

**SISTEM PAKAR UNTUK MENDIAGNOSA PENYAKIT PADA AYAM  
DENGAN METODE CERTAINTY FACTOR BERBASIS ANDROID**

**(Skripsi)**

**Oleh :**

**PUJA PUTRI ABDULLAH**



**JURUSAN ILMU KOMPUTER  
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
UNIVERSITAS LAMPUNG**

**2016**

## **ABSTRACT**

### **EXPERT SYSTEM TO DIAGNOSE CHICKEN DISEASES WITH CERTAINTY FACTOR BASED ON ANDROID**

**By**

**PUJA PUTRI ABDULLAH**

The research was conducted to create an expert system that is able to diagnose chicken diseases based on the experts / specialists knowledge. This study uses a calculation method called Certainty Factor (CF) to calculate the level of expertise and the expert system created on Android mobile device platform. The research data consist of symptoms data, chicken diseases data, and data rules. Chicken disease data is limited to diseases caused by bacteria (bacterial), totaling 19 types of diseases with 78 kinds of symptoms and 184 types of rules. Inference method that used is forward chaining by searching for rules based on user answers in the form of check mark ( ). Users answers are processed according to the rules and calculated using certainty factor method. The search process is complete when it gets a conclusion as percentage of the value from the biggest to the smallest CF value of any disease. The test results showed that : (1) Functional testing using Black Box Equivalence Partitioning (EP) get results as expected as the test scenarios in each test class. (2) Diagnosis testing by comparing the results of manual calculations and systems calculations showed 99 percent was appropriate and well-run. (3) Questionnaire testing with 31 respondents were divided into three groups of respondents indicated; first respondents group consisting of specialists of chicken diseases gets 63.33 percent of an average value (categorized good), the second respondents group consisting of breeder and students of majoring in animal husbandry gets 77.71 percent of an average value (categorized good), and the third respondents group consisting of students of majoring computer science gets 82 percent of an average value (categorized very good).

**Key Words** : Expert System, Certainty Factor, CF method, Chicken Disease, Likert Scale, Android.

## ABSTRAK

### SISTEM PAKAR UNTUK MENDIAGNOSA PENYAKIT PADA AYAM DENGAN METODE CERTAINTY FACTOR BERBASIS ANDROID

Oleh

**PUJA PUTRI ABDULLAH**

Penelitian ini dilakukan untuk membuat suatu sistem pakar yang mampu mendiagnosa penyakit pada ayam berdasarkan pengetahuan yang diberikan langsung dari pakar/ahlinya. Penelitian ini menggunakan metode perhitungan *Certainty Factor* (CF) dalam menghitung tingkat kepakaran dan dibuat pada *mobile device platform* Android. Data penelitian ini terdiri dari data gejala dan data penyakit ayam, serta data aturan. Pada penelitian ini data penyakit ayam dibatasi pada penyakit yang disebabkan oleh bakteri (*bacterial*) yang berjumlah 19 jenis penyakit dengan 78 jenis gejala dan 184 jenis aturan. Metode inferensi yang digunakan yaitu *forward chaining* (runut maju) dengan menelusuri aturan-aturan berdasarkan jawaban yang diberikan pengguna dalam bentuk tanda *check* ( ). Jawaban pengguna kemudian diproses berdasarkan aturan (*rule*) dan dihitung menggunakan metode *certainty factor*. Proses penelusuran dilakukan sampai didapatkan suatu kesimpulan berupa presentase nilai CF terbesar hingga terkecil dari setiap penyakit. Hasil pengujian menunjukkan bahwa: (1) Pengujian fungsional dengan menggunakan metode *Black Box Equivalence Partitioning* (EP) mendapatkan hasil sesuai dengan yang diharapkan pada skenario uji di setiap kelas uji. (2) Pengujian diagnosa dengan membandingkan hasil perhitungan manual dan sistem 99 persen sudah sesuai dan berjalan baik. (3) Pengujian kuesioner dengan 31 orang responden yang dibagi menjadi tiga kelompok responden menunjukkan; kelompok responden pertama yang terdiri dari pakar penyakit ayam mendapatkan rata-rata nilai sebesar 63,33 persen (dikategorikan baik), kelompok responden kedua yang terdiri dari para peternak dan mahasiswa Peternakan mendapatkan rata-rata nilai sebesar 77,71 persen (dikategorikan baik), dan kelompok responden ketiga yang terdiri mahasiswa Ilmu Komputer mendapatkan rata-rata nilai sebesar 82 persen (dikategorikan sangat baik).

Kata Kunci : Sistem Pakar, Certainty Factor, Metode CF, Penyakit Ayam, Skala Likert, Android.

**SISTEM PAKAR UNTUK MENDIAGNOSA PENYAKIT PADA AYAM  
DENGAN METODE CERTAINTY FACTOR BERBASIS ANDROID**

**Oleh :**

**PUJA PUTRI ABDULLAH**

**Skripsi**

Sebagai Salah Satu Syarat untuk Memperoleh Gelar  
**SARJANA KOMPUTER**

pada

Jurusan Ilmu Komputer  
Fakultas Matematika dan Ilmu Pengetahuan Alam



**JURUSAN ILMU KOMPUTER  
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
UNIVERSITAS LAMPUNG**

**2016**

Judul Skripsi : **SISTEM PAKAR UNTUK MENDIAGNOSA  
PENYAKIT PADA AYAM DENGAN  
METODE CERTAINTY FACTOR  
BERBASIS ANDROID**

Nama Mahasiswa : **Puja Putri Abdullah**

Nomor Pokok Mahasiswa : 1217051051

Jurusan : Ilmu Komputer

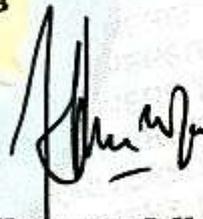
Fakultas : Matematika dan Ilmu Pengetahuan Alam

**MENYETUJUI**

1. Komisi Pembimbing

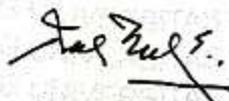


**Aristoteles, S.Si., M.Si.**  
NIP 19810521 200604 1 002



**Dr. Kusuma Adhianto, S.Pt., M.P.**  
NIP 19750611 200501 1 002

2. Ketua Jurusan Ilmu Komputer



**Dr. Ir. Kurnia Muludi, M.S.Sc.**  
NIP 19640616 198902 1 001

## MENGESAHKAN

### 1. Tim Penguji

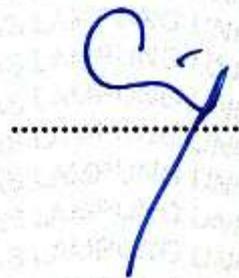
Ketua : **Aristoteles, S.Si., M.Si.**



Sekretaris : **Dr. Kusuma Adhianto, S.Pt., M.P.**



Penguji  
Bukan Pembimbing : **Rico Andrian, S.Si., M.Kom.**



### 2. Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam



**Prof. Warsito, S.Si., D.E.A., Ph.D.**  
NIP 19710212 199512 1 001

Tanggal Lulus Ujian Skripsi : **5 September 2016**

## PERNYATAAN

Saya yang bertanda tangan di bawah ini, menyatakan bahwa skripsi saya yang berjudul “Sistem Pakar Untuk Mendiagnosa Penyakit Pada Ayam Dengan Metode *Certainty Factor* Berbasis Android” merupakan karya saya sendiri dan bukan karya orang lain. Semua tulisan yang tertuang di skripsi ini telah mengikuti kaidah penulisan karya ilmiah Universitas Lampung. Apabila dikemudian hari terbukti skripsi saya merupakan hasil penjiplakan atau dibuat orang lain, maka saya bersedia menerima sanksi berupa pencabutan gelar yang telah saya terima.

Bandar Lampung, 30 Agustus 2016



**PUJA PUTRI ABDULLAH**

NPM. 1217051051

## RIWAYAT HIDUP



Penulis dilahirkan pada 31 Juli 1995 di Bukittinggi Sumatera Barat, sebagai anak pertama dari empat bersaudara dengan Ayah bernama Nuswanto dan Ibu bernama Eva Suryani.

Penulis menyelesaikan pendidikan Sekolah Dasar (SD) di SD Negeri 63 Surabayao Kec. Lubuk Basung Kab. Agam Sumatera Barat tahun 2006, menyelesaikan Sekolah Menengah Pertama (SMP) di Perguruan Diniyyah Puteri Kota Padang Panjang Sumatera Barat tahun 2009, kemudian melanjutkan jenjang Sekolah Menengah Atas (SMA) di SMA Negeri Agam Cendekia Kab. Agam Sumatera Barat dan lulus di tahun 2012.

Pada tahun 2012, penulis terdaftar sebagai mahasiswa Jurusan Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung. Adapun kegiatan yang dilakukan penulis selama menjadi mahasiswa antara lain:

1. Aktif sebagai aktivis Pers Mahasiswa di Unit Kegiatan Mahasiswa Fakultas (UKMF) Natural selama periode tahun 2012 sampai 2015.
2. Pernah mengikuti Karya Wisata Ilmiah (KWI) di Desa Sukaharjo IV Kabupaten Pringsewu pada bulan Januari sampai Februari 2013.
3. Aktif mengikuti berbagai seminar dan workshop seperti workshop writerpreneurship dan youngterpreneurship, seminar panas bumi, seminar teknologi, seminar kewirausahaan, dan berbagai lainnya.

4. Pernah mengikuti Pelatihan Jurnalistik Tingkat Lanjut (PJTL) yang diadakan oleh Lembaga Pers Mahasiswa (LPM) Teropong di Universitas Muhammadiyah Sumatera Utara (UMSU) pada bulan Mei 2014.
5. Pernah menjadi peserta Workshop Jurnalistik Mahasiswa (WJM) yang diadakan oleh Dewan Pers pada bulan Oktober 2014.
6. Melaksanakan kegiatan Kerja Praktik (KP) di Departemen IT (*Information Technology*) PT Tunas Dwipa Matra Bandar Lampung pada bulan Januari sampai Februari 2015.
7. Melaksanakan kegiatan Kuliah Kerja Nyata (KKN) di Desa Makarti Kecamatan Tumijajar Kabupaten Tulang Bawang Barat pada bulan Juli sampai September 2015.

## PERSEMBAHAN

*Teruntuk Papi Mamiku yang sangat kucintai, kupersembahkan skripsi ini*

*Terimakasih untuk kasih sayang, perhatian, pengorbanan, usaha, dukungan moril maupun materi, motivasi dan do'a-do'a yang tiada henti untuk kesuksesanku....*

*Teuntuk sahabat dan teman-teman tersayang,*

*Terimakasih untuk canda tawa, tangis dan perjuangan yang telah kita lewati bersama dan terima kasih untuk setiap rentetan kenangan yang telah terukir selama ini....*

## MOTTO

*“Dan bahwa manusia hanya memperoleh apa yang telah diusahakannya, dan sesungguhnya usahanya itu kelak akan diperlihatkan (kepadanya), kemudian akan diberi balasan kepadanya dengan balasan yang paling sempurna.”*

*(Q.S.An-Najm:39-41)*

*“Allah akan meninggikan orang-orang yang beriman di antaramu dan orang-orang yang diberi ilmu pengetahuan beberapa derajat.”*

*(Q.S.Al-Mujadalah:11)*

*“Ilmu lebih utama dari harta karena ilmu itu menjaga kamu, kalau harta kamulah yang menjaganya.”*

*(Ali bin Abi Thalib)*

*“Orang boleh pandai setinggi langit, tapi selama ia tidak menulis, ia akan hilang di dalam masyarakat dan dari sejarah. Menulis adalah bekerja untuk keabadian.”*

*(Pramoedya Ananta Toer)*

## SANWACANA

*Assalamualaikum wr. wb.*

Alhamdulillah, segala puji bagi Allah SWT yang telah melimpahkan rahmat, hidayah, kesehatan dan karunia-Nya sehingga penulis dapat menyelesaikan penulisan skripsi yang berjudul “Sistem Pakar Untuk Mendiagnosa Penyakit Pada Ayam Dengan Metode *Certainty Factor* Berbasis Android” dengan baik.

Terima kasih penulis ucapkan kepada semua pihak yang telah membantu dan berperan besar dalam penyusunan skripsi ini, seperti antara lain:

1. Kedua orang tua tercinta, Papi dan Mami, beserta keluarga besar yang selalu memberi do’a, motivasi dan kasih sayang yang tak terhingga.
2. Bapak Aristoteles, S.Si., M.Si. sebagai pembimbing utama dan pembimbing akademik, yang telah membimbing, memotivasi serta memberikan ide, kritik dan saran selama masa perkuliahan dan penyusunan skripsi sehingga penulis bisa sampai di tahap ini.
3. Bapak Dr. Kusuma Adhianto, S.Pt., M.P. sebagai pembimbing kedua yang telah membimbing dan memberikan bantuan, ide, kritik serta saran dalam penyusunan skripsi ini.
4. Bapak Rico Andrian, M.Kom. sebagai pembahas, yang telah memberikan komentar dan masukan yang bermanfaat untuk perbaikan dalam penyusunan skripsi ini.
5. Bapak Prof. Warsito, S.Si., D.E.A., Ph.D. selaku Dekan FMIPA Universitas Lampung.

6. Bapak Dr. Ir. Kurnia Muludi, M.S.Sc., selaku Ketua Jurusan Ilmu Komputer FMIPA Universitas Lampung.
7. Bapak Didik Kurniawan, S.Si., M.T., selaku Sekretaris Jurusan Ilmu Komputer FMIPA Universitas Lampung.
8. Bapak dan Ibu Dosen Jurusan Ilmu Komputer yang telah memberikan ilmu dan pengalaman hidup selama penulis menjadi mahasiswa.
9. Ibu Anita A.Md. dan Pak Irsan yang telah membantu segala urusan administrasi penulis di Jurusan Ilmu Komputer.
10. Keluarga Ilmu Komputer 2012 : Anita, Maya, Nafi, Riska, Concon, Erlina, Yuni, Deby, Kikin, Shandy, Eko, Ciwo, Owen, Indah, Febi, Lia, Ica, Nurul, Erika, Rani, Beta, Rizki, Haryati, Afriska, Rahman, Rayvicki, Abet, Dwi, Didin, Nikko, Roni, Bintang, Ridwan, Arif, Adit, Abi, Dipa, Ivan, Rio, Fildan, Varisa, Eka, Nila, Dian, Cindona, Taqiya, dan lain-lain. Terima kasih atas kebersamaannya selama ini.
11. Keluarga besar UKMF Natural yang telah memberikan berbagai pelajaran dan kenangan berharga selama proses berorganisasi.
12. Almamater Tercinta, Universitas Lampung yang telah memberikan penulis kesempatan untuk menempuh pendidikan perkuliahan S1 dengan baik.

## DAFTAR ISI

	Halaman
DAFTAR ISI .....	ix
DAFTAR GAMBAR .....	xiii
DAFTAR TABEL .....	xvi
BAB I PENDAHULUAN	
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	4
1.3 Batasan Masalah .....	4
1.4 Tujuan .....	5
1.5 Manfaat .....	5
BAB II TINJAUAN PUSTAKA	
2.1 Pengertian Kecerdasan Buatan .....	6
2.2 Sistem Pakar .....	7
2.1.1 Definisi Sistem Pakar .....	7
2.1.2 Ciri-ciri Sistem Pakar .....	7
2.1.3 Manfaat Sistem Pakar .....	8
2.1.4 Perbandingan Pakar dan Sistem Pakar .....	8
2.1.5 Komponen Sistem Pakar .....	9
2.1.6 Teknik Inferensi .....	11

2.3 Faktor Kepastian ( <i>Certainty Factor</i> ).....	12
2.4 Penyakit Pada Ayam.....	15
2.5 <i>Unified Modeling Language</i> (UML).....	17
2.6 Teknik Pengujian Perangkat Lunak.....	22
2.6.1 <i>Equivalence Partitioning</i> .....	23
2.6.2 Skala Likert.....	23
2.7 Sistem Operasi Android.....	25

### BAB III METODOLOGI PENELITIAN

3.1 Waktu dan Tempat Penelitian.....	26
3.2 Alat Pendukung.....	26
3.3 Tahapan Penelitian.....	27
3.3.1 Tahap Perumusan Masalah.....	28
3.3.2 Tahap Pengumpulan Data.....	28
3.3.3 Tahap Pengembangan Sistem.....	29
3.3.3.1 <i>Software Requirements Analysis</i> .....	29
3.3.3.2 <i>Design</i> .....	29
3.3.3.2.1 <i>Flowchart</i> .....	30
3.3.3.2.2 <i>Class Diagram</i> .....	31
3.3.3.2.3 <i>Sequence Diagram</i> .....	32
3.3.3.2.4 <i>Activity Diagram</i> .....	37
3.3.3.2.5 Rancangan Antarmuka ( <i>Interface</i> ).....	40
3.3.3.3 <i>Coding</i> .....	45
3.3.3.4 <i>Testing</i> .....	45
3.3.4 Analisis Hasil Pengujian.....	45

## BAB IV HASIL DAN PEMBAHASAN

4.1 Analisa Kebutuhan Data .....	46
4.2 Representasi Pengetahuan .....	47
4.3 Implementasi Sistem .....	47
4.4 Analisa Presentase Penyakit .....	48
4.5 Tampilan Sistem Pakar Penyakit Ayam .....	51
4.5.1 Tampilan Halaman <i>Splash Screen</i> .....	51
4.5.2 Tampilan Halaman Beranda .....	52
4.5.3 Tampilan Menu Utama .....	52
4.5.4 Tampilan Halaman Menu Diagnosis .....	53
4.5.5 Tampilan Halaman Pilih Gejala .....	54
4.5.6 Tampilan Halaman Hasil Diagnosis .....	55
4.5.7 Tampilan Halaman Informasi Penyakit .....	56
4.5.8 Tampilan Halaman Menu Data Penyakit .....	56
4.5.9 Tampilan Halaman Data Penyakit .....	57
4.5.10 Tampilan Halaman Menu Petunjuk .....	58
4.5.11 Tampilan Halaman Menu Tentang .....	58
4.6 Pengujian .....	59
4.6.1 Pengujian Internal .....	59
4.6.1.1 Pengujian Fungsional .....	59
4.6.1.1.1 Pengujian Versi Android .....	60
4.6.1.1.2 Pengujian Resolusi Layar dan Densitas Layar .....	61
4.6.1.1.3 Pengujian <i>User Interface</i> .....	62
4.6.1.1.4 Pengujian Fungsi dari Menu Aplikasi .....	63

4.6.1.2 Pengujian Diagnosa.....	66
4.6.2 Pengujian Eksternal.....	69
4.6.2.1 Analisa Hasil Kuesioner.....	72
BAB V PENUTUP	
5.1 Keimpulan.....	83
5.2 Saran.....	84
DAFTAR PUSTAKA.....	85
LAMPIRAN	
LAMPIRAN PENGUJIAN	

## DAFTAR GAMBAR

	Halaman
Gambar 2.1 Arsitektur Sistem Pakar.....	10
Gambar 3.1 Tahapan Penelitian.....	27
Gambar 3.2 Model <i>Use Case Diagram</i> .....	29
Gambar 3.3 <i>Flowchart</i> Sistem Pakar dan Pengguna.....	30
Gambar 3.4 Model <i>Class Diagram</i> .....	31
Gambar 3.5 <i>Sequence Diagram</i> Proses Melakukan Diagnosa.....	33
Gambar 3.6 <i>Sequence Diagram</i> Menampilkan Data Penyakit.....	34
Gambar 3.7 <i>Sequence Diagram</i> Mengakses Menu Bantuan.....	35
Gambar 3.8 <i>Sequence Diagram</i> Mengakses Menu Tentang.....	36
Gambar 3.9 <i>Activity Diagram</i> Proses Diagnosa Penyakit.....	37
Gambar 3.10 <i>Activity Diagram</i> Proses Mengakses Data Penyakit.....	38
Gambar 3.11 <i>Activity Diagram</i> Proses Mengakses Menu Bantuan.....	39
Gambar 3.12 <i>Activity Diagram</i> Proses Mengakses Menu Tentang.....	39
Gambar 3.13 Rancangan Halaman <i>Splash Screen</i> .....	40
Gambar 3.14 Rancangan Halaman Menu Utama.....	41
Gambar 3.15 Rancangan Halaman Beranda.....	41
Gambar 3.16 Rancangan Halaman Diagnosis.....	42
Gambar 3.17 Rancangan Halaman Pilih Gejala.....	42

Gambar 3.18 Rancangan Halaman Hasil Diagnosis.....	43
Gambar 3.19 Rancangan Halaman Data Penyakit.....	43
Gambar 3.20 Rancangan Halaman Bantuan.....	44
Gambar 3.21 Rancangan Halaman Tentang.....	44
Gambar 4.1 Tampilan Halaman <i>Splash Screen</i> .....	51
Gambar 4.2 Tampilan Halaman Beranda.....	52
Gambar 4.3 Tampilan Menu Utama.....	53
Gambar 4.4 Tampilan Halaman Menu Diagnosis.....	54
Gambar 4.5 Tampilan Halaman Pilih Gejala.....	54
Gambar 4.6 Tampilan Halaman Hasil Diagnosis.....	55
Gambar 4.7 Tampilan Halaman Informasi Penyakit.....	56
Gambar 4.8 Tampilan Halaman Menu Data Penyakit.....	57
Gambar 4.9 Tampilan Halaman Data Penyakit.....	57
Gambar 4.10 Tampilan Halaman Menu Petunjuk.....	58
Gambar 4.11 Tampilan Halaman Menu Tentang.....	59
Gambar 4.12 Grafik Hasil Pernyataan 1.....	74
Gambar 4.13 Grafik Hasil Pernyataan 2.....	75
Gambar 4.14 Grafik Hasil Pernyataan 3.....	76
Gambar 4.15 Grafik Hasil Pernyataan 4.....	77
Gambar 4.16 Grafik Hasil Pernyataan 5.....	78
Gambar 4.17 Grafik Hasil Pernyataan 6.....	79
Gambar 4.18 Grafik Hasil Pernyataan 7.....	80

Gambar 4.19 Grafik Hasil Pernyataan 8 .....81

## DAFTAR TABEL

	Halaman
Tabel 2.1 Perbandingan Pakar dan Sistem Pakar .....	9
Tabel 2.2 Certain Term CF .....	13
Tabel 2.3 Data Penyakit Ayam Penyebab Bakteri .....	16
Tabel 2.4 Simbol <i>Use Case Diagram</i> .....	18
Tabel 2.4 (Lanjutan) Simbol <i>Usecase Diagram</i> .....	19
Tabel 2.5 Simbol <i>Class Diagram</i> .....	20
Tabel 2.6 Simbol <i>Activity Diagram</i> .....	21
Tabel 2.7 Simbol <i>Sequence Diagram</i> .....	22
Tabel 4.1 Pengujian Versi Android .....	60
Tabel 4.2 Pengujian Resolusi Layar dan Densitas Layar .....	61
Tabel 4.2 (Lanjutan) Pengujian Resolusi Layar dan Densitas Layar .....	62
Tabel 4.3 Pengujian <i>User Interface</i> .....	63
Tabel 4.4 Pengujian Fungsi dari Menu Aplikasi .....	64
Tabel 4.4 (Lanjutan) Pengujian Fungsi dari Menu Aplikasi .....	65
Tabel 4.4 (Lanjutan) Pengujian Fungsi dari Menu Aplikasi .....	66
Tabel 4.5 Pengujian Diagnosa Sistem Pakar Penyakit Ayam .....	67
Tabel 4.5 (Lanjutan) Pengujian Diagnosa Sistem Pakar Penyakit Ayam .....	68
Tabel 4.6 Hasil Penilaian Responden Pakar Penyakit Ayam Terhadap Kuesioner Pengujian Sistem Pakar (Kelompok Responden I) .....	70
Tabel 4.7 Hasil Penilaian Responden Peternak dan Mahasiswa Peternakan Terhadap Kuesioner Pengujian Sistem Pakar (Kelompok Responden II) .....	71

Tabel 4.8 Hasil Penilaian Responden Mahasiswa Ilmu Komputer Terhadap Kuesioner Pengujian Sistem Pakar (Kelompok Responden III).....	72
Tabel 4.9 Kriteria Penilaian Responden.....	73

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Menurut data dari Direktorat Jenderal Peternakan dan Kesehatan Hewan Kementerian Pertanian (Ditjenak Kementan) pada tahun 2015 jenis ayam buras, ayam ras petelur dan ayam ras pedaging merupakan ternak dengan populasi terbanyak yang diusahakan oleh pengusaha ataupun peternak di Indonesia.

Dalam catatan sejarah perunggasan di Indonesia, pemeliharaan ayam pada musim kemarau, musim penghujan, maupun musim pancaroba sama-sama menguras energi bagi peternak. Pasalnya pada ketiga musim tersebut kemungkinan sebuah penyakit akan menyerang ternak ayam akan semakin tinggi.

Pada musim penghujan dimana kelembaban tinggi, memicu hampir semua bibit penyakit untuk tumbuh subur termasuk kejadian *mikotoksikosis* maupun infeksi cacing. Pada musim kemarau, infeksi saluran pernapasan dan *heat stress* tak bisa dielakkan lagi. Sedangkan pada musim pancaroba kemungkinan penyakit datang menjadi semakin tinggi karena pada musim tersebut ayam lebih kesulitan untuk beradaptasi. Oleh karena itu, untuk dapat mencegah ataupun menangani suatu penyakit diperlukan diagnosa dari para ahli, seperti dokter hewan.

Di Provinsi Lampung, keberadaan dokter hewan masih dinilai sangat minim dibandingkan dengan populasi ternak ayam yang tinggi. Kepala Dinas Peternakan dan Kesehatan Hewan Provinsi Lampung, Dessy Desmaniar Romas mengatakan, pada tahun 2014 pihaknya hanya memiliki 40 dokter hewan. Dari jumlah itu, 12 orang di antaranya sudah bersatus sebagai Pegawai Negeri Sipil (PNS). Sedangkan 28 orang lainnya masih tercatat sebagai Tenaga Harian Lepas (THL).

Proses diagnosa suatu penyakit pada ternak seharusnya dilakukan oleh seorang ahli di bidang tersebut. Namun, keterbatasan jumlah dan sulitnya berinteraksi langsung dengan tenaga ahli seperti dokter hewan, membuat sebagian besar peternak menangani sendiri permasalahan kesehatan dan penyakit ternaknya. Kurangnya pengetahuan tentang penanganan penyakit pada hewan ternak, bisa saja mengakibatkan peternak membuat kesalahan dalam mendiagnosa dan memberikan obat kepada hewan yang sedang sakit. Oleh karena itu, dibutuhkan suatu sistem pakar yang mampu mendiagnosa penyakit pada ternak ayam dengan pengetahuan yang berasal langsung dari ahlinya.

Sistem Pakar (*Expert System*) merupakan salah satu bagian dari ilmu kecerdasan buatan (*artificial intelligent*) yang cukup berkembang akhir-akhir ini. Berkat kemajuan teknologi yang terus mengalami kepesatan, sistem pakar sudah beralih menjadi suatu kebutuhan disaat mengalami kesulitan untuk menemui para pakar secara langsung.

Aplikasi sistem pakar dapat mewakili seorang pakar yang ahli dibidangnya untuk memberikan solusi terhadap permasalahan yang ada. Dengan aplikasi tersebut, pengetahuan pakar dapat disimpan tanpa batas waktu. Selain itu, sistem pakar

juga dapat meningkatkan produktifitas kerja, menghemat waktu dalam menyelesaikan masalah, penyederhanaan solusi untuk kasus-kasus yang kompleks dan berulang-ulang. Implementasi sistem pakar dapat dibangun dalam berbagai bentuk, seperti berbasis web ataupun *mobile*.

Pada era ini perkembangan teknologi berkembang sangat pesat, salah satunya perkembangan *Operation System* (OS) Android pada *handphone*. Menurut data dari Waiwai Marketing, jumlah pengguna Android di Indonesia mencapai angka 94 persen sedangkan pengguna OS lain seperti iOS (iPhone OS) hanya berkisar 6 persen. Hal ini menandakan bahwa Android merupakan *platform* yang paling banyak digunakan masyarakat di Indonesia. Selain itu Android dapat dimanfaatkan dalam proses implementasi aplikasi sistem pakar karena Android merupakan *platform* terbuka bagi pengembang untuk menciptakan berbagai aplikasi bisa dipakai oleh bermacam piranti bergerak.

Berdasarkan sebuah jurnal yang berjudul “Aplikasi Sistem Pakar Diagnosa Kerusakan Pada Sepeda Motor 4 Tak Menggunakan Metode *Certainty Factor* Berbasis Android” (Aryawan dkk, 2013), telah dilakukan penelitian tentang aplikasi sistem pakar untuk mendiagnosa sepeda motor berbasis android. Dalam hal ini penulis menggunakan metode perhitungan *certainty factor* dalam membangun sistem.

Dalam jurnal lain yang berjudul “Sistem Pakar Diagnosa Awal Penyakit Kulit Pada Sapi Bali dengan Menggunakan Metode *Forward Chaining* dan *Certainty Factor*” (Supartha dkk, 2014) telah dibangun sebuah sistem pakar diagnosa awal penyakit kulit pada sapi berbasis web. Dalam penelitian ini metode *forward*

*chaining* digunakan untuk pengecekan aturan (*rule*) berdasarkan inputan gejala dari pengguna dan diolah dengan basis pengetahuan. Sedangkan metode *certainty factor* digunakan untuk menghitung keakuratan hasil diagnosa penyakit.

Berdasarkan beberapa jurnal tersebut, maka akan dibuat “**Sistem Pakar Untuk Mendiagnosa Penyakit Pada Ayam Dengan Menggunakan Metode *Certainty Factor* Berbasis Android**”. Data-data yang dibutuhkan seperti nama penyakit, gejala penyakit, dan cara penanganannya didapatkan langsung dari ahli/pakar yang memahami ilmu tentang penyakit ayam serta buku dan jurnal yang mendukung.

## **1.2 Rumusan Masalah**

Berdasarkan latar belakang yang telah diuraikan, maka rumusan masalah penelitian ini adalah:

1. Bagaimana membuat sistem pakar untuk mendiagnosa penyakit pada ayam dengan menggunakan metode *certainty factor*?
2. Bagaimana membuat aplikasi sistem pakar berbasis android?

## **1.3 Batasan Masalah**

Batasan masalah dari analisis pengujian ini adalah sebagai berikut:

1. Sistem pakar yang dibuat dikhususkan untuk mendiagnosa penyakit ayam yang disebabkan oleh bakteri (*bacterial*).
2. Aplikasi sistem pakar dibuat pada perangkat *mobile device platform* android.
3. Sistem pakar dibuat dengan menggunakan metode perhitungan *certainty factor*.

#### **1.4 Tujuan Penelitian**

Tujuan dari penelitian ini adalah sebagai berikut:

1. Membuat suatu sistem pakar yang mampu mendiagnosa penyakit pada ayam yang disebabkan oleh bakteri.
2. Membuat aplikasi yang dapat memberikan pengetahuan tambahan kepada masyarakat khususnya peternak ayam tentang penyakit dan kesehatan pada ayam.

#### **1.5 Manfaat Penelitian**

Manfaat dari penelitian ini adalah membantu peternak ataupun pengusaha ayam dalam mendapatkan informasi penyakit ayam dan cara penanganannya berdasarkan gejala yang terlihat dengan menggunakan sistem pakar berbasis android.

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Pengertian Kecerdasan Buatan**

Kecerdasan buatan berasal dari bahasa Inggris “*Artificial Intelligence*” atau singkatan AI, yaitu *intelligence* adalah kata sifat yang berarti cerdas, sedangkan *artificial* artinya buatan. Kecerdasan buatan yang dimaksud di sini merujuk pada mesin yang mampu berpikir, menimbang tindakan yang akan diambil dan mampu mengambil keputusan seperti yang dilakukan oleh manusia (Sutojo dkk, 2011).

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

## **2.2 Sistem Pakar**

### **2.2.1 Definisi Sistem Pakar**

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

### **2.2.2 Ciri-ciri Sistem Pakar**

Menurut Sutojo (2011), ciri-ciri sistem pakar adalah :

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

### **2.2.3 Manfaat Sistem Pakar**

Menurut Sutojo (2011) sistem pakar menjadi populer karena sangat banyak kemampuan dan manfaat yang diberikannya, di antaranya :

1. Meningkatkan produktivitas, karena sistem pakar dapat bekerja lebih cepat daripada manusia.
2. Membuat seorang yang awam bekerja seperti layaknya seorang pakar.
3. Meningkatkan kualitas, dengan memberi nasihat yang konsisten dan mengurangi kesalahan.
4. Mampu menangkap pengetahuan dan kepakaran seseorang.
5. Dapat beroperasi di lingkungan yang berbahaya.
6. Memudahkan akses pengetahuan seorang pakar.
7. Handal, sistem pakar tidak pernah menjadi bosan dan kelelahan atau sakit.
8. Meningkatkan kapabilitas sistem komputer. Integrasi sistem pakar dengan sistem komputer lain membuat sistem lebih efektif dan mencakup lebih banyak aplikasi.
9. Mampu bekerja dengan informasi yang tidak lengkap atau tidak pasti.
10. Bisa digunakan sebagai media pelengkap dalam pelatihan. Pengguna pemula yang bekerja dengan sistem pakar akan menjadi lebih berpengalaman karena adanya fasilitas penjelas yang berfungsi sebagai guru.

### **2.2.4 Perbandingan Pakar dan Sistem Pakar**

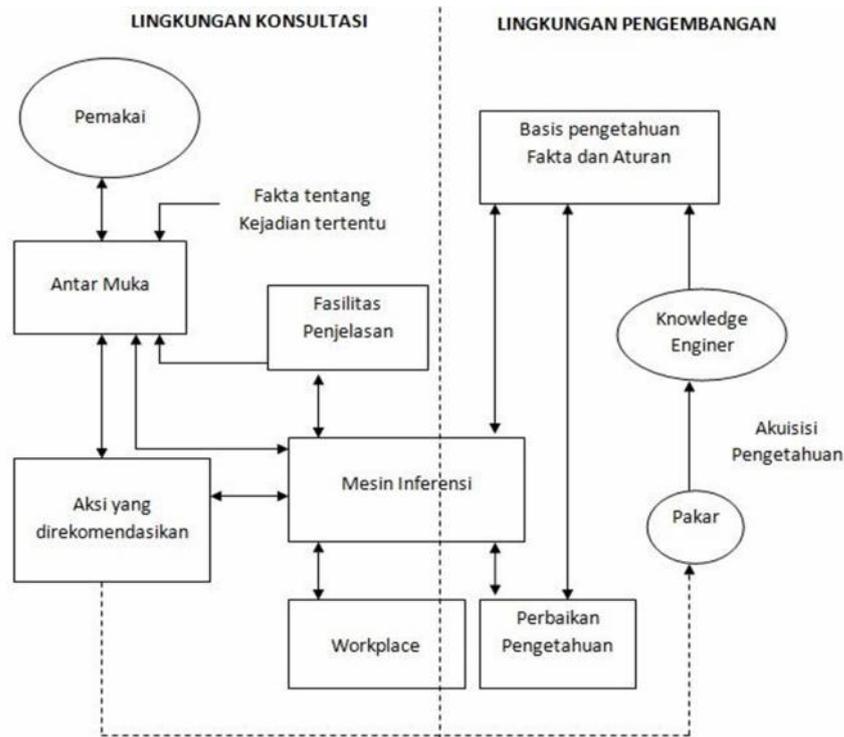
Perbandingan kemampuan antara seorang pakar dengan sistem pakar disajikan pada Tabel 2.1.

Tabel 2.1. Perbandingan Pakar dan Sistem Pakar

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

### 2.2.5 Komponen Sistem Pakar

Menurut Siswanto (2010), sistem pakar disusun oleh dua bagian utama, yaitu bagian lingkungan pengembangan (*development environment*) dan lingkungan konsultasi (*consultation environment*). Lingkungan pengembang digunakan oleh pembuat sistem pakar untuk membangun komponen-komponennya dan memperkenalkan pengetahuan ke dalam *knowledge base* (basis pengetahuan). Sedangkan lingkungan konsultasi digunakan oleh pengguna untuk berkonsultasi sehingga pengguna mendapatkan pengetahuan dan nasihat dari sistem pakar layaknya berkonsultasi dengan seorang pakar. Komponen-komponen sistem pakar (Rachmawati dkk, 2012) dapat dilihat pada Gambar 2.1.



Gambar 2.1 Arsitektur Sistem Pakar

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

1. *Knowledge Base* (Basis Pengetahuan)

Basis pengetahuan merupakan hasil *akuisisi* dan *representasi* pengetahuan dari seorang pakar. Basis pengetahuan berisi pengetahuan-pengetahuan dalam penyelesaian masalah. Basis pengetahuan terdiri dari dua elemen dasar yaitu fakta dan *rule* atau aturan.

2. *Inference Engine* (Mesin Inferensi)

Mesin inferensi adalah sebuah program yang berisi metodologi yang digunakan untuk melakukan penalaran terhadap informasi-informasi dalam basis pengetahuan untuk memformulasikan konklusi.

### 3. *User Interface* (Antar Muka Pengguna)

*User interface* adalah penghubung antar program sistem pakar dengan pengguna yang dapat dihubungkan via *desktop* ataupun *mobile*. Antarmuka digunakan sebagai media komunikasi antara pengguna dan sistem pakar.

#### 2.2.6 Teknik Inferensi

Mesin inferensi adalah bagian yang mengandung mekanisme fungsi berpikir dan pola-pola penalaran sistem yang digunakan seorang pakar. Mekanisme ini akan menganalisa suatu masalah tertentu dan selanjutnya mencari kesimpulan yang terbaik (Siswanto, 2010).

Menurut Sutojo dkk (2011) dan Siswanto (2010) Ada dua teknik penalaran (*inference*) yaitu sebagai berikut:

1. *Forward Chaining* (Pelacakan ke depan): Teknik ini memulai pencarian dengan fakta yang diketahui untuk menguji kebenaran hipotesa, kemudian mencocokkan fakta tersebut dengan bagian IF dari *rule* IF-THEN. Teknik ini cocok digunakan untuk menangani masalah peramalan (*prognosis*) dan pengendalian (*controlling*).
2. *Backward Chaining* (Pelacakan ke belakang): Teknik ini memulai pencarian dari kesimpulan (*goal*) dengan mencari sekumpulan hipotesa-hipotesa yang mendukung menuju fakta-fakta yang mendukung sekumpulan hipotesa tersebut.

### 2.3 Faktor Kepastian (*Certainty Factor*)

Teori *Certainty Factor* (CF) diusulkan oleh Shorliffe dan Buchanan pada 1975 untuk mengakomodasikan ketidakpastian pemikiran (*inexact reasoning*) seorang pakar. Untuk mengakomodasi hal ini digunakan untuk menggambarkan tingkat keyakinan pakar terhadap masalah yang sedang dihadapi (Sutojo dkk, 2011).

Dalam mengekspresikan derajat keyakinan, *certainty theory* menggunakan suatu nilai yang disebut *certainty factor* untuk mengasumsikan derajat keyakinan seorang pakar terhadap suatu data. Konsep ini kemudian diformulasikan dalam rumus dasar pada Persamaan (1) sebagai berikut.

$$CF_{\text{combine}}(CF_1, CF_2) = \begin{cases} CF_1 + CF_2(1 - CF_1) & \text{Kedua-duanya} > 0 \\ \frac{CF_1 + CF_2}{1 - \min(|CF_1|, |CF_2|)} & \text{Salah satu} < 0 \\ CF_1 + CF_2(1 + CF_1) & \text{Kedua-duanya} < 0 \end{cases} \dots\dots\dots(1)$$

(Kusrini, 2008).

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

Menurut Kusrini (2008), ada 2 macam faktor kepastian yang digunakan, yaitu:

1. Faktor kepastian yang diisikan oleh pakar bersama dengan aturan.
2. Faktor kepastian yang diberikan oleh pengguna.

a. Menggali hasil wawancara dengan pakar

Yaitu dengan menggali dari hasil wawancara dengan pakar yang bersangkutan.

Nilai CF (*rule*) didapat dari interpretasi "term" dari pakar menjadi nilai sebuah

MD/MB tertentu seperti terlihat pada Tabel 2.2.

Tabel 2.2 *Certain Term CF*

Certain Term	MD/MB
Tidak Tahu	0.00 - 0.29
Mungkin	0.30 - 0.49
Kemungkinan Besar	0.50 - 0.69
Hampir Pasti	0.70 - 0.89
Pasti	0.90 - 1.00

Contoh:

Jika batuk dan panas, maka "Hampir Pasti" penyakitnya adalah influenza.

Rule: **IF** (batuk **AND** panas) **THEN** penyakit = influenza (CF = 0,8)

b. Menggunakan metode perhitungan 'Net Belief'

Faktor Kepastian menunjukkan ukuran kepastian terhadap suatu fakta aturan.

Notasi Faktor Kepastian ditunjukkan pada Persamaan (2), (3), dan (4) berikut.

$$CF(H,E) = MB(H,E) - MD(H,E) \dots\dots\dots(2)$$

$$MB(H,E) = \begin{cases} 1 & P(H) = 1 \\ \frac{\max(P[(H|E),P(H)]-P(H)}{\max[1,0]-P(H)} & \text{lainnya} \end{cases} \dots\dots\dots(3)$$

$$MD(H,E) = \begin{cases} 1 & P(H) = 1 \\ \frac{\min(P[(H|E),P(H)]-P(H)}{\min[1,0]-P(H)} & \text{lainnya} \end{cases} \dots\dots\dots(4)$$

Keterangan :

$CF(H,E)$  : *Certainty factor* dari hipotesis H yang dipengaruhi oleh gejala (*evidence*) E. Besarnya CF berkisar antara 0 sampai dengan 1.

Nilai 0 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.

$P(H)$  : Probabilitas kebenaran hipotesis H

$P(H|E)$  : Probabilitas bahwa H benar karena fakta E

H : Hipotesa (dugaan)

E : Evidence (peristiwa atau fakta)

Contoh:

Seorang pakar spesialis penyakit mata menyatakan bahwa probabilitas seseorang berpenyakit katarak adalah 0,02. Dari data lapangan menunjukkan bahwa 100 orang penderita penyakit katarak, 40 orang memiliki gejala pandangan kabur. Dengan menganggap H = Katarak dan E = Pandangan Kabur, hitung faktor kepastian bahwa katarak disebabkan oleh kulup berminyak.

Jawab:

$$P(\text{Katarak}) = 0,02$$

$$P(\text{Katarak} | \text{Pandangan kabur}) = 40/100 = 0,4$$

$$MB(H,E) = \frac{\max(P[(H|E),P(H)]-P(H)}{\max[1,0]-P(H)}$$

$$MB(H,E) = \frac{\max [0,4,0,02] - 0.02}{1 - 0,02} = \frac{0,4 - 0.02}{1 - 0,02} = 0,39$$

$$MD(H,E) = \frac{\min (P[(H|E),P(H)] - P(H))}{\min[1,0] - P(H)}$$

$$MD(H,E) = \frac{\min [0,4,0,02] - 0.02}{0 - 0,02} = \frac{0,02 - 0.02}{- 0,02} = 0$$

$$CF = 0,39 - 0 = 0,39$$

Rule: **IF** (Gejala = Pandangan Kabur) **THEN** Penyakit = Katarak (CF = 0,39).

## 2.4 Penyakit Pada Ayam

Penyakit ayam merupakan kendala utama pada peternakan ayam intensif di lingkungan tropis seperti di Indonesia. Kerugian ekonomi akibat penyakit, khususnya penyakit menular, dapat digambarkan dalam bentuk kematian, meskipun yang lebih sering terjadi adalah bentuk penurunan produksi seperti pada kelompok penyakit pernafasan.

Secara garis besar, penyakit yang menyerang ternak ayam dapat ditimbulkan oleh 2 penyebab (Murtidjo, 1992), yaitu:

1. Penyebab hidup (*Living agent*), seperti:
  - a. Jasad renik (*mikroba*) : bakteri, virus, kapang, riketsia, protozoa binatang bersel satu.
  - b. Cacing : cacing bulat, pipih, dan cacing pita.
  - c. *Insekta* : kutu, lalat dan lain-lain.
2. Penyebab tidak hidup (*Nonliving agent*), seperti cekaman temperatur tinggi atau rendah, keracunan zat kimia atau nabati, defisiensi makanan dan kelebihan unsur makanan.

Menurut Tabbu (2015), secara umum terdapat 10 jenis penyakit pada ayam yang disebabkan oleh bakteri, seperti yang terlihat pada Tabel 2.3.

Tabel 2.3 Data Penyakit Ayam Penyebab Bakteri

No	Nama Penyakit	Bentuk Penyakit	Nama Bakteri
1	<i>Infectious coryza</i> (snot)		<i>Haemophilus paragallinarum</i>
2	<i>Fowl cholera</i> (kolera unggas)		<i>Pasteurella multocida</i>
3	<i>Kolibasilosis</i>	Kematian embrio, infeksi <i>Yolk Sac</i> , dan <i>Omfalitis</i> <i>Koliseptisemia</i>	<i>Escherichia coli</i>
4	<i>Salmonellosis</i>	<i>Pulorum</i> <i>Fowl typhoid</i> Infeksi <i>paratifoid</i> <i>Arizonosis</i>	<i>Salmonella</i>
5	<i>Klostridial</i>	<i>Enteritis ulseratifa</i> <i>Enteritis nekrotikan</i> <i>Dermatitis gangrenosa</i> <i>Botulisme</i>	<i>Clostridium sp.</i>
6	<i>Stafilokokosis</i>		<i>Staphylococcus aureus</i>
7	<i>Streptokokosis</i>		<i>Streptococcus sp.</i>
8	<i>Tuberculosis</i>		<i>Mycobacterium avium</i>
9	Infeksi <i>pseudomonas</i>		<i>Pseudomonas aeruginosa</i>
10	<i>Mikoplasmosis</i>	Infeksi <i>Mycoplasma gallisepticum</i> (CRD) Infeksi <i>Mycoplasma synoviae</i> (bentuk sinovitis) Infeksi <i>Mycoplasma synoviae</i> (bentuk pernapasan)	<i>Mycoplasma</i>

## **2.5 Unified Modeling Language (UML)**

Menurut Nugroho (2010), UML (*Unified Modeling Language*) adalah ‘bahasa’ pemodelan untuk sistem atau perangkat lunak yang berparadigma ‘berorientasi objek’. Pemodelan (modeling) sesungguhnya digunakan untuk penyederhanaan permasalahan-permasalahan yang kompleks sedemikian rupa sehingga lebih mudah dipelajari dan dipahami.

Menurut Widodo (2011), beberapa literatur menyebutkan bahwa UML menyediakan sembilan jenis diagram, yang lain menyebutkan delapan karena ada beberapa diagram yang digabung, misalnya diagram komunikasi, diagram urutan dan diagram pewaktuan digabung menjadi diagram interaksi. Namun demikian model-model itu dapat dikelompokkan berdasarkan sifatnya yaitu statis atau dinamis. Jenis diagram itu antara lain:

- A. Diagram Kelas (*Class Diagram*)
- B. Diagram Paket (*Package Diagram*)
- C. Diagram Use-Case (*Usecase Diagram*)
- D. Diagram Interaksi dan Sequence (*Sequence Diagram*)
- E. Diagram Komunikasi (*Communication Diagram*)
- F. Diagram Statechart (*Statechart Diagram*)
- G. Diagram Aktivitas (*Activity Diagram*)
- H. Diagram Komponen (*Component Diagram*)
- I. Diagram Deployment (*Deployment Diagram*)

Berikut penjelasan dari beberapa diagram yang digunakan pada penelitian ini.

### A. Diagram Use-Case (*Usecase Diagram*)

*Usecase diagram* menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah *usecase* merepresentasikan sebuah interaksi antara aktor dengan sistem. Seorang/sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu.

Tabel 2.4 Simbol *Usecase Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
2		<i>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> ).
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.

(Lanjutan) Tabel 2.4 Simbol *Usecase Diagram*

NO	GAMBAR	NAMA	KETERANGAN
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		<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.

## B. Diagram Kelas (*Class Diagram*)

*Class* adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi).

*Class diagram* menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain. *Class* memiliki tiga area pokok, yaitu nama (dan *stereotype*), atribut, dan metoda. Atribut dan metoda dapat memiliki salah satu sifat berikut :

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

Tabel 2.5 Simbol *Class Diagram*

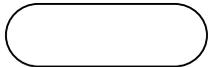
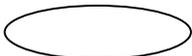
NO	GAMBAR	NAMA	KETERANGAN
1		<i>Generalization</i>	Hubungan dimana objek anak ( <i>descendent</i> ) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk ( <i>ancestor</i> ).
2		<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
3		<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
4		<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor
5		<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
6		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri ( <i>independent</i> ) akan memengaruhi elemen yang bergantung padanya elemen yang tidak mandiri
7		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya

### C. Diagram Aktivitas (*Activity Diagram*)

*Activity diagram* menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi.

*Activity diagram* dapat dibagi menjadi beberapa *object swimlane* untuk menggambarkan objek mana yang bertanggung jawab untuk aktivitas tertentu.

Tabel 2.6 Simbol *Activity Diagram*

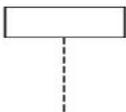
NO	GAMBAR	NAMA	KETERANGAN
1		<i>Activity</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain
2		<i>Action</i>	State dari sistem yang mencerminkan eksekusi dari suatu aksi
3		<i>Initial Node</i>	Bagaimana objek dibentuk atau diawali.
4		<i>Activity Final Node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

#### D. Diagram Interaksi dan Sequence (*Sequence Diagram*)

*Sequence diagram* menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence diagram* terdiri atas dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait).

Masing-masing objek, termasuk aktor, memiliki *lifeline* vertikal. *Message* digambarkan sebagai garis berpanah dari satu objek ke objek lainnya. *Activation bar* menunjukkan lamanya eksekusi sebuah proses, biasanya diawali dengan diterimanya sebuah *message*. Untuk objek-objek yang memiliki sifat khusus, standar UML mendefinisikan *icon* khusus untuk objek *boundary*, *controller* dan *persistent entity*.

Tabel 2.7 Simbol *Sequence Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		<i>LifeLine</i>	Objek <i>entity</i> , 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

## 2.6 Teknik Pengujian Perangkat Lunak

Ada dua macam pendekatan kasus uji yaitu *white-box* dan *black-box*. Pendekatan *white-box* adalah pengujian untuk memperlihatkan cara kerja dari produk secara rinci sesuai dengan spesifikasinya akan dites dengan menyediakan kasus uji yang akan mengerjakan kumpulan kondisi dan pengulangan secara spesifik. Sehingga melalui penggunaan metode ini akan dapat memperoleh kasus uji yang menjamin bahwa semua jalur independen pada suatu model telah digunakan minimal satu kali, penggunaan keputusan logis pada sisi benar dan salah, pengekseskuan semua *loop* dalam batasan dan batas operasional perekayasa, serta penggunaan struktur data internal guna menjamin validitasnya (Pressman, 2010).

Pendekatan *black-box* merupakan pendekatan pengujian untuk mengetahui apakah semua fungsi perangkat lunak telah berjalan semestinya sesuai dengan kebutuhan fungsional yang telah didefinisikan (Jiang, 2012). Kasus ini bertujuan untuk menunjukkan fungsi perangkat lunak tentang cara beroperasinya. Teknik

pengujian ini berfokus pada domain informasi dari perangkat lunak, yaitu melakukan kasus uji dengan mempartisi domain *input* dan *output* program.

Metode *black-box* memungkinkan perancang perangkat lunak mendapatkan serangkaian kondisi *input* yang sepenuhnya menggunakan semua persyaratan fungsional untuk suatu program. Pengujian ini berusaha menemukan kesalahan dalam kategori fungsi-fungsi yang tidak benar atau hilang, kesalahan *interface*, kesalahan dalam struktur data atau akses basis data eksternal, kesalahan kinerja, dan inisialisasi dan kesalahan terminal (Pressman, 2010).

### **2.6.1 Equivalence Partitioning**

*Equivalence Partitioning* (EP) merupakan metode *black box testing* yang membagi domain masukan dari program kedalam kelas-kelas sehingga *test case* dapat diperoleh. *Equivalence Partitioning* berusaha untuk mendefinisikan kasus uji yang menemukan sejumlah jenis kesalahan, dan mengurangi jumlah kasus uji yang harus dibuat. Kasus uji yang didesain untuk *Equivalence Partitioning* berdasarkan pada evaluasi dari kelas ekuivalensi untuk kondisi masukan yang menggambarkan kumpulan keadaan yang valid atau tidak. Kondisi masukan dapat berupa spesifikasi nilai numerik, kisaran nilai, kumpulan nilai yang berhubungan atau kondisi *Boolean* (Pressman, 2001).

### **2.6.2 Skala Likert**

Menurut Likert dalam Azwar S (2011), sikap dapat diukur dengan metode rating yang dijumlahkan merupakan metode penskalaan pernyataan sikap yang menggunakan distribusi respons sebagai dasar penentuan nilai skalanya. Nilai

skala setiap pernyataan tidak ditentukan oleh derajat *favourable* nya masing-masing akan tetapi ditentukan oleh distribusi *respons* setuju dan tidak setuju dari sekelompok responden yang bertindak sebagai kelompok uji coba (*pilot study*).

Skala Likert, yaitu skala yang berisi lima tingkat preferensi jawaban dengan pilihan sebagai berikut: 1 = sangat tidak setuju; 2 = tidak setuju; 3 = ragu-ragu atau netral; 4 = setuju; 5 = sangat setuju. Selanjutnya, penentuan kategori interval tinggi, sedang, atau rendah digunakan rumus dalam Persamaan (7) berikut.

$$I = \frac{100\%}{K} \dots\dots\dots(7)$$

Keterangan :

I = Interval;

K = Kategori jawaban

Untuk mendapatkan presentase hasil interpretasi, harus diketahui skor tertinggi (X) dan angka terendah (Y) untuk item penilaian dengan rumus pada Persamaan (8) berikut.

X = Skor terendah likert × jumlah responden (Angka Terendah 1)

Y = Skor tertinggi likert × jumlah responden (Angka Tertinggi 5) .....(8)

Penilaian interpretasi responden terhadap aplikasi sistem pakar ini adalah hasil nilai yang dihasilkan dengan menggunakan rumus index % pada Persamaan (9) berikut.

$$\text{Rumus Index \%} = \frac{\text{Total Skor}}{Y \times 100} \dots\dots\dots(9)$$

## 2.7 Sistem Operasi Android

Android merupakan sistem operasi untuk *mobile device* yang awalnya dikembangkan oleh Android Inc. Perusahaan ini kemudian dibeli oleh Google pada tahun 2005. Untuk mengembangkan Android dibentuklah kelompok yang bernama *Open Handset Alliance* (OHA) yang merupakan gabungan dari 34 perusahaan piranti keras, lunak, dan telekomunikasi termasuk Google, HTC, Intel, Motorola, Qualcomm, Tmobile, dan Nvidia.

Android dibuat menggunakan modifikasi kernel Linux 2.6. Aplikasi Android ditulis dengan bahasa Java yang menggunakan *Java Core Libraries*. Aplikasi Android dijalankan di atas VM bernama *Dalvik Virtual Machine*. Android menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri sehingga dapat digunakan oleh bermacam piranti penggerak (Wahana, 2012).

## **BAB III**

### **METODOLOGI PENELITIAN**

#### **3.1 Waktu dan Tempat Penelitian**

Penelitian ini dilakukan di Jurusan Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam dan Jurusan Peternakan Fakultas Pertanian Universitas Lampung. Waktu penelitian dilaksanakan pada periode semester genap Tahun Ajaran 2015/2016.

#### **3.2 Alat Pendukung**

Alat pendukung yang digunakan dalam penelitian ini adalah sebagai berikut:

##### **A. Perangkat Keras**

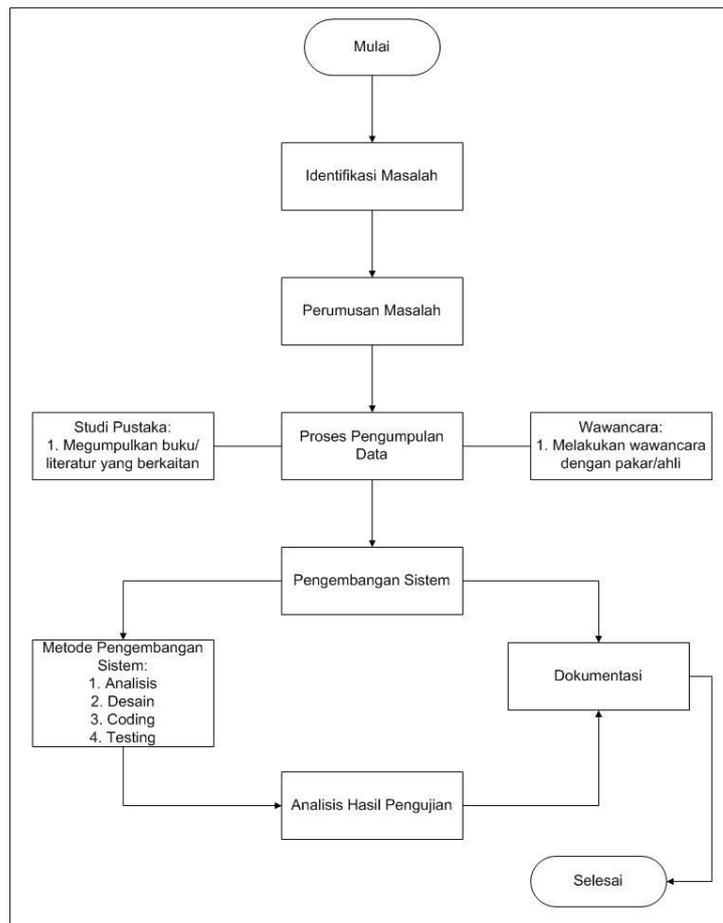
- Perangkat laptop ACER Aspire E1-431 dengan spesifikasi *Processor Intel (R) Celeron (R) CPU B820 @ 1.70 GHz , Harddisk 350 GB, RAM 2.00 GB.*
- *Smartphone* Android (Lenovo A316i *Jelly Bean 4.2.2 version*).

##### **B. Perangkat Lunak**

- Sistem Operasi Windows 7 32-bit.
- *Eclipse Luna SDK*, digunakan sebagai media pembuatan aplikasi.
- *Java Development Kit (JDK)*, merupakan *tools* pengembang bahasa pemrograman java.

- *Android SDK (Software Development Kit)*, merupakan *tools* pengembang pemrograman android.
- *Android Development Tool (ADT)*, *Plug-in* yang digunakan untuk mengintegrasikan *eclipse* menjadi lingkungan pengembangan android.
- *SQLite Manager*, sebagai *software* pembuatan dan akses *database*.
- *Microsoft Excel 2007*, digunakan untuk melakukan perhitungan dengan metode *certainty factor*.
- *Photoshop CS4*, digunakan untuk pembuatan tampilan *user interface* dan *editing* atribut gambar.

### 3.3 Tahapan Penelitian



Gambar 3.1 Tahapan Penelitian

Tahapan penelitian merupakan langkah-langkah yang akan dilakukan peneliti dalam melakukan penelitian. Langkah-langkah dalam penelitian ini meliputi identifikasi masalah, perumusan masalah, pengumpulan data, pengembangan sistem, dan analisis hasil pengujian sistem. Tahapan penelitian ini dapat dilihat pada Gambar 3.1.

### **3.3.1 Tahap Perumusan Masalah**

Tahapan ini merupakan proses merumuskan dan membatasi masalah yang akan diteliti. Perumusan dan pembatasan masalah diperlukan agar dapat lebih mengarahkan peneliti dalam membuat sistem sehingga proyek yang dikerjakan tidak keluar dari batasan yang telah ditetapkan sebelumnya.

### **3.3.2 Tahap Pengumpulan Data**

Tahapan pengumpulan data dilakukan dengan menggunakan dua metode yaitu melalui studi pustaka dan wawancara.

#### **a. Studi Pustaka**

Pada tahap ini data dikumpulkan melalui berbagai literatur seperti pada buku, jurnal, ataupun dokumen yang berkaitan dengan tema penelitian.

#### **b. Wawancara**

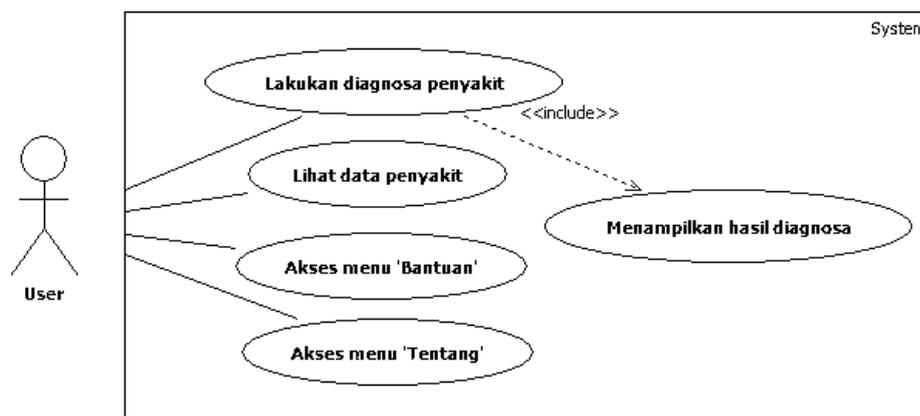
Pada metode ini dilakukan proses *interview* atau wawancara kepada para ahli/pakar. Hal ini bertujuan untuk mendapatkan data yang tidak ditemukan pada metode studi pustaka. Selanjutnya data-data yang telah dikumpulkan disusun menjadi basis aturan yang akan digunakan dalam sistem pakar.

### 3.3.3 Tahap Pengembangan Sistem

Dalam tahap pengembangan sistem digunakan metode *waterfall* yang terdiri dari empat tahap sebagai berikut.

#### 3.3.3.1 *Software Requirements Analysis*

Pada proses ini dilakukan pencarian kebutuhan sistem atau *software*. Dalam membangun aplikasi sistem pakar berbasis android ini telah dirancang model *use case diagram* untuk menginterpretasikan fungsi *interface* dari sisi pengguna (*user*). Desain *use case diagram* sistem dapat dilihat pada Gambar 3.2.



Gambar 3.2 Model *Use Case Diagram*

*Use Case Diagram* pada Gambar 3.2 menjelaskan bahwa aktor *user* memiliki empat aktivitas yang dapat dilakukan yaitu melakukan diagnosa penyakit, melihat informasi data penyakit, mengakses menu bantuan dan mengakses menu tentang.

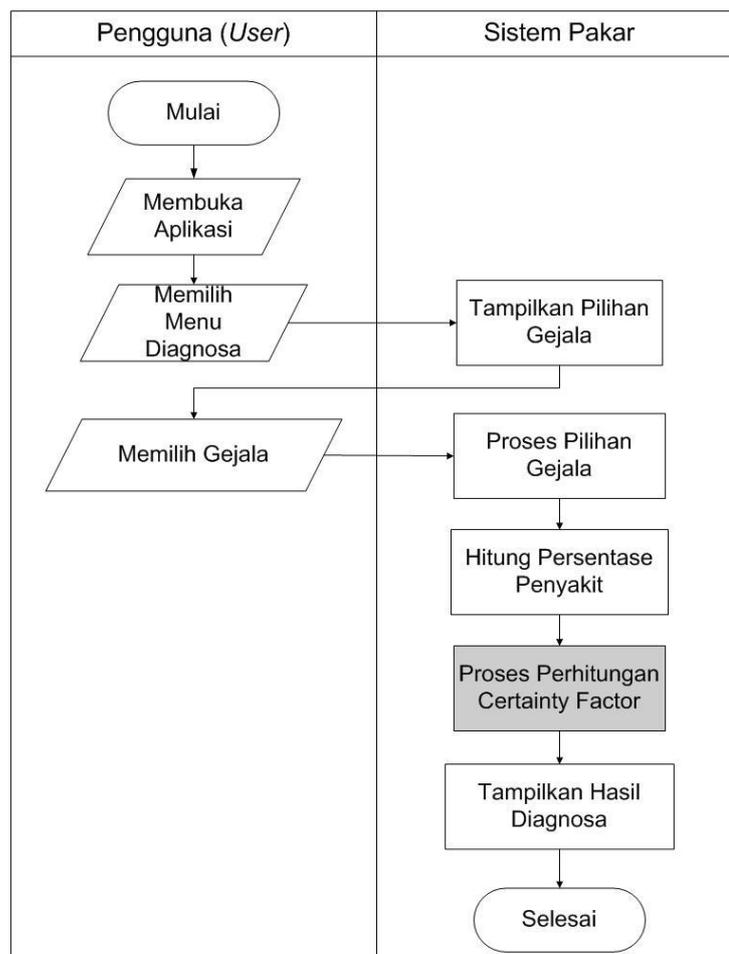
#### 3.3.3.2 *Design*

Pada proses ini dilakukan perancangan sistem dimana desain yang dibuat harus dapat mengimplementasikan kebutuhan yang telah disebutkan pada tahap sebelumnya. Perancangan pada sistem ini dibuat dengan menggunakan metode UML (*Unified Modelling language*). Desain perancangan dalam penelitian ini

menggunakan empat model perancangan yaitu *flowchart*, *class diagram*, *sequence diagram* dan *activity diagram*.

### 3.3.3.2.1 Flowchart

Pada sistem pakar diagnosa penyakit ayam terdapat satu aktor yang terlibat yaitu pengguna (*user*). Secara garis besar, proses-proses yang dilakukan pengguna dan sistem pakar dapat digambarkan dalam bentuk *flowchart*. Flowchart menunjukkan gambaran bisnis proses yang merupakan kumpulan proses yang berisi kumpulan aktivitas terstruktur dan saling berelasi satu sama lain untuk menghasilkan keluaran bisnis proses sistem pakar diagnosa penyakit pada ayam. *Flowchart* antara pengguna dan sistem pakar disajikan pada Gambar 3.3.

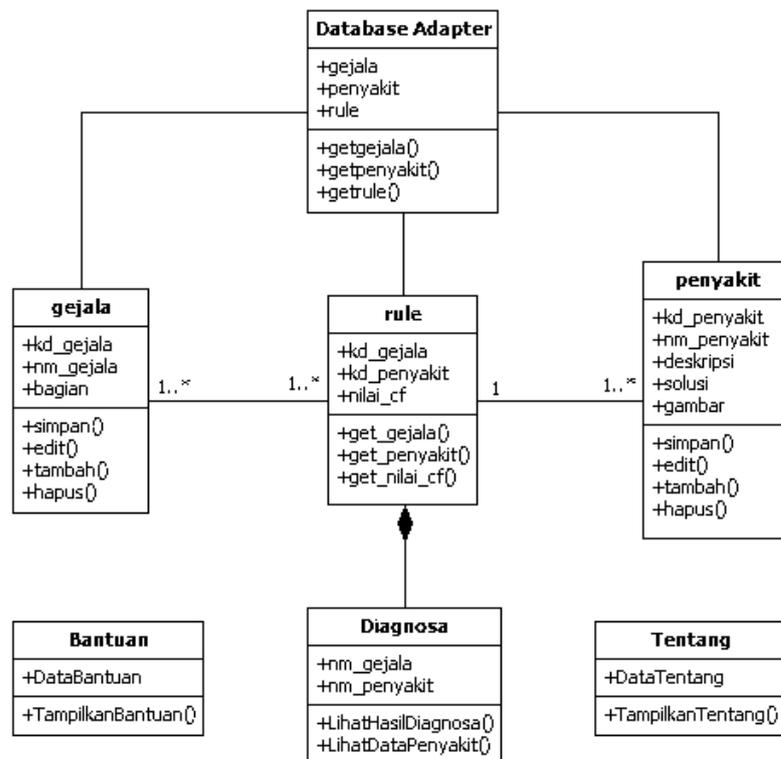


Gambar 3.3 *Flowchart* Sistem Pakar dan Pengguna

Gambar 3.3 menjelaskan untuk mengakses sistem pakar awalnya pengguna membuka aplikasi dan memilih menu diagnosa. Selanjutnya sistem akan menampilkan pilihan gejala yang telah tersimpan di *database*. Kemudian pengguna dapat memilih gejala penyakit yang terlihat pada ayam. Selanjutnya sistem akan memproses pilihan gejala dan menghitung persentase penyakit yang mungkin diderita ayam. Terakhir, sistem akan menampilkan hasil diagnosa yang berupa persentase penyakit, deskripsi penyakit, dan penanganan penyakit yang mungkin diderita ayam berdasarkan gejala yang telah dipilih.

### 3.3.3.2 Class Diagram

Merupakan diagram yang digunakan untuk menampilkan beberapa kelas serta paket-paket yang ada pada sistem aplikasi. Model *class diagram* sistem disajikan pada Gambar 3.4.



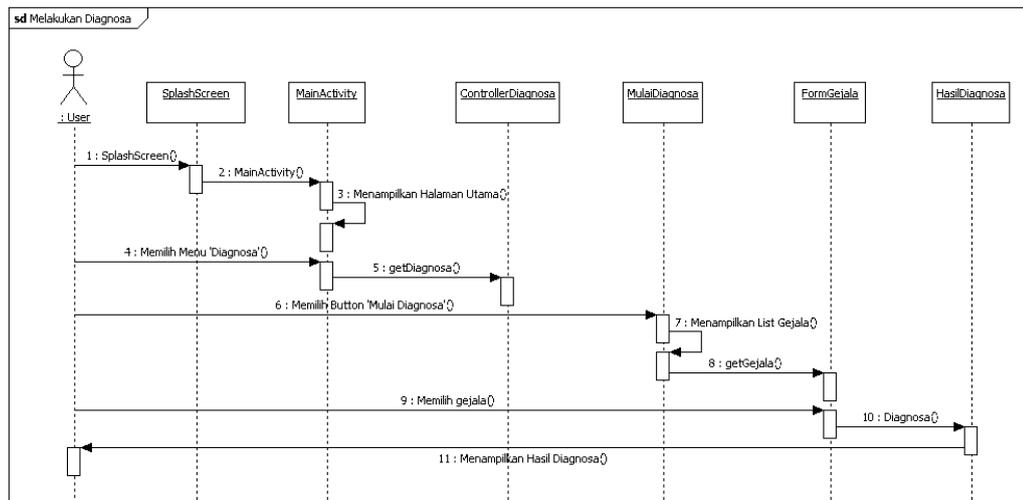
Gambar 3.4 Model *Class Diagram*

Gambar 3.4 menjelaskan bahwa *class diagram* dari Sistem Pakar Diagnosa Penyakit Ayam Berbasis Android memiliki 7 kelas yang terdiri dari *Database Adapter*, *Gejala*, *Relasi*, *Penyakit*, *Bantuan*, *Diagnosa*, dan *Tentang*.

1. *Database adapter* memiliki kelas turunan yang berupa kelas gejala, penyakit, dan relasi.
2. Kelas gejala memiliki hubungan asosiasi dengan kelas relasi dan memiliki nilai kardinalitas satu atau lebih. Hal ini berarti satu gejala dapat mempunyai satu atau lebih relasi. Begitupun halnya dengan kelas relasi dapat mempunyai satu atau lebih gejala.
3. Kelas penyakit memiliki hubungan asosiasi dengan kelas relasi dan memiliki nilai kardinalitas hanya satu. Hal ini berarti setiap penyakit hanya memiliki satu relasi. Sedangkan satu relasi dapat memiliki satu atau lebih penyakit.
4. Kelas diagnosa memiliki hubungan komposisi dengan kelas relasi. Hal ini berarti kelas diagnosa tidak dapat berdiri tanpa adanya kelas relasi. Jika kelas relasi hilang maka kelas diagnosa akan ikut hilang.
5. Kelas bantuan dan tentang berdiri sendiri dan tidak ada hubungan dengan kelas lain.

### **3.3.3.2.3 Sequence Diagram**

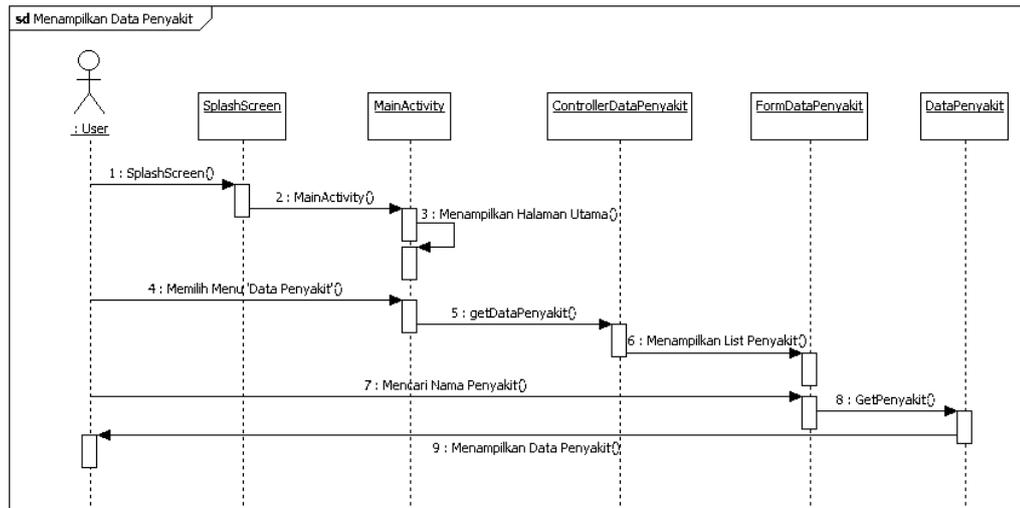
Diagram ini berguna untuk melihat spesifikasi dari sebuah pilihan *button* sehingga *user* dapat memilih *button* tersebut dan akan ditampilkan *sub-menu* dari masing-masing *button* tersebut. Pada sistem ini terdapat empat proses *sequence diagram* yang disajikan pada Gambar 3.5 sampai Gambar 3.8.



Gambar 3.5 *Sequence Diagram* Proses Melakukan Diagnosa

*Sequence Diagram* pada Gambar 3.5 terdiri dari 1 user dan 6 objek yaitu *splash screen*, *main activity*, *controller* diagnosa, *mulai* diagnosa, *form* gejala, dan hasil diagnosa. Proses Gambar 3.5 dijelaskan sebagai berikut.

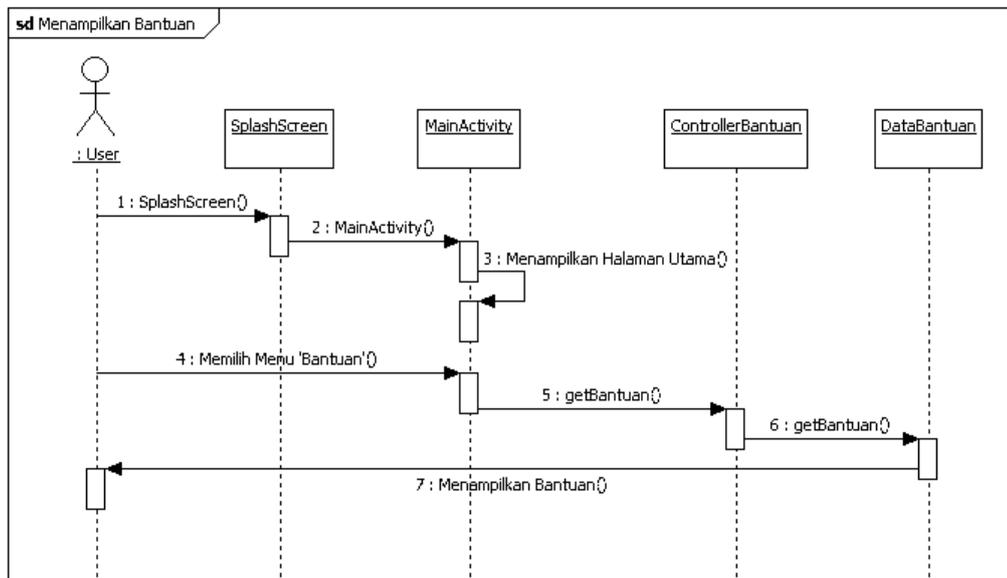
1. *User* mengakses aplikasi dengan melewati objek *splash screen* yang akan menampilkan *splash screen* selama beberapa detik.
2. Kemudian *user* masuk ke halaman utama (*main activity*). Objek melakukan *self stimulus* dengan menampilkan halaman utama sistem.
3. *User* memilih menu diagnosa dan objek *controller* diagnosa mendapatkan halaman yang dituju.
4. *User* memilih button mulai diagnosa dan objek mulai diagnosa menampilkan *list* gejala penyakit.
5. *User* memilih gejala dengan melakukan *checklist*, kemudian *form* gejala akan memproses pilihan tersebut untuk mendapatkan sebuah diagnosa.
6. Kemudian objek hasil diagnosa menampilkan hasil diagnosa penyakit.



Gambar 3.6 *Sequence Diagram* Menampilkan Data Penyakit

*Sequence Diagram* pada Gambar 3.6 terdiri dari 1 *user* dan 5 objek yaitu *splash screen*, *main activity*, *controller data penyakit*, *form data penyakit*, dan *data penyakit*. Proses pada Gambar 3.6 dijelaskan sebagai berikut.

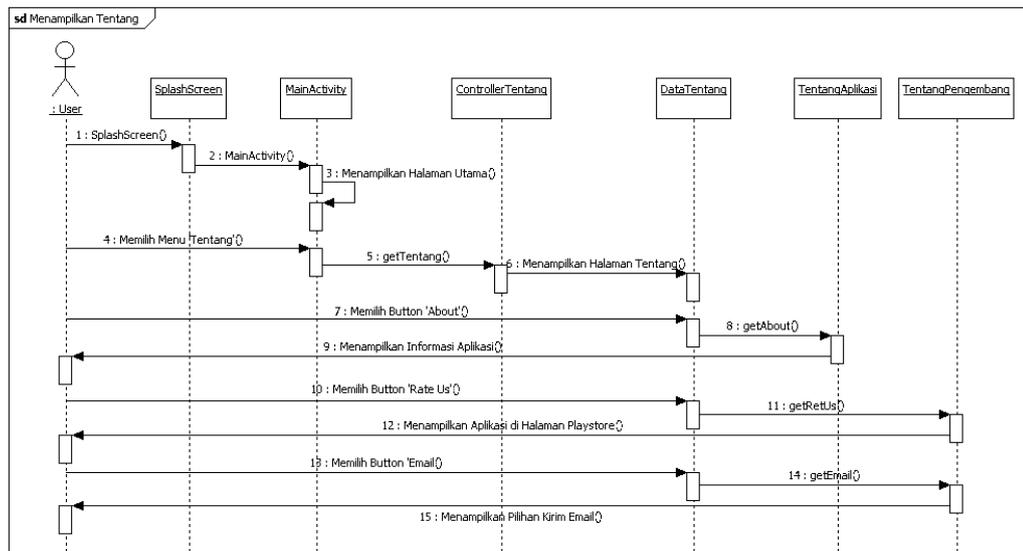
1. *User* mengakses aplikasi dengan melewati objek *splash screen* yang akan menampilkan *splash screen* selama beberapa detik.
2. Kemudian *user* masuk ke halaman utama (*main activity*). Objek melakukan *self stimulus* dengan menampilkan halaman utama sistem.
3. *User* memilih menu data penyakit dan objek *controller data penyakit* mendapatkan halaman yang dituju.
4. Objek *controller data penyakit* menampilkan *list* penyakit.
5. *User* mencari nama penyakit yang diinginkan dan *form data penyakit* mendapatkan nama penyakit tersebut.
6. *Form data penyakit* menampilkan data penyakit yang diinginkan.



Gambar 3.7 *Sequence Diagram* Mengakses Menu Bantuan

*Sequence Diagram* pada Gambar 3.7 terdiri dari 1 *user* dan 4 objek yaitu *splash screen*, *main activity*, *controller* bantuan, *form*, dan data bantuan. Proses pada Gambar 3.7 dijelaskan sebagai berikut.

1. *User* mengakses aplikasi dengan melewati objek *splash screen* yang akan menampilkan *splash screen* selama beberapa detik.
2. Kemudian *user* masuk ke halaman utama (*main activity*). Objek melakukan *self stimulus* dengan menampilkan halaman utama sistem.
3. *User* memilih menu bantuan dan objek *controller* bantuan mendapatkan halaman yang dituju.
4. Objek data bantuan menampilkan halaman bantuan.



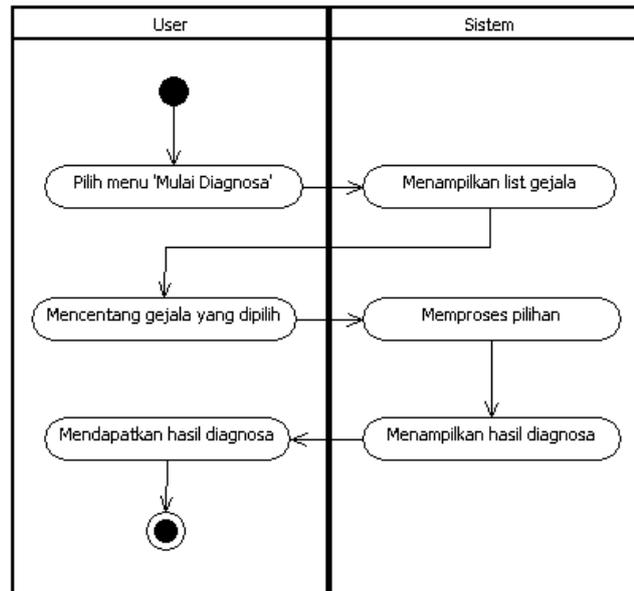
Gambar 3.8 *Sequence Diagram* Mengakses Menu Tentang

*Sequence Diagram* pada Gambar 3.8 terdiri dari 1 *user* dan 6 objek yaitu *splash screen*, *main activity*, *controller* tentang, *data* tentang, *tentang* aplikasi, dan *tentang* pengembang. Proses pada Gambar 3.8 dijelaskan sebagai berikut.

1. *User* mengakses aplikasi dengan melewati objek *splash screen* yang akan menampilkan *splash screen* selama beberapa detik.
2. Kemudian *user* masuk ke halaman utama (*main activity*). Objek melakukan *self stimulus* dengan menampilkan halaman utama sistem.
3. *User* memilih menu tentang dan objek *controller* tentang mendapatkan halaman yang dituju dan kemudian menampilkan halaman tentang.
4. *User* memilih *button* ‘about’ untuk mendapatkan informasi tentang aplikasi.
5. *User* memilih *button* ‘rate us’ dan *button* ‘email’ untuk memberikan review aplikasi dan berhubungan kontak dengan pengembang.

### 3.3.3.2.4 Activity Diagram

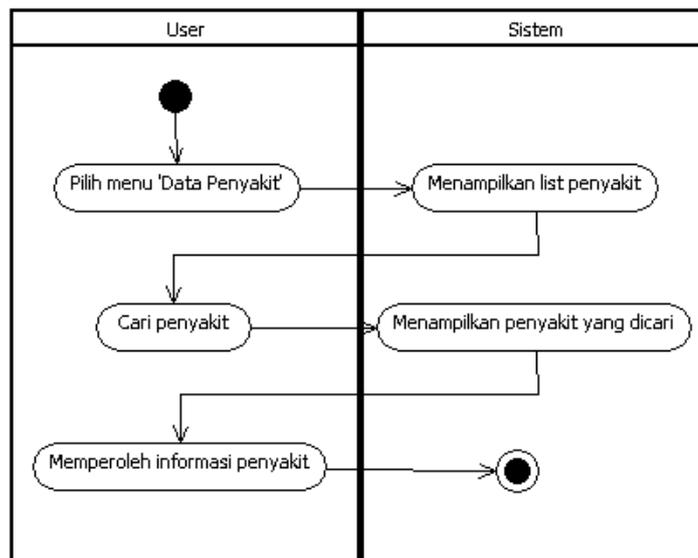
Merupakan diagram yang menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Pada sistem ini terdapat empat proses activity diagram yang disajikan pada Gambar 3.9 sampai Gambar 3.12.



Gambar 3.9 Activity Diagram Proses Diagnosa Penyakit

Activity Diagram pada Gambar 3.9 terdiri dari *user* dan sistem. Proses pada Gambar 3.9 dijelaskan sebagai berikut.

1. *User* mengawali aktivitas, memilih menu mulai diagnosa dan sistem akan menampilkan halaman *list* gejala.
2. Kemudian *user* mencentang gejala yang dipilih dan sistem akan memproses pilihan tersebut.
3. Selanjutnya sistem menampilkan hasil diagnosa dan *user* akan mendapatkan hasil diagnosa berdasarkan gejala yang dipilih.



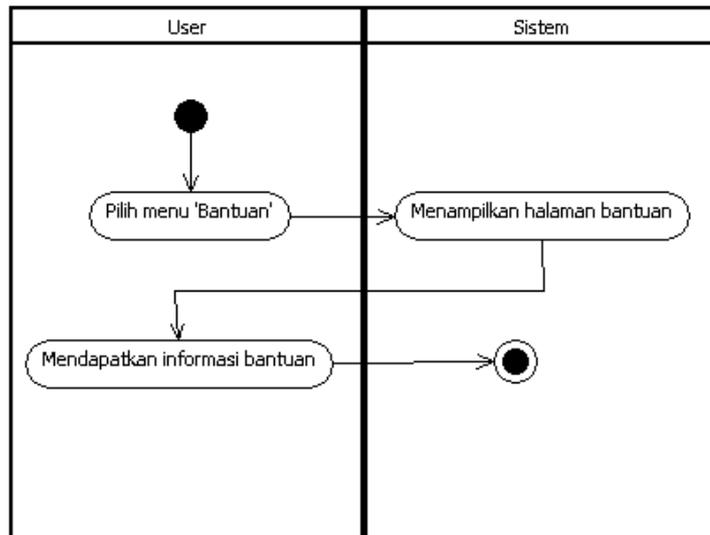
Gambar 3.10 *Activity Diagram* Proses Mengakses Data Penyakit

*Activity Diagram* pada Gambar 3.10 terdiri dari *user* dan sistem. Proses pada Gambar 3.10 dijelaskan sebagai berikut.

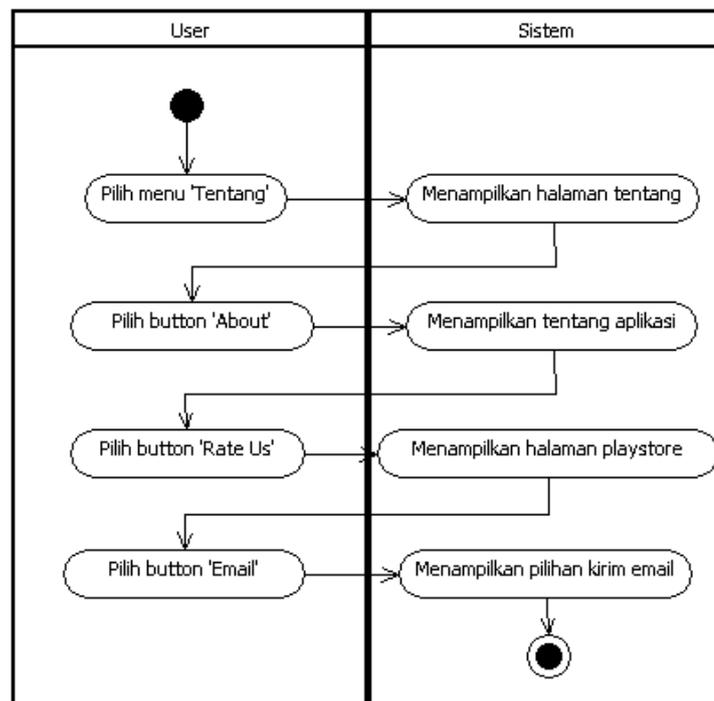
1. *User* mengawali aktivitas, memilih menu data penyakit dan sistem akan menampilkan *list* penyakit.
2. Kemudian *user* mencari penyakit yang dibutuhkan dan sistem menampilkan penyakit yang dicari.
3. Selanjutnya *user* akan memperoleh informasi data penyakit.

*Activity Diagram* pada Gambar 3.11 terdiri dari *user* dan sistem. Proses pada Gambar 3.11 dijelaskan sebagai berikut.

1. *User* mengawali aktivitas, memilih menu bantuan dan sistem akan menampilkan halaman bantuan.
2. *User* mendapatkan informasi bantuan.



Gambar 3.11 *Activity Diagram* Proses Mengakses Menu Bantuan



Gambar 3.12 *Activity Diagram* Proses Mengakses Menu Tentang

Sedangkan *Activity Diagram* pada Gambar 3.12 terdiri dari *user* dan sistem.

Proses pada Gambar 3.12 dijelaskan sebagai berikut.

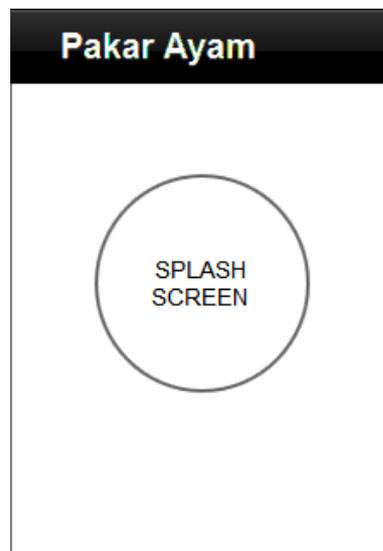
1. *User* mengawali aktivitas, memilih menu tentang dan sistem akan menampilkan halaman tentang.
2. *User* memilih *button* 'about' dan sistem akan menampilkan halaman yang berisi informasi tentang aplikasi.
3. *User* memilih *button* 'rate us' dan *button* 'email' untuk memberikan review aplikasi dan berhubungan kontak dengan pengembang.

#### **3.3.3.2.5 Rancangan Antarmuka (*Interface*)**

Antarmuka sistem dibuat sebagai penghubung antara sistem pakar dengan pengguna (*user*). Pada rancangan antarmuka sistem pakar ini terdapat beberapa halaman yang dapat diakses oleh pengguna (*user*), seperti berikut.

##### **A. Rancangan Halaman *Splash Screen***

Merupakan halaman *screen* kilat yang muncul ketika pertama kali membuka aplikasi. Rancangan halaman *splash screen* disajikan pada Gambar 3.13.



Gambar 3.13 Rancangan Halaman Splash Screen

## B. Rancangan Halaman Menu Utama

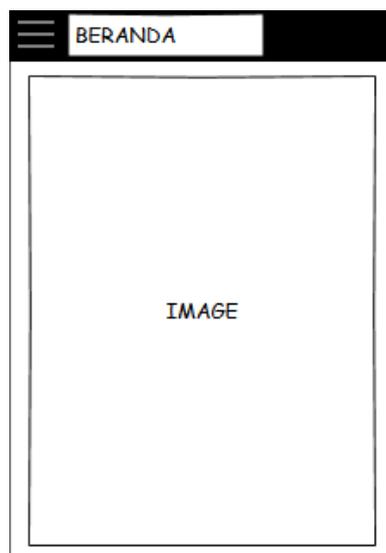
Merupakan halaman utama yang menampilkan beberapa pilihan menu kepada pengguna. Rancangan halaman utama disajikan pada Gambar 3.14.



Gambar 3.14 Rancangan Halaman Menu Utama

## C. Rancangan Halaman Beranda

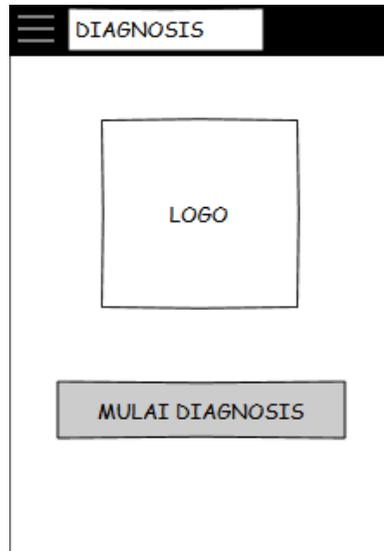
Merupakan halaman awal yang ditampilkan sistem ketika pengguna memilih menu beranda. Rancangan halaman beranda disajikan pada Gambar 3.15.



Gambar 3.15 Rancangan Halaman Beranda

#### D. Rancangan Halaman Diagnosis

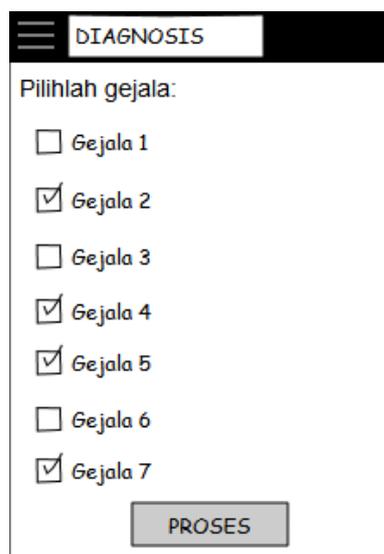
Merupakan halaman awal diagnosis yang ditampilkan ketika pengguna memilih menu diagnosis. Rancangan halaman diagnosis disajikan pada Gambar 3.16.



Gambar 3.16 Rancangan Halaman Diagnosis

#### E. Rancangan Halaman Pilih Gejala

Merupakan halaman yang menampilkan pilihan gejala yang dapat dipilih pengguna. Rancangan halaman mulai diagnosis disajikan pada Gambar 3.17.



Gambar 3.17 Rancangan Halaman Pilih Gejala

#### F. Rancangan Halaman Hasil Diagnosis

Merupakan halaman yang ditampilkan sistem setelah pengguna memilih gejala.

Rancangan halaman hasil diagnosis disajikan pada Gambar 3.18.



Gambar 3.18 Rancangan Halaman Hasil Diagnosis

#### G. Rancangan Halaman Data Penyakit

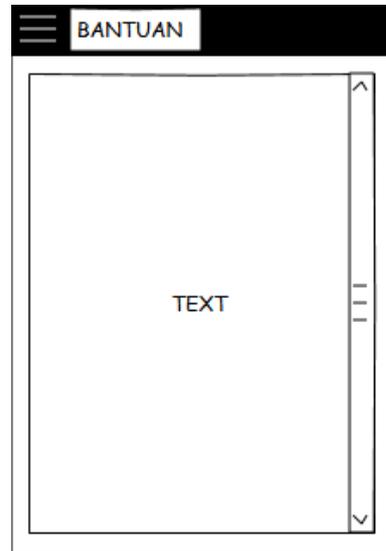
Merupakan halaman yang dapat menampilkan informasi penyakit beserta penjelasannya. Rancangan halaman data penyakit disajikan pada Gambar 3.19.



Gambar 3.19 Rancangan Halaman Data Penyakit

## H. Rancangan Halaman Bantuan

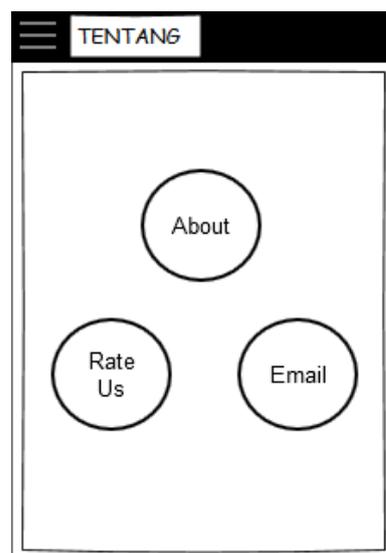
Merupakan halaman yang menampilkan informasi bantuan dalam penggunaan aplikasi. Rancangan halaman bantuan penyakit disajikan pada Gambar 3.20.



Gambar 3.20 Rancangan Halaman Bantuan

## I. Rancangan Halaman Tentang

Merupakan halaman yang menampilkan informasi tentang aplikasi dan pengembang. Rancangan halaman tentang disajikan pada Gambar 3.21.



Gambar 3.21 Rancangan Halaman Tentang

### **3.3.3.3 Coding**

Merupakan proses menerjemahkan desain yang telah dirancang menjadi bahasa yang dimengerti oleh komputer. Agar dapat dimengerti oleh mesin, dalam hal ini adalah komputer, maka desain rancangan harus diubah bentuknya ke dalam bahasa pemrograman seperti bahasa C, C++, PHP, Java, dan sebagainya. Dalam penelitian ini sistem dibangun menggunakan bahasa pemrograman android yang terdiri dari bahasa Java dan XML dengan menggunakan *framework Eclipse*.

### **3.3.3.4 Testing**

Setelah proses *coding* selesai, maka dilakukan proses pengujian dengan menggunakan metode *black box*. Pengujian *black box* merupakan pengujian aspek fundamental sistem tanpa memperhatikan struktur logika internal perangkat lunak. Proses ini dilakukan untuk memastikan sistem yang dibuat telah bekerja sesuai dengan ketentuan yang ditetapkan.

### **3.3.4 Analisis Hasil Pengujian**

Merupakan tahap analisa dari pengujian atau *testing* yang telah dilakukan pada tahap sebelumnya. Tahap ini merupakan proses akhir dari penyelesaian sistem. Analisis yang dilakukan bertujuan untuk membuat kesimpulan dari pengujian yang dilakukan, apakah masih terdapat kesalahan atau kekurangan di dalam sistem dan sebagainya.

## **BAB V**

### **PENUTUP**

#### **5.1 Kesimpulan**

Berdasarkan penelitian yang telah dilakukan dapat disimpulkan beberapa hal sebagai berikut :

1. Telah berhasil dibangun aplikasi ‘Pakar Ayam’ yang dapat membantu peternak, pengusaha, ataupun akademisi yang berhubungan dengan ternak ayam dalam melakukan diagnosa masalah penyakit ayam.
2. Sistem pakar yang dibangun dapat memberikan presentase hasil diagnosa penyakit berdasarkan fakta dan pengetahuan yang telah diberikan.
3. Presentase hasil diagnosa dengan menggunakan proses perhitungan *Certainty Factor* (CF) sangat dipengaruhi pada nilai CF yang diberikan oleh pakar.
4. Berdasarkan hasil pengujian *Equivalence Partitioning*, disimpulkan bahwa dari 31 orang total responden, 6 orang kelompok responden pertama yang terdiri dari Pakar Penyakit Ayam mendapatkan rata-rata nilai sebesar 63,33 persen. Hal ini menunjukkan bahwa penilaian terhadap aplikasi ‘Pakar Ayam’ oleh responden yang paham terhadap penyakit ayam tergolong ‘Baik’.

Dari 15 orang kelompok responden kedua yang terdiri dari peternak dan mahasiswa Peternakan mendapatkan rata-rata nilai sebesar 77,71 persen. Hal ini menunjukkan bahwa penilaian terhadap aplikasi 'Pakar Ayam' oleh responden yang berhubungan langsung terhadap penyakit ayam tergolong 'Baik'.

Sedangkan dari 10 orang kelompok responden ketiga yang terdiri dari mahasiswa Ilmu Komputer mendapatkan rata-rata nilai sebesar 82 persen. Hal ini menunjukkan bahwa penilaian terhadap aplikasi 'Pakar Ayam' oleh responden yang tidak paham dan tidak berhubungan terhadap penyakit ayam tergolong 'Sangat Baik' .

## **5.2 Saran**

Adapun saran yang diberikan adalah sebagai berikut :

1. Penambahan data penyakit ayam beserta gejala yang disebabkan oleh virus dan mikal.
2. Kelengkapan gambar penyakit ayam dan penyempurnaan desain *User Interface* (UI) aplikasi.
3. Penyederhanaan bahasa pada deskripsi dan solusi penyakit agar lebih mudah dimengerti.

## DAFTAR PUSTAKA

- Arhami, Muhammad. 2008. *Konsep Dasar Sistem Pakar*. Yogyakarta: Penerbit ANDI.
- Giarratano, J., & Riley, G. 2005. *Expert System Principles and Programming*. Boston : Pre-Press Company, Inc.
- Herlawati & Widodo. 2011. *Menggunakan UML*. Informatika. Bandung.
- Jiang, F., Y. Lu. 2012. Software testing model selection research based on yinyang testing theory. In: *IEEE Proceeding of International Conference on Computer Science and Information Processing (CISP)*, pp. 590-594.
- Komputer, Wahana. 2012. *Langkah Praktis Membangun Aplikasi Sederhana Platlayout Android*. Jakarta: PT Elex Media Komputindo.
- Kusrini. 2006. *Sistem Pakar Teori dan Aplikasi*. Yogyakarta: Penerbit ANDI.
- Nugroho, Adi. 2009. *Rekayasa Perangkat Lunak Menggunakan UML & Java*. Yogyakarta: Andi Offset.
- Nugroho, B. 2014. *Aplikasi Sistem Pakar*. Yogyakarta: Gava Media.
- Pressman, Roger S. 2001. *Software Engineering A Practitioner's Approach Fifth Edition*. McGraw-Hill Companies, Inc, New York.

- Pressman, R.S. 2010. *Software Engineering: A Practitioner's Approach*, 7<sup>th</sup> Edition. McGraw-Hill, New York.
- Rangga Tabbu, Charles. 2015. *Penyakit Ayam dan Penanggulangannya*. Yogyakarta: PT Kanisius.
- Siswanto, 2010. *Kecerdasan Tiruan Edisi 2*. Yogyakarta: Graha Ilmu.
- Sutojo, T., Edy M., dan Vincent S. 2011. *Kecerdasan Buatan*. Semarang: ANDI Yogyakarta.
- IlmuKomputer.Com. 2003. *Pengantar Unified Modeling Language (UML)*. [Online] Tersedia: <http://www.ilmukomputer.com/>. Diakses pada 1 Maret 2016.