



**IMPLEMENTASI METODE CERTAINTY FACTOR
UNTUK MENGIDENTIFIKASI PENYAKIT BURUNG
MURAI BATU**

SKRIPSI

Diajukan sebagai salah satu persyaratan untuk memperoleh gelar Sarjana
Pendidikan Program Studi Pendidikan Teknik Informatika dan Komputer

Oleh

Wahyu Himawan
NIM.5302412111

**JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNIK
UNIVERSITAS NEGERI SEMARANG
2019**


PERSETUJUAN PEMBIMBING

Nama : Wahyu Himawan
NIM : 5302412111
Program Studi : S-1 Pendidikan Teknik Informatika dan Komputer
Judul Skripsi : **IMPLEMENTASI METODE CERTAINTY FACTOR
UNTUK MENGIDENTIFIKASI PENYAKIT
BURUNG MURAI BATU**

Skripsi/TA ini telah disetujui oleh pembimbing untuk diajukan ke sidang panitia ujian skripsi Program Studi S-1 Pendidikan Teknik Informatika dan Komputer Fakultas Teknik Universitas Negeri Semarang.

Semarang, 6 Agustus 2019

Pembimbing I,


Dr. Djurjadi, M.T.
NIP. 196306281990021001

LEMBAR PENGESAHAN

Skripsi dengan judul "Implementasi Metode Certainty Factor untuk Mengidentifikasi Penyakit Burung Murai Batu" telah dipertahankan di hadapan Panitia Ujian Skripsi Fakultas Teknik UNNES pada 14 Agustus 2019.

Oleh:

Nama : Wahyu Himawan
NIM : 5302412111
Program Studi : S-1 Pendidikan Teknik Informatika dan Komputer

Panitia:

Ketua

Sekretaris



Dr. Ing. Dhidik Prastiyanto, S.T., M.T.

NIP. 197805312005011002

Penguji I



Ir. Ulfah Mediaty Arief, M.T.

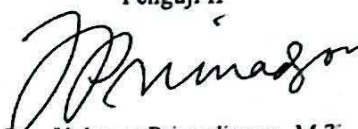
NIP. 196605051998022001

Penguji II



Ir. Ulfah Mediaty Arief, M.T., IPM

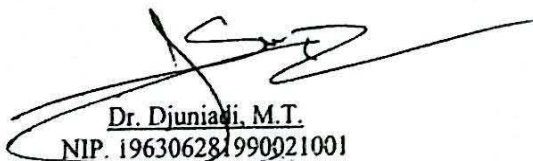
NIP. 196605051998022001



Drs. Yohanes Primadivono, M.T.

NIP. 196209021987031002

Penguji III/Pembimbing

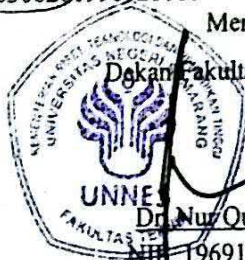


Dr. Djuniadi, M.T.

NIP. 196306281990021001

Mengetahui:

Dekan Fakultas Teknik UNNES



Dr. Nur Oudus, M.T., IPM

NIP. 196911301994031001

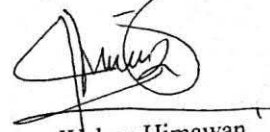
PERNYATAAN KEASLIAN

Dengan ini saya menyatakan bahwa :

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

Semarang, 5 Agustus 2019

Yang membuat pernyataan,



Wahyu Himawan

NIM. 5302412111

MOTTO DAN PERSEMBAHAN

- Sesungguhnya setiap perbuatan tergantung niatnya, dan sesungguhnya semua orang (akan dibalas) berdasarkan apa yang dia niatkan (HR Bukhari Muslim).
- Dalam setiap masalah yang kita hadapi terdapat suatu pelajaran yang dapat kita ambil, jangan terpaku pada masalahnya, namun pikirkan apa hikmahnya untuk kehidupanmu.
- Setiap orang memiliki zona waktunya masing-masing, belum tentu yang tercepat dianggap berhasil dan belum tentu yang terlambat dianggap gagal, begitu pula dengan perjalanan kehidupan.

Persembahan:

Skripsi ini saya persembahkan untuk:

1. Ibu dan ayah tercinta Ibu Suyati dan Bapak Sukarjo
2. Kedua kakak tercinta Ulya Himawati dan Anis Fuad
3. Keluarga besar Kos Robert House
4. Sahabat setia saya PTIK rombel 3
5. Teman-teman PTIK angkatan 2012
6. Teman-teman yang membantu saya menyusun skripsi Durrotun Nailul Muna Alfur Laila, Aji Sahbana dan Fajar Afif Dewantoro
7. Bapak Arimaz Hangga, S.T.,M.T. dan Alfa Faridh Suni, S.T.,M.T. yang banyak membantu saya dalam perkuliahan.

ABSTRAK

Himawan, Wahyu. 2019. Implementasi Metode *Certainty Factor* untuk Mengidentifikasi Penyakit Burung Murai Batu. Skripsi, Jurusan Teknik Elektro dan Informatika, Program Studi Pendidikan Teknik Informatika dan Komputer, Fakultas Teknik, Universitas Negeri Semarang. Dr. Djuniadi, M.T.

Burung murai batu merupakan kelompok burung yang di gemari di kalangan para pecinta kicauan karena memiliki suara atau spesifikasi kicauan yang sangat baik. Proses diagnosa jenis penyakit pada murai batu seharusnya dilakukan oleh seorang ahli di bidang tersebut. Namun, keterbatasan jumlah dan sulitnya berinteraksi langsung dengan tenaga ahli seperti peternak murai batu yang sudah berpengalaman dalam mendiagnosis dan menentukan jenis penyakit serta memberikan cara pengendalian guna mendapatkan solusinya, membuat sebagian besar peternak pemula dan pecinta burung menangani sendiri permasalahan tersebut. Maka dari itu dibuatlah Sistem pakar untuk mengidentifikasi penyakit pada burung murai batu yang dirancang menggunakan aplikasi berbasis web. Aplikasi yang dikembangkan diberi nama Sistem Pakar Murai Batu. Tujuan dari penelitian ini adalah mengembangkan aplikasi untuk mengidentifikasi penyakit pada burung murai batu yang mengimplementasikan metode *Certainty Factor*.

Metode penelitian yang digunakan adalah *waterfall* yang memiliki siklus *communication, planning, modeling, construction dan deployment*. Tahap pengujian pada penelitian ini dilakukan dalam rangka melakukan validation pada perangkat lunak. Proses validation dilakukan dengan *blackbox testing*.

Hasil penelitian ini adalah sebuah aplikasi Sistem Pakar Murai Batu yang telah mengimplementasikan metode *Certainty Factor*. Fitur yang terdapat dalam aplikasi Sistem Pakar Murai Batu antara lain: konsultasi, data hasil diagnosa penyakit, data gejala, data penyakit, tambah data, edit data, hapus data, serta panduan bagi pengguna dan pakar. Hasil pengujian dari aplikasi Sistem Pakar Murai Batu dengan uji *blackbox* menyatakan bahwa fungsionalitas pada sistem bekerja sesuai fungsinya dan dapat digunakan dengan baik.

Saran untuk penelitian ini adalah penambahan jenis burung yang dapat diidentifikasi penyakitnya seperti burung kenari, burung cucak hijau dan jenis burung kicau lainnya.

Kata kunci: Sistem Pakar, Murai Batu, *Certainty Factor*, Penyakit.

PRAKATA

Segala puji dan syukur penulis ucapkan kehadirat Allah SWT dan mengharapkan ridho yang telah melimpahkan rahmat-Nya sehingga penulis dapat menyelesaikan skripsi yang berjudul " Implementasi Metode Certainty Factor untuk Mengidentifikasi Penyakit Burung Murai Batu ". Skripsi ini disusun sebagai salah satu persyaratan meraih gelar Sarjana Pendidikan pada Program Studi S-1 Pendidikan Teknik Informatika dan Komputer Universitas Negeri Semarang. Shalawat serta salam senantiasa disampaikan kepada junjungan alam Nabi Muhammad SAW, mudah-mudahan kita semua mendapatkan safaat di yaumul akhir nanti, Aamiin.

Penyelesaian skripsi ini tidak lepas dari bantuan berbagai pihak, oleh karena itu pada kesempatan ini peneliti menyampaikan ucapan terima kasih serta penghargaan kepada :

1. Prof. Dr. Fathur Rokhman, M.Hum, Rektor Universitas Negeri Semarang atas kesempatan yang diberikan kepada peneliti untuk menempuh studi di Universitas Negeri Semarang.
2. Dekan Fakultas Teknik Universitas Negeri Semarang, Dr. Nur Qudus M.T., IPM.
3. Ketua Jurusan Teknik Elektro Fakultas Teknik Universitas Negeri Semarang, Dr. Ing. Dhidik Prastiyanto S.T., M.T.
4. Dr. Djuniadi, M.T. selaku pembimbing atas arahan dan motivasinya dalam penyusunan skripsi ini.
5. Drs. Riana Defi, M.Si selaku dosen wali.
6. Teman-teman Pendidikan Teknik Informatika dan Komputer angkatan 2012.
7. Semua pihak yang telah membantu peneliti dalam menyelesaikan skripsi ini yang tidak dapat penyusun sebutkan satu persatu.

Peneliti berharap semoga Skripsi ini dapat bermanfaat untuk peternak burung murai batu.

Semarang, Agustus 2019

Peneliti

DAFTAR ISI

HALAMAN JUDUL	i
LEMBAR PERSETUJUAN PEMBIMBING	ii
LEMBAR PENGESAHAN.....	iii
LEMBAR KEASLIAN KARYA ILMIAH.....	iv
MOTTO DAN PERSEMBAHAN.....	v
ABSTRAK	vi
PRAKATA	vii
DAFTAR ISI.....	viii
DAFTAR TABEL	xii
DAFTAR GAMBAR	xiv
DAFTAR LAMPIRAN.....	xvii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Batasan Masalah	4
1.3 Rumusan Masalah	5
1.4 Tujuan Penelitian	5
1.5 Manfaat Penelitian	6
1.6 Penegasan Istilah.....	6
BAB II LANDASAN TEORI	8
2.1 Penelitian Terdahulu	8
2.2 Sistem Pakar	8

2.2.1 Metode Pemecahan Masalah.....	11
2.2.2 Metode <i>Certainty Factor</i>	14
2.3 Murai Batu	15
2.4 Penyakit Unggas	19
2.4.1 Penyakit Berak Kapur.....	19
2.4.2 Penyakit Flu Burung.....	19
2.4.3 Penyakit Tetelo.....	19
2.4.4 Penyakit Produksi Telur.....	19
2.5 Rancangan Bangun Perangkat lunak.....	20
2.5.1 UML	20
2.5.2 Basis Data	26
2.5.2.1 <i>Database Management System</i>	27
2.5.2.2 Model Basis Data.....	27
2.5.2.3 Perancangan Basis Data.....	28
2.5.2.4 <i>Entity Relationship Diagram (ERD)</i>	29
2.5.3 MySQL	30
2.5.4 PHP	31
2.6 Pengujian Perangkat Lunak	32
2.6.1 <i>Blackbox Testing</i>	32
2.7 Kerangka Berfikir.....	40
BAB III METODE PENELITIAN	43
3.1 Waktu dan Tempat Penelitian.....	43
3.2 Desain Penelitian.....	43

3.3	Prosedur Pengembangan	46
3.3.1	Analisis Kebutuhan	48
3.3.1.1	Metode Pengumpulan Data	48
3.3.1.2	Identifikasi Masalah dan Kebutuhan Pengguna	50
3.3.1.3	Prinsip dan Konsep Analisis.....	52
3.3.2	Desain	55
3.3.2.1	Informasi Umum.....	55
3.3.2.2	Perancangan <i>Unified Modeling Language</i> (UML).....	55
3.3.2.3	Perancangan Antarmuka	81
3.3.2.4	<i>Flochart</i> Perancangan Sistem.....	84
3.3.2.5	Perancangan Basis Data	86
3.3.3	Implementasi	88
3.3.3.1	Persiapan Lingkungan Pengembangan	88
3.3.3.2	Desain Implementasi <i>Certainty Factor</i>	89
3.3.4	Pengujian.....	91
BAB IV HASIL PENELITIAN DAN PEMBAHASAN		92
4.1	Hasil Penelitian	92
4.1.1	Implementasi Kode.....	92
4.1.2	Data Gejala.....	99
4.1.3	Data Penyakit	100
4.1.4	Kaidah Produksi (<i>Rule</i>)	101
4.1.5	Antarmuka Aplikasi	104
4.2	Hasil Pengujian	132

4.2.1 Uji <i>Blackbox</i>	132
4.3 Pembahasan	135
BAB V PENUTUP	137
5.1 Simpulan.....	137
5.2 Saran.....	138
DAFTAR PUSTAKA	xvi
LAMPIRAN	xxii

DAFTAR TABEL

2.1 Tabel Nilai CF	15
2.2 Tabel Simbol-simbol ERD	29
2.3 Aspek <i>Software Quality</i>	36
3.1 Identifikasi Masalah dan Kebutuhan Pengguna	51
3.2 Spesifikasi Kebutuhan Perangkat Lunak(<i>Software</i>)	53
3.3 Spesifikasi Kebutuhan Perangkat Keras (<i>Hardware</i>)	54
3.3 Identifikasi <i>Use Case</i>	57
3.4 Narasi Menu Data Gejala	61
3.5 Narasi Menu Data Penyakit.....	62
3.6 Narasi Menu Konsultasi	62
3.7 Narasi Menu Bantuan.....	64
3.8 Narasi Menu Relasi	64
3.9 Narasi Menu Tambah Data Gejala.....	65
3.10 Narasi Menu Edit Data Gejala	66
3.11 Narasi Menu Tambah Data Penyakit	68
3.12 Narasi Menu Edit Data Penyakit.....	69
3.13 Narasi Menu Tambah Data Relasi	70
3.14 Narasi Menu Edit Data Relasi	71
3.15 Narasi Menu <i>Logout</i>	72
3.16 Narasi Menu Panduan Pengguna	73
3.17 Narasi Menu Panduan <i>Admin</i>	74

3.18 Narasi Menu Tentang Aplikasi	75
3.19 Tabel Data <i>Adminn</i>	86
3.20 Tabel Data Penyakit	87
3.21 Tabel Data Gejala.....	87
3.22 Tabel Relasi	88
4.1 Tabel Contoh Kasus <i>Certainty Factor</i>	94
4.2 Tabel Implementasi Pengkodean	96
4.3 Tabel Data Gejala	99
4.4 Tabel Data Penyakit	100
4.5 Tabel Kaidah Produksi(<i>Rule</i>)	101
4.6 Tabel Hasil Pengujian <i>Blackbox</i>	132

DAFTAR GAMBAR

2.1 Struktur Sistem Pakar Metode <i>Forward Chaining</i>	10
2.2 Cara Kerja <i>Forward Chaining</i>	13
2.3 Diagram Pelacakan <i>Forward Chaining</i> (Sembiring 2013 : 8)	13
2.4 Contoh Penggunaan <i>Use Case Diagram</i>	21
2.5 Contoh Penggunaan <i>Sequence Diagram</i>	22
2.6 Contoh Penggunaan <i>Activity Diagram</i>	23
2.7 Kerangka Berfikir	42
3.1 Model Sekuensial <i>Linear</i>	43
3.2 Langkah-Langkah Penelitian	47
3.3 <i>Use Case Diagram</i> Aktor <i>User</i>	59
3.4 <i>Use Case Diagram</i> Aktor <i>Admin</i>	60
3.5 <i>Activity Diagram</i> Data Gejala.....	76
3.6 <i>Activity Diagram</i> Data Penyakit	77
3.7 <i>Activity Diagram</i> Relasi	77
3.8 <i>Activity Diagram</i> Konsultasi	77
3.9 <i>Activity Diagram</i> Bantuan	78
3.10 <i>Activity Diagram</i> Tambah Data	78
3.11 <i>Activity Diagram</i> Edit Data	79
3.12 <i>Activity Diagram</i> Logout	79
3.13 <i>Activity Diagram</i> Panduan Pengguna.....	80
3.14 <i>Activity Diagram</i> Panduan Admin	80

3.15 <i>Activity Diagram</i> Tentang Aplikasi	80
3.16 Rancangan <i>Interface</i> Halaman Utama.....	81
3.17 Rancangan <i>Interface</i> Halaman Data Gejala.....	81
3.18 Rancangan <i>Interface</i> Halaman Penyakit	82
3.19 Rancangan <i>Interface</i> Halaman Konsultasi	82
3.20 Rancangan <i>Interface</i> Menu Relasi	83
3.21 Rancangan <i>Interface</i> Menu Bantuan.....	83
3.22 Rancangan <i>Interface</i> Menu <i>Login Logout</i>	84
3.23 <i>Flowchart</i> Rancangan Sistem	85
3.24 Tampilan Aplikasi <i>Xampp Control</i>	89
4.1 Spesifikasi Perangkat Komputer.....	92
4.2 Antarmuka Beranda <i>Desktop</i>	105
4.3 Antarmuka Beranda <i>Mobile</i>	106
4.4 Antarmuka Data Gejala <i>Desktop</i>	107
4.5 Antarmuka Data Gejala <i>Mobile</i>	107
4.6 Antarmuka Data Penyakit <i>Desktop</i>	108
4.7 Antarmuka Data Penyakit <i>Mobile</i>	109
4.8 Antarmuka Konsultasi <i>Desktop</i>	110
4.9 Antarmuka Konsultasi <i>Mobile</i>	111
4.10 Antarmuka Hasil Konsultasi <i>Desktop</i>	112
4.11 Antarmuka Hasil Konsultasi <i>Mobile</i>	113
4.12 Antarmuka Bantuan <i>Desktop</i>	114
4.13 Antarmuka Bantuan <i>Mobile</i>	115

4.14 Antarmuka <i>Login Desktop</i>	116
4.15 Antarmuka <i>Login Mobile</i>	116
4.16 Antarmuka Menu <i>Administrator Desktop</i>	117
4.17 Antarmuka Menu <i>Administrator Mobile</i>	118
4.18 Antarmuka Tambah Data Gejala <i>Desktop</i>	119
4.19 Antarmuka Tambah Data Gejala <i>Mobile</i>	119
4.20 Antarmuka Edit Data Gejala <i>Desktop</i>	120
4.21 Antarmuka Edit Data Gejala <i>Mobile</i>	121
4.22 Antarmuka Tambah Data Penyakit <i>Desktop</i>	122
4.23 Antarmuka Tambah Data Penyakit <i>Mobile</i>	122
4.24 Antarmuka Edit Data Penyakit <i>Desktop</i>	123
4.25 Antarmuka Edit Data Penyakit <i>Mobile</i>	124
4.26 Antarmuka Tambah Data Relasi <i>Desktop</i>	125
4.27 Antarmuka Tambah Data Relasi <i>Mobile</i>	125
4.28 Antarmuka Edit Data Relasi <i>Desktop</i>	126
4.29 Antarmuka Edit Data Relasi <i>Mobile</i>	127
4.30 Antarmuka Panduan Pengguna <i>Desktop</i>	128
4.31 Antarmuka Panduan Pengguna <i>Mobile</i>	128
4.32 Antarmuka Panduan Admin <i>Desktop</i>	129
4.33 Antarmuka Panduan Admin <i>Mobile</i>	130
4.34 Antarmuka Tentang Aplikasi <i>Desktop</i>	131
4.35 Antarmuka Tentang Aplikasi <i>Mobile</i>	132

DAFTAR LAMPIRAN

	Halaman
Lampiran 1 Surat Pengajuan Judul Skripsi	139
Lampiran 2 Surat Usulan Topik	140
Lampiran 3 Surat Usulan Dosen Pembimbing	141
Lampiran 4 Surat Keputusan Dosen Pembimbing.....	142
Lampiran 5 Surat Ijin Penelitian.....	143
Lampiran 6 Dokumentasi Penelitian.....	144

BAB I

PENDAHULUAN

1.1 Latar Belakang

Seiring dengan perkembangan ilmu ekologi, bioteknologi, dan pelestarian satwa, sekarang banyak jenis burung hias maupun ocehan. Murai batu merupakan burung teritori. Hidup di hutan-hutan dataran rendah hingga ke ketinggian 1500 meter dpl. Paling sering dijumpai pada di ketinggian 500-600 meter dpl. Tempat yang paling disukainya adalah hutan tropis yang lembab, hutan sekunder dan perkebunan. Burung ini tinggal dekat permukaan tanah, biasanya di rumpun-rumpun bambu dan pohon-pohon pendek.

Habitat asli burung murai batu tersebar mulai dari Asia bagian selatan mencakup India, Nepal, Burma dan Srilangka. Hingga ke Asia Tenggara mencakup Malaysia, Indonesia (Sunda Besar), Thailand dan Indocina. Terkadang dijumpai juga di Cina bagian selatan. Murai batu ditemukan juga di kepulauan Hawaii. Burung ini dibawa ke Hawaii dari Malaysia pada tahun 1930. Makanan burung murai batu di alam bebas adalah serangga kecil seperti semut, kutu, kaki seribu, cacing, dan belatung. Burung ini juga menyukai buah-buahan sejenis buah berry.

Murai batu (*Copsychus malabaricus*), juga dikenal sebagai Kucica hutan. Termasuk ke dalam famili *Muscicapidae* atau burung cacing. Tersebar di seluruh pulau Sumatra, Semenanjung Malaysia, dan sebagian pulau Jawa. Beberapa pakar menganggap ras dari Kalimantan Utara Murai alis putih (*Copsychus malabaricus stricklandii*) sebagai spesies tersendiri. Di habitat aslinya murai batu cenderung

memilih hutan alam yang rapat atau hutan sekunder. Murai batu merupakan kelompok burung yang dikenal sebagai teritorial dan sangat kuat dalam mempertahankan wilayahnya (*Thruses*). Burung murai batu memiliki suara kicauan yang bagus sehingga mendapat penghargaan terbaik atas nyanyian nya yang sangat indah pada tahun 1947 (*The Best Song Birds – Delacour, 1947*). Burung murai batu merupakan kelompok burung yang di gemari di kalangan para pecinta kicauan karena memiliki suara atau spesifikasi kicauan yang sangat baik. Untuk daerah Sumatra khususnya, banyak sekali di temukan burung jenis murai batu, seperti burung murai batu aceh, burung murai batu medan , burung murai nias dan murai batu lampung.

Proses diagnose jenis penyakit pada murai batu seharusnya dilakukan oleh seorang ahli di bidang tersebut. Namun, keterbatasan jumlah dan sulitnya beriteraksi langsung dengan tenaga ahli seperti peternak murai batu yang sudah berpengalaman dalam mendiagnosis dan menentukan jenis penyakit serta memberikan cara pengendalian guna mendapatkan solusinya, membuat sebagian besar peternak pemula dan pecinta burung menangani sendiri permasalahan tersebut. Kurangnya pengetahuan tentang penyakit murai batu bisa saja mengakibatkan peternak membuat kesalahan dalam menganalisa hasil diagnose tersebut. Oleh karena itu, dibutuhkan suatu sistem pakar yang mampu menentukan jenis penyakit murai batu dengan pengetahuan yang berasal langsung dari ahlinya.

Sistem pakar memiliki beberapa ciri khusus, seperti yang diungkapkan Sutojo (2001: 162) beberapa ciri tersebut adalah terbatas dalam domain keahlian tertentu dan keluarannya bersifat anjuran. 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*. Sehingga sistem pakar diharapkan membantu para peternak murai batu karena bisa memberikan anjuran dalam menentukan jenis penyakit murai batu.

Untuk membantu tercapainya hal tersebut maka dibuat suatu aplikasi sistem pakar yang mampu menentukan jenis penyakit murai batu dan dapat digunakan oleh seseorang khususnya peternak murai batu. Dalam membuat sistem pakar diperlukan 4 macam komponen, yaitu *Knowledge Base*, *Inference Engine*, *User interface*, *Development Engine* (Siswanto, 2010:129). Metode yang digunakan dalam fitur analisa adalah metode *Certainty Factor*. *Certainty factor* diperkenalkan oleh Shortliffe Buchanan dalam pembuatan MYCIN. *Certainty factor* (CF) merupakan nilai parameter klinis yang diberikan MYCIN untuk menunjukkan besarnya kepercayaan.

Berdasarkan sebuah jurnal yang berjudul “Sistem Pakar Pendeteksi Penyakit Pada Burung Murai Menggunakan Metode *Forward Chaining*” (Audrey, 2016), telah dilakukan penelitian tentang aplikasi sistem pakar untuk menentukan jenis penyakit pada murai batu. Dalam hal ini penulis menggunakan metode *forward chaining* digunakan untuk pengecekan aturan (*rule*) berdasarkan inputan jenis indukan dari pengguna dan diolah dengan basis pengetahuan.

Berdasarkan penelitian yang telah dilakukan oleh Audrey (2016) yang mengimplementasikan *Forward Chaining* dalam pembuatan sistem pakar namun belum ada sistem pakar yang dapat menentukan jenis penyakit murai batu dengan mengimplementasikan metode *Certainty Factor*. Sedangkan metode *Certainty Factor* digunakan untuk menghitung keakuratan jenis penyakit murai batu dari hasil diagnosa, maka judul penelitian ini adalah “Implementasi Metode *Certainty Factor* Untuk Mengidentifikasi Penyakit Burung Murai Batu”. Data-data yang dibutuhkan seperti gejala, jenis penyakit, didapatkan langsung dari ahli / pakar yang memahami ilmu tentang murai batu serta buku dan jurnal yang mendukung.

1.1 Identifikasi Masalah

Berdasarkan uraian latar belakang masalah tersebut di atas, teridentifikasi beberapa masalah sebagai berikut:

1. Proses diagnosa jenis penyakit pada murai batu seharusnya dilakukan oleh seorang ahli, namun pada kenyataannya para peternak menangani sendiri permasalahan tersebut.
2. Jumlah tenaga ahli penyakit burung murai batu masih terbatas.
3. Peternak kesulitan berinteraksi dengan tenaga ahli.
4. Peternak seringkali membuat kesalahan diagnosa penyakit burung murai batu.

1.2 Pembatasan Masalah

Berdasarkan permasalahan yang teridentifikasi tersebut di atas, batasan masalah yang akan dibahas dalam penelitian ini yakni bagaimana

mengimplementasikan metode *certainty factor* untuk mengidentifikasi penyakit burung murai batu.

1.3 Rumusan Masalah

Berdasarkan latar belakang, dalam menentukan jenis penyakit murai batu masih kesulitan untuk menentukan tingkat kepercayaan hasil yang mereka dapatkan. Merujuk pada permasalahan tersebut maka rumusan masalah dalam penelitian ini adalah:

1. Bagaimana mengimplementasikan metode *Certainty Factor* pada sistem pakar untuk menentukan jenis penyakit murai batu?
2. Bagaimana tingkat kualitas aplikasi yang dikembangkan berdasarkan pengujian sistem dengan *Blackbox testing*?

1.4 Tujuan

Berdasarkan rumusan masalah tersebut di atas, tujuan dari penelitian ini adalah sebagai berikut:

1. Untuk mengetahui bagaimana mengimplementasikan metode *Certainty Factor* pada sistem pakar untuk menentukan jenis penyakit murai batu.
2. Untuk mengetahui bagaimana tingkat kualitas aplikasi yang dikembangkan berdasarkan pengujian sistem dengan *Blackbox testing*.

1.5 Manfaat

Penelitian ini diharapkan dapat memberikan manfaat baik bagi pengguna, peneliti, dan semua pihak yang terkait. Manfaat tersebut antara lain:

1. Bagi Pengguna

Hasil dari penelitian ini bermanfaat bagi pengguna yaitu para pengguna, khususnya peternak murai batu karena dengan aplikasi ini dapat memudahkan peternak dalam mendapatkan jenis penyakit dengan tepat. Hasil dari konsultasi, memberikan hasil berupa nilai yang dapat digunakan untuk mengukur tingkat kepercayaan hasil gejala atas penyakit yang dihasilkan.

2. Bagi Penulis

Penelitian ini memberikan manfaat kepada penulis karena penulis akan dapat mengembangkan kemampuannya dalam hal merancang dan membangun sebuah aplikasi.

BAB II

KAJIAN PUSTAKA DAN LANDASAN TEORI

2.1 Kajian Pustaka

Menurut penelitian yang telah dilakukan oleh Audrey (2016) dengan judul “Sistem Pakar Pendeteksi Penyakit Pada Burung Murai Menggunakan Metode *Forward Chaining*”. Skripsi Jurusan Teknik Sistem Informasi, Fakultas Teknik, Universitas Nusantara Persatuan Guru Republik Indonesia. Hasil dari penelitian ini adalah dibuatnya aplikasi berbasis web yang dapat menentukan jenis penyakit pada burung murai menggunakan metode *Forward Chaining*.

Dari penelitian yang relevan tersebut, belum ada yang melakukan penelitian tentang aplikasi berbasis web yang menentukan jenis penyakit murai batu menggunakan metode *Certainty Factor*, oleh karena itu penelitian ini akan mengembangkan penelitian yang pernah dilakukan tersebut dengan membuat aplikasi berbasis web menentukan jenis penyakit murai batu menggunakan metode *Certainty Factor*.

2.2 Landasan Teori

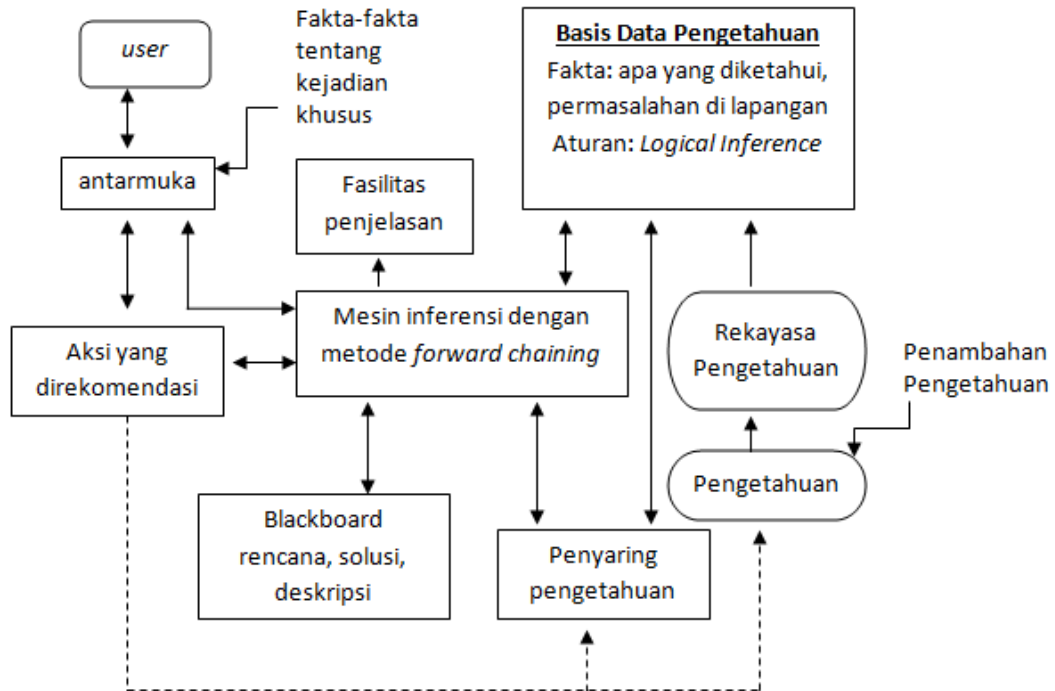
2.2.1 Sistem Pakar

Sistem pakar merupakan sebuah perangkat lunak komputer yang memiliki basis pengetahuan untuk tujuan tertentu dan menggunakan penalaran yang menyerupai seorang pakar dalam memecahkan masalah (Sembiring, 2013:7). Sistem pakar merupakan sistem yang memanfaatkan pengetahuan manusia.

Pengetahuan tersebut direkam dalam komputer untuk memecahkan persoalan yang biasanya memerlukan keahlian manusia (Turban et.al, 2005: 9). Sistem pakar dirancang agar dapat menyelesaikan permasalahan tertentu sesuai bagaimana cara manusia menyelesaikan permasalahan tersebut, secara otomatis.

Sistem pakar yang mencoba memecahkan masalah yang biasanya hanya bisa dipecahkan oleh seorang pakar, dipandang berhasil ketika mampu mengambil keputusan seperti yang dilakukan oleh pakar aslinya baik dari sisi proses pengambilan keputusannya maupun hasil keputusan yang diperoleh. Menurut Kusrini (2008: 3) ada banyak cara untuk merepresentasikan pengetahuan, di antaranya adalah logika (*logic*), jaringan semantik (*semantic nets*), *Object-Atribut-Value* (OAV), bingkai (*frame*), dan kaidah produksi (*production rule*).

Menurut Honggowibowo (2009: 188), sistem pakar dapat ditampilkan dalam dua macam, yaitu: pengembangan dan konsultasi. Pengembangan digunakan untuk membangun sistem pakar berbasis komponen dan memasukkan pengetahuan ke dalam basis data pengetahuan. Konsultasi digunakan oleh *user* untuk memperoleh pengetahuan dan berkonsultasi. Komponen-komponen yang ada pada sistem. Mesin inferensi merupakan otak dari aplikasi sistem pakar. Bagian inilah yang menuntun *user* untuk memasukkan fakta sehingga diperoleh suatu kesimpulan. Apa yang dilakukan oleh mesin inferensi ini didasarkan pada pengetahuan yang ada dalam basis pengetahuan. Menurut Turban (2005: 10) seperti yang ditampilkan pada gambar 2.1. Rincian dari masing-masing komponen sistem pakar sebagai berikut:



Gambar 2.1 Struktur Sistem Pakar Metode *Forward Chaining*

1. Basis pengetahuan (*knowledge base*). Berisi pengetahuan-pengetahuan yang dibutuhkan untuk memahami, memformulasikan dan memecahkan persoalan.
2. Motor inferensi (*inferensi engine*). *Forward Chaining* merupakan *group* dari multiple inferensi yang melakukan pencarian dari suatu masalah kepada solusinya.
3. *Blackboard*. Merupakan area kerja memori yang disimpan sebagai database untuk deskripsi persoalan terbaru yang ditetapkan oleh data input dan digunakan juga untuk perekaman hipotesis dan keputusan sementara.
4. Subsistem akuisisi pengetahuan. Akuisisi pengetahuan adalah akumulasi, transfer dan informasi keahlian pemecahan masalah dari pakar atau sumber pengetahuan terdokumentasi ke program komputer untuk membangun atau memperluas basis data pengetahuan.

5. Antarmuka pengguna. Digunakan untuk media komunikasi antara *user* dan program.
6. Subsistem penjelasan. Digunakan untuk melacak respon dan memberikan penjelasan tentang kelakuan sistem pakar secara interaktif melalui pertanyaan.
7. Sistem penyaring pengetahuan.

2.2.1.1 Metode Pemecahan Masalah

Forward *chaining* merupakan mesin inferensi *group multiple* dari inferensi yang melakukan pencarian dari suatu masalah untuk mendapatkan solusinya. Sistem pakar yang dibuat menggunakan framework *codeigniter* dan MySQL sebagai basis datanya. Menurut Kusri (2006: 35) inferensi merupakan proses untuk menghasilkan informasi dari fakta yang diketahui atau diasumsikan. Inferensi adalah konklusi logis (*logical conclusion*) atau implikasi berdasarkan informasi yang tersedia. Dalam sistem pakar, proses inferensi dilakukan dalam suatu modul yang disebut *inferensi engine* (mesin inferensi). Ada dua metode inferensi yang penting dalam sistem pakar, yaitu runut maju (*forward chaining*) dan runut balik (*backward chaining*).

Forward Chaining adalah *data-driven* karena inferensi dimulai dengan informasi yang tersedia dan baru konklusi diperoleh. Contoh sederhana dari *forward chaining* seperti berikut ini: misalkan anda sedang mengemudi dan tiba-tiba anda melihat mobil polisi dengan cahaya kelap- kelip dan bunyi sirine. Dengan *forward chaining* mungkin anda akan berkesimpulan bahwa polisi ingin

anda atau seseorang untuk berhenti. Itu adalah fakta awal yang mendukung dua kemungkinan konklusi. Jika mobil polisi membuntuti dibelakang anda atau polisi melambatkan tangan memberhentikan anda, maka kesimpulan lebih lanjut adalah polisi ingin anda yang berhenti (Putra et al, 2013: 42).

Operasi dari metode *forward chaining* dimulai dengan memasukkan sekumpulan fakta yang diketahui ke dalam memori kerja (*working memory*), kemudian menurunkan fakta baru berdasarkan aturan yang premisnya cocok dengan fakta yang diketahui. Proses ini dilanjutkan sampai dengan mencapai *goal* atau tidak ada lagi aturan yang premisnya cocok dengan fakta yang diketahui. Langkah-langkah yang harus dilakukan dalam membuat sistem *forward chaining* berbasis aturan, yaitu:

1. Pendefinisian Masalah

Tahap ini meliputi pemilihan domain masalah dan akuisisi pengetahuan.

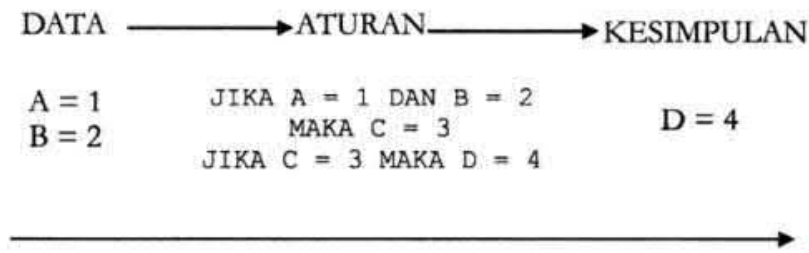
2. Pendefinisian Data Input

Sistem *forward chaining* memerlukan data awal untuk memulai inferensi.

3. Pendefinisian Struktur Pengendalian Data

Aplikasi yang kompleks memerlukan premis tambahan untuk membantu mengendalikan.

Berikut ini menunjukkan bagaimana cara kerja metode inferensi *forward chaining*

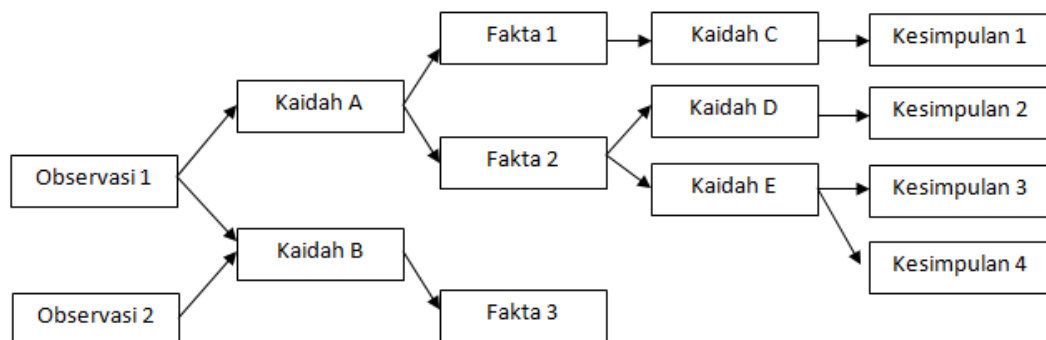


Gambar 2.2 Cara Kerja *Forward Chaining*

Metode *forward chaining* cocok digunakan untuk menangani masalah pengendalian (*controlling*) dan peramalan (*prognosis*). Berikut ini model penyelesaian dengan menggunakan metode *forward chaining*:

JIKA Penderita terkena penyakit *epilepsy idiopatik* dengan CF 0,4 - 0,6

MAKA Berikan obat *carbamazepine*



Gambar 2.3 Diagram Pelacakan *Forward Chaining* (Sembiring, 2013: 8)

2.2.1.2 Metode *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.

Implementasi *Certainty Factor* berfokus pada penghitungan keluaran hasil prosentasi Penyakit. Metode ini digunakan agar pengguna dapat menentukan Jenis Penyakit mana yang lebih dominan dengan data Gejala yang sama, sehingga pengguna tidak bingung Jenis Penyakit mana yang sesuai dengan Gejala yang dipilih. Rumus dari *Certainty Factor* gabungan menurut David Heckerman (1992:4) adalah:

Rumus CF

$$CF = MB - MD$$

CF : Faktor Kepastian (*Certainty Factor*)

MB : Ukuran Kepercayaan (*Measure of Believe*)

MD : Ukuran Ketidakpercayaan (*Measure of Disbelieve*)

Rumus CF Gabungan Bernilai Positif

$$CF_{total} = CF_n + CF_{n+1}(1 - CF_n)$$

CF Total : CF akhir

CF_n : CF ke-n

CF_{n+1} : CF ke-n+1

Contoh kasus :

Berikut ini merupakan daftar Gejala yang dipilih oleh pengguna.

Tabel 2.1 contoh kasus certainty factor

Kode	Nama Gejala
G001	Volume Menurun
G002	Sering Minum Air Setelah Berkicau

Dilihat pada tabel 2.1 didapat aturan sebagai berikut.

Penyakit : Suara

Serak (P001)

Rule 1 : *IF* G001 *THEN* H001

(CF = 0,57) Rule 2 : *IF* G002

THEN H001 (CF = 0,67)

Dari *rule* tersebut dapat diketahui nilai CF masing-masing Gejala. Dari nilai CF tersebut dapat dihitung CF gabungan. CF1 digabungkan dengan CF2. Lebih jelasnya akan diberikan contoh penggunaan rumus CF gabungan bernilai positif sesuai dengan persamaan (1).

Diketahui :

$$CF1 = 0,57$$

$$CF2 = 0,67$$

Perhitungan CF gabungan :

$$CF_{gabungan} = CF1 + CF2 (1 - CF1) \dots\dots\dots (1)$$

$$CF_{gabungan} = 0,57 + 0,67 (1 - 0,57)$$

$$CF_{gabungan} = 0,57 + 0,67 (0,43)$$

$$CF_{gabungan} = 0,57 + 0,67 (0,43)$$

$$CF_{gabungan} = 0,57 + 0,288$$

$$CF_{gabungan} = 0,858$$

Jadi kemungkinan Jenis Penyakit adalah Suara serak dengan nilai CF 0.858 (pasti) yang sesuai dengan tabel 2.1.

Pada perhitungan nilai *Certainty Factor* gabungan, maka kedua nilai digabungkan. Apabila ada jenis Gejala lain yang menyebabkan Penyakit yang sama maka nilainya akan digabungkan dengan nilai *Certainty Factor* yang telah digabungkan sebelumnya membentuk nilai gabungan baru.

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 Kusri (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.

Faktor kepastian yang diisikan oleh pakar bersama dengan aturan yaitu dengan menggali dari hasil wawancara dengan pakar yang bersangkutan. Nilai CF (*rule*) didapat dari interpretasi "*term*" dari pakar menjadi nilai sebuah *CF* tertentu seperti terlihat pada Tabel 2.2 menurut Sutojo (2011).

Tabel 2.2 Tabel Nilai CF

Uncertain Term	CF
Unknown (tidak tahu)	0.0 - 0.29
Maybe (mungkin)	0.3 - 0.49
Probably (kemungkinan besar)	0.5 - 0.69
Almost certainly (hampir pasti)	0.7 - 0.89
Definitely (pasti)	0.9 - 1.00

Contoh Kasus:

Jika gejala sering minum setelah berkicau dan volume kicau menurun drastis, maka “*Almost certainly* (hampir pasti)” jenis penyakitnya adalah suara serak.

Rule : IF (sering minum setelah berkicau AND volume kicau menurun drastis)
THEN jenis penyakitnya = suara serak (CF = 0,8).

2.2.2 Murai Batu

Burung Murai Batu atau di sebut juga Kucica hutan dengan nama latin (*Copsychus malabaricus*). Merupakan salah satu burung faforit para kicau mania di Indonesia tidak heran karena burung ini dalam segi suara cukup variatif dan bagus sehingga mendapat penghargaan terbaik atas nyanyian nya yang sangat indah pada tahun 1947 (*The Best Song Birds – Delacour, 1947*). Karena cukup banyak penggemarnya menjadikan keberadaanya di alam liar terancam akibat perburuan liar,untungnya sekarang sudah mulai bermunculan para penangkar burung ini selain dapat menyelamatkan populasi Murai Batu di alam liar dan di tunjang dari stabilnya harga burung ini di pasaran sehingga mendatangkan nilai ekonomis juga bagi para penangkarnya.

Burung ini Termasuk ke dalam famili Muscicapidae atau burung cacing. Penyebaranya mencakup seluruh pulau Sumatra, Semenanjung Malaysia, Pulau Kalimantan dan sebagian pulau Jawa. Namun beberapa pakar menganggap ras dari Kalimantan Utara masuk dalam Kucica Alis-putih (*Copsychus malabaricus stricklandii*) sebagai spesies tersendiri. Di habitat aslinya Kucica Hutan atau sering di sebut Burung murai batu cenderung memilih hutan alam yang rapat atau hutan

sekunder. Murai batu merupakan kelompok burung yang dikenal sebagai teritorial dan sangat kuat dalam mempertahankan wilayahnya (*Thruses*).

Burung Murai Batu yang banyak beredar di para Kicau Mania di Indonesia ada beberapa Jenis Burung Murai Batu dan umumnya diberikan nama sesuai daerah asalnya, di bawah ini akan saya sebutkan Macam Burung Murai Batu lengkap, antara lain :

1. Murai Batu Medan

Burung ini mendapatkan namanya dari asal daerahnya yaitu Medan. Power kicauan yang dimiliki keras dan melengking juga Burung Murai Batu Medan di kenal sebagai Fighter sejati yang mempunyai mental pemberani. Saat mendengar kicauan burung lain dia akan mengembangkan dada dan perutnya nialai plus ini sering disebut dengan istilah Ndegek dan Ngobra.

Ciri khusus Muari Batu jenis ini ekornya melengkung dan panjang dengan ukuran sekitar 27 s/d 30 cm , ada strip hitam tegas merata kanan kiri dari pangkal ekor secara Horisontal membentuk huruf U.

2. Murai Batu Nias

Jenis Muari Batu ini tidak cukup sulit di kenali, ciri-cirinya mempunyai ekor panjang sampai 17-20cm yang bewarna hitam pekat. Kekuatan kicauanya keras dan memiliki mental cukup kuat.Kadang juga konsumen ada yang menyebutnya Murai Batu Aceh.

Bulu ekor hitam dari pangkal sampai ujung kadang ada titik putih di ujung ekor , disebut juga Murai Batu *Black Tail*.

3. Murai Batu Lampung/Jambi

Ciri-ciri dari jenis murai batu ini adalah mempunyai ekor yang lebih pendek dan kaku daripada jenis lain. Panjangnya bisa antara 12 s/d 18 cm. Karakter suara yang dikeluarkannya cenderung berulang-ulang, namun semua terdengar cukup merdu dan keras. Nilai plus jenis ini yaitu mempunyai stamina yang kuat sehingga tidak mudah letih jika saat berkicau.

Bulu putih ada semburat hitam tipis/sedikit di pangkal ekor ke 4 ada hitam membelah ekor secara vertical

4. Murai Batu Aceh

Jenis ini merupakan Murai yang sering di temukan di wilayah pulau Sumatera bagian Utara, khususnya daerah Provinsi Aceh. Ciri-ciri Jenis Murai Batu Aceh antara lain panjang ekornya beragam rata-rata 19-30cm. Selain itu, kicauannya sering mengerluarkan suara ngeroll panjang dan gemar sekali mengeluarkan suara nembak. Murai ini juga termasuk Fighter yang cukup baik meskipun ukuran bodi nya relatif lebih kecil.

Ciri Bulu Ekor Murai Batu Aceh yaitu Strip Horisontal merata kanan dan kiri.

5. Murai Batu Jawa (Larwo)

Sepintas tidak ada perbedaan mencolok antara Murai Batu Jawa atau juga bisa di sebut larwo (lar dowo) ini dengan murai batu jenis lain yang berasal dari daerah Sumatera ataupun Kalimantan. Namun kalau kita perhatikan perbedaannya yaitu: ukuran tubuhnya yang lebih kecil dari murai batu Sumatera dan juga batas garis dada yang berwarna hitam yang berakhir di perutnya. Sementara murai batu Sumatera dan Kalimantan rata-rata batas

hitamnya sampai bagian dada saja. Perbedaan lainnya adalah performa ketika bersuara, yakni bulu-bulu di kepalanya akan berdiri seperti jambul dan kicaunya cenderung ngeban.

6. Murai Batu Kalimantan (Borneo)

Burung Murai Batu asal Pulau Kalimantan (Borneo) juga mempunyai penampilan dan kemampuan kicau yang mengagumkan. Di habitat aslinya, ia dikenal sebagai burung agresif yang selalu ngotot jika mendengar kicaun burung lain. Burung murai yang satu ini mempunyai ciri panjang ekor yang bisa tumbuh 8 hingga 13. Meskipun ukuran ekornya relatif lebih kecil dadanya akan mengembang ketika berkicau. Ada bulu putih polos dari pangkal ekor sampai ke ujung ekor kadang ada semburat hitam tipis.

2.2.3 Penyakit Unggas

2.2.3.1 Penyakit Berak Kapur

Penyakit berak kapur adalah penyakit dengan nama latin *pullorum disease*, Gejala yang muncul pada penyakit ini antara lain adalah nafsu makan berkurang, badan kurus, bulu kusam dan berkerut, produksi telur berkurang, berak berwarna keputih-putihan.

2.2.3.2 Penyakit Flu Burung

Penyakit flu burung memiliki nama latin *Avian Influenza*, Pertama kali terjadi di Italia sekitar tahun 1800 dan menyebar luas sampai tahun 1930, Gejala yang muncul pada penyakit ini adalah nafas sesak, nafas ngorok, Nampak membiru,

keluar cairan berbusa dari mata, kepala bengkak, dan menyebabkan kematian secara mendadak.

2.2.3.3 Penyakit Tetelo (*Newcastle Disease*)

Penyakit ini pertama ditemukan oleh Doyle pada tahun 1927, di daerah *Newcastle on Tyne*, Inggris dan memiliki gejala sebagai berikut, tanpa lesu, mencret kehijau-hijauan, sempoyongan, kepala berputar.

2.2.3.4 Penyakit Produksi Telur (*Egg Drop Syndrome*)

Merupakan penyakit yang pertamakali dilaporkan oleh *Van Eck* pada tahun 1976 di *Netherlands* pada pakar kesehatan ternak, Gejala yang dialami ayam ketika mengidap penyakit ini diantaranya nafas cepat, produksi telur menurun, kualitas telur jelek, mencret kehijau-hijauan.

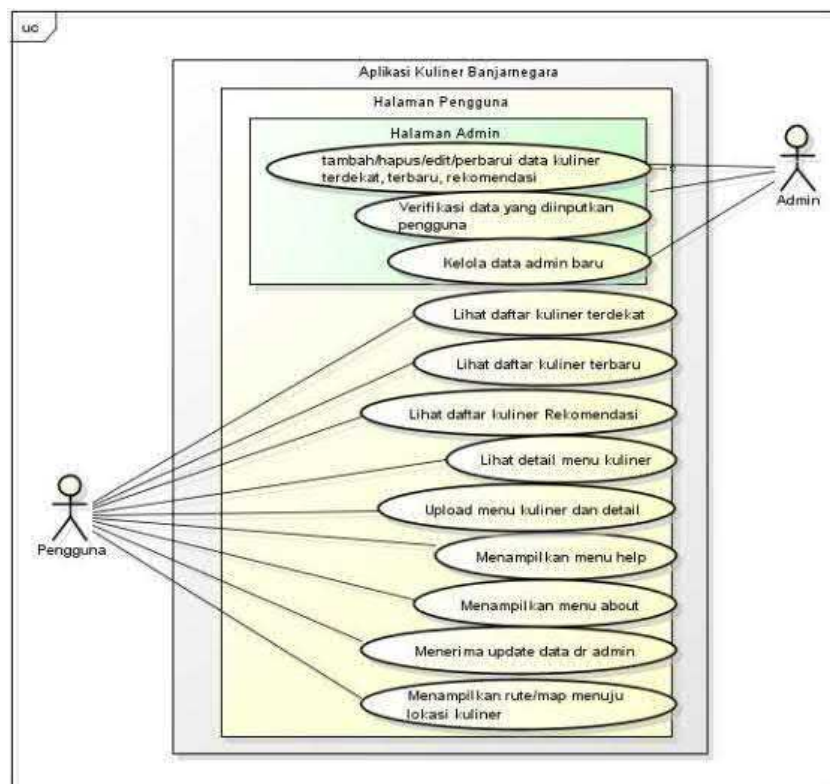
2.2.4 Rancang Bangun Perangkat Lunak

2.2.4.1 *Unified Modeling Language* (UML)

Unified Modeling Language (UML) adalah sebuah bahasa standard untuk menulis *blueprint* sebuah perangkat lunak. UML dapat digunakan untuk menggambarkan, menetapkan, membangun dan mendokumentasikan artefak dari sistem perangkat lunak intensif (Pressman, 2010:841). Beberapa diagram yang digunakan untuk pemodelan perangkat lunak yaitu:

1. Use Case Diagram

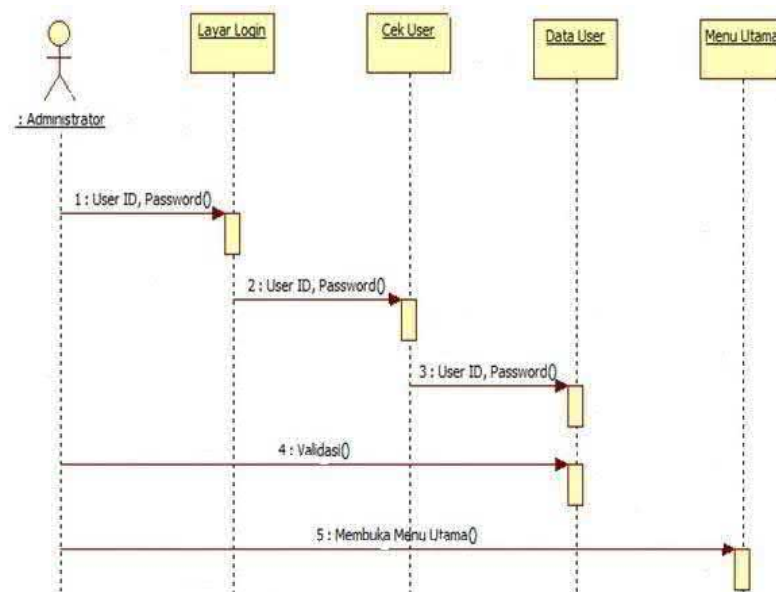
Use case diagram digunakan untuk membantu dalam menentukan fungsi dan fitur perangkat lunak yang berasal dari perspektif pengguna. Sebuah *use case* mendeskripsikan bagaimana seorang pengguna berinteraksi dengan sistem dengan mendefinisikan langkah-langkah yang diperlukan untuk mencapai tujuan tertentu. Sebuah UML *use case diagram* menggambarkan keseluruhan permasalahan pengguna dan bagaimana mereka berhubungan. Pada diagram ini, sosok tongkat mewakili seorang aktor, dan permasalahan pengguna ditampilkan oval, antara aktor dan permasalahan pengguna dihubungkan dengan garis (Pressman, 2010:843). Contoh penggunaan *use case diagram* dapat dilihat pada gambar 2.4 berikut ini.



Gambar 2.4 Contoh Penggunaan *Use Case Diagram*

2. Sequence Diagram

Sequence diagram digunakan untuk menunjukkan komunikasi dinamis antara objek selama menjalankan tugasnya. Ini menunjukkan urutan temporal dimana pesan dikirim antara objek untuk menyelesaikan tugas. *Sequence diagram* menunjukkan pemanggilan metode menggunakan panah horizontal dari *the caller* menuju ke *the called*, yang diberikan label sesuai dengan nama metode dan opsional yang terdiri dari parameter, tipe parameter dan tipe *return* (Pressman, 2010:848-849). Contoh penggunaan *sequence diagram* dapat dilihat pada gambar 2.5 berikut ini.

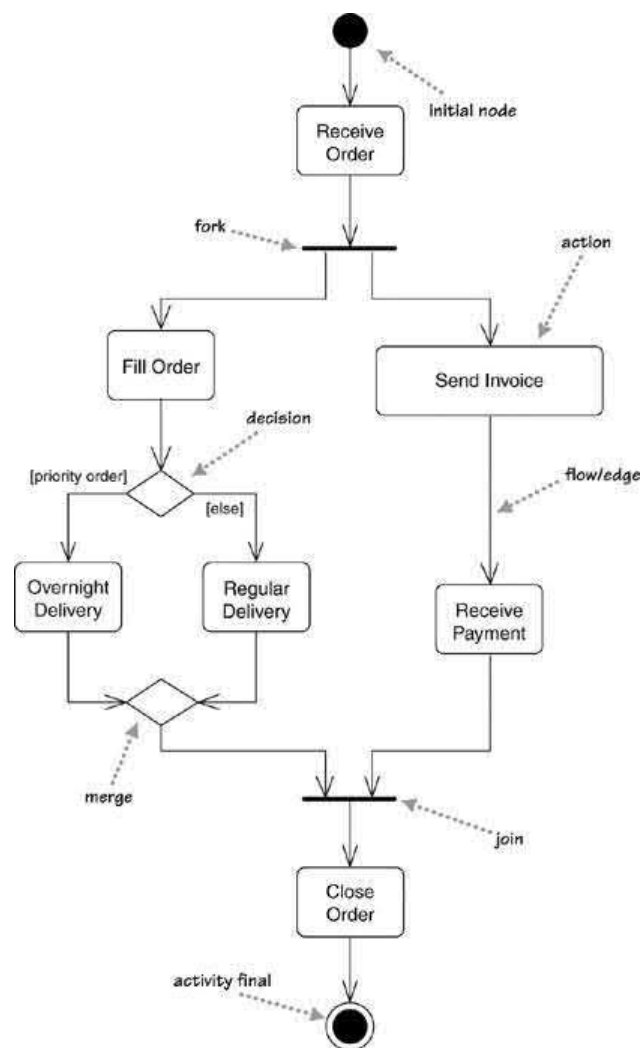


Gambar 2.5 Contoh Penggunaan *Sequence Diagram*

Pada contoh *sequence diagram* diatas, bisa dilihat bahwa yang menjadi aktor adalah administrator. *Activation boxes* biasanya memiliki garis yang memberitahu aktifitas yang terjadi ketika aktor atau objek berinteraksi ke objek lain.

3. Activity Diagram

Activity diagram menggambarkan perilaku dinamis sistem atau bagian dari sistem melalui aliran kontrol antara tindakan sistem. *Activity diagram* mirip dengan *flowchart* namun pada *activity diagram* dapat menunjukkan arus secara bersamaan (Pressman, 2010:853). Contoh penggunaan *activity diagram* dapat dilihat pada gambar 2.6 berikut ini.



Gambar 2.6 Contoh Penggunaan *Activity Diagram*

Pada contoh *activity diagram* diatas, pertama - tama pegawai menerima pesanan, lalu pegawai bisa langsung mengisi pesanan (*fill order*) dan mengirimkan

tagihan (*send invoice*) saat pegawai mengisi pesanan, pelanggan bisa memilih prioritas pesanan. Jika iya maka akan masuk ke *overnight delivery* tetapi jika tidak akan masuk ke *regular delivery*.

Setelah semua selesai maka pesanan akan ditutup dan pelanggan bisa menunggu pesannya (*close order*). *Decision* digunakan untuk aktifitas conditional seperti ya atau tidak. Berikut beberapa tambahan definisi diagram-diagram yang terdapat dalam UML Menurut Sri Dharwiyanti (2003) yaitu:

a. *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 : 1. Nama (dan *stereotype*) 2. Atribut 3. Metoda Atribut dan metoda dapat memiliki salah satu sifat berikut:

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

b. *Statechart* Diagram

Statechart diagram menggambarkan transisi dan perubahan keadaan (dari satu state ke state lainnya) suatu objek pada sistem sebagai akibat dari stimuli yang diterima. Pada umumnya *Statechart* diagram menggambarkan class tertentu (satu class dapat memiliki lebih dari satu *Statechart* diagram).

c. *Collaborative* Diagram

Collaboration diagram juga menggambarkan interaksi antar objek seperti *sequence* diagram, tetapi lebih *menekankan* pada peran masing-masing objek dan bukan pada waktu penyampaian message. Setiap message memiliki *sequence number*, di mana *message* dari level tertinggi memiliki nomor 1. *Messages* dari level yang sama memiliki *prefiks* yang sama

d. *Component* Diagram

Component diagram menggambarkan struktur dan hubungan antar komponen piranti lunak, termasuk ketergantungan (*dependency*) di antaranya.

Komponen piranti lunak adalah modul berisi *code*, baik berisi *source code* maupun *binary code*, baik *library* maupun *executable*, baik yang muncul pada *compile time*, *link time*, maupun *run time*. Umumnya komponen terbentuk dari beberapa *class* dan/atau *package*, tapi dapat juga dari komponen-komponen yang lebih kecil. Komponen dapat juga berupa *interface*, yaitu kumpulan layanan yang disediakan sebuah komponen untuk komponen lain.

e. *Deployment Diagram*

Deployment/physical diagram menggambarkan detail bagaimana komponen di-*deploy* dalam infrastruktur sistem, di mana komponen akan terletak (pada mesin, server atau piranti keras apa), bagaimana kemampuan jaringan pada lokasi tersebut, spesifikasi server, dan hal- hal lain yang bersifat fisik.

Sebuah node adalah server, workstation, atau piranti keras lain yang digunakan untuk men-*deploy* komponen dalam lingkungan sebenarnya. Hubungan antar node (misalnya TCP/IP) dan requirement dapat juga didefinisikan dalam diagram ini.

Pada penelitian ini dipakai *Use case Diagram* dan *Activity Diagram* saja karena aplikasi yang di kembangkan berbasis website.

2.2.4.2 Basis Data

Basis data diartikan sebagai kumpulan data tentang suatu benda atau kejadian yang saling berhubungan satu sama lain. Menurut Elmasri R.(1994):

1. Basis data merupakan penyajian suatu aspek dari dunia nyata ("*real world*" atau "*miniworld*"), Misalnya basis data perbankan, perpustakaan, pertanahan, perpajakan dan sebagainya.

2. Basis data merupakan kumpulan data dari berbagai sumber yang secara logika mempunyai arti implisit. Sehingga data yang terkumpul secara acak dan tanpa mempunyai arti, tidak dapat disebut basis data.
3. Basis data perlu dirancang, dibangun, dan data dikumpulkan untuk suatu tujuan. Basis data dapat digunakan oleh beberapa pemakai dan beberapa aplikasi yang sesuai dengan kepentingan pemakai (Waljiyanto, 2003).

2.2.4.3.1 Database management System (DBMS)

Database Management System adalah kumpulan *file* yang saling berkaitan bersama dengan program pengelolaannya. Menurut Kadir (2003:17) pengertian DBMS adalah suatu program komputer yang digunakan untuk memasukkan, mengubah, menghapus, memanipulasi dan memperoleh data/informasi dengan praktis dan efisien.

2.2.4.3.2 Model Basis Data

Model basis data menyatakan hubungan antar rekaman yang tersimpan dalam basis data. Beberapa literatur menggunakan istilah struktur data logis untuk menyatakan keadaan ini. Menurut Abdul Kadir (2003:22), model dasar yang paling umum ada 3 macam yaitu:

1. Model Hirarkis

Model hirarkis biasanya disebut model pohon karena menyerupai pohon yang dibalik. Contoh produk DBMS yang menggunakan Model Hirarkis adalah IMS

(*Information Management System*), yang dikembangkan oleh 2 perusahaan yaitu IBM dan *Rockell International Corporation*.

2. Model Jaringan

Model jaringan distandarisasikan pada tahun 1971 oleh *Data Base Task Group/DBTG*. Model ini juga disebut model CODASYL (*Confrence on Data Systems Languages*), karena DBTG adalah bagian dari CODASYL.

3. Model Relasional

Model Relasional merupakan model yang paling sederhana sehingga mudah digunakan dan dipahami oleh pengguna, serta yang paling populer saat ini. Model ini menggunakan sekumpulan tabel berdimensi dua, dengan masing-masing relasi tersusun atas tupel atau baris dan atribut. Relasi ini dirancang sedemikian rupa sehingga dapat menghilangkan kemubaziran data dan menggunakan kunci tamu untuk berhubungan dengan relasi lain. Contoh produk DBMS yang menggunakan model relasional adalah CAIDBMS/DB.

2.2.4.3.3 Perancangan Basis Data

Proses perancangan data, terlepas dari masalah yang ditangani, dibagi menjadi tiga tahapan yaitu :

1. Perancangan Basis Data secara Konseptual

Perancangan basis data secara konseptual merupakan upaya untuk membuat model yang masih bersifat konsep.

2. Perancangan Basis Data secara Logis

Perancangan basis data secara logis merupakan tahapan untuk memetakan model konseptual ke basis data yang akan dipakai (model relasional, hirarkis atau jaringan).

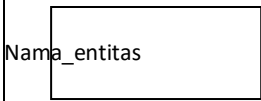
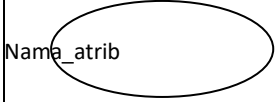
3. Perancangan Basis Data secara Fisis

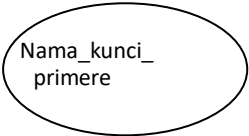
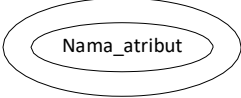
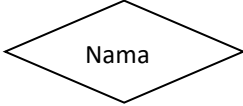

Perancangan basis data secara fisis merupakan tahapan untuk menuangkan perancangan basis data yang bersifat logis menjadi Basis data fisis yang tersimpan pada media penyimpanan eksternal. Perancangan basis data merupakan langkah unruk menentukan basis data yang diterapkan dapat mewakili seluruh kebutuhan pengguna. Penyusunan basis data ini berlandaskan kamus aliran data yang telah dibuat pada tahapan sebelumnya.

2.2.4.3.4 *Entity Relationship Diagram (ERD)*

Pemodelan awal basis data yang paling banyak digunakan adalah *Entity Relationship Diagram (ERD)*. ERD dikembangkan berdasarkan teori himpunan dalam bidang matematika. ERD digunakan untuk pemodelan basis data relasional (Rosa dan Shalahuddin, 2011). Simbol-simbol pada ERD ditunjukkan pada tabel 2.2 dibawah ini :

Tabel 2.3 Simbol-simbol ERD

Nama	Keterangan	Simbol
Entitas/ <i>entity</i>	Entitas merupakan data inti yang akan disimpan; bakal table pada basis data	
Atribut	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas.	

Nama	Keterangan	Simbol
Atribut kunci primer	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas dan digunakan sebagai kunci akses <i>record</i> yang diinginkan; biasanya berupa id.	
Atribut Multi nilai / multi value	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas	
Relasi	Relasi yang menghubungkan antar entitas; biasa diawali dengan kata kerja	
Asosiasi/ Association	Penghubung antara relasi dan entitas dimana dikedua ujungnya memiliki <i>multiplicity</i> kemungkinan jumlah pemakai	

2.2.4.3 MySQL

MySQL adalah sistem management *database* relasional. Suatu relasional menyimpan data dalam tabel-tabel terpisah. Hal ini memungkinkan kecepatan dan fleksibilitas. Tabel-tabel yang dihubungkan dengan relasi yang ditentukan membuatnya bisa mengkombinasi data dari beberapa tabel dari suatu permintaan. Bagian SQL dari kata MySQL berasal dari “*Structured Query Language*” bahasa paling umum yang dipergunakan untuk mengakses *database*. Konektivitas, kecepatan dan keamannya membuat MySQL cocok untuk pengaksesan *database* pada internet. MySQL merupakan sistem *client/server* yang terdiri dari SQL server *multithreaded* yang memungkinkan *backend* yang berbeda, sejumlah program *client* dan *library* yang berbeda, *tool* administratif, dan beberapa antarmuka pemrograman. MySQL juga tersedia sebagai *library* yang bisa digabungkan di aplikasi (Utdirartatmo, 2002).

2.2.4.4 PHP (*Hypertext Preprocessor*)

PHP adalah bahasa *scripting* yang menyatu dengan HTML dan dijalankan pada *server side*. Artinya semua sintaks yang kita berikan akan sepenuhnya dijalankan pada *server* sedangkan yang dikirimkan ke *browser* hanya hasilnya saja (Anonim, 2003). Menurut Sutarman (2007) PHP merupakan bahasa pemrograman yang dapat disisipkan dalam *script* HTML. Banyak sintaks di dalamnya yang mirip dengan bahasa C, Java dan Perl.

PHP bersifat bebas dipakai, tidak perlu membayar apapun untuk menggunakan perangkat lunak ini. Kode PHP diawali dengan tanda "<?php" dan diakhiri dengan ">". Pasangan keduanya berfungsi sebagai tag kode PHP. Berdasarkan tag ini, *server* dapat memahami kode PHP dan kemudian memprosesnya. Salah satu kelebihan dari PHP adalah mampu berkomunikasi dengan berbagai *database* yang terkenal. Dengan demikian, menampilkan data yang bersifat dinamis, yang diambil dari *database*, merupakan hal yang mudah diimplementasikan. PHP juga sangat cocok untuk membangun halaman-halaman web dinamis. (Abdul Kadir, 2002). Beberapa keunggulan PHP adalah:

1. PHP memiliki tingkat akses yang lebih cepat.
2. PHP memiliki tingkat *lifecycle* yang cepat sehingga selalu mengikuti perkembangan teknologi internet.
3. PHP memiliki tingkat keamanan yang tinggi.
4. PHP mampu berjalan di beberapa *server* yang ada, misalnya Apache, Microsoft IIS, PWS, AOLserver, ptpd, dan Xitami.

5. PHP mendukung akses ke beberapa *database* yang sudah ada, baik yang bersifat gratis ataupun komersial. *Database* itu antara lain MySQL, PostgreSQL, mSQL, Infomix, dan MicrosoftSQLserver.

2.2.5 Pengujian Perangkat Lunak

Pengujian perangkat lunak adalah satu elemen dari suatu topik yang luas yang sering disebut dengan *verification and validation testing (V&V)*. Verifikasi mengacu pada serangkaian kegiatan yang memastikan perangkat lunak dapat melakukan suatu fungsi tertentu yang telah ditentukan. Validasi mengacu pada serangkaian kegiatan berbeda yang memastikan perangkat lunak telah sesuai dengan kebutuhan pengguna (Pressman, 2010:450-451). Teknik pengujian perangkat lunak yang digunakan pada penelitian ini adalah *validation testing*. *Validation testing* dilakukan dengan *blackbox testing*.

Blackbox Testing

Menurut Pressman (2010, 495) *Black-Box testing* berfokus pada persyaratan fungsional perangkat lunak yang memungkinkan engineers untuk memperoleh set kondisi input yang sepenuhnya akan melaksanakan persyaratan fungsional untuk sebuah program. *Black-Box testing* berusaha untuk menemukan kesalahan dalam kategori berikut:

1. Fungsi yang tidak benar atau fungsi yang hilang
2. Kesalahan antarmuka
3. Kesalahan dalam struktur data atau akses database eksternal

4. Kesalahan perilaku (*behavior*) atau kesalahan kinerja
5. Inisialisasi dan pemutusan kesalahan

Pengujian *Black Box Testing* memiliki beberapa ciri sebagai berikut:

1. *Black box testing* berfokus pada kebutuhan fungsional pada software, berdasarkan pada spesifikasi kebutuhan dari software.
2. *Black box testing* bukan teknik alternatif daripada *white box testing*. Lebih daripada itu, ia merupakan pendekatan pelengkap dalam mencakup error dengan kelas yang berbeda dari metode *white box testing*.
3. *Black box testing* melakukan pengujian tanpa pengetahuan detail struktur internal dari sistem atau komponen yang dites. juga disebut sebagai *behavioral testing*, *specification-based testing*, *input/output testing* atau *functional testing*.

Menurut Crosby (1979: 34) mendefinisikan kualitas atau mutu sebagai “*conformance to requirements*”, selama seseorang dapat berdebat tentang perbedaan antara kebutuhan, keinginan dan kemauannya, definisi kualitas harus mempertimbangkan perspektif pemakai tersebut. Kunci utama yang penting bagi mereka adalah bagaimana prioritasnya tentang metode yang dibangun, dibungkus untuk mendukung sebuah produk.

Dari teori di atas sebuah perangkat lunak dikatakan berkualitas apabila memenuhi tiga ketentuan pokok yaitu :

1. Perangkat lunak dapat memenuhi kebutuhan pemakai. Apabila sebuah perangkat lunak tidak dapat memenuhi kebutuhan pengguna, maka perangkat lunak tersebut dikatakan tidak atau kurang memiliki kualitas.

2. Perangkat lunak memenuhi standar pengembangan *software*. Apabila sebuah perangkat lunak dikembangkan dengan tidak mengikuti metodologi standar, maka hampir dapat dipastikan bahwa kualitas yang baik akan sulit tercapai.
3. Perangkat lunak memenuhi sejumlah kriteria implisit. Apabila sebuah perangkat lunak tidak memenuhi salah satu kriteria implisit tersebut, maka perangkat lunak tersebut tidak dapat dikatakan memiliki kualitas yang baik.

Pengujian *blackbox* pengujian yang bertujuan untuk mengamati fungsionalitas dari perangkat lunak yang dikembangkan. Aspek *Functionality* memiliki lima sub karakteristik sebagai berikut:

1. *Suitability*

Kemampuan perangkat lunak untuk menyediakan serangkaian fungsi yang sesuai untuk tugas tertentu dan tujuan pengguna.

2. *Accuracy*

Kemampuan perangkat lunak dalam memberikan hasil yang presisi dan benar sesuai dengan kebutuhan.

3. *Security*

Kemampuan perangkat lunak untuk mencegah akses yang tidak diinginkan, menghadapi penyusup maupun otorisasi dalam modifikasi data.

4. *Interoperability*

Kemampuan perangkat lunak untuk berinteraksi dengan satu atau lebih sistem tertentu.

5. *Compliance*

Kemampuan perangkat lunak dalam memenuhi standar dan kebutuhan sesuai peraturan yang berlaku.

2.3 Kerangka Berpikir

Murai batu merupakan burung teritori. Hidup di hutan-hutan dataran rendah hingga ke ketinggian 1500 meter dpl. Paling sering dijumpai pada di ketinggian 500-600 meter dpl. Tempat yang paling disukainya adalah hutan tropis yang lembab, hutan sekunder dan perkebunan. Burung ini tinggal dekat permukaan tanah, biasanya di rumpun-rumpun bambu dan pohon-pohon pendek.

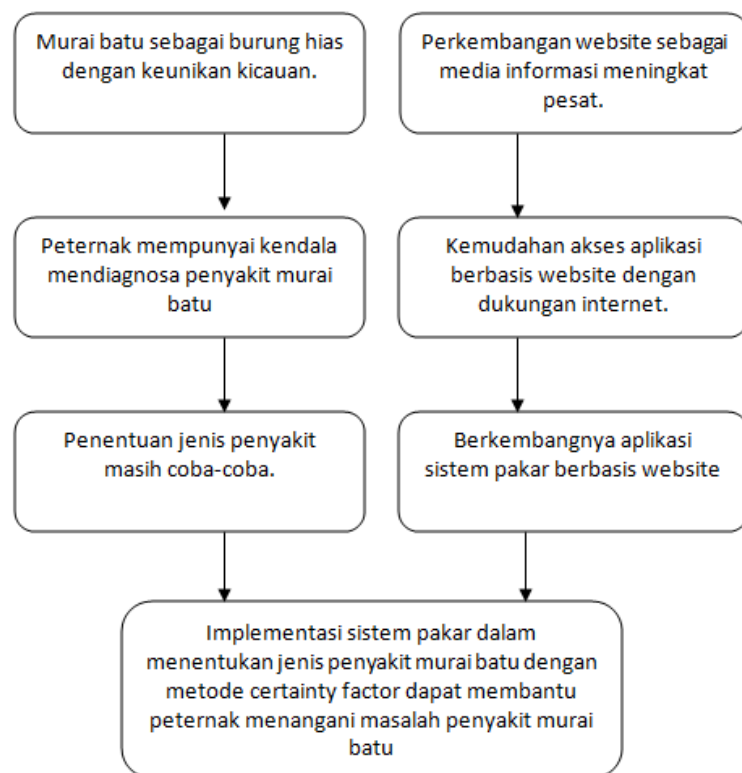
Dewasa ini beberapa jenis murai batu sudah banyak dibudidayakan di Indonesia. Tujuannya untuk menghasilkan murai batu dengan kualitas kicauan yang semakin lebih baik dan lebih unik dari sebelumnya.

Bunyinya amat kuat dan jelas dengan berbagai frasa dan sering kali mengikuti burung lain, burung ini juga mempunyai harga yang cukup mahal oleh sebab itu kebutuhan informasi sangat di butuhkan para masyarakat awam untuk mengatasi masalah tentang penyakit burung Murai Batu yang hanya bisa diselesaikan dengan bantuan para ahli .

Pada saat ini, internet sangat dekat dengan kehidupan sehari – hari, bahkan hampir setiap orang selalu menggunakan internet. Sehingga perlu dibuat suatu aplikasi berbasis website yang mampu memberikan rekomendasi atau diagnose penyakit pada burung murai tersebut.

Dalam pembuatan aplikasi sistem pakar menentukan jenis penyakit murai batu ini berbentuk sebuah *website*, karena *website* mudah untuk diakses secara cepat. Pengguna hanya membutuhkan *web browser* dan sebuah perangkat komputer atau *smartphone* yang terhubung ke internet untuk dapat menggunakan aplikasi sistem pakar menentukan jenis penyakit murai batu ini.

Dengan adanya aplikasi sistem pakar ini, diharapkan peternak murai batu lebih bisa mengetahui jenis penyakit seperti apa yang sedang dialami oleh murai batu dan bagaimana cara menanganinya.



Gambar 2.7 Kerangka Berpikir

BAB V

PENUTUP

Bab ini menjelaskan kesimpulan dan saran dari penelitian yang telah dilakukan ketika mengembangkan aplikasi untuk mengidentifikasi penyakit pada burung Murai batu dengan metode *certainty factor*.

5.1 Kesimpulan

1. Berdasarkan hasil observasi dan pengamatan pada penyakit burung Murai Batu, terdapat 7 kemungkinan penyakit yang dapat menyerang burung murai batu. gejala yang muncul dari penyakit tersebut juga beragam, terdapat 24 kemungkinan gejala yang dapat digunakan untuk mengidentifikasi penyakit yang menyerang pada burung murai batu. untuk dapat mengidentifikasi penyakit yang diderita oleh burung murai batu, digunakan metode *certainty factor*, dimana metode ini mengasumsikan derajat keyakinan seorang pakar terhadap suatu data. dalam penelitian ini, terdapat 30 *rule* yang digunakan untuk mengasumsikan derajat keyakinan seorang pakar terhadap penyakit yang diderita oleh burung murai batu.
2. Aplikasi pakar untuk mengidentifikasi penyakit pada burung murai batu ini dikembangkan dengan metode *System Development Life Cycle* (SDLC). selain itu, pada penelitian ini dilakukan pengujian, yaitu pengujian *black box* untuk pengujian fungsi-fungsi yang telah dikembangkan, hasil dari pengujian *black box* pada penelitian

ini adalah sesuai. semua fungsi yang telah dikembangkan berjalan sesuai dengan yang telah dirancang sebelumnya.

3. 5.2 Saran

Aplikasi untuk mengidentifikasi penyakit pada burung Murai batu dengan metode *certainty factor* ini dibuat untuk memberikan analisis mengenai berbagai kemungkinan penyakit yang mungkin menyerang burung murai batu. disarankan untuk penelitian selanjutnya menggunakan metode lain dari *certainty factor*, misalnya dengan metode *fuzzy*. kedua disarankan untuk menggunakan obyek lain seperti identifikasi penyakit pada burung kenari, jalak, dan jenis burung kicau lainnya. ketiga pengujian terhadap aplikasi. pengujian pada aplikasi dapat ditambahkan pengujian *white box*, dimana pada pengujian ini dapat dilihat alur logika yang digunakan saat mengembangkan sistem. selain itu, dapat juga ditambahkan pengujian terhadap *user experience* pada aplikasi.

DAFTAR PUSTAKA

- Ariawan, J., E. Tekat B.W., Novryan A. 2016. Sistem pakar menentukan gen anakan pada *lovebird*. *Jurnal Sisfotek Global* 6(3): 114-121.
- Dewanto, A., M. Sitanggang. 2009. *Buku Pintar Merawat dan Melatih Burung Kicauan*. Jakarta: Agromedia Pustaka.
- Handono, B., S. Gunarso, R. Turut. 2013. *Lovebird Sukses Menangkarkan dan Memelihara*. Jakarta: Penebar Swadaya.
- Heckerman, D. 1992. *The Certainty-Factor Model*. Los Angeles: University Of Southern California.
- Honggowibowo, A.S. 2009. Sistem pakar diagnosis penyakit tanaman padi berbasis web dengan forward dan backward chaining . *Jurnal Telkomnika* 7(3): 187-194.
- JIMP. 2016. *Jurnal Informatika Merdeka Pasuruan Vol.2, No.1 ISSN : 2503-1945*
- Kusrini. 2006. *Sistem Pakar, Teori dan Aplikasi*. Yogyakarta: Andi Offset.
- Kusrini. 2008. *Aplikasi Sistem Pakar Menentukan Faktor Kepastian Pengguna dengan Metode Kuantifikasi Pertanyaan*. Yogyakarta: Andi Offset.
- Mauli, H., 2013. *Kiat Sukses Budidaya Burung Lovebird*. Jakarta: Bonjer Media.
- Munandar, A., Suherman, dan Sumiati. 2012. The Use of Certainty Factor with Multiple Rules for Diagnosing Internal Disease. *International Journal of Application or Innovation in Engineering & Management* 1(1): 58-64.
- Prasetyo, Bambang. 2017. *26 Burung Kicau Unggulan*. Yogyakarta: Andi.
- Pressman, R. S. 1997. *Rekayasa Perangkat Lunak: Pendekatan Praktisi*. Edisi Satu. Yogyakarta: Andi.
- Pressman, R. S. 2010. *Software Engineering: A Practitioner's Approach*. 7th ed. Mc. GrawHill. New York.
- Rohajawati, S., Supriyati, R. 2010. Sistem Pakar: Diagnosis Penyakit Unggas Dengan Metode Certainty Factor. *Jurnal CommiT* 4(1): 41-46.
- Siswanto. 2010. *Kecerdasan Tiruan, Edisi 2*. Yogyakarta: Graha Ilmu.
- Soenanto, H. 2002. *Teknik Menangkarkan Lovebird*. Semarang: Effhar.

Sofa, R., Dini D., Ate S. 2012. Pembangunan aplikasi sistem pakar untuk diagnosis penyakit tanaman padi. *Jurnal Algoritma Sekolah Tinggi Teknologi Garut* 9(3): 1-8.

Sutojo, T., Mulyanto, E., dan Suhartono V. 2011. *Kecerdasan Buatan*. Yogyakarta: Andi.

Turban E., Aronson J.E., Liang T.P. 2005. *Decision Support Systems and Intelligent Systems (Sistem Pendukung Keputusan dan Sistem Cerdas)*. Yogyakarta: Andi Offset.