_Instalasi:/xampp/htdocs/.
6. Kemudian jalankan XAMPP Control Panel dan start Apache dan MySQL seperti pada Gambar 4.20.



**Gambar 4.20 XAMPP Control Panel**

7. Kemudian buka browser dan buka alamat <http://localhost/phpmyadmin> dan import database sistem yang berada di dalam folder spkayam. Setelah itu ketikan alamat <http://localhost/spkayam> di browser kesayangan, lihat pada Gambar 4.21.



**Gambar 4.21 Mengakses Aplikasi Melalui peramban**

## 4.2 Uji Coba Sistem dan Program

Uji coba sistem dan program dilakukan untuk mengetahui apakah sistem aplikasi dapat berjalan dengan baik sesuai yang diharapkan atau

tidak. Pengujian sistem juga bertujuan untuk menghindari kesalahan yang mungkin terjadi saat pembuatan sistem aplikasi.

#### **4.2.1 *Black-box Testing***

Metode pengujian perangkat lunak dari aplikasi yang dibuat, dan akan dijelaskan pada Tabel 4.1 sampai 4.8.

**Tabel 4.1 *Black-box Testing* Menu Utama**

Input	Output	Status
Tombol Beranda	Menuju ke menu beranda	Sesuai
Tombol Diagnosa	Menuju ke menu diagnosa	Sesuai
Tombol Keterangan	Menuju ke menu Keterangan	Sesuai
Tombol Login	Menuju ke menu Login	Sesuai
Tombol Tentang	Menuju ke menu Tentang	Sesuai

#### **4.2 *Black Box Testing* Menu Mulai Diagnosa**

Input	Output	Status
Daftar Gejala	Memberikan pilihan daftar gejala penyakit	Sesuai
Tombol Lihat Hasil	Menuju ke menu lihat hasil, memproses data	Sesuai

**Tabel 4.3 *Black Box Testing* Menu Pakar**

Input	Output	Status
Tombol Beranda	Menuju ke menu Beranda	Sesuai
Tombol Admin	Menuju ke menu Admin	Sesuai

Tombol Penyakit	Menuju ke menu Penyakit	Sesuai
Tombol Gejala	Menuju ke menu Gejala	Sesuai
Tombol Pengetahuan	Menuju ke menu Pengetahuan	Sesuai
Tombol Ubah Password	Menuju ke menu Ubah Password	Sesuai
Tombol Logout	Menuju ke menu Logout	Sesuai

**Tabel 4.4 Black Box Testing Menu Admin**

Input	Output	Status
Daftar Admin	Memberikan daftar Admin	Sesuai
Tombol Tambah Admin	Menuju ke menu tambah admin, memproses data	Sesuai
Tombol Edit Admin	Menuju ke menu edit admin, memproses data	Sesuai
Tombol Hapus Admin	Menuju ke menu edit admin, menghapus data	Sesuai
Tombol Cari	Menuju ke menu pencarian admin, memproses data	Sesuai

**Tabel 4.5 Black Box Testing Menu Penyakit**

Input	Output	Status
Daftar Penyakit	Memberikan daftar Penyakit	Sesuai
Tombol Tambah Penyakit	Menuju ke menu tambah Penyakit, memproses data	Sesuai
Tombol Edit Penyakit	Menuju ke menu edit Penyakit, memproses data	Sesuai
Tombol Hapus	Menuju ke menu edit	Sesuai

Penyakit	Penyakit, menghapus data	
Tombol Cari	Menuju ke menu pencarian Penyakit, memproses data	Sesuai

**Tabel 4.6 Black Box Testing Menu Gejala**

Input	Output	Status
Daftar Gejala	Memberikan daftar Gejala	Sesuai
Tombol Tambah Gejala	Menuju ke menu tambah Gejala, memproses data	Sesuai
Tombol Edit Gejala	Menuju ke menu edit Gejala, memproses data	Sesuai
Tombol Hapus Gejala	Menuju ke menu edit Gejala, menghapus data	Sesuai
Tombol Cari	Menuju ke menu pencarian Gejala, memproses data	Sesuai

**Tabel 4.7 Black Box Testing Menu Pengetahuan**

Input	Output	Status
Daftar Pengetahuan	Memberikan daftar Pengetahuan	Sesuai
Tombol Tambah Pengetahuan	Menuju ke menu tambah Pengetahuan, memproses data	Sesuai
Tombol Edit Pengetahuan	Menuju ke menu edit Pengetahuan, memproses data	Sesuai
Tombol Hapus Pengetahuan	Menuju ke menu edit Pengetahuan, menghapus data	Sesuai

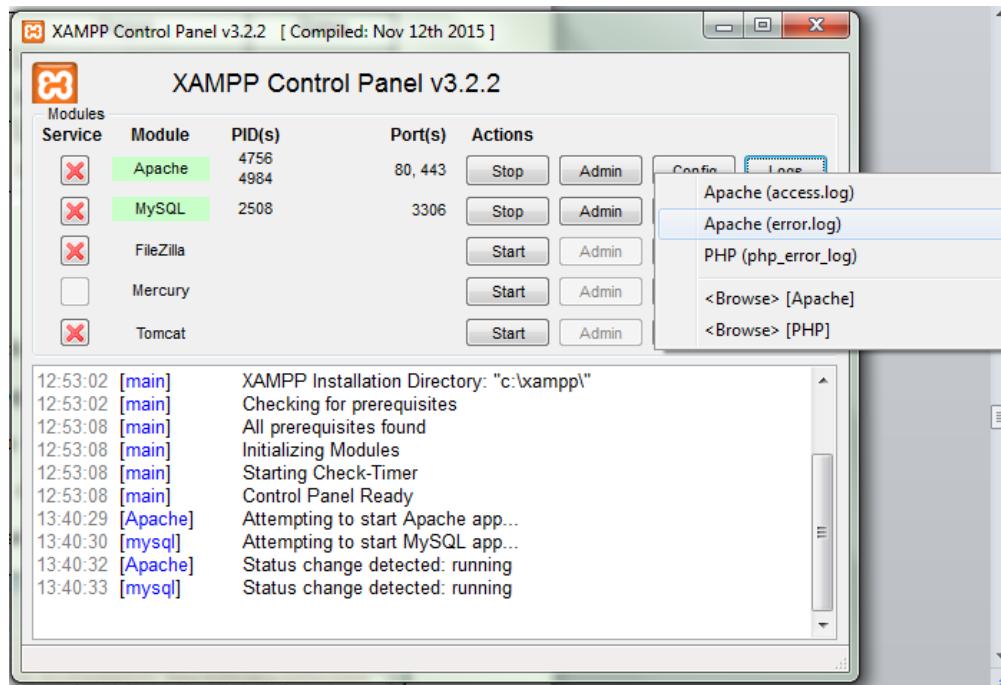
Tombol Cari	Menuju ke menu pencarian Pengetahuan, memproses data	Sesuai
-------------	--	--------

**Tabel 4.8 Black Box Testing Ubah Password**

Input	Output	Status
Data Pengetahuan	Memberikan data Pengetahuan	Sesuai
Tombol Simpan	Menyimpan data yang dimasukkan ke database	Sesuai

#### 4.2.2 White-box Testing

*White-box Testing* dilakukan dengan cara melihat kedalam modul untuk meneliti kode program dan untuk menganalisa, apakah terdapat kesalahan dalam program tersebut atau tidak. Dalam aplikasi XAMPP, *white box testing* dapat dilihat dari *logcat*, yang terdapat error pada program jika program tersebut tidak sesuai dengan aturan yang sesuai, Silahkan menuju XAMPP Control Panel dan masuk pada tab Logs dan lihat pada error.log seperti pada Gambar 4.22. Untuk Gambar 4.23 Merupakan proses pengecekan error pada apache dengan mengklik tombol log pada menu XAMPP.



Gambar 4.22 Proses Pengecekan *error log* Apache

This screenshot shows a Notepad window titled 'error - Notepad' containing a large amount of Apache error log data. The log entries are timestamped and show various PHP warnings and errors, such as 'PHP Warning: file\_get\_contents(): php\_network\_getaddresses: getaddrinfo failed: No such host is known.', 'PHP Warning: file\_get\_contents(): failed to open stream: php\_network\_getaddresses: getaddrinfo failed: No such host is known.', and numerous syntax errors like 'Parse error: syntax error, unexpected \$', 'Parse error: syntax error, unexpected 'for' (T\_FOR)', and 'Parse error: syntax error, unexpected ''http://mu-det1.blogspot.c...'', expecting T\_CONSTANT\_ENCAPSED\_STRING or T\_WHITESPACE'.

Gambar 4.23 White-box Testing log Apache

### 4.2.3 Kesalahan Kode Program (*Syntax Error*)

Kesalahan dalam penulisan program dapat terjadi apabila *programmer* menuliskan kode program yang tidak sesuai dengan aturan *syntax*. Dalam beberapa bahasa pemrograman berlaku *case sensitive*

yaitu besar kecil huruf diperhitungkan, merupakan salah satu sebab kesalahan kode.

#### **4.2.4 Kesalahan Proses (*Run Time Error*)**

Kesalahan proses (*run time error*) terjadi saat program dijalankan. Hal ini menyebabkan proses program berhenti sebelum selesai pada waktunya karena *compiler* menemukan kondisi yang belum terpenuhi dan tidak layak untuk dikerjakan.

#### **4.2.5 Kesalahan Logika (*Logical Error*)**

Kesalahan logika (*Logical Error*) terjadi akibat kesalahan dalam logika menulis kode program. Hal ini menyebabkan hasil dari program tidak sesuai dengan yang diharapkan. Untuk mengetahui kesalahan logika adalah dengan melakukan pengecekan keseluruhan kode program sampai menemukan logika yang tidak sesuai.

#### **4.2.6 Kesalahan Koneksi Database (*Database Connection*)**

Kesalahan koneksi database (*Database Connection*) ini umum terjadi karena modul mysql tidak di jalankan atau database di dalam phpmyadmin tidak sesuai dengan sistem, dapat juga nama database atau password tidak sesuai dengan file koneksi pada sistem.

### **4.3 Pengujian Hasil Diagnosa**

Dari seluruh kemungkinan hasil jawaban yang diambil 15 sampel untuk menguji keakuratan hasil diagnosa sistem dengan hasil diagnosa pakar. Hasil pengujian tersebut dapat dilihat pada Tabel 4.9 berikut ini:

**Tabel 4.9 Pengujian Hasil Diagnosa**

NO	GEJALA	HASIL			KESI MPU LAN
		CF USER	SIST EM	PAK AR	
1	1. Nafsu makan berkurang 2. Nafas sesak / megap-megap 3. Nafas ngorok basah 4. Batuk	1. Hampir pasti ya 2. Kemungkinan besar ya 3. Mungkin ya 4. Pasti ya	BAM (100 %)	BAM	S
2	1. Telur lebih kecil 2. Kelumpuhan pada tembolok 3. Batuk berdarah	1. Kemungkinan besar ya 2. Tidak tahu 3. Hampir pasti ya	IL (80%)	IL	S
3	1. Diare 2. Produksi telur menurun	1. Hampir pasti tidak 2. Kemungkinan besar ya	EDS (41%)	EDS	S
4	1. Terdapat kotoran putih menempel disekitar 2. Mati secara mendadak	1. Kemungkinan besar ya 2. Kemungkinan besar ya	AI (60%)	BK	T
5	1. Perut membesar 2. Bergerombol di sudut kandang	1. Hampir pasti ya 2. Mungkin ya 3. Tidak tahu	BA (80%)	BA	S

	3. Mematuk daerah kloaka				
6	1. Duduk dengan sikap membungkuk 2. Kelihatan mengantuk dengan bulu berdiri 3. Badan kurus	1. Hampir pasti ya 2. Kemungkinan besar ya 3. Kemungkinan besar ya	GD (64%)	BA	T
7	1. Tampak lesu 2. Kepala bengkak 3. Kepala terputar 4. Sayap menggantung 5. Kerabang telur kasar	1. Hampir pasti ya 2. Mungkin ya 3. Hampir pasti ya 4. Mungkin tidak 5. Mungkin ya	ND (54%)	ND	S
8	1. Keluar cairan dari mata dan hidung 2. Kotoran kuning kehijauan 3. Batuk berdarah 4. Kelihatan mengantuk dengan bulu berdiri	1. Pasti ya 2. Tidak tahu 3. Mungkin ya 4. Kemungkinan besar ya	SA (80%)	SA	S
9	1. Terdapat kotoran putih menempel disekitar anus 2. Kotoran atau feses berdarah	1. Mungkin ya 2. Hampir pasti ya	BD (80%)	BD	S
10	1. Nafsu makan berkurang 2. Nafas sesak / megap-megap	1. Mungkin ya 2. Mungkin ya 3. Kemungkinan besar tidak	EDS (36%)	EDS	S

	3. Nafas ngorok basah 4. Diare	4. Kemungkinan besar ya			
11	1. Perut membesar 2. Sayap menggantung	1. Tidak tahu tidak 2. Kemungkinan besar ya	TA (36%)	TA	S
12	1. Mematuk daerah kloaka 2. Gangguan kesadaran	1. Hampir pasti ya 2. Kemungkinan besar ya	GD (80%)	GD	S
13	1. Kepala bengkak 2. Mati Secara Mendadak 3. Putih telur encer	1. Hampir pasti ya 2. Pasti ya 3. Mungkin Tidak	AI (100 %)	AI	S
14	1. Mencret kehijau-hijauan 2. Nampak membiru 3. Pembengkakan pial 4. Jengger pucat 5. Kaki dan sayap lumpuh	1. Mungkin ya 2. Mungkin ya 3. Mungkin tidak 4. Tidak tahu 5. Hampir pasti ya	MD (48%)	MD	S
15	1. Bersin-bersin 2. Produksi telur menurun 3. Mati secara mendadak 4. Putih telur encer 5. Tidur paruhnya di letakkan di lantai 6. Terdapat lendir bercampur darah pada rongga mulut	1. Hampir pasti ya 2. Mungkin ya 3. Mungkin tidak 4. Tidak tahu 5. Mungkin ya 6. Kemungkinan besar ya	IL (60%)	IL	S

Keterangan :

S = Sama

T = Tidak Sama

BK = Berak Kapur (Pullorum Disease)

KA = Kolera Ayam (Fowl Cholera)

AI = Flu Burung (Avian Influenza)

ND = Tetelo (Newcastle Disease)

TA = Tipus Ayam (Fowl Typhoid)

BD = Berak Darah (Coccidosis)

BD = Gumboro (Gumboro Disease)

SA = Salesma Ayam (Infectious Coryza)

BAM = Batuk Ayam Menahun (Infectious Bronchitis)

BA = Busung Ayam (Lymphoid Leukosis)

SA = Salesma Ayam (Infectious Coryza)

IL = Batuk Darah (Infectious Laryngotrac)

MD = Mareks (Mareks Disease)

EDS = Produksi Telur (Egg Drop Syndrome 76/EDS 76)

Dari tabel 4.4 dapat dilihat bahwa dari 15 sampel menghasilkan 13 hasil yang sama dan 1 hasil yang berbeda dengan hasil di lapangan. Maka dari itu pengujian sampel di atas sistem ini menghasilkan akurasi ketepatan sebesar 86.67% dan dikatakan layak oleh pakar.

Nilai akurasi =  $13/15 * 100\% = 86.67\%$

#### **4.4 Implementasi Kode Program**

Implementasi kode program adalah suatu tahapan dalam pembuatan sistem yang bertujuan agar sistem yang digunakan sesuai dengan rancangan yang telah dibuat. Tujuan dari implementasi kode program ini adalah untuk memperjelas pembaca tentang program-program yang digunakan dalam pembuatan aplikasi. Pada implementasi kode program, akan dibagi kedalam beberapa class dasar diantaranya, Class Menu\_Beranda merupakan menu utama yang memuat tombol untuk menjalankan aktivitas selanjutnya, Menu\_Pakar adalah menu yg diakses oleh pakar untuk pengetahuan sistem, DB Adapter berfungsi sebagai pengontrol untuk mengambil data dari database yang dibutuhkan pada class lain, Class Diagnosa digunakan oleh user untuk memilih gejala yang dirasakan dengan class kondisi yang di sediakan dan berisi tentang logika perhitungan certainty factor, Class Hasil menampilkan informasi penyakit yang berasal dari hasil dari perhitungan certainty factor berdasarkan gejala yang terpilih, untuk lebih jelasnya bisa dilihat pada lampiran.

## **BAB V**

### **PENUTUP**

#### **1.7      Kesimpulan**

Berdasarkan hasil pembahasan perancangan aplikasi sistem pakar ayam menggunakan metode certainty factor berbasis *website responsive*, maka diambil kesimpulan sebagai berikut:

1. Aplikasi sistem pakar sudah berjalan dengan metode *Certainty Factor* dan untuk melakukan diagnosa pada ayam di user dapat memilih nilai sesuai Interpretasi Certainty Factor yang telah diberikan dan di proses bersama CF pakar.
2. Berdasarkan pengujian yang dilakukan nilai ke akuratan sistem dengan pakar hampir sama, dari 15 sampel menghasilkan 13 hasil yang sama dan 2 hasil yang berbeda dengan hasil di lapangan, di sistem juga memberikan deskripsi hasil diagnosa lengkap beserta saran dan tindakan. Dan *Inference engine* bekerja dengan baik, sesuai dengan *rule* yang telah diprogramkan sebelumnya.
3. Tingkat responsif sudah teruji dengan perangkat tester Mozilla Quantum *Responsive Design Mode* dan semua class telah mengikuti perubahan tampilan, Dan telah sesuai dari perangkat terkecil / *mobile* hingga yang besar / *desktop*.

## 5.2 Saran

Dibutuhkan kritik atau saran demi terciptanya aplikasi yang lebih baik dan dapat dimanfaatkan kedepannya. Saran yang ingin penulis sampaikan antara lain:

1. Penambahan fitur *chatting online* antara pengguna dengan dokter spesialis ayam atau pakar yang ahli dalam bidang peternakan.
2. Konversi dari website ke aplikasi mobile sehingga lebih praktis untuk pengguna mobile.
3. Penambahan deskripsi dan keterangan gambar saat melakukan diagnosa akan lebih membantu dalam pemilihan gejala yang diderita.

## DAFTAR PUSTAKA

- [1] Badan Pusat Statistik. 2014. Jumlah Perusahaan Peternakan Ternak Besar dan Kecil Menurut Badan Hukum/ Usaha, 2000-2014, (Online), (<http://www.bps.go.id/linkTableDinamis/view/id/1086>, diakses 10 Januari 2017).
- [2] Effendi, Mella R. 2013. Sistem Pakar Untuk Mendiagnosa Penyakit Ayam di Koperasi Serba Usaha (KSU) Jati Mekar.
- [3] Yanto, Taofhik A. 2013. Perancangan Sistem Pakar Diagnosis Penyakit Ayam dengan Menggunakan Metode Forward Chaining.
- [4] Siti dan Rina. 2010. Sistem Pakar: Diagnosis Penyakit Unggas dengan Metode Certainty Factor.
- [5] *University of Illinois Extension*. 2013. What Is a Chicken? - Incubation and Embryology - *University of Illinois Extension*, (Online), (<https://extension.illinois.edu/eggs/res08-whatis.html>, diakses 10 Januari 2017).
- [6] Handayani, L. dan Sutikno, T. 2008. Sistem Pakar untuk Diagnosa Penyakit THT Berbasis Web. dengan e2gLite *Expert System Shell* , Jurnal Teknologi Industri, Vol. XII, No 1.
- [7] Martin, J and Oxman, S. 1988. *Building Expert Systems: A Tutorial*. Prentice Hall Press, Edinburgh, 199–203.
- [8] Kusrini, 2006. Sistem Pakar, Teori dan Aplikasi, Yogyakarta: ANDI Offset Yogyakarta.
- [9] Kusrini, 2008. Aplikasi Sistem Pakar, Yogyakarta: ANDI Offset Yogyakarta.
- [10] Giarratano, J. & Riley, G., 2005. Expert Sistem: Principles. and Programming, 4th Edition, Boston: PWS Publishing Company.

- [11] Desiani, A dan Arhami, M., 2006. Konsep Dasar Sistem Pakar, Yogyakarta: ANDI Offset Yogyakarta.
- [12] Herawan Hayadi, Kasman Rukun. 2016. What is Expert System, Yogyakarta: Deepublish.
- [13] Budhi, Gregorius S. dan Intan, Rolly. 2008. Penerapan Probabilitas Penggunaan Fakta guna menentukan Certainty Factor sebuah Rule pada Rule Base Expert System. UK Petra Surabaya jurusan Teknik Informatika.
- [14] Kusrini, 2007. Strategi Perancangan dan Pengelolaan Basis Data, Yogyakarta: ANDI Offset Yogyakarta.
- [15] Fathansyah. 2012. Basis Data. Bandung: Informatika.
- [16] A. S., Rosa dan Shalahuddin, M. 2013. Rekayasa Perangkat Lunak Terstruktur. Dan Berorientasi Objek. Bandung: Informatika.
- [17] Booch, G., Maksimchuk, R.A., Engle, M.W., Conallen, J., Houston, K.A., Young, B.J., 2007. Object-Oriented Analysis and Design with Applications, 3rd Edition, Addison-Wesley Professional.
- [18] Sakur, Stendy B, 2011. PHP 5 Pemrograman Berorientasi Objek - Konsep dan Implementasi, Yogyakarta: ANDI Offset Yogyakarta.
- [19] Indonesia. Departemen Pendidikan Nasional, Pusat Bahasa (Indonesia) 2008. Kamus besar bahasa Indonesia Pusat Bahasa, Gramedia Pustaka Utama.

- [20] Sunyoto, Andi, 2007. Pemograman database dengan visual basic dan Microsoft. SQL, Yogyakarta: Andi Offset. Yogyakarta.
- [21] Kadir, Abdul, 2001, Web Dinamis Menggunakan PHP, Yogyakarta: Andi Offset. Yogyakarta.

## Lampiran 1 :

### Kode program diagnosa :

```

<title>Diagnosa - Chirexs 1.0</title>
<?php
switch ($_GET['act']) {

    default:
        if ($_POST['submit']) {
            $arcolor = array('#ffffff', '#cc66ff', '#019AFF', '#00CBFD',
'#00FEFE', '#A4F804', '#FFFC00', '#FDCC01', '#FD9A01', '#FB6700');
            date_default_timezone_set("Asia/Jakarta");
            $inptanggal = date('Y-m-d H:i:s');

            $arobot = array('0', '1', '0.8', '0.6', '0.4', '-0.2', '-0.4',
'-0.6', '-0.8', '-1');
            $argejala = array();

            for ($i = 0; $i < count($_POST['kondisi']); $i++) {
                $arkondisi = explode("_", $_POST['kondisi'][$i]);
                if (strlen($_POST['kondisi'][$i]) > 1) {
                    $argejala += array($arkondisi[0] => $arkondisi[1]);
                }
            }

            $sqlkondisi = mysql_query("SELECT * FROM kondisi order by
id+0");
            while ($rkondisi = mysql_fetch_array($sqlkondisi)) {
                $arkondisitext[$rkondisi['id']] = $rkondisi['kondisi'];
            }

            $sqlpkt = mysql_query("SELECT * FROM penyakit order by
kode_penyakit+0");
            while ($rpkt = mysql_fetch_array($sqlpkt)) {
                $arpkt[$rpkt['kode_penyakit']] = $rpkt['nama_penyakit'];
                $ardpkt[$rpkt['kode_penyakit']] = $rpkt['det_penyakit'];
                $arspkt[$rpkt['kode_penyakit']] = $rpkt['srn_penyakit'];
                $argpkt[$rpkt['kode_penyakit']] = $rpkt['gambar'];
            }

            //print_r($arkondisitext);
// ----- perhitungan certainty factor (CF) -----
// ----- START -----
            $sqlpenyakit = mysql_query("SELECT * FROM penyakit order by
kode_penyakit");
            $arpenyakit = array();
            while ($rpenyakit = mysql_fetch_array($sqlpenyakit)) {
                $scftotal_temp = 0;
                $cf = 0;
                $sqlgejala = mysql_query("SELECT * FROM basis_pengetahuan
where kode_penyakit=$rpenyakit[kode_penyakit]");
                $scflama = 0;
                while ($rgejala = mysql_fetch_array($sqlgejala)) {
                    $arkondisi = explode("_", $_POST['kondisi'][0]);
                    $gejala = $arkondisi[0];

```

```

        for ($i = 0; $i < count($_POST['kondisi']); $i++) {
            $arkondisi = explode("_", $_POST['kondisi'][$i]);
            $gejala = $arkondisi[0];
            if ($rgejala['kode_gejala'] == $gejala) {
                $cf = ($rgejala['mb'] - $rgejala['md']) *
$arbobot[$arkondisi[1]];
                if (($cf >= 0) && ($cf * $cflama >= 0)) {
                    $cflama = $cflama + ($cf * (1 - $cflama));
                }
                if ($cf * $cflama < 0) {
                    $cflama = ($cflama + $cf) / (1 - Math . Min(Math .
abs($cflama), Math . abs($cf)));
                }
                if (($cf < 0) && ($cf * $cflama >= 0)) {
                    $cflama = $cflama + ($cf * (1 + $cflama));
                }
            }
        }
        if ($cflama > 0) {
            $arpenyakit += array($rpenyakit[kode_penyakit] =>
number_format($cflama, 4));
        }
    }

arsort($arpenyakit);

$inpgejala = serialize($argejala);
$inppenyakit = serialize($arpenyakit);

$np1 = 0;
foreach ($arpenyakit as $key1 => $value1) {
    $np1++;
    $idpkt1[$np1] = $key1;
    $vlpkt1[$np1] = $value1;
}

mysql_query("INSERT INTO hasil(
    tanggal,
    gejala,
    penyakit,
    hasil_id,
    hasil_nilai
)
VALUES(
    '$inptanggal',
    '$inpgejala',
    '$inppenyakit',
    '$idpkt1[1]',
    '$vlpkt1[1]'
) ");
// ----- END -----
echo "<div class='content'>
<h2 class='text text-primary'>Hasil Diagnosis
&nbsp;&nbsp;<button id='print' onClick='window.print();' data-

```

```

toggle='tooltip' data-placement='right' title='Klik tombol ini
untuk mencetak hasil diagnosa'><i class='fa fa-print'></i>
Cetak</button> </h2>
    <hr><table class='table table-bordered table-striped
diagnosa'>
        <thead>
            <tr>
                <th width=8%>No</th>
                <th width=10%>Kode</th>
                <th>Gejala yang dialami (keluhan)</th>
                <th width=20%>Pilihan</th>
            </tr>;
        <tbody>
            $ig = 0;
            &foreach ($argejala as $key => $value) {
                $kondisi = $value;
                $ig++;
                $gejala = $key;
                $sql4 = mysql_query("SELECT * FROM gejala where
kode_gejala = '$key'");
                $r4 = mysql_fetch_array($sql4);
                echo '<tr><td>' . $ig . '</td>';
                echo '<td>G' . str_pad($r4[kode_gejala], 3, '0',
STR_PAD_LEFT) . '</td>';
                echo '<td><span class="hasil text text-primary">' .
$r4[nama_gejala] . "</span></td>";
                echo '<td><span class="kondisipilih" style="color:' .
$arcolor[$kondisi] . '">' . $arkondisitext[$kondisi] .
"</span></td></tr>";
            }
            $np = 0;
            &foreach ($arpenyakit as $key => $value) {
                $np++;
                $idpkt[$np] = $key;
                $nmpkt[$np] = $arpkt[$key];
                $vlpkt[$np] = $value;
            }
            if ($argpkt[$idpkt[1]]) {
                $gambar = 'gambar/penyakit/' . $argpkt[$idpkt[1]];
            } else {
                $gambar = 'gambar/noimage.png';
            }
            echo "</table><div class='well well-small'><img class='card-
img-top img-bordered-sm' style='float:right; margin-left:15px;'>
src=''" . $gambar . "' height=200><h3>Hasil Diagnosa</h3>";
            echo "<div class='callout callout-default'>Jenis penyakit
yang diderita adalah <b><h3 class='text text-success'>" .
$nmpkt[1] . "</b> / " . round($vlpkt[1], 2) . " % (" . $vlpkt[1] .
")<br></h3>";
            echo "</div></div><div class='box box-info box-solid'><div
class='box-header with-border'><h3 class='box-
title'>Detail</h3></div><div class='box-body'><h4>";
            echo $ardpkt[$idpkt[1]];
            echo "</h4></div></div>
            <div class='box box-warning box-solid'><div class='box-
header with-border'><h3 class='box-title'>Saran</h3></div><div
class='box-body'><h4>";
            echo $arspkt[$idpkt[1]];
            echo "</h4></div></div>

```

```

<div class='box box-danger box-solid'><div class='box-
header with-border'><h3 class='box-title'>Kemungkinan
lain:</h3></div><div class='box-body'><h4>" ;
    for ($ipl = 2; $ipl < count($idpkt); $ipl++) {
        echo " <h4><i class='fa fa-caret-square-o-right'></i> " .
$nmppkt[$ipl] . "</b> / " . round($vlpkt[$ipl], 2) . " % (" .
$vlpkt[$ipl] . ")<br></h4>";
    }
    echo "</div></div>
    </div>";
} else {
    echo "
        <h2 class='text text-primary'>Diagnosa Penyakit</h2> <hr>
        <div class='alert alert-success alert-dismissible'>
            <button type='button' class='close' data-
dismiss='alert' aria-hidden='true'>x</button>
            <h4><i class='icon fa fa-exclamation-
triangle'></i>Perhatian !
        </h4>
        Silahkan memilih gejala sesuai dengan kondisi ayam
        anda, anda dapat memilih kepastian kondisi ayam dari pasti tidak
        sampai pasti ya, jika sudah tekan tombol proses (<i class='fa fa-
        search-plus'></i>) di bawah untuk melihat hasil.
    </div>
    <form name=text_form method=POST action='diagnosa' >
        <table class='table table-bordered table-striped
        konsultasi'><tbody class='pilihkondisi'>
            <tr><th>No</th><th>Kode</th><th>Gejala</th><th
        width='20%'>Pilih Kondisi</th></tr>" ;

        $sql3 = mysql_query("SELECT * FROM gejala order by
kode_gejala");
        $i = 0;
        while ($r3 = mysql_fetch_array($sql3)) {
            $i++;
            echo "<tr><td class=opsi>$i</td>";
            echo "<td class=opsi>G" . str_pad($r3[kode_gejala], 3,
'0', STR_PAD_LEFT) . "</td>";
            echo "<td class=gejala>$r3[nama_gejala]</td>";
            echo '<td class="opsi"><select name="kondisi[]" id="sl' .
$i . '" class="opsikondisi"/><option data-id="0" value="0">Pilih
        jika sesuai</option>';
            $s = "select * from kondisi order by id";
            $q = mysql_query($s) or die($s);
            while ($rw = mysql_fetch_array($q)) {
                ?
                <option data-id=<?php echo $rw['id']; ?>" value=<?php
                echo $r3['kode_gejala'] . '_' . $rw['id']; ?>><?php echo
                $rw['kondisi']; ?></option>
                ?
            }
            echo '</select></td>';
        ?
        <script type="text/javascript">
            $(document).ready(function () {

```

```

        var arcolor = new Array('#ffffff', '#cc66ff',
'#019AFF', '#00CBFD', '#00FEFE', '#A4F804', '#FFFC00', '#FDCCD01',
'#FD9A01', '#FB6700');
        setColor();
        $('.pilihkondisi').on('change', 'tr td select#sl<?php
echo $i; ?>', function () {
        setColor();
    });
        function setColor()
{
    var selectedItem = $('tr td select#sl<?php echo $i;
?> :selected');
    var color = arcolor[selectedItem.data("id")];
    $('tr td select#sl<?php echo $i;
?>.opsikondisi').css('background-color', color);
    console.log(color);
}
});
</script>
<?php
echo "</tr>";
}
echo "
<input class='float' type=submit data-toggle='tooltip'
data-placement='top' title='Klik disini untuk melihat hasil
diagnosa' name=submit value='&#xf00e;' style='font-family:Arial,
FontAwesome'>
</tbody></table></form>";
}
break;
}
?>
```

## Lampiran 2 :

### Kode program Pengetahuan Pakar :

```

<title>Pengetahuan - Chirexs 1.0</title>
<?php

session_start();
if (!isset($_SESSION['username']) &&
isset($_SESSION['password'])) {
    header('location:index.php');
    exit();
} else {
    ?>
<script type="text/javascript">
function Blank_TextField_Validator()
{
if (text_form.kode_penyakit.value == "")
{
    alert("Pilih dulu penyakit !");
    text_form.kode_penyakit.focus();
    return (false);
}
```

```

        }
        if (text_form.kode_gejala.value == "")
        {
            alert("Pilih dulu gejala !");
            text_form.kode_gejala.focus();
            return (false);
        }
        if (text_form.mb.value == "")
        {
            alert("Isi dulu MB !");
            text_form.mb.focus();
            return (false);
        }
        if (text_form.md.value == "")
        {
            alert("Isi dulu MD !");
            text_form.md.focus();
            return (false);
        }
        return (true);
    }
    function Blank_TextField_Validator_Cari()
    {
        if (text_form.keyword.value == "")
        {
            alert("Isi dulu keyword pencarian !");
            text_form.keyword.focus();
            return (false);
        }
        return (true);
    }
    -->
</script>
<?php
include "config/fungsi_alert.php";
$aksi="modul/pengetahuan/aksi_pengetahuan.php";
switch($_GET[act]){
    // Tampil pengetahuan
    default:
        $offset=$_GET['offset'];
        //jumlah data yang ditampilkan perpage
        $limit = 15;
        if (empty ($offset)) {
            $offset = 0;
        }
        $stampil=mysql_query("SELECT * FROM basis_pengetahuan ORDER BY
kode_pengetahuan");
        echo "<form method=POST action='?module=pengetahuan'
name=text_form onsubmit='return Blank_TextField_Validator_Cari()'
        <br><br><table class='table table-bordered'>
        <tr><td><input class='btn bg-olive margin' type=button
name=tambah value='Tambah Basis Pengetahuan'
onclick=\"window.location.href='pengetahuan/tambahpengetahuan';\">
<input type=text name='keyword' style='margin-left: 10px;' placeholder='Ketik dan tekan cari...' class='form-control'>
        </td></tr></table></form>";
}

```

```

value='$_POST[keyword]' /> <input class='btn bg-olive margin'
type=submit value=' Cari ' name=Go></td> </tr>
    </table></form>";
    $baris=mysql_num_rows($tampil);
    if ($_POST[Go]){
        $numrows = mysql_num_rows(mysql_query("SELECT * FROM
basis_pengetahuan b,penyakit p where
b.kode_penyakit=p.kode_penyakit AND p.nama_penyakit like
'%$_POST[keyword]%'"));
        if ($numrows > 0){
            echo "<div class='alert alert-success alert-
dismissible'>
                <h4><i class='icon fa fa-check'></i> Sukses!</h4>
                Pengetahuan yang anda cari di temukan.
            </div>";
            $i = 1;
        }
        echo" <table class='table table-bordered' style='overflow-
x-auto' cellpadding='0' cellspacing='0'>
            <thead>
                <tr>
                    <th>No</th>
                    <th>Penyakit</th>
                    <th>Gejala</th>
                    <th>MB</th>
                    <th>MD</th>
                    <th width='21%'>Aksi</th>
                </tr>
            </thead>
            <tbody>";
        $hasil = mysql_query("SELECT * FROM basis_pengetahuan
b,penyakit p where b.kode_penyakit=p.kode_penyakit AND
p.nama_penyakit like '%$_POST[keyword]%'");
        $no = 1;
        $counter = 1;
        while ($r=mysql_fetch_array($hasil)){
            if ($counter % 2 == 0) $warna = "dark";
            else $warna = "light";
            $sql = mysql_query("SELECT * FROM gejala where kode_gejala =
'$r[kode_gejala]'");
            $rgejala=mysql_fetch_array($sql);
            echo "<tr class='".$warna."'>
                <td align=center>$no</td>
                <td>$r[nama_penyakit]</td>
                <td>$rgejala[nama_gejala]</td>
                <td align=center>$r[mb]</td>
                <td align=center>$r[md]</td>
                <td align=center><a type='button' class='btn btn-
success margin'
href=pengetahuan/editpengetahuan/$r[kode_pengetahuan]><i class='fa
fa-pencil-square-o' aria-hidden='true'></i> Ubah </a> &ampnbsp
                <a type='button' class='btn btn-danger margin'
href=\"JavaScript: confirmIt('Anda yakin akan menghapusnya
?', '$aksi?module=pengetahuan&act=hapus&id=$r[kode_pengetahuan]', ''
,'','','','u','n','Self','Self')\" onMouseOver=\"self.status='';
return true\\" onMouseOut=\"self.status=''; return true\\"><i
class='fa fa-trash-o' aria-hidden='true'></i> Hapus</a>
            </tr>";
            $counter++;
        }
    }
}

```

```

        </td></tr>";
    $no++;
    $counter++;
}
echo "</tbody></table>";
}
else{
echo "<div class='alert alert-danger alert-
dismissible'>
<h4><i class='icon fa fa-ban'></i> Gagal!</h4>
Maaf, Pengetahuan yang anda cari tidak ditemukan ,
silahkan inputkan dengan benar dan cari kembali.
</div>";
}
}else{

if($baris>0){
echo" <table class='table table-bordered' style='overflow-
x:auto' cellpadding='0' cellspacing='0'>
<thead>
<tr>
<th>No</th>
<th>Penyakit</th>
<th>Gejala</th>
<th>MB</th>
<th>MD</th>
<th width='21%'>Aksi</th>
</tr>
</thead>
<tbody>
";
$hasil = mysql_query("SELECT * FROM basis_pengetahuan ORDER BY
kode_pengetahuan limit $offset,$limit");
$no = 1;
$no = 1 + $offset;
$counter = 1;
while ($r=mysql_fetch_array($hasil)){
if ($counter % 2 == 0) $warna = "dark";
else $warna = "light";
$sql = mysql_query("SELECT * FROM gejala where kode_gejala =
'$r[kode_gejala]' ");
$rgejala=mysql_fetch_array($sql);
$sql2 = mysql_query("SELECT * FROM penyakit where
kode_penyakit = '$r[kode_penyakit]' ");
$rpenyakit=mysql_fetch_array($sql2);
echo "<tr class='".$warna."'>
<td align=center>$no</td>
<td>$rpenyakit[nama_penyakit]</td>
<td>$rgejala[nama_gejala]</td>
<td align=center>$r[mb]</td>
<td align=center>$r[md]</td>
<td align=center>
<a type='button' class='btn btn-success margin'
href=pengetahuan/editpengetahuan/$r[kode_pengetahuan]><i class='fa
fa-pencil-square-o' aria-hidden='true'></i> Ubah </a> &ampnbsp

```

```

        <a type='button' class='btn btn-danger margin'
        href=\"JavaScript: confirmIt('Anda yakin akan menghapusnya
        ?','$aksi?module=pengetahuan&act=hapus&id=$r[kode_pengetahuan]', ''
        , '' , '' , 'u' , 'n' , 'Self' , 'Self')\" onMouseOver=\"self.status='';
        return true\" onMouseOut=\"self.status=''; return true\"
            <i class='fa fa-trash-o' aria-hidden='true'></i>
Hapus</a>
        </td></tr>";
    $no++;
    $counter++;
}
echo "</tbody></table>";
echo "<div class=paging>";

if ($offset!=0) {
    $prevoffset = $offset-10;
    echo "<span class=prevnext> <a
href=index.php?module=pengetahuan&offset=$prevoffset>Back</a></spa
n>";
}
else {
    echo "<span class=disabled>Back</span>"; //cetak halaman
tanpa link
}
//hitung jumlah halaman
$halaman = intval($baris/$limit); //Pembulatan

if ($baris%$limit){
    $halaman++;
}
for ($i=1;$i<=$halaman;$i++) {
    $newoffset = $limit * ($i-1);
    if($offset!=$newoffset){
        echo "<a
href=index.php?module=pengetahuan&offset=$newoffset>$i</a>";
        //cetak halaman
    }
    else {
        echo "<span class=current>".$i."</span>"; //cetak
halaman tanpa link
    }
}

//cek halaman akhir
if(!((($offset/$limit)+1==$halaman) && $halaman !=1)) {

    //jika bukan halaman terakhir maka berikan next
    $newoffset = $offset + $limit;
    echo "<span class=prevnext><a
href=index.php?module=pengetahuan&offset=$newoffset>Next</a>";
}
else {
    echo "<span class=disabled>Next</span>"; //cetak halaman
tanpa link
}

```

```

        echo "</div>";
    }else{
        echo "<br><b>Data Kosong !</b>";
    }
}
break;

case "tambahpengetahuan":
    echo "<br>
<br>
<form name=text_form method=POST
action='$_aksi?module=pengetahuan&act=input' onsubmit='return
Blank_TextField_Validator()'>
    <br><br><table class='table table-bordered'>
        <tr><td width=120>Penyakit</td><td><select class='form-
control' name='kode_penyakit' id='kode_penyakit'><option
value=''>- Pilih Penyakit -</option>;
        $hasil4 = mysql_query("SELECT * FROM penyakit order by
nama_penyakit");
        while($r4=mysql_fetch_array($hasil4)){
            echo "<option
value='$_r4[kode_penyakit]'>$_r4[nama_penyakit]</option>";
        }
        echo "</select></td></tr>
        <tr><td>Gejala</td><td><select class='form-control'
name='kode_gejala' id='kode_gejala'><option value=''>- Pilih
Gejala -</option>;
        $hasil4 = mysql_query("SELECT * FROM gejala order by
nama_gejala");
        while($r4=mysql_fetch_array($hasil4)){
            echo "<option
value='$_r4[kode_gejala]'>$_r4[nama_gejala]</option>";
        }
        echo "</select></td></tr>
        <tr><td>MB</td><td><input autocomplete='off'
placeholder='Masukkan MB' type=text class='form-control' name='mb'
size=15 ></td></tr>
        <tr><td>MD</td><td><input autocomplete='off'
placeholder='Masukkan MD' type=text class='form-control' name='md'
size=15 ></td></tr>
        <tr><td></td><td><input class='btn btn-success'
type=submit name=submit value='Simpan' >
            <input class='btn btn-danger' type=button name=batal
value='Batal'
onclick=\"window.location.href='?module=pengetahuan';\"></td></tr>
    </table></form>";
}
break;

case "editpengetahuan":
    $edit=mysql_query("SELECT * FROM basis_pengetahuan WHERE
kode_pengetahuan='$_GET[id]'");
    $r=mysql_fetch_array($edit);

    echo "<br>
<br>

```

```

<form name=text_form method=POST
action='$_aksi?module=pengetahuan&act=update' onsubmit='return
Blank_TextField_Validator()'>
    <input type=hidden name=id value='$_r[kode_pengetahuan]'>
    <br><br><table class='table table-bordered'>
        <tr><td width=120>Penyakit</td><td><select class='form-
control' name='kode_penyakit' id='kode_penyakit'>;
        $hasil4 = mysql_query("SELECT * FROM penyakit order by
nama_penyakit");
        while($r4=mysql_fetch_array($hasil4)){
            echo "<option value='$_r4[kode_penyakit]' ";
            if($_r[kode_penyakit]==$r4[kode_penyakit]) echo "selected";
            echo ">$_r4[nama_penyakit]</option>";
        }
        echo "</select></td></tr>
        <tr><td>Gejala</td><td><select class='form-control'
name='kode_gejala' id='kode_gejala'>;
        $hasil4 = mysql_query("SELECT * FROM gejala order by
nama_gejala");
        while($r4=mysql_fetch_array($hasil4)){
            echo "<option value='$_r4[kode_gejala]' ";
            if($_r[kode_gejala]==$r4[kode_gejala]) echo "selected";
            echo ">$_r4[nama_gejala]</option>";
        }
        echo "</select></td></tr>
        <tr><td>MB</td><td><input autocomplete='off'
placeholder='Masukkan MB' type=text class='form-control' name='mb'
size=15 value='$_r[mb]'></td></tr>
        <tr><td>MD</td><td><input autocomplete='off'
placeholder='Masukkan MD' type=text class='form-control' name='md'
size=15 value='$_r[md]'></td></tr>
        <tr><td></td><td><input class='btn btn-success'
type=submit name=submit value='Simpan' >
            <input class='btn btn-danger' type=button name=batal
value='Batal'
onclick=\"window.location.href='?module=pengetahuan';\"
        </td></tr>
    </table></form>";
    break;
}
?>
<?php } ?>

```