



**SISTEM PAKAR DIAGNOSA PENYAKIT KAMBING DAN DOMBA
MENGUNAKAN METODE *CERTAINTY FACTOR* (STUDI KASUS UNIT
PELAKSANA TEKNIS PEMBIBITAN TERNAK (UPT) DAN HIJAU
MAKANAN TERNAK (HMT) GARAHAN)**

SKRIPSI

diajukan guna melengkapi tugas akhir dan memenuhi salah satu syarat
untuk menyelesaikan Program Studi Sistem Informasi (S1)
dan mencapai gelar Sarjana Komputer

Oleh

Maya Elfrida Bimantari

NIM 122410101095

PROGRAM STUDI SISTEM INFORMASI

UNIVERSITAS JEMBER

2017



**SISTEM PAKAR DIAGNOSA PENYAKIT KAMBING DAN DOMBA
MENGUNAKAN METODE *CERTAINTY FACTOR* (STUDI KASUS UNIT
PELAKSANA TEKNIS PEMBIBITAN TERNAK (UPT) DAN HIJAU
MAKANAN TERNAK (HMT) GARAHAN)**

SKRIPSI

diajukan guna melengkapi tugas akhir dan memenuhi salah satu syarat
untuk menyelesaikan Program Studi Sistem Informasi (S1)
dan mencapai gelar Sarjana Komputer

Oleh

**Maya Elfrida Bimantari
NIM 122410101095**

PROGRAM STUDI SISTEM INFORMASI

UNIVERSITAS JEMBER

2017

PERSEMBAHAN

skripsi ini saya persembahkan untuk:

1. Ayahanda dan Ibunda tercinta;
2. Kakak tercinta dan keluarga besar yang tersayang;
3. Sahabatku bersama dukungan dan doanya;
4. Guru-guru sejak taman kanak-kanak sampai dengan perguruan tinggi;
5. Almamater Program Studi Sistem Informasi Universitas Jember.

MOTTO

Tidak seorangpun mengetahui apa yang disembunyikan untuk mereka yaitu (bermacam-macam kenikmatan) yang menyenangkan hati sebagai balasan terhadap apa yang mereka kerjakan(Q.S. As-Sajadah:17)

Bila Kamu Tidak Memiliki Keinginan untuk meraih sesuatu, tidak berhasrat untuk menggapai kesuksesan (apapun pengertain kesuksesan bagimu), kamu sebenarnya sedang mempersiapkan dirimu menuju keruntuhan. Milan Kundera

Belajar dari kemarin, hidup untuk sekarang, berharap untuk besok. Hal paling penting adalah jangan berhrneti untuk bertanya . albert Einstein

PERNYATAAN

Saya yang bertanda tangan di bawah ini:

nama : Maya Elfrida Bimantari

NIM : 122410101095

menyatakan dengan sesungguhnya bahwa karya ilmiah yang berjudul “Sistem Pakar Diagnosa Penyakit Kambing dan Domba menggunakan Metode *Certainty factor*”, adalah benar-benar hasil karya saya sendiri, kecuali jika dalam pengutipan substansi disebutkan sumbernya, belum pernah diajukan pada institusi mana dan bukan karya jiplakan. Saya bertanggung jawab atas keabsahan dan kebenaran isinya sesuai dengan sikap ilmiah yang harus dijunjung tinggi.

Demikian pernyataan ini saya buat dengan sebenarnya, tanpa adanya tekanan dan paksaan dari pihak manapun serta bersedia mendapatkan sanksi akademik jika dikemudian hari pernyataan ini tidak benar.

Jember, 21 Juli 2017

Yang menyatakan,

SKRIPSI

**SISTEM PAKAR DIAGNOSA PENYAKIT KAMBING DAN DOMBA
MENGUNAKAN METODE *CERTAINTY FACTOR* (STUDI KASUS UNIT
PELAKSANA TEKNIS PEMBIBITAN TERNAK (UPT) DAN HIJAU
MAKANAN TERNAK (HMT) GARAHAN)**

oleh:

Maya Elfrida Bimantari

NIM 122410101095

Pembimbing :

Dosen Pembimbing Utama : Anang Andrianto., S.T.,M.T

Dosen Pembimbing Pendamping : Muhammad Arief Hidayat.,S.Kom.,M.Kom

PENGESAHAN PEMBIMBING

Skripsi berjudul “Sistem Pakar Diagnosa Penyakit Kambing dan Domba Menggunakan Metode *Certainty factor* (Studi Kasus Unit Pelaksana Teknis Pembibitan Ternak (UPT) Dan Hijau Makanan Ternak (HMT) Garahan)” telah diuji dan disahkan pada :

hari, tanggal : Jumat, 21 Juli 2017

tempat : Program Studi Sistem Informasi Universitas Jember.

Disetujui oleh:

Pembimbing I,

Pembimbing II,

Anang Andrianto, ST., MT
NIP. 196906151997021002

Muhamad Arief H., S.Kom., M.Kom
NIP 198101232010121003

PENGESAHAN PENGUJI

Skripsi berjudul “Sistem Pakar Diagnosa Penyakit Kambing dan Domba Menggunakan Metode *Certainty factor* (Studi Kasus Unit Pelaksana Teknis Pembibitan Ternak (UPT) Dan Hijau Makanan Ternak (HMT) Garahan)” telah diuji dan disahkan pada :

hari, tanggal : Jumat, 21 Juli 2017

tempat : Program Studi Sistem Informasi Universitas Jember.

Tim Penguji,

Penguji I,

Penguji II,

Nelly Oktavia Adiwijaya S.Si.,MT.NIP.
NIP. 198410242009122008

Yanuar Nurdiansyah ST,.M.Cs.NIP.
NIP. 198201012010121004

Ketua Program Studi,

Prof. Drs. Slamin, M.Comp.Sc., Ph.D
NIP. 196704201992011001

RINGKASAN

“Sistem Pakar Diagnosa Penyakit Kambing dan Domba Menggunakan Metode *Certainty factor* (Studi Kasus Unit Pelaksana Teknis Pembibitan Ternak (UPT) Dan Hijau Makanan Ternak (HMT) Garahan)”; Maya Elfrida Bimantari, 122410101095; 2017: 196 halaman; Program Studi Sistem Informasi Universitas Jember.

Kambing dan domba merupakan salah satu ternak yang banyak dibudidayakan oleh masyarakat. Dinas Peternakan UPT PT dan HMT juga memiliki tugas untuk mengembangkan peternak binaan dengan melakukan kerja sama dengan masyarakat dalam melakukan budidaya ternak terutama kambing dan domba. Hasil usaha dan kerja sama dapat memenuhi tujuan dan hasil maksimal dengan memperhatikan manajemen pemeliharaan terutama dalam bidang kesehatan ternak. Terbatasnya tenaga medis mengakibatkan kurang optimalnya dalam penanganan penyakit. Pengetahuan masyarakat sekitar mengenai penyakit dan penanganan penyakit kambing dan domba menjadi faktor yang mempengaruhi proses budidaya ternak. Pengetahuan penyakit yang terbatas mengakibatkan penanganan menjadi lambat dan melakukan penanganan yang tidak tepat sehingga mengakibatkan parahnya penyakit.

Sistem pakar diagnose penyakit kambing dan domba merupakan aplikasi yang dapat membantu dalam menentukan penyakit pada kambing dan domba. Sistem pakar ini menerapkan metode *certainty factor* yang merupakan salah satu metode dalam sistem pakar. *Certainty factor* digunakan dalam perhitungan persentase tingkat keyakinan penyakit yang diderita. Nilai *certainty factor* didapat dari nilai MB dan nilai MD yang dimasukan oleh pakar pada setiap gejala. Nilai akhir *certainty factor* didapatkan dari perhitungan kombinasi dari masing-masing gejala. Hasil perhitungan persentase penyakit akan ditampilkan pada sistem, nilai tertinggi merupakan nilai yang merupakan penyakit yang terpilih dan ditampilkan detail informasi penyakit tersebut. Basis pengetahuan yang terdapat pada sistem ini meliputi penyakit, gejala

dan kaidah atau aturan yang digunakan pada proses perhitungan dalam menentukan penyakit.

Pelaksanaan penelitian ini terdiri dari tiga tahap yaitu tahap pengumpulan data, tahap analisis dan tahap pengembangan sistem. Tahap pengumpulan data dilakukan melalui studi literatur dan wawancara kepada narasumber. Tahap analisis dilakukan untuk menganalisis data yang didapatkan. Tahap pengembangan sistem dilakukan dengan mengadopsi model waterfall yang dimulai dari analisis kebutuhan fungsional dan non-fungsional sistem, pembuatan desain sistem, penulisan kode program, pengujian sistem dan perawatan. Sistem ini dibangun menggunakan bahasa pemrograman Page Hypertext Pre-Processor (PHP) dan framework Code Igniter (CI). Berdasarkan hasil pengujian terhadap sistem, sistem yang dibangun dapat memberikan hasil persentase keyakinan pasien menderita demam tifoid serta saran penanganan berdasarkan gejala yang dialami oleh pasien.

PRAKATA

Puji syukur kepada Allah SWT, atas segala limpahan rahmat, hidayat dan karuniaNya maka penulis dapat menyelesaikan skripsi yang berjudul berjudul “Sistem Pakar Diagnosa Penyakit Kambing dan Domba Menggunakan Metode *Certainty factor* (Studi Kasus Unit Pelaksana Teknis Pembibitan Ternak (UPT) Dan Hijau Makanan Ternak (HMT) Garahan)”. Skripsi ini disusun untuk memenuhi salah satu syarat menyelesaikan pendidikan Strata Satu (S1) pada Program Studi Sistem Informasi Universitas Jember.

Penyusunan skripsi ini tidak lepas dari dukungan dan bantuan berbagai pihak. Oleh karena itu penulis menyampaikan terima kasih kepada:

1. Allah SWT, yang telah memberikan kesempatan untuk hidup menyelesaikan skripsi ini, dan kekuatan yang diberikan sehingga skripsi ini selesai;
2. Ibu dan Bapak atas segala kesabaran, keikhlasan, limpahan kasih sayang, dan doa serta perjuangan yang tiada henti hingga saya bisa meraih semua ini;
3. Prof. Drs. Slamir, M.Comp.Sc.,Ph.D, selaku Ketua Program Studi Sistem Informasi Universitas Jember
4. Anang Andrianto, ST., MT selaku Dosen Pembimbing Utama, Muhammad Arief H, S.Kom.,M.Kom selaku Dosen Pembimbing Anggota yang telah sabar, memberikan banyak arahan, dan bimbingan dalam penyelesaian skripsi ini;
5. Yanuar Nurdiansyah ST.,M.Cs.NIP selaku Dosen Pembimbing Akademik;
6. Seluruh dosen dan staf karyawan di Program Studi Sistem Informasi Universitas Jember;
7. Seluruh pihak Unit Pelaksana Teknis Pembibitan Ternak (UPT) Dan Hijau Makanan Ternak (HMT) Garahan yang telah membantu penulis dalam menyelesaikan penelitian ini.

8. Kakak-kakak saya dan keluargabesar tercinta;
9. Sahabat- sahabat lusi, zalfa, dan dewi sri yang telah memberi semangat dan memotivasi saya. Seftya, Lintang , Ghau, Esa, Puput, Dwi, Kurnia, Zalfa, Bagus, Mada, dan Zalfa serta Teman-teman FORMATION(SI-2012) yang banyak membantu penulis saat pengerjaan skripsi.
10. Penghuni kosan mastrip 24 mba tiara, bundo ifka dan mantan penghuni citra zooketi, Bu Nur yang telah menjadi tempat sharing dan yang selalu memotivasi serta mengarahkan penulis. Dan untuk nicsap teruslah tersenyum.
11. Serta semua pihak yang tidak dapat disebutkan satu per satu yang telah membantu baik tenaga maupun pikiran dalam pelaksanaan kegiatan penelitian dan penyusunan skripsi ini.

Dengan harapan bahwa penelitian ini nantinya akan terus berlanjut dan berkembang kelak, penulis juga menerima segala kritik dan saran dari semua pihak demi kesempurnaan skripsi ini. Akhirnya penulis berharap, semoga skripsi ini dapat bermanfaat.

Jember, 14 Juli 2017

Penulis

DAFTAR ISI

PERSEMBAHAN	ii
MOTTO.....	iii
PERNYATAAN.....	iv
SKRIPSI.....	v
PENGESAHAN PEMBIMBING.....	vi
PENGESAHAN PENGUJI.....	vii
RINGKASAN	viii
PRAKATA.....	x
DAFTAR ISI.....	xii
DAFTAR GAMBAR	xvi
DAFTAR TABEL.....	xix
BAB I. PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah	4
1.3 Tujuan dan Manfaat.....	4
1.3.1 Tujuan	4
1.3.2 Manfaat	4
1.4 Batasan Masalah.....	5
1.5 Sistematika Penulisan.....	5
BAB 2. TINJAUAN PUSTAKA	7
2.1 Penelitian Terdahulu.....	7
2.2 Domba Dan Kambing.....	8
2.3 Penyakit Kambing dan Domba.....	8
2.4. Sistem Pakar	12
2.5 <i>Certainty factor</i>	14
BAB 3. METOLOGI PENELITIAN	18

3.1	Jenis Penelitian	18
3.2	Tempat dan Waktu Penelitian	18
3.3	Tahapan Penelitian	18
3.3.1.	<i>Requirement and Definition</i> (Analisa kebutuhan).....	19
3.3.2.	<i>Software sistem and design</i>	20
3.3.3.	<i>Implementation and unit testing</i>	21
3.3.4.	<i>Integration and unit Testing</i>	21
3.3.1.	<i>Operation and maintance</i>	22
BAB 4. PENGEMBANGAN SISTEM		23
4.1	Analisis Kebutuhan Sistem.....	23
4.1.1.	SOP (<i>Statment Of Purpose</i>)	24
4.1.2.	Kebutuhan Fungsional	24
4.1.3.	Kebutuhan Non Fungsional.....	25
4.1.4.	Fungsi Sistem	25
4.2.	Desain Sistem	26
4.2.1	<i>Business Process</i>	26
4.2.2	<i>Use Case Diagram</i>	27
4.2.3	<i>Use Case Scenario</i>	31
4.2.4	<i>Sequence Diagram</i>	44
4.2.5.	<i>Activity Diagram</i>	49
4.2.6	<i>Class Diagram</i>	54
4.2.7	<i>Entity Relationship Diagram (ERD)</i>	56
4.3	Penulisan Kode Program	56
4.4	Pengujian Sistem	62
4.4.1	Pengujian <i>White box</i>	62
4.4.2	Pengujian <i>Black Box</i>	72
BAB 5. HASIL DAN PEMBAHASAN		73
5.1	Hasil Pengembangan Sistem	73
5.1.1	Halaman Home utama sistem.....	73

5.1.2	Halaman Konsultasi (<i>user Umum</i>)	74
5.1.3	Halaman <i>Login</i>	75
5.1.4	Tampilan Home Pakar dan <i>User Anggota</i>	75
5.1.5	Halaman Mengelola Penyakit	76
5.1.6	Halaman Mengelola Gejala.....	78
5.1.7	Fitur <i>Rule</i>	80
5.1.8	Fitur <i>User</i>	81
5.1.9	Fitur <i>Record</i> penyakit.....	82
5.1.10	Fitur Konsultasi	83
5.1.11	Fitur Informasi	84
5.1.12	Fitur <i>User Profil</i>	85
5.2	Hasil Implementasi Metode <i>Certainty factor</i> Pada Sistem Pakar Daignosa Penyakit Kambing dan Domba Dengan Metode <i>Certainty factor</i>	86
5.2.1.	Hasil Pembuatan Metode <i>Certainty factor</i> Kedalam Kode Pemrograman	86
5.2.2.	Hasil Diagnosa Pada Sistem Pakar Diagnosa Penyakit Kambing Dan Domba Dengan Metode <i>Certainty factor</i>	89
5.3	Pembahasan Perancangan dan Pengembangan Sistem Pakar Diagnosa Penyakit kambing dan Domba	90
5.3.1.	Halaman home	91
5.3.2.	Halaman konsultasi	91
5.3.3.	Halaman <i>Login</i>	91
5.3.4.	Halaman mengelola penyakit	92
5.3.5.	Halaman Gejala.....	92
5.3.6.	Halaman <i>rule</i>	92
5.3.7.	Fitur <i>User</i>	92
5.3.8.	Tampilan <i>Record</i>	93
5.3.9.	Konsultasi.....	93
5.3.10.	Informasi Penyakit.....	94
5.3.11.	<i>User profil</i>	94

5.3.12. Tampilan Home	94
5.3. Pembahasan Implementasi Metode <i>Certainty factor</i> Pada Sistem Pakar Daiagnosa Penyakit Kambing dan Domba.	95
5.3.1. Pembahasan Metode <i>Certainty factor</i> Dalam Sistem Pakar	95
5.3.2. Pembahasan Sistem Pakar dengan Metode <i>Certainty factor</i> dalam Diagnosa Penyakit Kambing dan Domba Dengan.....	95
5.3.3. Pembahasan Hasil Metode <i>Certainty factor</i> Pada Sistem dengan Perhitungan Secara Manual.....	97
5.4. Pengujian Sistem	100
BAB 6. PENUTUP.....	106
6.1 Kesimpulan.....	106
6.2 Saran.....	107
Daftar Pustaka	108
LAMPIRAN	109
Lampiran A. Perancangan Sistem	109
A. 1 Skenario Sistem.....	109
A. 2 Activity Sistem	126
A. 3 Sequence Sistem.....	130
Lampiran B. Kode Program	134
Lampiran C. Pengujian.....	159
C. 1. <i>White box</i>	159
C. 2. <i>Black Box</i>	166
Lampiran D. Hasil Analisi Data.....	174

DAFTAR GAMBAR

Gambar 3. 1 Alur Model Waterfall	19
Gambar 4. 1 <i>Business Proses</i>	26
Gambar 4. 2 Use Case Diagram.....	27
Gambar 4. 3 <i>Sequence Diagram</i> Mengelola Penyakit	45
Gambar 4. 4 <i>Sequence Diagram</i> Mengelola Data Gejala	46
Gambar 4. 5 <i>Sequence Diagram</i> Mengelola Data <i>Rule</i> atau Pembobotan.....	47
Gambar 4. 6 <i>Sequence Diagram</i> Konsultasi <i>User Member</i>	48
Gambar 4. 7 <i>Activity Diagram</i> Mengelola Data Penyakit.....	50
Gambar 4. 8 <i>Activity Diagram</i> Mengelola Gejala.....	51
Gambar 4. 9 <i>Activity Diagram</i> Mengelola <i>Rule</i>	52
Gambar 4. 10 <i>Activity Diagram</i> Melakukan Konsultasi	53
Gambar 4. 11 <i>Class Diagram</i>	55
Gambar 4. 12 <i>Entity Relationship Diagram</i>	56
Gambar 4. 13 Kode Program C_konsultasi.....	60
Gambar 4. 14 ListingProgram <i>function__construct()</i>	64
Gambar 4. 15 Listing Program <i>Function</i> Index().....	64
Gambar 4. 16 Listing Program <i>Function</i> Halaman Hasil.....	64
Gambar 4. 17 Diagram Alir <i>function__construct()</i>	66
Gambar 4. 18 Diagram Alir <i>function_halamanhasil()</i>	66
Gambar 4. 19 Diagram Alir <i>function_index()</i>	67
Gambar 4. 20 Diagram Alir <i>function_inputHitungCF()</i>	67
Gambar 5. 1 Halaman Utama.....	74
Gambar 5. 2 Halaman Konsultasi	75
Gambar 5. 3 Halaman <i>Login</i>	75
Gambar 5. 4 Halaman Home Pakar.....	76

Gambar 5. 5 Halaman Home <i>user</i> anggota	76
Gambar 5. 6 Halaman Fitur Penyakit.....	77
Gambar 5. 7 <i>Pop up</i> tambah penyakit.....	77
Gambar 5. 8 Halaman <i>Edit</i> Penyakit.....	78
Gambar 5. 9 Halaman <i>Detail</i> Penyakit	78
Gambar 5. 10 Halaman Gejala	79
Gambar 5. 11 Halaman <i>Edit</i> Gejala	79
Gambar 5. 12 Halaman Menampilkan Notifikasi Gagal Hapus	80
Gambar 5. 13 Halaman pada fitur <i>rule</i>	80
Gambar 5. 14 <i>Pop up form</i> tambah <i>rule</i>	81
Gambar 5. 15 Halaman <i>User</i>	81
Gambar 5. 16 Menampilkan <i>Pop up</i> Kelola <i>User</i>	82
Gambar 5. 17 Halaman <i>Record User</i>	82
Gambar 5. 18 Halaman <i>Record</i> Pakar.....	83
Gambar 5. 19 Halaman Konsultasi	83
Gambar 5. 20 Halaman hasil konsultasi.....	84
Gambar 5. 21 Halaman Informasi Penyakit.....	85
Gambar 5. 22 Halaman <i>Detail</i> Penyakit	85
Gambar 5. 23 Halaman <i>User Profil</i>	86
Gambar 5. 24 <i>Input</i> gejala.....	87
Gambar 5. 25 Gambar Proses Metode <i>certainty factor</i>	87
Gambar 5. 26 gambar perhtiungan berdasarkan nilai cf	88
Gambar 5. 27 Proses <i>input</i> dan menampilkan data pada halaman Hasil akhir	88
Gambar 5. 28 Perbandingan nilai akhir pada <i>user</i> non member	89
Gambar 5. 29 Pemilihan gejala	89
Gambar 5. 30 Hasil Konsultasi	90
Gambar 5. 31 Halaman Hasil Perhitungan Sistem.....	100

A.2. 1 Gambar <i>Activity Login</i>	126
A.2. 2 Gambar <i>Activity Login User Anggota</i>	127
A.2. 3 <i>Activity Diagram Record Konsultasi</i>	128
A.2. 4 <i>Activity Diagram</i> Melihat Informasi Penyakit	128
A.2. 5 <i>Activity Diagram</i> Mengelola data <i>User</i>	129
A.3. 1 <i>sequence login</i>	130
A.3. 2 <i>Sequence Login User Anggota</i>	130
A.3. 3 <i>Sequence Record Pakar</i>	131
A.3. 4 <i>Sequence Record User Anggota</i>	131
A.3. 5 <i>Sequence</i> Mengelola Data <i>user</i>	132
A.3. 6 <i>Sequence</i> Informasi Penyakit	133
A.3. 7 <i>Sequence Logout User Anggota</i>	133
A.3. 8 <i>Sequence Logout Pakar</i>	133
Gambar B. 1 Kode Program Login	134
Gambar B. 2 Kode Program Model M_penyakit	136
Gambar B. 3 Kode Program Model M_Gejala	138
Gambar B. 4 Kode Program Model M_Rule	141
Gambar B. 5 Kode Program Model Class M_Konsultasi	146
Gambar B. 6Kode Program Controler Welcome	149
Gambar B. 7 Kode Program Controller/User.....	150
Gambar B. 8 Kode Program Controller/admin	151
Gambar B. 9 Kode Program Controller/Penyakit	153
Gambar B. 10 Kode Program Controller/p_gejala.....	154
Gambar B. 11 Kode Program Controller P_rule	156
Gambar B. 12 Kode Program Controller c_konsultasi	159

DAFTAR TABEL

Tabel 2. 1 Aturan Nilai <i>Certainty factor</i>	Error! Bookmark not defined.
Tabel 4. 1 Data Penyakit dan Gejala.....	23
Tabel 4. 2 Definisi Aktor	28
Tabel 4. 3 Definisi <i>Usecase</i>	29
Tabel 4. 4 Skenario <i>Login</i> Pakar.....	31
Tabel 4. 5 Skenario Mengelola Data Penyakit.....	33
Tabel 4. 6 <i>Scenario</i> Mengelola <i>rule</i>	38
Tabel 4. 7 <i>Scenario</i> Konsultasi	42
Tabel 4. 8 Tabel Basis <i>Test Function__construct()</i>	69
Tabel 4. 9 Tabel Basis Tes <i>function_index()</i>	69
Tabel 5. 1 Gejala Pilih.....	97
Tabel 5. 2 Pembobotan Gejala Penyakit ORF.....	98
Tabel 5. 3 Pembobotan Gejala Penyakit Kuku dan Mulut.....	98
Tabel A.1. 2. Skenario Login	109
Tabel A.1. 3 Skenario Mengelola Gejala.....	111
Tabel A.1. 4 Skenario Mengelola Data User	116
Tabel A.1. 5 Skenario Menampilkan Record.....	121
Tabel A.1. 6 Menampilkan Informasi penyakit	123
Tabel A.1. 7 Melihat Detail Informasi	124
Tabel A.1. 8 Skenario logout	125
Tabel A.1. 9 Skenario Logout pakar	125
Tabel C.1. 1 Test Case Controller P_penyakit function Edit.....	160
Tabel C.1. 2 Test Case Controller P_Penyakit Function Submittedit.....	161
Tabel C.1. 3 Test case controller p_penyakit function hapus	162
Tabel C.1. 4 Test Case Controller p_penyakit function Insertpenyakit	164
Tabel C.2. 1 Tabel Uji <i>Black Box</i>	166

Tabel D. 1 Tabel Pembobotan..... 174



BAB I. PENDAHULUAN

Bab ini merupakan langkah awal dari penulisan tugas akhir ini. Bab ini berisi latar belakang, rumusan masalah, tujuan dan manfaat, batasan masalah, metodologi penelitian, dan sistematika penulisan.

1.1 Latar Belakang

Unit Pelaksana Teknis Pembibitan Ternak (UPT) PT dan Hijau Makanan Ternak (HMT) Garahan adalah badan lembaga milik negara yang berada di bawah pengawasan dinas peternakan Provinsi Jawa Timur. UPT PT dan HMT memiliki tugas sebagai lembaga yang mengawasi dan menyuplai ketersediaan ternak dimasyarakat terutama untuk ternak kambing dan domba. Selain menyediakan bibit, Dinas Peternakan UPT PT dan HMT juga memiliki tugas untuk mengembangkan peternak binaan dengan melakukan kerja sama dengan masyarakat. Hasil usaha dan kerja sama dapat memenuhi tujuan dan hasil maksimal dengan memperhatikan manajemen pemeliharaan terutama dalam bidang kesehatan ternak.

Kesehatan ternak merupakan faktor paling penting dalam pembudidayaan ternak. Kesehatan ternak berhubungan dengan cara pemeliharaan ternak yang meliputi kecukupan nutrisi atau pemberian kecukupan gizi ternak dan kondisi kandang/perkandangan (Bahri, Beriajaya, & Adjid, Tanpa Tahun). Hal yang perlu untuk diperhatikan adalah pemahaman terhadap penyakit ternak, ini menjadi prioritas karena dengan pemahaman dan pengetahuan yang memadai dapat mengantisipasi dan mampu melakukan tindakan penanganan serta pencegahan penyakit maupun penyebaran penyakit yang akut.

Agen penyebab penyakit pada kambing dan domba dapat digolongkan menjadi beberapa meliputi penyakit oleh kuman, virus, parasit, dan kondisi tubuh kambing itu sendiri terkait perlakuan dan kondisi fisik ternak. Penyakit yang berbahaya dan patut untuk diwaspadai adalah penyakit yang disebabkan oleh virus,

kuman dan parasit karena memiliki sifat menular yang dan penyebaran yang cepat (Triakoso, 2011). Pengendalian yang tepat diperlukan untuk mencegah penyakit untuk berkembang lebih luas dan parah serta menimbulkan kerugian materiil.

Pengendalian penyakit adalah usaha dalam mengendalikan penyakit yang bersifat menular maupun tidak menular dengan cara melakukan tindakan pencegahan dan pengobatan yang sesuai dengan prosedur terkait penyakit yang diderita (Ruhyat, 2001). Penanganan penyakit yang salah akan menyebabkan meluasnya penularan hingga menyebabkan endemic pada penyakit tertentu. Prosedur penanganan yang tepat diperlukan meliputi pemberian diagnosa serta penanganan, vaksin dan pemberian obat berdasarkan pada penyakit yang diderita dengan sesuai dosis yang tepat. Selain melakukan pengobatan perlu dilakukan pencegahan agar penyakit tersebut tidak kambuh dan menular. Pengambilan keputusan dalam penentuan penyakit yang diderita merupakan faktor yang sangat penting. Hasil diagnosa atau penentuan penyakit merupakan tahapan untuk langkah penanganan selanjutnya.

Diagnosa penyakit merupakan cara untuk menentukan penyakit yang diderita oleh hewan atau ternak. Diagnosa merupakan suatu proses penentuan penyakit dengan cara menentukan dan mengamati perubahan yang terjadi pada ternak melalui tanda-tanda atau gejala yang terlihat, hingga dapat diketahui penyakit dan penyebabnya (Ruhyat, 2001). Gejala penyakit yang terjadi pada kambing dan domba sebagian besar memiliki ciri-ciri tertentu serta memiliki gejala yang sama dengan penyakit lainnya. Penentuan jenis penyakit atau diagnosa yang diderita memerlukan pertimbangan karena penyakit yang diderita kambing beragam. Penentuan penyakit dibutuhkan peran seorang pakar dalam memberikan langkah penanganan tepat berdasarkan pada penyakit yang diderita.

Di UPT PT dan HMT Garahan memiliki tenaga kesehatan ternak namun jumlah yang terbatas. Tenaga medis yang mencukupi sangat diperlukan karena masyarakat sebagian besar memiliki budidaya ternak kambing dan domba.

Kurangnya tenaga medis terutama dokter hewan mengakibatkan pelayanan bagi masyarakat peternak tidak optimal.

Masyarakat belum memiliki pengetahuan yang cukup terhadap beragamnya jenis penyakit pada kambing dan domba karena kurang sosialisasi pengetahuan terhadap penyakit. Penanganan yang tidak sesuai terhadap penyakit yang diderita ternak mengakibatkan semakin parahnya penyakit. Kerja seorang ahli dibutuhkan dalam membantu memberikan langkah penanganan dan informasi dalam penanggulangan penyakit. Pengetahuan masyarakat dan penanganannya masih terbatas sehingga membutuhkan suatu alat bantu dalam mempermudah mendapatkan informasi penyakit domba dan kambing. Semakin maju teknologi dan perkembangan sistem informasi dapat dimanfaatkan untuk mengatasi masalah yang terjadi. Penerapan sistem pakar merupakan salah satu perkembangan sistem informasi dalam membantu memberikan solusi dan informasi penyakit serta penanganannya yang berfungsi membantu kinerja pakar dan *user*.

Sistem pakar penentuan diagnosa penyakit domba dan kambing dapat digunakan sebagai salah satu alat bantu dalam penentuan penyakit kambing dan domba. Metode yang digunakan dalam sistem pakar ini adalah metode *Certainty factor*. Metode *certainty factor* adalah metode yang mendefinisikan kepastian terhadap sebuah fakta untuk menggambarkan tingkat keyakinan pakar, Sehingga akan didapatkan persentase keyakinan terhadap penyakit yang diderita berdasarkan dari proses perhitungan yang diambil dari nilai bobot. Pada *certainty factor* bobot setiap gejala didapatkan dari pakar yang menggambarkan besarnya kemungkinan terjadinya gejala terhadap suatu penyakit. Aplikasi yang akan dibuat pada penelitian ini adalah aplikasi berbasis web, pengguna sistem digolongkan menjadi 3 meliputi pakar, *user* dan *user member*. *User member* merupakan *user* yang telah terdaftar yang merupakan anggota dari peternak binaan.

1.2 Rumusan Masalah

Berdasarkan permasalahan yang telah dijabarkan pada latar belakang, didapatkan rumusan masalah sebagai berikut:

1. Bagaimana penentuan penyakit kambing dan domba pada sistem pakar menggunakan metode *certainty factor*?
2. Bagaimana merancang dan mengembangkan sistem pakar penyakit kambing dan domba menggunakan dengan model *waterfall*?

1.3 Tujuan dan Manfaat

Tujuan dan manfaat yang dipaparkan merupakan berbagai hasil yang ingin dicapai dan dapat digunakan dari beberapa aspek penelitian ini terselesaikan.

1.3.1 Tujuan

Tujuan dari penelitian ini adalah

1. Mengetahui perancang dan pembangun sebuah sistem pakar diagnosa dan penanganan penyakit kambing dengan menerapkan metode *waterfall*.
2. Mengetahui penerapan metode *certainty factor* dalam sebuah sistem pakar penentuan penyakit kambing dan domba.

1.3.2 Manfaat

Adapun manfaat yang ingin dicapai dalam penelitian ini adalah:

1. Bagi Penulis
Mengetahui proses penerapan dari metode *certainty factor* pada sistem pakar penentuan penyakit kambing dan domba.
2. Bagi Pihak Lain
Penelitian ini dapat digunakan sebagai bahan referensi peneliti lain yang ingin mengembangkan hasil penelitian ini dikemudian hari.
3. Bagi Akademis

Memberikan kontribusi serta masukan bagi siapa saja yang membutuhkan informasi yang berhubungan dengan judul penelitian ini. Bersama itu penelitian ini merupakan suatu upaya untuk menambah judul penelitian di Program Studi Sistem Informasi Universitas Jember.

4. Bagi Instansi

Penelitian ini diharapkan dapat memberikan kontribusi pada instansi untuk mempermudah pengambilan keputusan pada penyakit kambing dan domba.

1.4 Batasan Masalah

Sistem digunakan untuk menentukan penyakit pada hewan kambing dan domba

1. Penentuan penyakit pada domba dan kambing hanya berdasarkan dari gejala fisik yang dapat diamati.
2. Sistem ini memberikan keputusan berdasarkan besar persentase keyakinan penyakit yang mungkin diderita dan sistem meberikan informasi penanganan.
3. Sistem ini hanya membantu dalam pendampingan pemberian solusi mungkin untuk penyakit yang memilki detaksi yang mirip lebih abik dilakukan pemeriksaan secara laboatorium.

1.5 Sistematika Penulisan

Sistematika penulisan dan kerunutan skripsi ini disusun sebagai berikut:

1. Pendahuluan

Bab ini menjelaskan tentang latar belakang, perumusan masalah, tujuan, dan manfaat, ruang lingkup studi dan sistematika penulisan.

2. Tinjauan Pustaka

Bab ini menjelaskan tentang materi, informasi, tinjauan pustaka, dan studi terdahulu yang menjadi kerangka pemikiran dalam penelitian.

3. Metodologi Penelitian

Bab ini menjelaskan tentang metode penelitian yang digunakan dalam penelitian.

4. Pengembangan Sistem

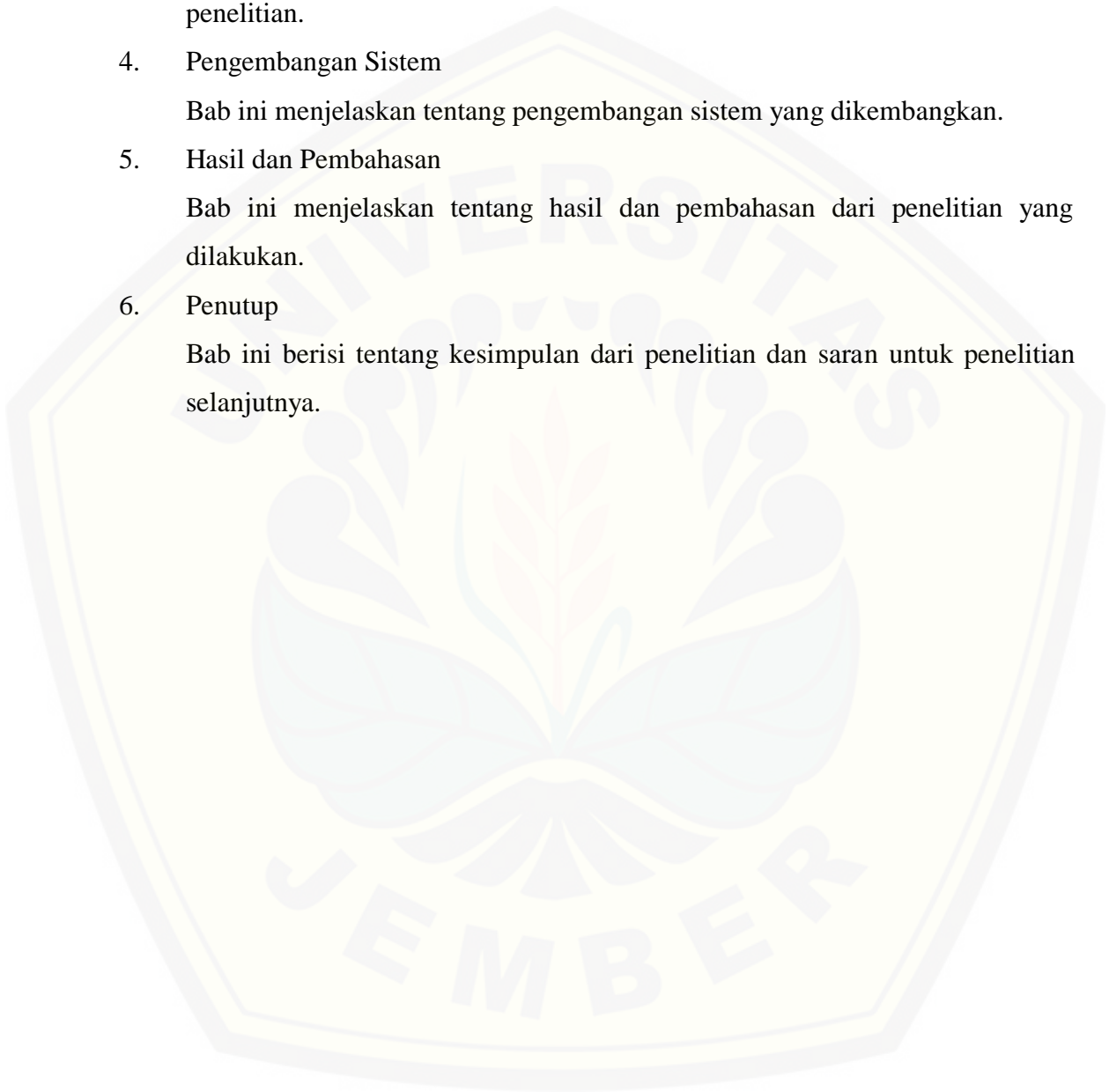
Bab ini menjelaskan tentang pengembangan sistem yang dikembangkan.

5. Hasil dan Pembahasan

Bab ini menjelaskan tentang hasil dan pembahasan dari penelitian yang dilakukan.

6. Penutup

Bab ini berisi tentang kesimpulan dari penelitian dan saran untuk penelitian selanjutnya.



BAB 2. TINJAUAN PUSTAKA

Pada bagian ini dipaparkan tinjauan yang berkaitan dengan masalah yang dibahas, kajian teori yang berkaitan dengan masalah, kerangka pemikiran yang merupakan sintesis dari kajian teori yang dikaitkan dengan permasalahan yang dihadapi. Teori-teori ini diambil dari buku, literature, jurnal, dan internet.

2.1 Penelitian Terdahulu

Penelitian terdahulu berjudul “Penerapan Sistem Pakar dengan Metode *Certainty factor* pada Aplikasi Dokter Virtual Berbasis Android (Rakhman, 2012). Penelitian ini membahas penerapan metode *certainty factor* dalam pendeteksi penyakit mata. Metode ini diterapkan karena dapat memberikan jawaban pada permasalahan perhitungan bobot dari gejala yang dipilih dan nilai keyakinan diberikan berdasarkan penilaian dari pakar. Selain itu aplikasi yang dibangun berbasis android atau mobile karena perkembangan mobile sangat tinggi penggunaanya. Aplikasi yang dirancang memiliki performa yang diandalkan dalam melakukan diagnosa penyakit mata, akurasi pada aplikasi tersebut belum ditentukan karena belum melibatkan pihak medis dalam penelitian tersebut.

Penelitian yang lain dilakukan “Sistem Pakar Diagnosis Kambing Berbasis WEB Menggunakan *Certainty factor*” (Mira, 2014). Penelitian ini membangun sebuah sistem pakar yang digunakan untuk diagnosa penyakit pada kambing. Halaman *user* pada aplikasi, terdapat modul konsultasi dan pencarian, sedangkan pada halaman pakar terdapat modul master penyakit, modul master gejala, dan modul master aturan. Penerapan perhitungan *certainty factor* pada penelitian tersebut menggunakan nilai CF Pakar dan Nilai CF *user* yang ditentukan oleh *user*. Informasi yang disajikan dari jurnal tersebut hanya meliputi diagnosa penyakit. Hasil *output* dari sistem adalah menampilkan jenis penyakit dengan nilai kepastian yang tertinggi.

Berdasarkan dari penelitian yang pernah dilakukan maka aplikasi yang akan dibuat pada sistem pakar penentuan penyakit kambing dan domba menggunakan metode *certainty factor* yang digunakan untuk penentuan penyakit yang diderita. Sistem yang akan dibuat ditambahkan beberapa modul yang belum terdapat pada penelitian sebelumnya yaitu *record* kesehatan ternak dan menambahkan informasi penanganan dan obat serta langkah pencegahan untuk menghindari penyakit.

2.2 Domba Dan Kambing

Kambing dan domba merupakan salah satu hewan mamalia yang banyak dibudidayakan oleh masyarakat. Domba dan kambing merupakan jenis ternak potong yang tergolong ternak ruminansia kecil karena memiliki pencernaan pakan yang khas didalam rumen. Sistem pencernaan yang khas inilah yang menyebabkan domba mampu mengubah pakan-pakan berkualitas rendah menjadi produk bergizi tinggi, seperti daging dan susu, serta hasil ikutan yang berkualitas tinggi seperti kulit dan wol (Abidin, 2002). Manfaat yang dihasilkan inilah yang membuat sebagian masyarakat melakukan budidaya kambing dan domba. Selain dari segi pemanfaatan, pemeliharaan kambing dan domba tergolong mudah.

Mayoritas masyarakat beranggapan bahwa kambing dan domba adalah hewan yang sama namun domba dan kambing memiliki perbedaan. Kambing dan Domba memiliki perbedaan dari bentuk fisik domba memiliki kelenjar bawah mata, tanduk berpenampang segitiga, bulu sangat baik sebagai bahan wol. Kambing tidak memiliki kelenjar bawah mata, tanduk berpenampang bulat dan lurus, serta bulu kurang dimanfaatkan (Subangkit, 2011).

2.3 Penyakit Kambing dan Domba

Penyakit pada hewan kambing dan domba terdiri dari dua jenis yaitu penyakit menular dan tidak menular. Penyakit menular merupakan penyakit yang disebabkan oleh virus, bakteri dan parasit, sedangkan penyakit tidak menular adalah penyakit yang disebabkan kondisi tubuh ternak sendiri seperti kurang mineral, kurang gizi dan

keracunan. Penyakit pada hewan ternak perlu dilakukan penanganan yang tepat terutama penyakit menular karena dapat menular kepada ternak lainnya. Penularan penyakit pada kambing dan domba dapat menular melalui kontak langsung, kontak dengan penyebab penyakit dan serangga.

Penyakit menular merupakan penyakit yang disebabkan oleh virus, bakteri dan parasite. Penyakit menular dapat mengakibatkan kematian pada ternak yang umumnya terjadi pada penyakit yang disebabkan oleh virus dan bakteri. Penyakit asal parasit kebanyakan bersifat kronis (Darmono & Hardiman, 2011). Penyakit infeksius utama yang sering menyerang kambing di Indonesia meliputi kelompok penyakit bakterial (pink eye, pneumonia, dan foot root), kelompok penyakit viral (orf), kelompok penyakit parasite (skabies, cacingan) (Bahri, Beriajaya, & Adjid, Tanpa Tahun). Detail mengenai penyakit menular tersebut dapat dilihat pada uraian berikut:

1. Penyakit Orf
 - a. Penyakit ORF merupakan penyakit yang disebabkan oleh virus yang termasuk dalam golongan parapoks. Penyakit ini adalah penyakit yang menular
 - b. Penyebab penyakit ini adalah tingkat kelembapan lingkungan yang tinggi dan kondisi ternak dalam keadaan stress. Penyakit ini menular dengan cepat dari ternak terinfeksi ke ternak yang sehat melalui kontak langsung. Penularan dapat juga terjadi akibat hewan yang peka mengkonsumsi pakan yang tercemar oleh keropeng bungkul orf.
 - c. Gejala : Terdapat lesi atau keropeng disekitar bibir, terdapat bintik-bintik merah sekitar bibir, terjadi bintik merak disekitar tubuh, nafsu makan ternak terlihat menurun.
2. Penyakit *Pink Eye*
 - a. Pink eye merupakan penyakit peradangan mata akut yang disebabkan oleh bakteri bernama *Rickettsia*.

- b. Penyebab terjadinya penyakit ini dapat dipengaruhi oleh musim panas karena banyaknya debu dan meningkatnya populasi lalat *musca autumnalis* selain itu penyakit ini dapat terjadi pada saat waktu ternak dalam perjalanan (transportasi) pemindahan tempat sehingga menimbulkan iritasi, kepadatan populasi pada kandang ternak serta perubahan iklim yang ekstrim dapat mempengaruhi timbulnya penyakit ini.
 - c. Gejala yang dialami pada ternak penderita penyakit pink eye adalah mata berwarna merah hingga mengalami peradangan, kelopak mata membengkak, mata ternak mengeluarkan cairan (berair).
3. Penyakit Kudis Menular (*Scabies*)
- a. Skabies atau kudis merupakan penyakit yang menyerang jaringan kulit pada hewan. ektoparasit utama penyakit ini menyerang bagian kulit ternak ruminansia, terutama kambing dan kelinci bahkan dapat menular ke manusia (*zoonosis*).
 - b. Penyebab dari penyakit ini adalah tungau *sarcoptes* yang umumnya terdapat pada lorong tanduk. Kejadian kudis pada ternak telah tersebar terutama pada keadaan kekurangan pakan, dimusim kemarau dan di lingkungan kandang yang kotor
 - c. Gejala: Kulit ternak menebal mengeras hingga berlipat-lipat, terdapat keropeng pada tubuh ternak, bulu ternak kusam, ternak terlihat menggosok-gosokan badannya.
4. *Foot Root*
- a. Penyakit *foot root* merupakan pembusukan kaki atau borok ceracak yang tergolong penyakit bakterial dan disebabkan oleh *Bacteroides (Fusobacterium) nodosus*. Kondisi kandang yang basah dan kotor juga sering dikaitkan dengan kejadian penyakit ini.

- b. Penyakit foot root dapat dicegah dengan selalu memperhatikan kondisi kandang. Ternak diusahakan selalu berada di lantai yang kering dan dilakukan pemotongan kuku. Lumpur dan kotoran sebagai pemicu penyakit ini, selalu dibersihkan agar tidak terselip diantara kuku
- c. Gejala dari penyakit *foot root* adalah kakai ternak pincang, terjadi peradangan pada kaki ternak, timbul nanah pada kaki ternak yang terluka, terjadi abses disekitar kuku ternak dan ternaka mengalami nafsu makan

5. Penyakit Kuku dan Mulut

Gejala penyakit kuku dan mulut meliputi ternak mengalami peningkatan salivasi, melepuh terjadi disekitar bagian kuku, melepuh atau terjadi benjolan berair disekitar bibir, nafsu makan ternak menurun.

Penyakit yang terdapat pada kambing selain penyakit menular adalah penyakit yang tidak menular. Penyakit yang tidak menular mayoritas disebabkan oleh pemeliharaan dan lingkungan peternakan. Jenis dari penyakit yang non infeksius sebagai berikut:

1. Kembung

Penyakit yang terjadi karena mengembangnya rumen akibat terisi oleh gas yang berlebihan. Perut kembung atau timpani adalah suatu keadaan mengembangnya rumen akibat terisi oleh gas yang berlebihan. Hal ini terjadi ketika esophagus mengalami sumbatan sehingga menghambat pengeluaran gas.

Gejala dari penyakit kembung adalah nafsu makan ternak menurun, frekuensi nafas menurun, perut ternak membengkak dan kembung, selaput lendir berwarna kebiruan.

2. Diare

Diare adalah gejala abnormalitas sistem pencernaan dan sering terjadi pada anak kambing. Gejala ini tidak hanya menyebabkan kekurangan penyerapan sari-sari makanan, tetapi ternak juga akan mengalami kehilangan cairan dalam

jumlah banyak. Gejala yang dialami ternak adalah kotoran ternak berwarna dan berbentuk cair.

3. Mastitis

Mastiti adalah penyakit pembengkakan ambing pada ternak. Mastitis adalah salah satu penyakit radang ambing yang menyebabkan ambing menjadi abnormal diakibatkan infeksi, biasanya penyakit ini akan bersifat akut, sub akut bahkan akan mengakibatkan kronis.

Gejala dari penyakit ini adalah peningkatan frekuensi pernafasa pada ternak, pembengkakan pada ambing ternak, suhu badan ternak mengalami peningkatan, dan nafsu makan ternak menurun.

4. Penyakit *Myasis*

Myasis atau belatungan adalah infestasi larva lalat ke dalam jaringan tubuh hewan hidup. Penyakit ini dapat menyerang semua hewan termasuk unggas dan manusia. Penyakit ini memiliki gejala meliputi terdapat luka traumatic pasca melahirkan, terjadi peradangan pada tubuh yang terluka, terdapat cairan pada luka, dan suhu tubuh ternak meningkat.

2.4. Sistem Pakar

Sistem pakar adalah sistem yang menggunakan pengetahuan manusia yang dimasukan dalam sebuah komputer dan digunakan untuk menyelesaikan masalah yang pada umumnya membutuhkan kepakaran atau keahlian manusia (Dhani, 2014). Pengertian lain diungkapkan oleh Durkin, sistem pakar adalah program komputer yang dirancang untuk memodelkan kemampuan penyelesaian masalah yang dilakukan seorang pakar (Faizal, Tanpa tahun). Pengertian diatas dapat disimpulkan bahwa sistem pakar merupakan sekumpulan fungsi dalam sebuah komputer yang dapat membantu menyelesaikan permasalahan dengan mengadopsi pengetahuan pakar .

Pengetahuan yang diimplementasikan kedalam sebuah sistem meliputi pengetahuan yang dimiliki pakar terkait bidang yang dikuasai, fakta dan teknik penalaran dalam menyelesaikan sebuah masalah. Pemodelan kemampuan penyelesaian masalah dilakukan dengan memperoleh informasi dari pakar, kemudian dilakukan pemodelan atau pengolahan sesuai dengan metode yang digunakan untuk mendapatkan hasil akhir sebuah solusi dari permasalahan. Proses pengolahan hasil akhir pemecahan masalah dilakukan dengan cara mengkombinasi kaidah penarikan kesimpulan (*inference rules*) yang diberikan oleh satu atau lebih pakar dalam bidang tertentu. Kombinasi dari kedua hal tersebut disimpan dalam komputer, yang selanjutnya digunakan dalam proses pengambilan keputusan untuk menyelesaikan masalah tertentu.

Perancangan dan pembangunan sistem pakar memiliki 3 modul utama meliputi modul *knowledge base*, *working memory* dan *inference engine* (Homsah & Ariwibowo, 2011). Selain modul utama terdapat modul lainnya pada sistem pakar yaitu *Interface*, *developer interface*, *explanation interface* dan *external program*. Penjabaran dari modul yang terdapat pada sebuah sistem pakar adalah, sebagai berikut:

1. *Knowledge base* (Basis Pengetahuan).

Berisi pengetahuan-pengetahuan dalam penyelesaian masalah. Terdapat dua bagian dasar dalam sebuah sistem pakar yaitu fakta dan aturan. Fakta merupakan objek area dalam permasalahan tertentu, sedangkan aturan merupakan cara dimana untuk mendapatkan sebuah fakta yang baru dari fakta yang ada.

2. *Inference engine* (mesin Inferensi)

Mesin inferensi merupakan perangkat lunak yang melakukan penalaran dengan menggunakan pengetahuan yang ada untuk menghasilkan suatu kesimpulan atau hasil akhir. Dalam komponen ini dilakukan pemodelan

proses berpikir manusia. Mesin inferensi ini data yang telah *diinput* akan diproses. Mesin inferensi dibagi menjadi dua cara yaitu inferensi secara *backward chaining* dan *forward chaining*. *Forward chaining* merupakan sistem pencarian yang runtut kedepan sedangkan *backward chaining* merupakan pencaarian runtut kebelakang.

3. *Working Memory*

Working memory merupakan penyimpanan fakta yang diketahui dari hasil menjawab pertanyaan.

4. Antarmuka Pengguna (*User Interface*)

Pada bagian ini terjadi dialog antara program dan pengguna. Antarmuka merupakan penghubung yang menyajikan hasil dari sistem yang dapat dimengerti oleh pengguna.

5. *Explanation facility*

Memberikan penjelasan saat *user* mengetahui apakah alasan yang diberikan sebuah solusi.

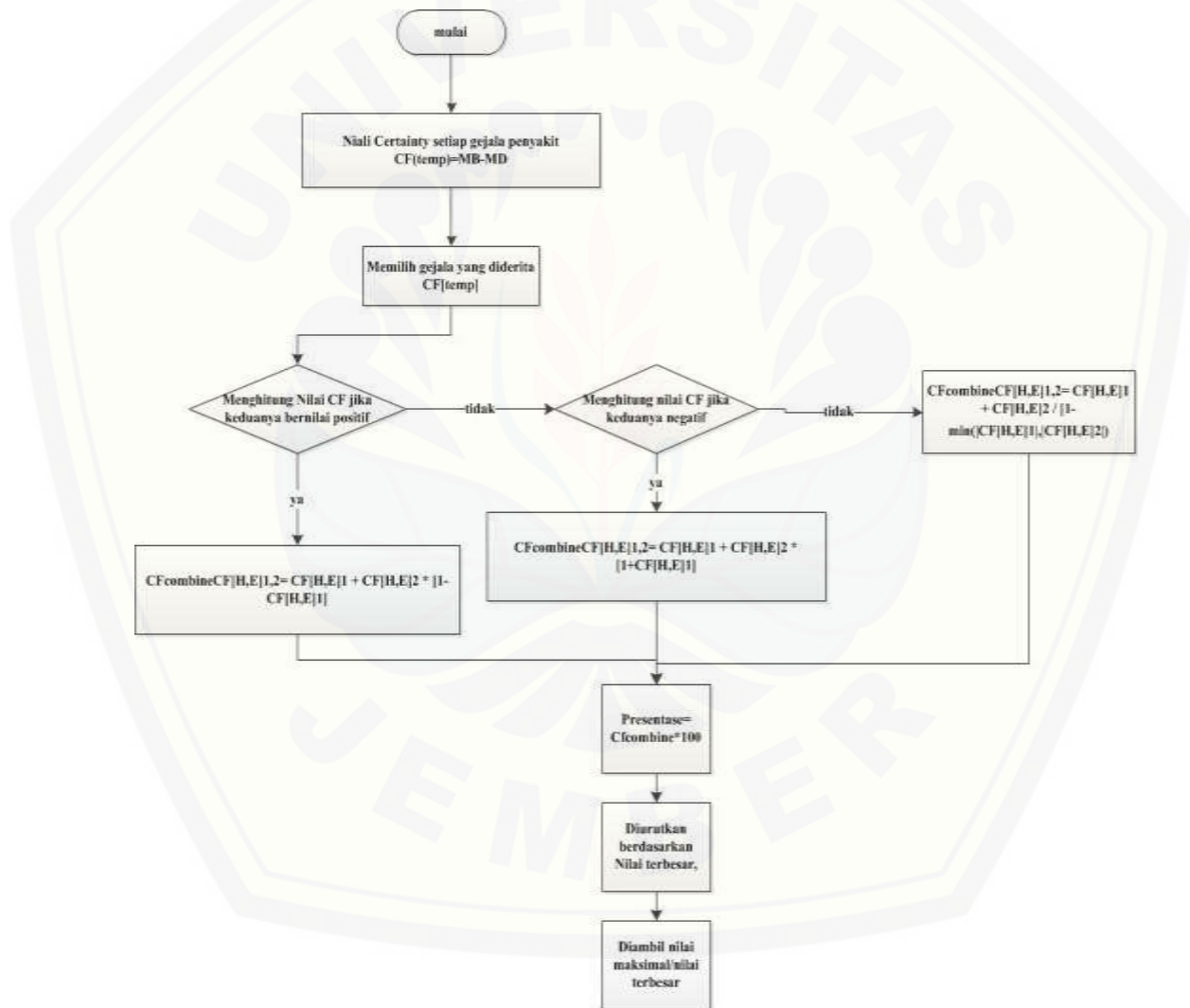
6. *External programs.*

Berbagai program seperti *database*, *spreadsheets*, *algorithms*, dan lainnya yang berfungsi untuk mendukung sistem.

2.5 *Certainty factor*

Certainty factor merupakan metode yang menggunakan faktor kepastian yang dapat berupa probabilitas. *Certainty factor* adalah metode yang menggambarkan kepercayaan dalam kejadian (atau fakta atau hipotesis) berdasarkan kejadian (atau pada penilaian seorang pakar) (Rohajawati & Supriyati, 2010). Setiap nilai dalam *certainty factor* memiliki nilai keyakinan tersendiri yang menunjukkan ukuran kepastian terhadap suatu fakta.

Perhitungan nilai *certainty factor* dilakukan dengan membandingkan dua perhitungan saja, jika lebih akan dilakukan perhitungan secara bertahap. Proses Pada nilai *certainty factor* akan dihitung setiap nilai *certainty factor* yang didapatkan dari pengurangan nilai mb md, untuk lebih jelas proses perhitungan dalam mencari nilai md dan mb akan dijelaskan pada gambar 2.1. Nilai hasil pada metode *Certainty factor* yang memiliki hasil mendekati 1 memiliki nilai kepercayaan yang terbesar.



Gambar 2. 1 Alur Metode *Certainty factor*

Alur proses perhitungan dari gambar 2.1 akan diuraikan sebagai berikut:

1. Proses perhitungan pertama dengan menjumlah nilai mb dan mb untuk mendapatkan nilai CF pada masing – masing *rule* dari penyakit gejala

$$CF[h, e] = MB[h, e] - MD[h, e] \quad \dots\dots\dots \text{Persamaan 2.1}$$

Keterangan:

$CF[h, e]$ = Faktor kepastian

$MB[h, e]$ = *measure of belief*, ukuran kepercayaan atau tingkat keyakinan terhadap hipotesis (h), jika diberikan *evidence* (e) antara 0 dan 1

$MD[h, e]$ = *measure of disbelief*, ukuran ketidakpercayaan atau tingkat keyakinan terhadap hipotesis (h), jika diberikan *evidence* (e) antara 0 dan 1.

2. Berdasarkan gejala yang dipilih didapatkan nilai *certainty factor*. Nilai CF pakar yang sudah ditemukan nantinya akan digunakan untuk menghitung nilai CF kombinasi dengan membandingkan setiap dua nilai *certainty factor*. Selanjutnya menentukan nilai kombinasi dari setiap aturan yang sudah ada dapat dilihat pada persamaan 2.2.

$$CF_{combine} CF[H, E]_{1,2} = CF[H, E]_1 + CF[H, E]_2 * [1 - CF[H, E]_1] \quad \dots$$

Persamaan 2.2

Persamaan 2 digunakan jika nilai dari kedua CF bernilai positif. Apabila nilai $CF[H, E]_1$ dan $CF[H, E]_2$ bernilai negatif atau kurang dari nol digunakan pada persamaan 2.3.

$$CF_{combine} CF[H, E]_{1,2} = CF[H, E]_1 + CF[H, E]_2 * [1 + CF[H, E]_1] \quad \dots$$

Persamaan 2.3

Persamaan 4 digunakan jika salah satu nilai CF bernilai negataif

$$CF_{combine}CF[H,E]_{1,2} = \frac{CF[H,E]_1 + CF[H,E]_2}{1 - \min(|CF[H,E]_1|, |CF[H,E]_2|)} \dots \text{Persamaan 2.4}$$



BAB 3. METOLOGI PENELITIAN

Metodologi penelitian adalah gambaran sistematis yang berurutan dalam menganalisa suatu permasalahan dan menjawab permasalahan tersebut, sehingga didapatkan hasil berupa tujuan penelitian. Metode penelitian kali ini dijabarkan menjadi beberapa bagian sebagai berikut :

3.1 Jenis Penelitian

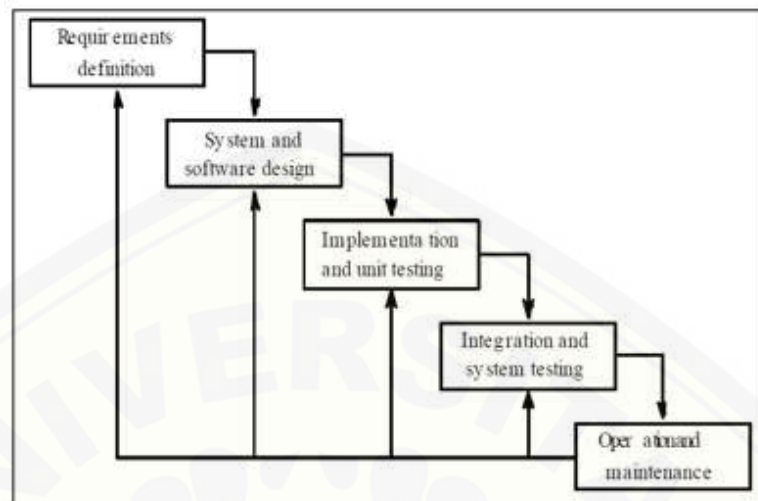
Jenis penelitian yang digunakan dalam penelitian adalah pengembangan. Tujuan dari penelitian ini membangun suatu sistem yang digunakan untuk membantu pakar dalam melakukan diagnosa penyakit yang terjadi pada domba dan kambing.

3.2 Tempat dan Waktu Penelitian

Tempat yang dilaksanakan untuk penelitian adalah di Unit Pelaksana Teknis Pembibitan Ternak (UPT) PT dan Hijau Makanan Ternak (HMT) Garahan kabupaten Jember. Waktu penelitian dilakukan pada minggu pertama bulan Agustus hingga September.

3.3 Tahapan Penelitian

Tahap penelitian yang digunakan dalam pengembangan sistem ini adalah kosep pengembangan metode *Software development life cycle* (SDLC). Terdapat beberapa model yang dapat diterapkan dalam metode pengembangan sistem, namun pada penelitian ini SDLC yang digunakan adalah model *waterfall*. *Waterfall* adalah model yang terstruktur dilakukan terurut mulai dari mengumpulkan kebutuhan, mendesain sistem, mengimplementasikan sistem, *testing* dan *maintenance*. Setiap tahapan dalam model *waterfall* harus diselesaikan terlebih dahulu tiap tahapnya sebelum berlanjut pada tahap berikut. Struktur *waterfall* dapat dilihat pada gambar3.1.



Gambar 3. 1 Alur Model *Waterfall*

3.3.1. *Requirement and Definition*(Analisa kebutuhan)

Tahap analisis kebutuhan merupakan tahap pertama pada proses perancangan yang harus terpenuhi. Kebutuhan yang diperlukan oleh sistem dijabarkan pada proses ini. Informasi kebutuhan yang telah didapat kemudian dikelompokkan menjadi kebutuhan fungsional dan kebutuhan *non fungsional*. Selain pengelompokan kebutuhan, analisis kebutuhan juga digunakan untuk menentukan fungsi maupun fitur-fitur yang dibutuhkan. Teknik yang digunakan untuk mengumpulkan kebutuhan data dalam perancangan sistem ini terdiri dari 2 metode yaitu berikut:

1. Studi Pustaka

Pengumpulan data yang digunakan dalam mengimplementasi sistem pakar ini adalah data sekunder yang didapat dari studi pustaka pada penelitian – penelitian terdahulu yang didapat dari jurnal, *e-book*, karya tulis, skripsi dan buku. Teknik ini digunakan untuk mengumpulkan informasi gejala penyakit yang terjadi pada domba dan kambing. Pencarian data sekunder ini untuk menunjang kelengkapan data dan menambah referensi terkait data diagnosa. Selain itu, tahap ini digunakan untuk mencari informasi terkait metode *certainty factor* dan penerapannya dalam sistem pakar.

2. Wawancara

Wawancara merupakan teknik dari pendekatan komunikasi. Wawancara merupakan komunikasi dua arah untuk mendapatkan data dari responden. Wawancara adalah berupa percakapan langsung antar dua pihak atau lebih untuk mendapatkan informasi secara lisan dengan tujuan untuk memperoleh data yang dapat menjelaskan ataupun menjawab suatu permasalahan penelitian (Hasibuan, 2007). Wawancara dalam penelitian dilakukan pada pakar atau pihak yang ahli dalam bidang kesehatan hewan dan para ahli yang ada di UPT dan HMT Dinas Peternakan. Wawancara yang dilakukan guna mendapatkan gejala yang terjadi pada ternak dan penyakit yang pernah terjadi dilingkungan perternakan , pencegahan, pengobatan dan pembobotan pada masing – masing hubungan gejala dan penyakit. Pembobotan yang dilakukan oleh pakar adalah pemberian nilai md dan mb.

3.3.2. *Software sistem and design*

Pembuatan desain sistem menggunakan *Unified Modeling Language* (UML) yang dirancang menggunakan konsep *Object-Oriented Programming* (OOP). Pembuatan desain sistem berdasarkan dari kebutuhan yang telah didefinisikan pada *requirement definition* atau analisis kebutuhan. Berikut pemodelan UML yang akan digunakan antara lain:

1. *Business Process*, digunakan untuk menggambarkan aktifitas dan proses yang lengkap. Bisnis proses didapatkan dari hasil analisa data yang digunakan. Bisnis proses menggambarkan data yang masuk pada sistem dan hasil luaran pada sistem.
2. *Usecase Diagram*, digunakan untuk menggambarkan fungsional yang disediakan sistem. Setelah melakukan bisnis proses selanjutnya adalah membuat *use case* yang menggambarkan rangkaian sistem secara teratur yang dilakukan oleh actor.

3. *Scenario*, digunakan untuk menjelaskan alur sistem yang ada fitur pada *usecase*. Setiap fitur yang terdapat pada *usecase* akan dijabarkan alur dan *output* dari sistem.
4. *Activity Diagram*, digunakan untuk mendeskripsikan aktifitas suatu operasi yang terdapat pada *scenario*. *Activity diagram* menggambarkan aktifitas pada sistem yang menjabarkan aliran sistem dari awal, *decision* yang terjadi dan akhir dari sistem.
5. *Sequence Diagram*, digunakan untuk menggambarkan aliran logika interaksi dan pesan yang dikirim antar obyek. Melalui *activity diagram* dapat diketahui aliran logika pada sistem.
6. *Class Diagram*, digunakan untuk menggambarkan struktur dan deskripsi *class* dalam sistem. *Class* yang terdapat pada *class diagram* didapatkan dari *class* yang sudah terdapat pada *sequence diagram*.
7. *Entity Relationship Diagram*, digunakan untuk menunjukkan relasi antar obyek. Setelah dibentuknya relasi antar objek akan dilakukan penerapan kode program

3.3.3. *Implementation and unit testing*

Tahap implementasi merupakan tahap penerapan coding berdasarkan desain yang telah dibuat. Hal yang dilakukan dalam implementasi antara lain:

1. Penulisan program menggunakan bahasa pemrograman *Page Hyper Text Pre-Processor* (PHP) dengan *framework Code Igniter* (CI).
2. *Database* yang digunakan adalah MySQL dengan menggunakan aplikasi XAMPP.

3.3.4. *Integration and unit Testing*

Tujuan dari tahap ini adalah untuk mengetahui apakah sistem yang dibuat sudah sesuai dengan kebutuhan serta mencari kesalahan atau *bug*. Pengujian perangkat lunak dilakukan dengan dua metode, yakni:

1. White box Testing

Merupakan metode pengujian perangkat lunak dengan melihat modul yang telah dibuat dengan program yang ada.

2. Black Box Testing

Merupakan metode pengujian perangkat lunak dengan memeriksa fungsionalitas dari aplikasi yang berkaitan dengan struktur internal kerja.

- 3.3.1. *Operation and maintance*

Pemeliharaan dilakukan ketika adanya permintaan perubahan atau perbaikan pada sistem oleh *user*. Pemeliharaan ini dilakukan untuk melengkapi kekurangan dari sistem. Selain itu pemeliharaan juga dilakukan untuk menemukan kekurangan serta *bug* yang terjadi pada sistem.

BAB 4. PENGEMBANGAN SISTEM

Bab ini membahas pengembangan sistem pakar dengan menggunakan metode *certainty factor*. Model *waterfall* digunakan dalam pengembangan sistem pakar ini. Tahapan awal dimulai dengan analisis kebutuhan fungsional dan non fungsional sistem, pembuatan desain sistem, penulisan kode program dan pengujian sistem.

4.1 Analisis Kebutuhan Sistem

Tahap ini merupakan awal dari proses pengembangan sebuah sistem dan merupakan bagian terpenting. Kebutuhan yang digunakan pada pengembangan sistem diuraikan pada tahap analisis kebutuhan sistem, kebutuhan tersebut meliputi kebutuhan fungsional dan non fungsional. Kebutuhan-kebutuhan tersebut didapatkan dari pengamatan pada sebuah masalah dan didapat dari hasil pengolahan masalah. Pengumpulan data kebutuhan sistem dapat dilakukan dengan melakukan studi literatur dan wawancara.

Pada proses pengambilan data, hasil yang diperoleh dari wawancara dan studi literatur adalah data gejala dan informasi penyakit yang dapat dilihat pada tabel 4.1 Untuk lebih lengkap dapat dilihat pada lampiran D.

Tabel 4. 1 Data Penyakit dan Gejala

No	Penyakit	Gejala
1	Scabies	Menggaruk dan Menggosokan Badan
		Nafsu makan menurun
		Timbul Caran Pada bekas Luka
		terdapat keropeng pada tubuh ternak
		Kulit Menebal
2	ORF	Muncul Bintik Merah Di sekitar bibir/ muncul lesi disekitar bibir
		Terdapat Benjolan Bernanah

Selain itu, dari hasil wawancara juga didapatkan pembobotan masing- masing gejala yang memiliki nilai MB(nilai kepercayaan) dan MD(nilai ketidak percayaan) pada setiap gejala yang berhubungan dengan penyakit tertentu. Data mengenai *detail* pembobotan dapat dilihat pada lampiran D. Selain itu, hasil dari studi literature yang dilakukan didapatkan penerapan metode *certainty factor* kedalam sistem pakar dan alur dalam implementasi kedalam sistem pakar.

4.1.1. SOP (*Statment Of Purpose*)

Sistem pakar diagnosa penyakit pada kambing dan domba merupakan sistem pakar yang menerapkan metode *certainty factor* dalam penentuan penyakit. Tujuan dari sistem pakar ini adalah membantu kerja seorang pakar dalam menentukan penyakit yang diderita oleh ternak kambing dan domba. Aplikasi ini menyediakan pengetahuan gejala, informasi penyakit dan konsultasi serta pembobotan aturan yang dimasukan atau dikelola oleh pakar. Halaman konsultasi digunakan *user* yang sudah terdaftar maupun belum terdaftar untuk melakukan diagnosis. Sistem ini memberikan hasil persentase penyakit yang diderita oleh ternak kambing dan domba dengan menghitung bobot dari setiap gejala terpilih. Gejala yang didapat dari *user* selanjutnya akan dihitung dengan menggunakan metode *certainty factor* hingga diperoleh persentase besarnya penyakit yang diderita. Penyakit yang memiliki nilai terbesar akan ditampilkan informasi terkait *detail* penyakit berupa penanganan, penyebab dan pencegahan.

4.1.2. Kebutuhan Fungsional

Kebutuhan fungsional merupakan kebutuhan pokok pada sistem yang menjabarkan keadaan sistem dalam melakukan proses penerimaan *output* dan *input* serta memroses keduanya. Kebutuhan fungsional sistem pada penentuan penyakit pada kambing dan domba adalah:

- 1 Sistem dalam mengelola data penyakit (*view, edit dan delete*).
- 2 Sistem mampu mengelola data gejala(*view, edit dan delete*).

- 3 Sistem dapat memproses penghitungan metode *certainty factor*.
- 4 Sistem dapat mengelola rule dari metode *certainty factor* (view, edit dan delete).
- 5 Sistem mampu mengelola hasil konsultasi dari *user*.
- 6 Sistem mampu menyimpan *record* konsultasi.
- 7 Sistem mampu mengelola data *user*.

4.1.3. Kebutuhan Non Fungsional

Kebutuhan non – fungsional merupakan hal yang dibutuhkan oleh sistem untuk mendukung aktivitas sistem. Sistem ini memiliki kebutuhan fungsional sebagai berikut, yaitu:

1. Sistem menggunakan *username* dan *password* untuk autentifikasi untuk akses sistem.
2. Sistem dapat diakses oleh banyak pengguna secara bersamaan.

4.1.4. Fungsi Sistem

Pada sistem pakar penentuan penyakit kambing dan domba terdapat fitur *login* sesuai dengan hak akses dari setiap pengguna. Sistem akan melakukan proses autentifikasi *username* dan *password*. Selanjutnya sistem akan menampilkan tampilan sesuai dengan hak akses pengguna yang meliputi:

1. Admin

Pakar merupakan pengguna sistem yang telah terdaftar, dan ketika *login* berhasil maka sistem akan menampilkan halaman *home* admin. Pakar dapat mengakses halaman penyakit, gejala, *user*, *record* dan *log out* serta *list* terakhir hasil konsultasi.

2. Anggota (*member*)

Member merupakan pengguna sistem yang telah terdaftar. Pengguna member ketika *login* akan menampilkan halaman *home* pengguna member, halaman konsultasi,

record penyakit, mengelola profil melihat informasi penyakit dan mendapat informasi terkait penyakit.

3. User non member/anggota

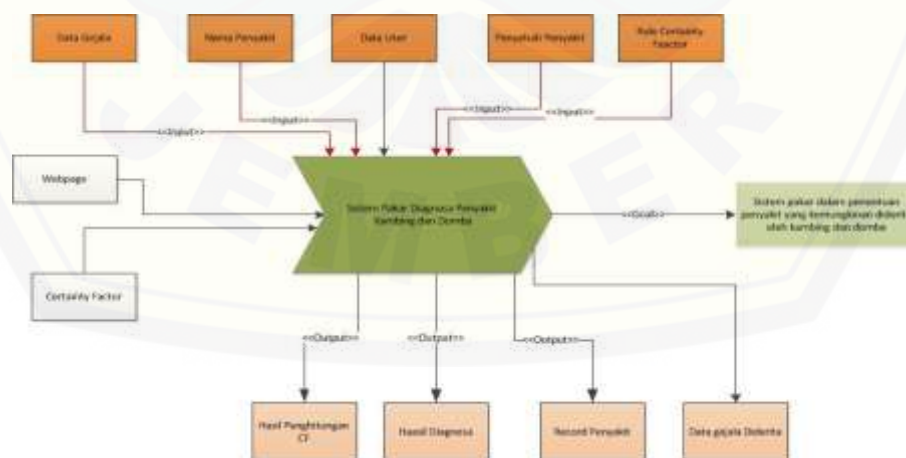
Kategori *user* yang tidak terdaftar pengguna yang ingin mengetahui mengenai penyakit kambing dan domba. Halaman yang dapat diakses meliputi *home*, informasi penyakit dan halaman konsultasi. Pada halaman konsultasi *user* umum dapat melakukan konsultasi tanpa mengakses sistem dan mampu untuk melakukan konsultasi tapi hasil konsultasi tersebut tidak disimpan.

4.2. Desain Sistem

Desain sistem yang dibuat meliputi *business process*, *use case diagram*, *use case scenario*, *sequence diagram*, *activity diagram*, *class diagram*, dan ERD.

4.2.1 Business Process

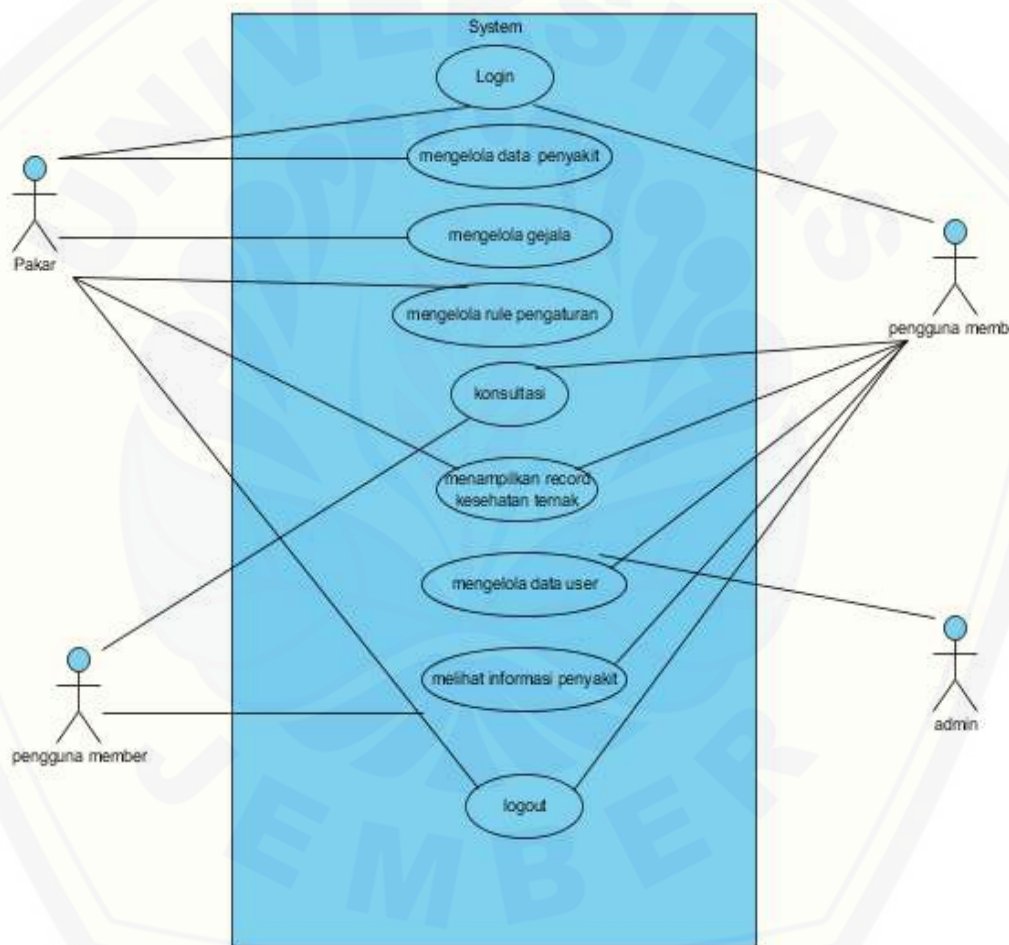
Bussines proses adalah aktivitas yang bertujuan untuk menghasilkan sebuah produk. Komponen yang terdapat pada business proses adalah proses *input* data, data luaran yang dihasilkan atau disebut dengan *output* dan tujuan dari sistem yang akan dicapai. Pada Sistem pakar penentuan penyakit kambing dan domba memiliki beberapa data *input* dan *output* yang dijelaskan pada gambar 4. 1.



Gambar 4. 1 *Business Proses*

4.2.2 Use Case Diagram

Use Case Diagram merupakan pemodelan yang dibuat untuk dapat menggambarkan interaksi antara aktor dengan sistem informasi. *Use case diagram* digunakan untuk mengetahui aksi yang dilakukan oleh *user* sesuai dengan hak akses yang dimiliki oleh setiap *user* atau *actor*. *Use Case Diagram* sistem ini dapat dilihat pada Gambar 4.2.



Gambar 4. 2 Use Case Diagram

Penjelasan tentang definisi aktor dan definisi *use case* dalam *use case diagram* pada Gambar 4.2 akan dijelaskan di bawah ini.

1. Definisi Aktor

Aktor merupakan pengguna yang terlibat dalam sistem. Pada sistem ini aktor dikelompokkan menjadi 3 yaitu, *user* umum, *user* terdaftar dan admin yang sekaligus menjadi pakar. Fungsi dan kegiatan aktor pada sistem penentuan penyakit kambing dan domba akan dijelaskan pada tabel 4.2:

Tabel 4. 2 Definisi Aktor

Aktor	Deskripsi
Pakar	Aktor yang memiliki hak akses utama pada sistem. Aktor ini dapat melakukan pengelolaan data gejala, penyakit, <i>user</i> dan <i>rule</i> (aturan <i>certainty factor</i>) pada sistem. Selain itu pakar juga dapat mengelola yang meliputi proses <i>edit</i> , hapus dan tambah <i>user</i> pada sistem. Pakar juga dapat melihat hasil konsultasi oleh member yang nantinya berguna untuk melakukan pengawasan terhadap kesehatan ternak.
Pengguna terdaftar	Aktor yang menggunakan sistem dalam melakukan konsultasi terkait gejala yang dapat dilihat dan memasukan gejala derita dalam sistem. Selain itu selain melakukan konsultasi pengguna terdaftar dapat menyimpan hasil konsultasi dan melihat <i>record</i> .

Pengguna tidak terdaftar	Pengguna tidak terdaftar adalah pengguna yang hanya dapat melakukan konsultasi dan hasil konsultasi tidak disimpan kedalam <i>database</i> .
Admin	Admin adalah petugas pelayanan yang memiliki tugas dalam melakukan pengolahan hak akses user. Admin juga berhak dalam mengelola data pakar yang dapat terlibat pada pengelolaan data.

2. Definisi *Use Case*

Use case merupakan bagian dari desain sistem yang menjelaskan *detail* dari *use case* diagram, definisi *use case* dapat dilihat pada table 4.3.

Tabel 4. 3 Definisi *Use case*

No	Use Case	Deskripsi
1.	Login	Menggambarkan proses autentifikasi untuk masuk ke sistem.
2.	Mengelola Data Penyakit	Menggambarkan proses pengolahan data yang meliputi tambah data baru, <i>edit</i> data, dan melihat <i>detail</i> data.
3.	Mengelola data gejala	Menggambarkan proses pengolahan data yang meliputi tambah data baru, <i>edit</i> data, dan melihat <i>detail</i> data..

4.	Mengelola Rule	Menggambarkan proses pengolahan data yang meliputi tambah data baru, <i>edit</i> data, dan melihat <i>detail</i> data.
5.	Melihat record	Menggambarkan proses melihat data hasil konsultasi yang telah dilakukan oleh <i>user</i> member atau anggota.
6.	Konsultasi	Menggambarkan proses melakukan perhitungan prediksi penyakit dengan melakukan pemilihan gejala yang diderita yang dilakukan oleh <i>user</i> . <i>User</i> bukan member dapat melakukan konsultasi tetapi hasil konsultasi tidak dapat disimpan.
7.	Informasi penyakit	Menggambarkan proses melihat <i>detail</i> informasi penyakit.
8.	Mengelola <i>Member</i> atau Anggota	Menggambarkan proses pengolahan data yang meliputi tambah data baru, <i>edit</i> data, dan melihat <i>detail</i> data. Sedangkan <i>user</i> dapat melakukan perubahan dan melihat <i>detail</i> data profil atau pribadi.
9.	Log out	Menggambarkan proses <i>user</i> keluar dari sistem

4.2.3 Use Case Scenario

Use case scenario digunakan untuk menjelaskan alur sistem sesuai dengan yang ada pada *Use case diagram* seperti pada Gambar 4.2.

1. Skenario Use Case Login

Penjelasan urutan aksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif skenario *use case login* dijelaskan pada lampiran tabel 4.4.

Tabel 4. 4 Skenario *Login Pakar*

Name	<i>Login</i>
Aktor	Pakar
Entry Condition	Pakar membuka halaman sistem
Exit Condition	Pengguna telah melakukan <i>Login</i> dan membuka halaman <i>home</i>
Skenario Utama	
<i>Login</i>	
Aksi Aktor	Reaksi Sistem
1. Buka halaman web sistem	
	2. Menampilkan halaman sistem pakar penentuan penyakit kambing dengan <i>detail</i> tampilan sebagai berikut: <ol style="list-style-type: none"> a. <i>Footer</i> yang berisi informasi Instansi b. <i>header</i> menu (<i>home</i>, <i>konsultasi</i>, <i>informasi</i> dan <i>login</i>) c. <i>content</i> berisi keterangan penjelasan sistem pakar
3. Klik menu <i>Login</i>	
	4. Menampilkan halaman <i>login</i> , yang berisi sebagai berikut: <ol style="list-style-type: none"> a. menampilkan <i>pop up</i> form <i>Login Password</i> dan <i>Username</i>

	<ul style="list-style-type: none"> b. <i>Button login</i> c. <i>Footer</i> yang berisi informasi Instansi d. <i>header</i> menu (home, konsultasi, informasi dan <i>login</i>) e. <i>button login</i>
5. Isi form <i>Login</i>	
a. <i>Username</i>	
b. <i>Password</i>	
6. Klik <i>Login</i>	
	7. Melakukan proses pemeriksaan kolom dan autentifikasi
	8. Menampilkan pesan”selamat datang admin” dan menampilkan tombol ok
9. Klik ok	
	10. Menampilkan halaman home serta <i>header</i> menu Gejala, Penyakit, <i>Rule</i> , <i>Record</i> , <i>User</i> , home dan log out . Menampilkan tabel <i>record</i> terakhir diambil dari data konsultasi, <i>user</i> , hasil_konsultasi berisi: <ul style="list-style-type: none"> a. Nama <i>user</i> b. Tanggal konsultasi c. Penyakit terbesar yang kemungkinan di derita
Scenario Alternatif	
“password atau <i>username</i> salah”	
Aksi Aktor	Aksi Sistem
	8.a.Menampilkan pesan <i>Login</i> “Kombinasi <i>Username</i> dan Password salah”.
9.a.Klik <i>button</i> ok	
	10.a. Menampilkan halaman home
Skenario Alternatif	

“kolom form <i>Login</i> kosong”	
Aksi Aktor	Aksi Sistem
	8.b Menampilkan pesan “ <i>please fill out the field</i> pada kolom yang kosong

2. Mengelola Data Penyakit

Use case scenario mengelola data penyakit digunakan untuk menjelaskan alur sistem sesuai dengan yang ada pada *Use case diagram* seperti pada Gambar 4.5.

Tabel 4. 5 Skenario Mengelola Data Penyakit

Name	Mengelola Data Penyakit
Aktor	Pakar
Entry Condition	Pakar sudah melakukan <i>login</i> dan membuka halaman menu penyakit
Exit Condition	Berhasil menambahkan penyakit
Skenario Utama	
“Menambah Penyakit”	
Aksi Aktor	Reaksi Sistem
1. Memilih menu penyakit	
	2. Menampilkan halaman penyakit yang meliputi: <ol style="list-style-type: none"> a. tabel <i>list</i> penyakit (id penyakit, nama penyakit) yang tersimpan <i>button action</i> hapus , dan <i>edit</i> b. <i>Button</i> tambah penyakit c. Navbar menu home. Gejala, penyakit, <i>rule</i>, <i>record</i> penyakit, <i>user</i> dan <i>logout</i> d. <i>Footer</i>

3. Klik tombol tambah penyakit	
	4. Menampilkan <i>pop up form</i> penyakit a. Nama penyakit b. Jenis penyakit c. Kategori penyakit d. Penanganan e. Pengobatan f. Obat g. Pencegahan h. Table penyakit yang sudah disimpan
5. Mengisi <i>form</i> penyakit a. Nama penyakit b. Jenis penyakit c. Kategori penyakit d. Penanganan e. Pengobatan f. Obat g. Pencegahan	
6. Klik tombol simpan	
	7. Melakukan proses pemeriksaan data yang akan disimpan
	8. Meampilkan pesan “Berhasil Disimpan” dan <i>button ok</i>
9. Klik <i>button ok</i>	
	10. Menampilkan halaman penyakit yang meliputi: a. tabel <i>list</i> penyakit (id penyakit, nama penyakit) yang tersimpan <i>button action</i> hapus , dan <i>edit</i> b. <i>Button</i> tambah penyakit c. <i>Navbar menu home</i> . Gejala, penyakit, <i>rule</i> , <i>record</i> penyakit, <i>user</i> dan <i>logout</i> d. <i>Footer</i>
Scenario Alternatif	

“form kosong tidak lengkap”	
Aksi Aktor	Reaksi Sistem
	8.a.Menampilkan pesan “ <i>please fill out the field</i> ” pada kolom yang kosong
Mengelola Data Penyakit(Mengubah)	
Aktor	Pakar memilih tombol <i>edit</i> yang berada pada halaman menu penyakit
Entry Condition	Pakar telah melakukan <i>login</i> dan telah melakukan edit data penyakit
Exit Condition	Berhasil mengubah data penyakit
Skenario Utama	
“Mengubah Penyakit”	
1. Klik tombol <i>edit</i>	
	2. Menampilkan form <i>edit</i> penyakit <ol style="list-style-type: none"> a. Nama penyakit b. Jenis penyakit c. Kategori penyakit d. Penanganan e. Pengobatan f. Obat g. Pencegahan h. <i>Button</i> simpan i. Navigasi menu gejala, penyakit <i>rule</i>, <i>record</i> penyakit, <i>user</i> dan <i>logout</i>
3. Mengubah data penyakit <ol style="list-style-type: none"> a. Nama penyakit b. Jenis penyakit c. Kategori penyakit d. Penanganan e. Pengobatan f. Obat g. Pencegahan 	
4. Klik <i>button</i> simpan	
	5. Melakukan proses pemeriksaan

	data yang akan disimpan
	6. Menampilkan pesan berhasil mengubah data dan <i>button</i> ok
7. Klik <i>button</i> ok	
	8. Menampilkan halaman penyakit yang meliputi: <ol style="list-style-type: none"> a. Tabel <i>list</i> penyakit(kode penyakit dan nama penyakit) yang telah di <i>update button action</i> hapus, dan <i>edit</i> b. Tambah penyakit c. <i>Navbar</i> menu <i>home</i>. Gejala, penyakit, <i>rule</i>, <i>record</i> penyakit,<i>user</i> dan <i>logout</i> d. <i>Footer</i>
Alternatif (Form Kosong)	
	6.a.Menampilkan pesan “ <i>please fill out the field</i> “ pada kolom yang belum terisi
Name	Mengelola Data Penyakit (Menghapus)
Aktor	Pakar
Entry Condition	Pakar telah melakukan <i>login</i> dan memilih tombol hapus
Exit Condition	Pakar berhasil menghapus data penyakit
Skenario Utama “Hapus Penyakit”	
1. Klik tombol hapus	
	2. Melakukan proses pemeriksaan data belum digunakan
	3. Menampilkan pesan “berhasil hapus data” dan <i>button</i> ok
4. Klik <i>button</i> ok	

	5. Menampilkan halaman penyakit dan table dengan data yang baru.
Skenario Alternatif	
“Gagal Hapus, Data digunakan Pada tabel Lain”	
	3.a. Manampilkan Pesan “gagal hapus data” dan <i>button</i> ok
4.a. Klik ok	
	5.a. Menampilkan halaman menu gejala, tabel <i>list</i> gejala yang tersimpan, <i>action button</i> (<i>edit</i> dan <i>hapus</i>) dan navigasi menu <i>home</i> . Gejala, penyakit, <i>rule</i> , <i>record</i> dan <i>logout</i>
Name	Mengelola Penyakit(Detail Penyakit)
Aktor	Pakar
Entry Condition	Pakar telah melakukan login dan memilih <i>button detail</i> pada halaman penyakit
Exit Condition	Berhasil menampilkan data detail penyakit
Skenario Utama	
Aksi Aktor	Reaksi Sistem
1. Klik menu penyakit	
	2. Menampilkan halaman menu penyakit, meliputi: <ul style="list-style-type: none"> a. <i>list</i> nomor penyakit dan nama penyakit <i>action detail</i>, <i>hapus</i> dan <i>edit</i> b. tambah penyakit c. menampilkan <i>header</i> menu <i>home</i>, penyakit, gejala, penyakit, <i>rule</i>, <i>user</i>, <i>record</i> penyakit dan <i>log uot</i> d. <i>footer</i>
3. Klik <i>detail</i>	

	<p>4. Menampilkan informasi <i>detail record</i> kesehatan ternak yang dipilih yang meliputi :</p> <ul style="list-style-type: none"> a. no /id penyakit b. Nama penyakit c. Jenis penyakit d. Kategori penyakit e. Penanganan f. Pengobatan g. Obat h. Pencegahan <p>Navigasi menu gejala, penyakit <i>rule</i>, <i>record</i> penyakit, <i>user</i> dan <i>logout</i></p>
--	---

3. *Scenario* mengelola gejala

Use case scenario mengelola gejala digunakan untuk menjelaskan alur sistem sesuai dengan yang ada pada *scenario diagram* yang dijelaskan pada Lampiran A.1.

4. *Scenario* Mengelola *rule*

Scenario mengelola *rule* digunakan untuk menjelaskan alur sistem mengelola data aturan atau *rule* yang meliputi tambah data, *edit* data, hapus data dan *view* data aturan. *Scenario* mengelola *rule* akan dijelaskan pada tabel 4.6.

Tabel 4. 6 *Scenario* Mengelola *rule*

Name	Mengelola Nilai <i>Rule</i> CF
Aktor	Pakar
Entry Condition	Pakar telah melakukan <i>login</i> dan membuka halaman <i>rule</i>
Exit Condition	Berhasil menambah data <i>rule</i>

Skenario Utama <i>“input nilai CF”</i>	
Aksi Aktor	Reaksi Sistem
1. Memilih menu <i>rule</i>	
	2. Menampilkan halaman <i>rule</i> , a. table data gabungan yang berisi nama penyakit, nama gejala, nilai md dan mb. b. <i>Button</i> tambah <i>rule</i> c. Terdapat <i>action edit</i> dan hapus d. navigasi menu home. Gejala, penyakit, <i>rule</i> , <i>record</i> penyakit, <i>user</i> dan <i>logout</i> e. <i>Footer</i>
3. Klik tambah <i>rule</i>	
	4. Menampilkan form nilai CF a. <i>Dropdown</i> nama penyakit b. <i>Dropdown</i> nama gejala c. Field nilai mb d. Filed nilai md e. <i>Button</i> simpan f. Menu navigasi gejala, penyakit, <i>rule record</i> , <i>user</i> dan <i>logout</i>
5. Mengisi data <i>rule</i> a. memilih <i>dropdown</i> nama gejala b. memilih nama penyakit, c. nilai mb d. nilai md	
6. Klik tombol simpan	
	7. Melakukan proses pemeriksaan data yang akan disimpan
	8. Menampilkan pesan “Berhasil Menyimpan” dan <i>button ok</i>
9. Klik <i>button ok</i>	
	10. Menampilkan halaman <i>rule</i> , a. tabel data gabungan yang

	<p>berisi nama penyakit, nama gejala, nilai md dan mb</p> <p>b. Terdapat action <i>edit</i> dan hapus</p> <p>c. navigasi menu home. Gejala, penyakit, <i>rule</i>, <i>record</i>, <i>user</i> dan <i>logout</i></p> <p>d. footer</p>
Scenario Alternatif “Nilai CF Kosong”	
Aksi Aktor	Reaksi Sistem
	8.a. Menampilkan pesan <i>please fill out this field</i>
Skenario Alternatif “Nilai CF Sudah Dimasukan”	
	8.b Menampilkan pesan Gagal <i>Input Nilai</i>
Name	Mengelola Nilai <i>Rule</i> CF (Mengubah nilai CF)
Aktor	Pakar
Entry Condition	Pakar telah melakukan <i>login</i> dan membuka halaman form edit rule
Exit Condition	Berhasil mengubah nilai rule
Skenario Utama “Mengubah Nilai CF”	
1. Klik tombol <i>edit</i>	
	<p>2. Menampilkan form nilai CF, meliputi:</p> <p>a. <i>Dropdown</i> nama penyakit</p> <p>b. <i>Dropdown</i> nama gejala</p> <p>c. Field nilai mb</p> <p>d. Filed nilai md</p> <p>e. <i>Button</i> simpan</p> <p>Menampilkan header gejala, penyakit, <i>rule record</i>, <i>dropdown</i></p>

	berisi <i>user</i> dan <i>logout</i> serta Menampilkan footer
4. Mengubah nilai <i>rule</i> a. nilai <i>mb</i> b. nilai <i>md</i>	
5. Klik <i>simpan</i>	
	6. Melakukan proses pemeriksaan data yang akan disimpan
	7. Menampilkan pesan “berhasil simpan data” dan <i>button ok</i>
8. Klik <i>button ok</i>	
	9. Menampilkan halaman <i>rule</i> , a. table data gabungan yang berisi nama penyakit, nama gejala, nilai <i>md</i> dan <i>mb</i> . b. Terdapat <i>action edit</i> dan hapus c. navigasi menu home. Gejala, penyakit, <i>rule</i> , <i>record</i> , <i>user</i> dan <i>logout</i> d. footer
Name	Mengelola Nilai <i>Rule CF</i>
Aktor	Pakar
Entry Condition	Pakar telah melakukan <i>login</i> dan memilih tombol hapus
Exit Condition	Berhasil menghapus data <i>rule</i>
Skenario Utama “Hapus Penyakit”	
1. Klik tombol hapus	
	2. Melakukan proses pemeriksaan data belum digunakan
	3. Menampilkan pesan “berhasil hapus data” dan <i>button ok</i>
4. Klik <i>button ok</i>	
	5. Menampilkan halaman penyakit

	dan table dengan data yang baru.
Skenario Alternatif	
“Gagal Hapus, Data digunakan Pada tabel Lain”	
	3.a. Menampilkan pesan “gagal hapus data” dan <i>button</i> ok
4.a. Klik ok	
	5.a. Menampilkan halaman menu gejala, tabel <i>list</i> gejala yang tersimpan, <i>action button</i> (<i>edit</i> dan <i>hapus</i>) dan navigasi menu home. Gejala, penyakit, <i>rule</i> , <i>record</i> dan <i>logout</i>

5. *Scenario* Konsultasi

Scenario konsultasi digunakan untuk melakukan konsultasi dengan memilih gejala yang diderita oleh ternak. Gejala yang dipilih nantinya akan menghasilkan perhitungan metode *certainty factor* yang akan dijelaskan pada *scenario* pada tabel 4.7.

Tabel 4. 7 *Scenario* Konsultasi

Name	Konsultasi
Aktor	Member
Entry Condition	Member telah melakukan <i>login</i> dan membuka halaman konsultasi
Exit Condition	Member berhasil melakukan konsultasi dan berhasil membuka halaman diagnosa
Skenario Utama	
Aksi Aktor	Reaksi Sistem
1. Klik menu Konsultasi	
	2. Menampilkan halaman konsultasi dan <i>list</i> navigasi manu home,

	konsultasi, <i>record</i> , informasi, <i>dropdown user</i> profil <i>user</i> dan <i>logout</i> serta form konsultasi : a. Jenis kambing b. Menampilkan <i>list</i> nama gejala c. Serta menampilkan <i>button</i> simpan dan reset
3. Mengisi form konsultasi a. Jenis kambing b. Memilih nama gejala	
4. Klik tombol simpan	
	5. Melakukan proses pemeriksaan data yang akan disimpan
	6. Menampilkan halaman hasil Konsultasi, a. <i>detail</i> informasi penyakit(nama penyakit, jenis penyakit, penyebab, obat, penanganan dan pencegahan) yang memiliki hasil persentase dari penyakit tertinggi b. hasil perhitungan dari penyakit yang kemungkinan di derita serta nama penyakit.
Scenario Alternatif	
“Data yang dipilih kosong”	
Aksi Aktor	Reaksi Sistem
	6.a. Menampilkan <i>pop up</i> pesan “gagal tambah data konsultasi”

6. *Scenario* melihat record ternak

Use case scenario digunakan untuk menjelaskan alur sistem sesuai dengan yang ada pada *Use case diagram* yang dijelaskan pada Lampiran A.1.

7. *Scenario* mengelola user

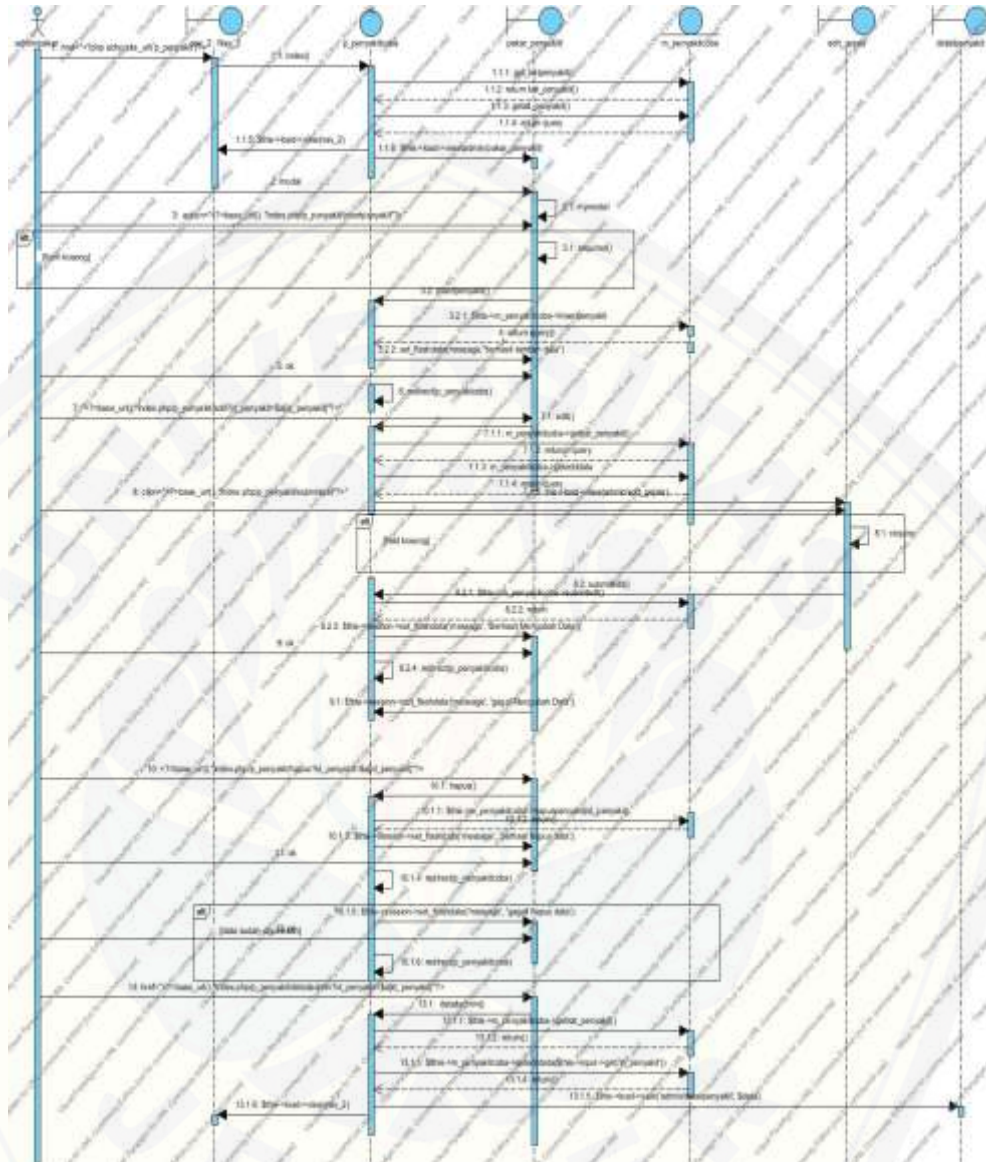
Use case scenario digunakan untuk menjelaskan alur sistem sesuai dengan yang ada pada *Use case diagram* yang dijelaskan pada Lampiran A.1.

8. *Scenario* melihat informasi penyakit
Use case scenario digunakan untuk menjelaskan alur sistem sesuai dengan yang ada pada *Use case diagram* yang dijelaskan pada Lampiran A.1.
9. *Scenario* logout
Use case scenario digunakan untuk menjelaskan alur sistem sesuai dengan yang ada pada *Use case diagram* yang dijelaskan pada Lampiran A.1.

4.2.4 *Sequence Diagram*

Sequence diagram adalah diagram yang digunakan untuk menggambarkan interaksi yang terjadi antarobjek di dalam sistem yang disusun.

1. *Sequence Diagram* Login
Penggambaran *sequence diagram login* digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat seperti yang ditunjukkan pada lampiran A.3 .
2. *Sequence* Mengelola penyakit
Penggambaran *sequence diagram mengelola penyakit* digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat seperti yang ditunjukkan pada Gambar4.3 .



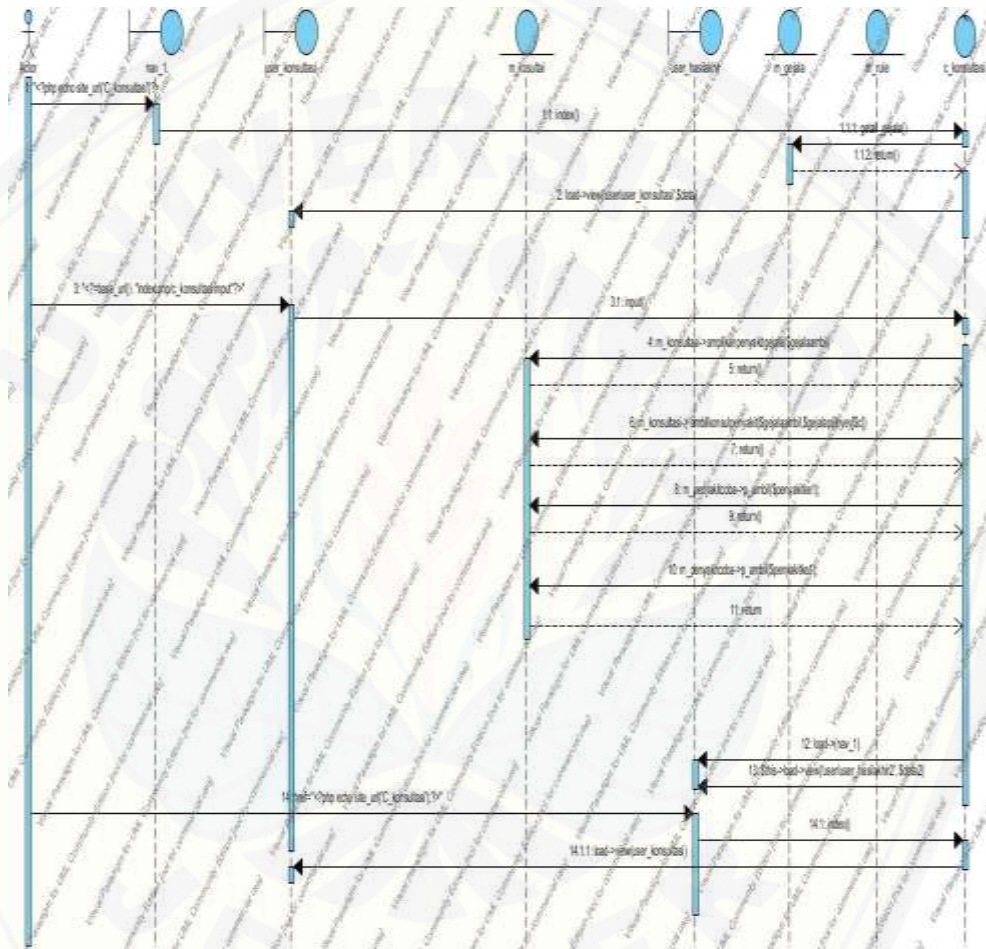
Gambar 4. 3 *Sequence Diagram* Mengelola Penyakit

3. *Sequence* mengelola gejala

Penggambaran *sequence diagram* mengelola gejala digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat seperti yang ditunjukkan pada Gambar 4.4. .

5. Sequence konsultasi

Penggambaran *sequence diagram user member* digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat seperti yang ditunjukkan pada Gambar 4.6.



Gambar 4. 6 *Sequence Diagram* Konsultasi *User Member*

6. *Sequence* menampilkan *record* konsultasi

Penggambaran *sequence diagram record* digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat seperti yang ditunjukkan pada lampiran A.3.

7. *Sequence* Mengelola data *user*

Penggambaran *sequence diagram user* digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat seperti yang ditunjukkan pada lampiran A.3.

8. *Sequence* Melihat informasi penyakit

Penggambaran *sequence diagram* melihat informasi penyakit digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat seperti yang ditunjukkan pada lampiran A.3.

9. *Sequence* logout

Penggambaran *sequence diagram logout* digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat seperti yang ditunjukkan pada lampiran.

4.2.5. *Activity* Diagram

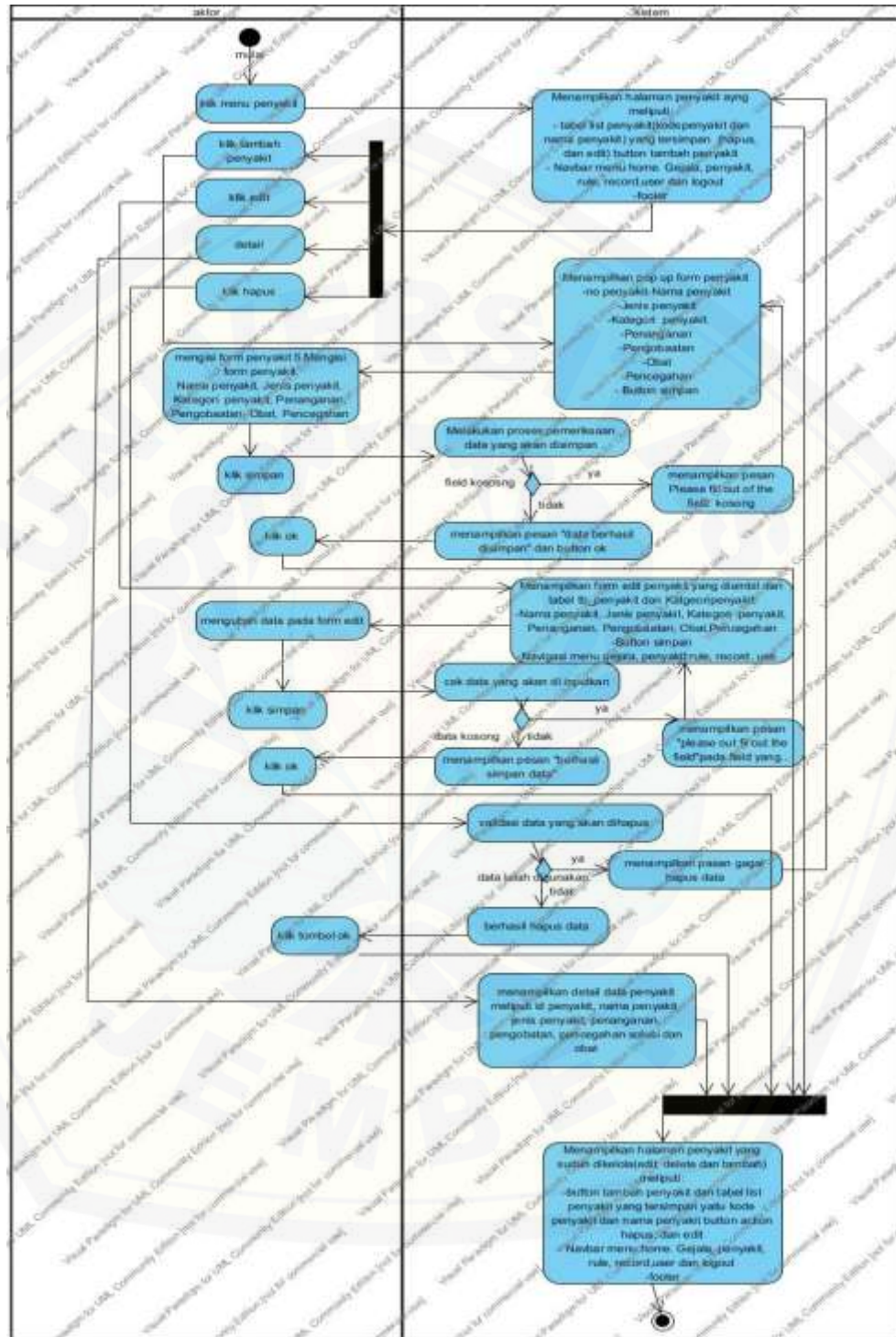
Activity diagram adalah dokumentasi desain yang menggambarkan aliran aktivitas dalam sistem informasi diagnosa penyakit pada kambing dan domba menggunakan metode *certainty factor*. Berikut penjelasan seluruh dan contoh *activity diagram* pada sistem pakar diagnosa penyakit kambing dan domba:

1. *Activity* Diagram Login

Alur *Activity diagram view login* menjelaskan tentang proses *login* kesistem yang dilakukan oleh pakar dan *user member*, penjelasan *activity* dapat dilihat pada lampiran A.2.

2. *Activity* Diagram Mengelola data penyakit

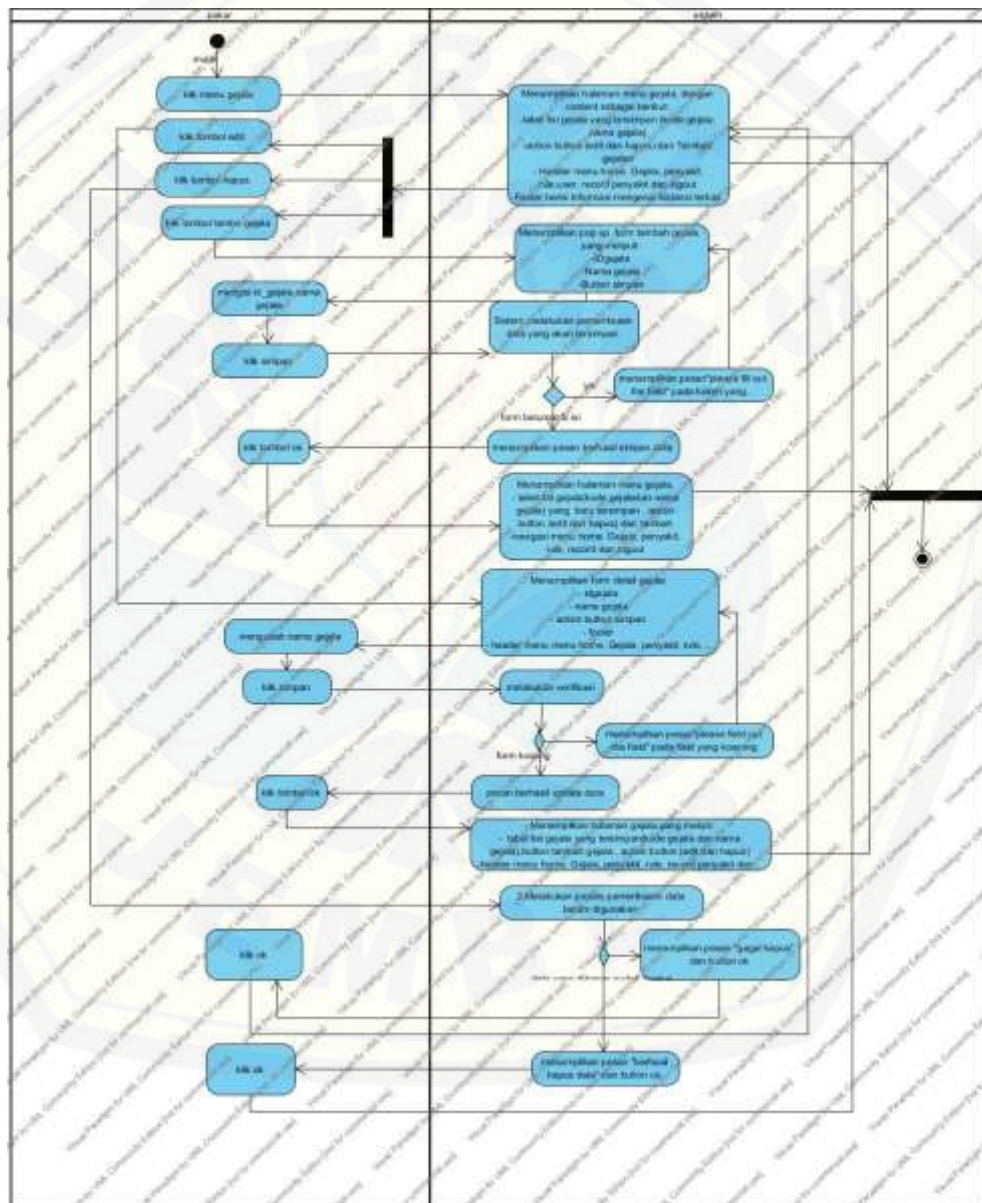
Activity diagram Mengelola Data Penyakit menggambarkan aktivitas pada sistem yang dilakukan oleh pakar. Aktivitas tersebut meliputi tambah, *edit*, hapus dan lihat *detail* penyakit . Proses mengelola data hanya dapat dilakukan oleh pakar. Alur dari *activity diagram* mengelola data penyakit dapat dilihat pada gambar 4. 7.



Gambar 4. 7 Activity Diagram Mengelola Data Penyakit

3. Activity Diagram Mengelola Gejala

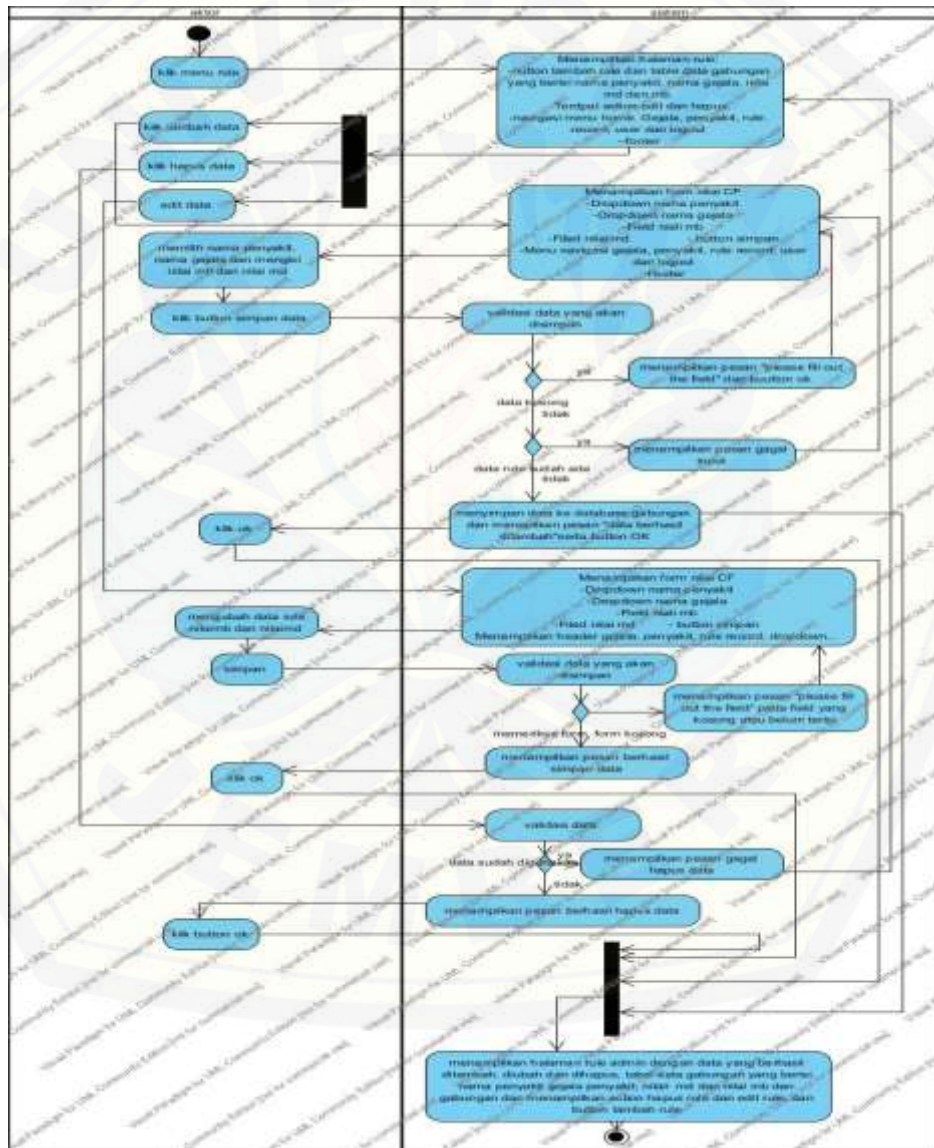
Activity diagram mengelola gejala adalah gambaran aktivitas pengolahan data yang dilakukan oleh pakar yang meliputi aktifitas tambah data gejala, edit data gejala dan hapus data gejala. Alur activity diagram mengelola gejala dapat dilihat pada gambar 4.8.



Gambar 4. 8 Activity Diagram Mengelola Gejala

4. Activity Diagram Mengelola Rule

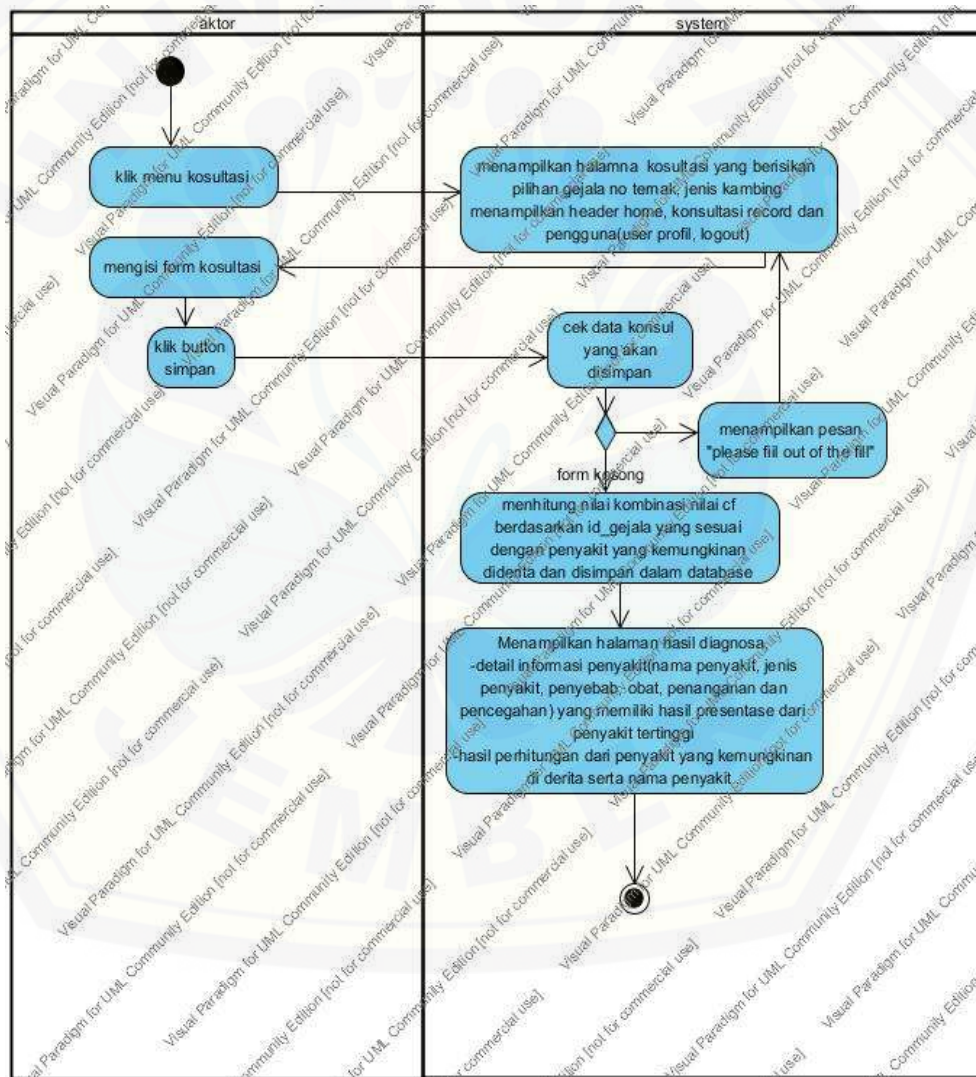
Activity diagram mengelola rule atau aturan adalah gambaran aktivitas pengolahan data yang dilakukan oleh pakar yang meliputi aktivitas tambah data, edit dan hapus data aturan atau rule. Alur activity diagram mengelola rule atau aturan dapat dilihat pada gambar 4.9.



Gambar 4. 9 Activity Diagram Mengelola Rule

5. Activity Diagram Konsultasi

Activity diagram Konsultasi adalah gambaran aktivitas pemilihan gejala penyakit yang dilakukan oleh *user*. Penyakit yang dipilih tersebut akan digunakan untuk melakukan perhitungan dengan melihat bobot gejala pada masing – masing penyakit yang dimiliki oleh gejala terpilih. Alur *activity diagram* mengelola gejala dapat dilihat pada gambar 4.10.



Gambar 4. 10 *Activity Diagram* Melakukan Konsultasi

6. *Activity Diagram Melihat Record Hasil Konsultasi*

Activity diagram record hasil konsultasi berfungsi menampilkan hasil *record* konsultasi yang pernah dilakukan. *Record* konsultasi merupakan hasil dari konsultasi yang dilakukan oleh *user* anggota. Penjelasan mengenai *activity diagram record* konsultasi dapat dilihat pada lampiran A.2.

7. *Activity Diagram Mengelola Data User*

Activity diagram mengelola data user adalah aktivitas sistem yang dilakukan oleh pakar. Aktivitas tersebut meliputi menambah *user*, *edit user*, hapus *user* dan melihat *detail user*. Data *user* digunakan sebagai acuan pengaturan hak akses. Penjelasan mengenai *activity diagram* mengelola data *user* konsultasi dapat dilihat pada lampiran A.2.

8. *Activity Diagram Melihat Informasi Detail Penyakit*

Activity diagram melihat *detail* informasi penyakit adalah gambaran tentang aktivitas sistem yaitu menampilkan *detail* penyakit. *detail* penyakit menampilkan informasi lengkap terkait penyakit yang terdapat pada sistem. Penjelasan mengenai *activity diagram* melihat *detail* penyakit dapat dilihat pada lampiran A.2.

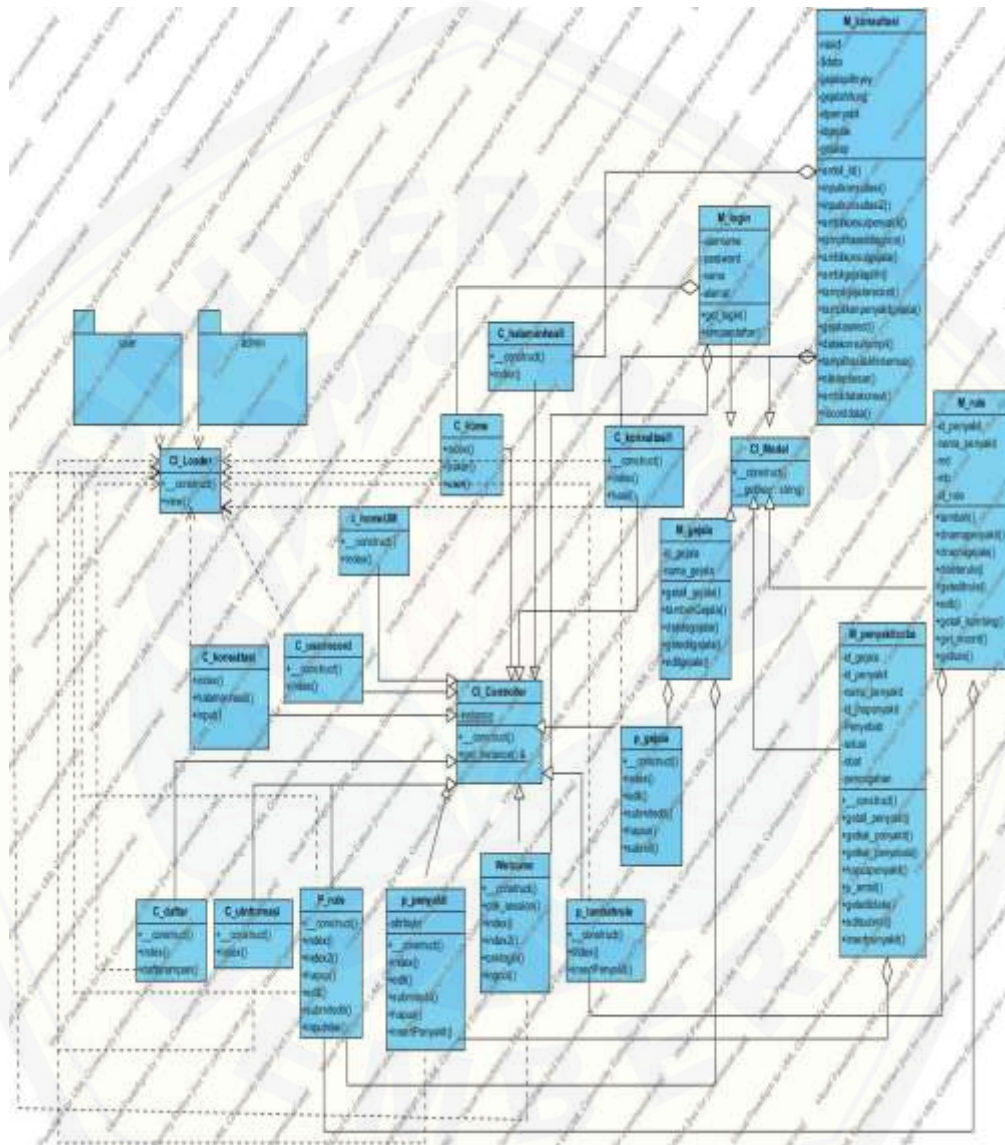
9. *Activity Diagram Log out*

Penjelasan mengenai *activity diagram* log out penyakit dapat dilihat pada lampiran A.2. Log out digunakan untuk keluar dari sistem. setelah melakukan log out akan arahkan menuju kehalaman home awal.

4.2.6 *Class Diagram*

Class Diagram merupakan gambaran dari berbagai *class* yang saling terhubung sesuai dengan model penerapan sistem atau coding yang dalam kasus ini menggunakan *Model-View-Controller* atau MVC berdasarkan *framework Codeigniter* sehingga pada *class diagram* ini terdapat tiga *package*, yaitu *model*, *controller*, dan

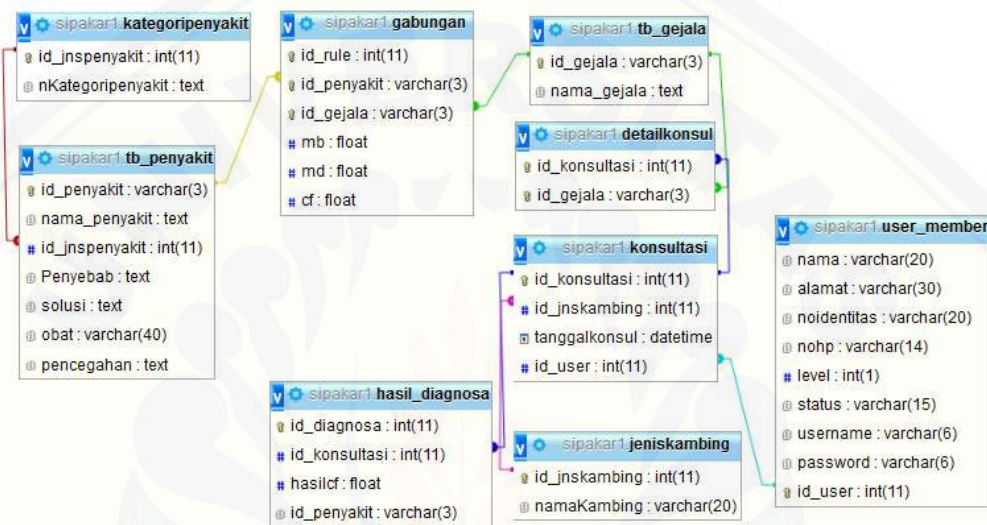
view. membutuhkan masukan atau *input* yang didapat dari view. Gambaran class diagram akan dipaparkan pada Gambar 4.11.



Gambar 4. 11 Class Diagram

4.2.7 Entity Relationship Diagram (ERD)

ERD merupakan gambaran komponen dan struktur *database* yang digunakan dalam pembangunan sistem. ERD pada sistem pakar diagnosa penyakit pada kambing dan domba menggunakan metode *certainty factor* dapat dilihat pada Gambar 4.12.



Gambar 4. 12 Entity Relationship Diagram

4.3 Penulisan Kode Program

Tahapan penulisan kode program merupakan implementasi *design* yang telah dibuat kedalam bentuk baris kode-kode program. Bahasa pemrograman yang digunakan dalam sistem pakar diagnosa penyakit kambing dan domba dengan menggunakan metode *certainty factor* adalah bahasa PHP, *Hyper Text Markup Language* (HTML), JavaScript, dan *Structured Query Language* (SQL). Framework yang digunakan adalah *Codeigniter* dan pengaturan tampilan digunakan CSS.

Penerapan kode pemrograman perhitungan metode *certainty factor* (CF) diterapkan pada file controller dengan nama *class* konsultasi. Model yang digunakan

adalah file model `m_konsultasi` dan `M_gejala`. Halaman *view* `v_konsultasi` merupakan *class* yang digunakan untuk menampilkan halaman konsultasi menampilkan *list* gejala penyakit, sedangkan `v_hasilakhir` adalah *class* yang menampilkan hasil akhir proses perhitungannya. Kode program metode *certainty factor* dapat dilihat pada gambar 4.13

```
1 <?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
2
3 class C_konsultasi extends CI_Controller {
4
5     public function __construct(){
6         parent::__construct();
7         $this->load->model('m_gejala');
8         $this->load->model('m_rule');
9         $this->load->model('m_konsultasi');
10        $this->load->model('m_penakitcoba');
11    }
12
13    public function index(){
14        $this->load->view('nav3');
15        $data['gejala']= $this->m_gejala->getall_gejala();
16        $data['kambing']=$this->m_rule->getall_kambing();
17        $this->load->view('user/user_konsultasi',$data);
18    }
19    public function halamanhasil(){
20        $this->load->view('nav3');
21        $this->load->view('user/halamanhasil');
```

```
22 }
23
24 public function inputHitungCF(){
25
26 $data['nomer']=$this->m_konsultasi->ambil_id();
27 $data['idKambing']=$this->input->post('id_jnskambing');
28 $data['nomorkambing']=$this->input->post('nomorkambing');
29 $data['idgejala']=$this->input->post('idgejala');
30 $idgejala= $data['idgejala'];
31 $data['total']= $data['idgejala'];
32 $nomor= $data['nomer'];
33 $tambah = $this->m_konsultasi->inputkonsultasi($data);
34
35 $gejalaambil= "";
36 $gejalaambil = implode(",", $idgejala);
    $gejalapilih= $this->m_konsultasi->tampilkanpenyakitgejala($gejalaambil)-
37 >result_array();
38
39 $data1['tampilgejala']= $this->m_konsultasi->gejalaselect($data);
40 for ($i=0; $i < count($gejalapilih); $i++) {
41 foreach ($gejalapilih as $gejalapilih2) {
42 $gejalapilih2['id_penyakit'];
43 $namapenyakit1 []= $gejalapilih2['nama_penyakit'];
44 }
45 }
46 $gejalahitung =count($gejalapilih);
47 for($c=0; $c<$gejalahitung; $c++){
```

```
48 $cf= array();
    $pilih = $this->m_konsultasi->
49 ambilkonsulpenyakit($gejalaambil,$gejalapilihyyey[$c]);
50 for($j=0; $j<count($pilih);$j++){
51 foreach ($pilih->result_array() as $gabungan) {
52 $cf[]= $gabungan['cf'];
53 $gejalapilih1= $gabungan['id_gejala'];
54
55 }
56 }
57 $nilaicf= 0;
58 for($m = 0; $m<count($cf); $m++){
59 if($nilaicf<0 && $cf[$m]<0){
60 $nilaicf = $nilaicf + ($cf[$m] * (1 + $nilaicf ));
61 }
62 elseif($nilaicf>=0 && $cf[$m]>=0){
63 $nilaicf = $cf[$m] + ($nilaicf * (1 - $cf[$m] ));
64 }
65 else {
66 $nilaicf = ($cf[$m] + $nilaicf) / 1 - (min(abs($nilaicf), abs($cf[$m])));
67 }
68
69 }
70
71 $nilai= $nilaicf*100;
72 $nilaicf2 []= $nilai;
73 $data2['cari']=$namapenyakit1;
```



```
74 $data2['coba']=$nilaicf2;
75 }
    $input= $this->m_konsultasi-
76 >inputkonsultasi2($data,$nilaicf2,$gejalapilihney,$gejalahitung);
77 $data1['datakonsul']= $this->m_konsultasi->tampilhasilakhirsemua($nomor);
78 $data1['hasil']= $this->m_konsultasi->datakonsultampil($nomor);
79 $data1['maks']= $this->m_konsultasi->nilaiterbesar($nomor);
80 $this->load->view('nav3', $data1);
81 $this->load->view('user/user_hasilakhir', $data1);
82
83 }
84
85
86 }
87 ?>
```

Gambar 4. 13 Kode Program C_konsultasi

Pada Tabel 4.13. merupakan implementasi metode *certainty factor* kedalam kode program yang terdapat pada *class* c_konsultasi. *Class* tersebut terdapat *method construct*, *input*, *index* dan halaman hasil. *Method construct* pada bari ke-5 hingga ke-11 berisi kode program yang digunakan untuk memanggil *class* lain terdapat pada *class* c_konsultasi, *class* yang dipanggil meliputi *class* m_gejala, m_rule, m_konsultasi dan m_penyakit. *Method index* merupakan *method* yang digunakan untuk menampilkan halaman konsultasi yang berisi nama gejala yang didapat dengan memanggil metode *getall_gejala* yang terdapat pada kelas m_gejala.

Metode *inputHitungCF()* merupakan *method* yang didalamnya terdapat proses perhitungan metode *certainty factor*. Pada awal metode pada baris ke-26 hingga baris

ke-33 terdapat fungsi *post* yang digunakan untuk mengambil data yang akan disimpan pada *database*. Fungsi *post* ini menampung data pada *form input* yang terdapat pada halaman *view user_konsultasi*. Baris ke- 33 berfungsi memanggil model yang digunakan untuk proses penyimpanan data ke *database*, model yang dipanggil terdapat pada *class m_konsultasi* dengan nama *function input konsultasi* dengan parameter yang disimpan bernama *\$data* yang merupakan variable penampung data yang akan disimpan.

Baris ke-35 hingga baris ke-45 digunakan untuk mendapatkan data penyakit berdasarkan pada gejala yang dipilih. Variabel *\$idgejala* yang merupakan gejala yang dipilih pada *form konsultasi*. *\$idgejala* tersebut kemudian ditampung dengan menggunakan fungsi *implode* yang digunakan untuk mendapatkan data penyakit yang berhubungan dengan gejala yang dipilih oleh *user*. Proses selanjutnya baris ke-46 menghitung jumlah penyakit yang ditemukan pada proses sebelumnya untuk melakukan proses perulangan dalam mencari nilai *certainty factor* pada setiap penyakit.

Pada proses menghitung nilai kombinasi *certainty factor* dengan melihat nilai *cf* masing gejala berdasarkan pada penyakit. pada baris 47 dan 48 melakukan perulangan dan membandingkan gejala yang ada pada penyakit dengan gejala yang dipilih *user* untuk selanjutnya mengambil nilai *cf* pada setiap gejala berdasarkan penyakit. Selanjutnya pada baris berikut mengambil nilai *cf* pada gejala berdasarkan penyakit.

Pada baris 58 hingga baris ke- 68 melakukan pemeriksaan nilai jika nilai *cf* memiliki nilai positif akan dilakukan proses perhitungan pada baris 60. jika salah satu dari nilai *cf* bernilai negatif atau memiliki nilai yang berbeda akan dilakukan proses perhitungan baris 66. Nilai akhir *certainty factor* yang telah diketahui dikalikan dengan nilai 100 untuk mendapatkan hasil persentase. Hasil akhir yang diketahui disimpan pada *database* dengan memanggil *class model m_konsultasi*

dengan metode *input* konsultasi² dengan parameter yang disimpan \$data, \$nilaif2, \$gejalapilih^{ey}, \$gejalahitung yang dituliskan pada baris ke- 76.

Baris ke- 77 hingga 81 berfungsi untuk menampilkan halaman hasil dari konsultasi. baris 77 hingga 79 memanggil data pada model m_konsultasi, data yang dipanggil pertama terletak pada metod tampilhasilakhirsemua yang bertujuan menampilkan hasil perhitungan *certainty factor*. Pada baris berikutnya menampilkan *detail* konsultasi data gejala yang dipilih. Selanjutnya yang terakhir menampilkan hasil konsultasi dengan nilai *certainty factor* yang tertinggi untuk ditampilkan *detail* informasi penyakit.

4.4 Pengujian Sistem

Pengujian dilakukan untuk menguji dan mengevaluasi sistem dengan menggunakan metode pengujian sistem. Proses pengujian dilakukan dengan pengujian *whitebox* terlebih dahulu kemudian akan dilanjutkan dengan pengujian *blackbox*.

4.4.1 Pengujian *White box*

Pengujian *white box* adalah pengujian perangkat lunak dengan melihat modul yang telah dibuat dengan program yang ada. Pengujian *white box* adalah pengujian yang dilakukan dengan melakukan pengecekan pada perancangan menggunakan desain sistem yang kemudian dibagi kedalam beberapa kasus pengujian. Pengujian dilakukan untuk menilai fungsi dari sistem hasil pengolahan sistem sesuai dengan yang diinginkan. Pengujian *white box* dapat dilakukan dengan cara sebagai berikut:

1. *Listing Program*

Pada tahap *listing* program akan dihitung kumpulan kode program yang akan diujikan. kode program akan dijelaskan pada lampiran, untuk contoh dapat dilihat pada lampiran.

2. Diagram alir

Diagram merupakan notasi yang digunakan untuk merepresentasikan aliran control yang digambarkan dari hasil penomoran dari *listing* program. Diagram alir ini menggambarkan logika alur kontrol logika pada program. Diagram alir digambarkan dengan *node* (simpul) yang dihubungkan dengan *edge* (garis).

3. Kompleksitas siklomatik (*Cyclomatic Complexity*)

Kompleksitas siklomatik merupakan pengujian perangkat lunak yang menyediakan ukuran kuantitatif dari kompleksitas logis suatu program. Setiap node dan garis pada kompleksitas siklomatik akan memiliki nilai yang disebut dengan jalur independen. Perhitungan kompleksitas siklomatik menggunakan persamaan 3 sebagai berikut :

$$V(G) = E - N + 2 \quad \dots \text{Persamaan 4.1}$$

Keterangan:

$V(G)$ = Kompleksitas siklomatik

E = Jumlah edge (garis)

N = Jumlah node (simpul)

Kompleksitas siklomatik pada bagian metode dapat dilihat pada sub bab C. Kompleksitas Siklomatik (*Cyclomatic Complexity*).

4. Jalur independen (*Independent Path*)

Jalur independen adalah setiap jalur yang melalui program, menunjukkan setidaknya satu kumpulan pernyataan-pernyataan pemrosesan atau kondisi baru. Jalur independen dalam grafik alir bergerak setidaknya satu *edge* yang belum dilintasi sebelum jalur tersebut didefinisi.

5. Pengujian Basis Set (*Test Case*)

Pengujian Basis Set menggambarkan pelaksanaan jalur. Data yang dieksekusi dimasukkan ke dalam grafik alir apakah sudah melewati basis set yang tersedia.

Sistem telah memenuhi syarat kelayakan perangkat lunak jika salah satu jalur yang dieksekusi setidaknya satu kali.

Pengujian *white box* secara *detail* akan terletak pada lampiran berikut ini adalah contoh pengujian *white box* yang dilakukan pada class dan function yang merupakan implementasi kode program metode *certainty factor*.

1. Listing Program *User_konsultasi*

```
5 public function __construct(){
6     parent::__construct();
7     $this->load->model('m_gejala');
8     $this->load->model('m_rule');
9     $this->load->model('m_konsultasi');
10    $this->load->model('m_penakitcoba');
11
12 }
```

Gambar 4. 14 Listing Program *function __construct()*

```
14 public function index(){
15     $this->load->view('nav3');
16     $data['gejala'] = $this->m_gejala->getall_gejala();
17     $data['kambing'] = $this->m_rule->getall_kambing();
18     $this->load->view('user/user_konsultasi',$data);
19 }
```

Gambar 4. 15 Listing Program *Function Index()*

```
20 public function halamanhasil(){
21     $this->load->view('nav3');
22     $this->load->view('user/halamanhasil');
23 }
```

Gambar 4. 16 Listing Program *Function Halaman Hasil*

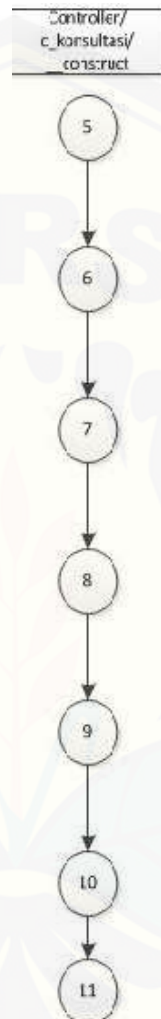
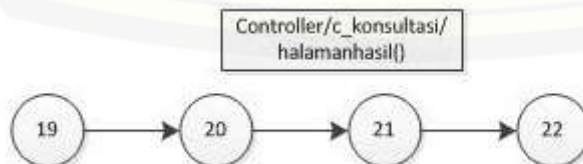
```

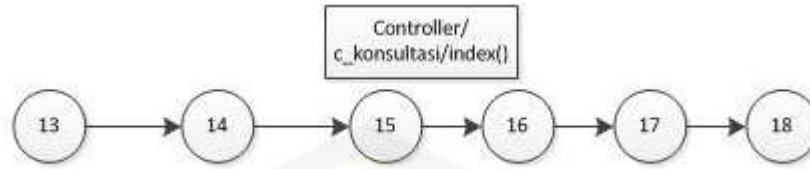
25 public function inputhitungCF(){
26
27     $data['nomer'] = $this->m_konsultasi->ambil_id();
28     $data['idxkambing'] = $this->input->post('id_inkambing');
29     $data['nomorkambing'] = $this->input->post('nomorkambing');
30     $data['idgejala'] = $this->input->post('idgejala');
31     $idgejala = $data['idgejala'];
32     $data['total'] = $data['idgejala'];
33     $nomor = $data['nomer'];
34     $tambah = $this->m_konsultasi->inputkonsultasi($data);
35
36     $gejalaambil = "";
37     $gejalaambil = implode(", ", $idgejala);
38     $gejalapilih = $this->m_konsultasi->tampilkanpenyakitgejala($gejalaambil)->result_array();
39     echo count($gejalapilih);
40     $data1['tampilgejala'] = $this->m_konsultasi->gejalaselect($data);
41     for ($i=0; $i < count($gejalapilih); $i++) {
42         foreach ($gejalapilih as $gejalapilih2) {
43             $gejalapilih2['id_penyakit'];
44             $namapenyakit1 [] = $gejalapilih2['nama_penyakit'];
45         }
46     }
47     $gejalahitung = count($gejalapilih);
48     for($c=0; $c < $gejalahitung; $c++){
49         $pilih = $this->m_konsultasi->ambilkonsulpenyakit($gejalaambil,$gejalapilih2[$c]);
50         for($j=0; $j < count($pilih);$j++){
51             foreach ($pilih->result_array() as $gabungan) {
52                 $cf[] = $gabungan['cf'];
53                 $mb[] = $gabungan['mb'];
54                 $md[] = $gabungan['md'];
55                 $gejalapilih1 = $gabungan['id_gejala'];
56             }
57         }
58     }
59     $nilaimb = 0;
60     $nilaicf = 0;
61     for($m = 0; $m < count($md); $m++){
62         if($nilaicf <= $cf[$m]){
63             $nilaicf = $nilaicf + ($cf[$m] * (1 - $nilaicf));
64         }
65         elseif($nilaicf <= $cf[$m]){
66             $nilaicf = $nilaicf + ($cf[$m] * (1 - $nilaicf));
67         }
68         else {
69             $nilaicf = ($cf[$m] + $nilaicf) / 1 + (min(abs($nilaicf), abs($cf[$m])));
70         }
71     }
72
73     $nilai = $nilaicf*100;
74     $nilaicf2 [] = $nilai;
75     $data2['cari'] = $namapenyakit1;
76     $data2['coba'] = $nilaicf2;
77
78 }
79 $input = $this->m_konsultasi->inputkonsultasi2($data,$nilaicf2,$gejalapilih2,$gejalahitung);
80 $data1['datakonsul'] = $this->m_konsultasi->tampilhasilakhirsemua($nomor);
81 $data1['hasil'] = $this->m_konsultasi->datakonsultampil($nomor);
82 $data1['maks'] = $this->m_konsultasi->nilaiterbesar($nomor);
83 $this->load->view('nav3', $data1);
84 $this->load->view('user/user_hasilakhir', $data1);
85

```

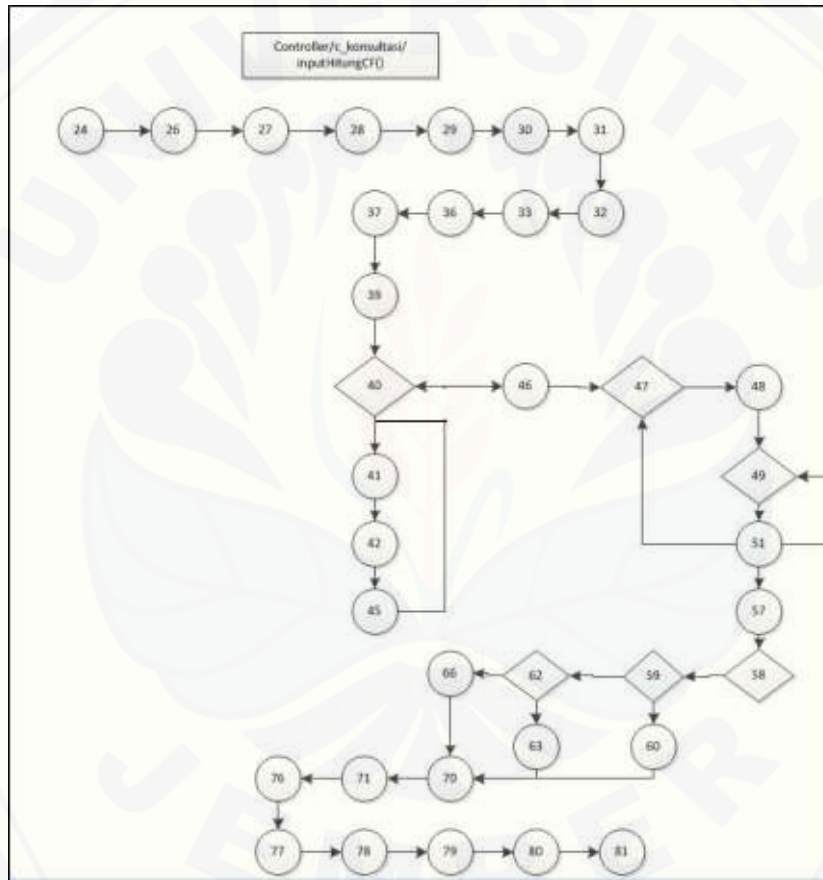
Gambar 4. 17 Listing Program function metode certainty factor

2. Diagram Alir

Gambar 4. 18 Diagram Alir *function__construct()*Gambar 4. 19 Diagram Alir *function_halamanhasil()*



Gambar 4. 20 Diagram Alir *function_index()*



Gambar 4. 21 Diagram Alir *function_inputHitungCF()*

3. Komplexitas Siklometri (CC)

a. *Function_construct*

$$V(G) = E - N + 2$$

$$V(G) = 6 - 7 + 2 = 1$$

b. *Function_index()*

$$V(G) = E - N + 2$$

$$V(G) = 6 - 7 + 2 = 1$$

c. *Function_hasilakhir*

$$V(G) = E - N + 2$$

$$V(G) = 6 - 7 + 2 = 1$$

d. *Function_inputHitungCF()*

$$V(G) = E - N + 2$$

$$V(G) = 40 - 36 + 2 = 6$$

4. Jalur Independen (Independent Path)

a. *Function__construct*

$$\text{Jalur 1: } 5 - 6 - 7 - 8 - 9 - 10 - 11$$

b. *Function_index*

$$\text{Jalur 1: } 13 - 14 - 15 - 16 - 17 - 18$$

c. *Function_hasilakhir*

$$\text{Jalur 1 : } 19 - 20 - 21 - 22$$

d. *Function_proses*

$$\begin{aligned} \text{Jalur 1 : } & 25 - 26 - 27 - 28 - 29 - 30 - 31 - 32 - 33 - 36 - 37 - 39 - 40 - 41 \\ & - 42 - 45 - 40 \end{aligned}$$

$$\begin{aligned} \text{Jalur 2 : } & 25 - 26 - 27 - 28 - 29 - 30 - 31 - 32 - 33 - 36 - 37 - 39 - 40 - 46 \\ & - 47 - 48 - 49 - 51 - 47 \end{aligned}$$

$$\begin{aligned} \text{Jalur 3 : } & 25 - 26 - 27 - 28 - 29 - 30 - 31 - 32 - 33 - 36 - 37 - 39 - 40 - 46 \\ & - 47 - 48 - 49 - 58 - 60 - 70 - 71 - 76 - 77 - 78 - 80 - 81 \end{aligned}$$

$$\begin{aligned} \text{Jalur 4 : } & 25 - 26 - 27 - 28 - 29 - 30 - 31 - 32 - 33 - 36 - 37 - 39 - 40 - 46 \\ & - 47 - 48 - 49 - 62 - 63 - 70 - 71 - 76 - 77 - 78 - 80 - 81 \end{aligned}$$

$$\begin{aligned} \text{Jalur 5: } & 25 - 26 - 27 - 28 - 29 - 30 - 31 - 32 - 33 - 36 - 37 - 39 - 40 - 46 \\ & - 47 - 48 - 49 - 62 - 70 - 71 - 76 - 77 - 78 - 80 - 81 \end{aligned}$$

Jalur 6 : 25 – 26 – 27 – 28 – 29 – 30 – 31 – 32 – 33- 36 – 37 – 39 – 40 – 46
 – 47 – 48 – 49 – 51 – 49

5. Pengujian Basis Tes

a. *Function__construct()*

Tabel 4. 8 Tabel Basis *Test Function__construct()*

b. *Function_index()*

Tabel 4. 9 Tabel Basis Tes *function_index()*

Test Case function index()	
Jalur1	
Test Case function __construct()	
Jalur1	
Test Case	Jika <i>function __construct()</i> dijalankan pada baris 5-11
Target yang diharapkan	Berhasil memanggil <i>class</i> model <i>m_rule</i> , <i>m_gejala</i> , <i>m_konsultasi</i> dan <i>m_penyakitcoba</i>
Hasil pengujian	Benar
Path/ Jalur	5 – 6 – 7 – 8 – 9 – 10 – 11
Test Case	
Jika <i>function index()</i> dijalankan pada baris 13-18	
Target yang diharapkan	Berhasil mengambil data dengan memanggil <i>function</i> <i>getall_gejala()</i> pada <i>m_gejala</i> dan <i>getall_kambing</i> pada <i>class</i> models <i>m_rule</i> dalam bentuk array <i>\$data[pola]</i> dan menampilkannya dalam <i>class user_konsultasi</i>
Hasil pengujian	Benar
Path/ Jalur	13 – 14 – 15 – 16 – 17 – 18

c. *Function_inputHitungCF()*

Tabel 4. 10 Tabel Basis Tes *function_inputHitungCF()*

Test Case function_inputHitungCF()	
Jalur 1	
Test Case	Jika <i>function input</i> dijalankan pada baris 24-40
Target yang diharapkan	Berhasil mengambil mengambil data yang diinputkan oleh pengguna selanjutnya disimpan kedalam <i>database</i> dengan memanggil model <i>m_konsultasi method inputkonsultasi</i> . Mencari penyakit yang sesuai dengan gejala yang dipilih oleh <i>user</i> dan diketahui jumlah pepenyakit yang diketahui dan nama penyakitnya.
Hasil pengujian	Benar
Path/ Jalur	25 – 26 – 27 – 28 – 29 – 30 – 31 – 32 – 33- 36 – 37 – 39 – 40 – 41 – 42 – 45 – 40
Jalur 2	
Test Case	<i>function inputHitungCF(25-57)</i>
Target yang diharapkan	Pada fungsi ini melakukan pencocokan gejala dan penyakit yang sudah diketahui pada proses sebelumnya sehingga diketahui bobot nilai <i>certainty factor</i> . Perulangan pada proses dilakukan sebanyak gejala yang dipilih dan penyakit yang diketahui.
Hasil pengujian	Benar
Path/ Jalur	25 – 26 – 27 – 28 – 29 – 30 – 31 – 32 – 33- 36 – 37 – 39 – 40 – 46 – 47 – 48 – 49 – 51 – 57
Jalur 3	
Test Case	jika <i>function inputHitungCF(25-81)</i>
Target yang diharapkan	Berhasil mengambil data dengan memanggil <i>function</i>

	ambil konsultasi penyakit() pada model m_konsultasi. Melakukan perulangan sebanyak gejala yang dipilih berdasarkan pada penyakit sesuai dengan database gabungan. Mengambil nilai cf setiap gejala yang sesuai dengan penyakit. jika kedua nilai cf bernilai negatif lanjut pada proses 60.
Hasil pengujian	Benar
Path/ Jalur	25 – 26 – 27 – 28 – 29 – 30 – 31 – 32 – 33- 36 – 37 – 39 – 40 – 46 – 47 – 48 – 49 – 58 – 60 – 70 – 71 – 76 – 77 – 78 – 80 – 81

Jalur4

Test Case	Jika <i>function</i> inputHitung() dijalankan pada baris 25-81
Target yang diharapkan	Nilai CF yang diambil pada tahap sebelumnya, nilai cf yang sudah dikelompokkan berdasarkan penyakit di lakukan pengecekan nilai tersebut bernilai negatif atau positif. Jika kedua nilai CF memiliki nilai positif maka berlanjut pada baris 62.
Hasil pengujian	Benar
Path/ Jalur	25 – 26 – 27 – 28 – 29 – 30 – 31 – 32 – 33- 36 – 37 – 39 – 40 – 46 – 47 – 48 – 49 – 62 – 63 – 70 – 71 – 76 – 77 – 78 – 80 – 81

Jalur5

Test Case	Jika <i>function</i> InputHitung() dijalankan pada baris 25-81
Target yang diharapkan	Nilai CF yang diambil pada tahap sebelumnya, nilai cf yang sudah dikelompokkan berdasarkan penyakit di lakukan

	pengecekan nilai tersebut bernilai negatif atau positif. Jika kedua nilai CF memiliki nilai yang berbeda lanjut pada proses 66.
Hasil pengujian	Benar
Path/ Jalur	25 – 26 – 27 – 28 – 29 – 30 – 31 – 32 – 33- 36 – 37 – 39 – 40 – 46 – 47 – 48 – 49 – 66 – 70 – 71 – 76 – 77 – 78 – 80 – 81
Jalur 6	
Test Case	Jika <i>function</i> inputHitung() dijalankan pada baris 25-81
Target yang diharapkan	Melakukan perulangan dan mengambil nilai cf berdasarkan penyakit.
Hasil pengujian	Benar
Path/ Jalur	25 – 26 – 27 – 28 – 29 – 30 – 31 – 32 – 33- 36 – 37 – 39 – 40 – 46 – 47 – 48 – 49 – 62 – 70 – 71 – 76 – 77 – 78 – 80 – 81

4.4.2 Pengujian *Black Box*

Pengujian *black box* merupakan pengujian yang dilakukan oleh *user* dengan menjalankan program secara langsung dan menganalisis *input* dan *output* yang dihasilkan sistem. Pengujian *black box* dapat dilihat pada lampiran C.2.

BAB 6. PENUTUP

Bab ini berisi mengenai kesimpulan dan saran dari peneliti tentang penelitian yang telah dilakukan. Kesimpulan dan saran tersebut diharapkan dapat digunakan sebagai acuan pada penelitian selanjutnya.

6.1 Kesimpulan

Kesimpulan dari penelitian yang dilakukan adalah:

1. Sistem pakar ini terbagi atas 4 pengguna yaitu admin yang merupakan petugas pelayanan, pakar, *user* anggota dan non anggota. Pakar merupakan pengguna yang memiliki hak akses tertinggi. Kegiatan pengelolaan data penyakit, gejala dan aturan yang merupakan pemberian nilai pembobotan gejala serta mengelola hak akses *user* anggota dilakukan oleh pakar. Sistem pakar ini menggunakan model *waterfall* dalam pengembangan dan perancangannya. Tahap pengembangan sistem meliputi analisa data dan persyaratan, desain sistem, penulisan kode program, pengujian dan pemeliharaan. Analisa data dan persyaratan merupakan tahapan awal yang digunakan untuk menjabarkan kebutuhan sistem yang datanya didapat dari proses studi literatur dan hasil wawancara. desain sistem proses implementasi kedalam bentuk bagan alur dan gambaran sistem yang akan digunakan sebagai acuan implementasi kode program. Penulisan kode program menggunakan bahasa pemrograman *PHP*, *framework Codeigniter*, dan *database* yang digunakan *MySQL*. Tahap pengujian dilakukan menggunakan metode *white box* dan *black box*. Tahap pemeliharaan bertujuan untuk mengetahui fungsionalitas sistem dan kesesuaian sistem serta .

2. Hasil konsultasi dengan menerapkan metode *certainty factor* pada sistem pakar diagnosa penyakit kambing dan domba mampu menampilkan hasil prenetase penyakit berdasarkan pada gejala yang dipilih. Semakin banyak gejala yang dipilih pada sistem akan menghasil lebih banyak penyakit. Hasil konsultasi yang dilakukan oleh *user* member akan tersimpan pada *database* sedangkan *user* umum tidak akan disimpan. Nilai akhir *certainty factor* didapatkan dari menghitung kombinasi nilai *cf* setiap gejala yang dipilih dikelompokan berdasarkan penyakit. Nilai *CF* didapatkan dari pengurangan bobot *mb* dan *md* yang dimasukan oleh pakar. Hasil pada halaman diagnosa adalah menampilkan informasi penyakit yang tertinggi dan menampilkan tabel persentase penyakit lain. Persentase hasil akhir tersebut bergantung pada bobot yang diberikan oleh seorang pakar dan pemilihan gejala dari *user*. Hasil dari diagnosa dilakukan pemeriksaan oleh pakar untuk memastikan bahwa hasil luaran sistem telah sesuai dengan hasil dari pakar.

6.2 Saran

Beberapa saran berikut diharapkan dapat memberikan perbaikan dalam penelitian selanjutnya, yaitu:

1. Sistem pakar yang akan dikembangkan selanjutnya dapat menggunakan dan menerapkan metode yang lain atau menggabungkan dengan metode lain.
2. Pengembangan lebih lanjut dapat dilakukan dengan menambahkan data penyakit dan data gejala untuk menunjang kelengkapan sistem dan keakuraatan dari sistem pakar.

Daftar Pustaka

- Abidin, S. d. (2002). *Beternak Kambing Dan domba*. Jakarta.
- Bahri , S., Beriajaya, & Adjid, R. (Tanpa Tahun). *Manajemen Kesehatan Dalam Usaha Ternak*.
- Darmono, & hardiman. (2011). Penyakit Utama Yang Sering Ditemukan Pada Ruminansia Keci(Kambing dan Domba). *Workshop Nasional Disversifikasi Pangan Daging Ruminansia Kecil 2011*, 33.
- Dhani, S. (2014). *Rancang Bangun Sistem Pakar Untuk Mendiagnosa Penyakit Degeneratif* .
- Faizal, E. (Tanpa tahun). *Penerapan Dhemster Shafer Pada Sistem Cerdas Pendeteksi Penyakit Kandungan*.
- Halim S. et al. (2011). Penerapan Metode *Certainty factor* dalam Sistem Pakar Pendeteksi Resiko Osteoporosis dan Osteoarthritis.
- Mira, O. (2014). *Sistem Pakar Diagnosis Kambing Berbasis Web Menggunakan Metode Certainty factor*.
- Rakhman, E. (2012). Penerapan Sistem Pakar Dengan Metode *Certainty factor* pada Aplikasi Dokter Virtual Berbasis Android.
- Rohajawati, S., & Supriyati, R. (2010). *SISTEM PAKAR DIAGNOSIS PENYAKIT UNGGAS DENGAN METODE CERTAINTY FACTOR*. *CommIT*.
- Ruhyat, K. (2001). Dr. In R. Kartasudjana , *Teknik Kesehatan Ternak* (p. 16). Bandung: DEPARTEMEN PENDIDIKAN NASIONAL .
- Subangkit, M. (2011). *Teknik Pembibitan Kambing dan Domba*. Jakarta: Penebar Swadaya.
- Triakoso, N. (2011). *Aspek Klinik dan Penularan Penyakit Pada Ternak*, 2.

LAMPIRAN

Lampiran A. Perancangan Sistem

Lampiran dari hasil Perancangan sistem paka diagnose penyakit kambing dan domba dengan menggunakan metode *certainty factor*. Uraian mengenai perancangan akan diuraikan pada perancangan berikut:

A. 1 Skenario Sistem

Berisikan hasil dari pembuatan skenario pada sistem pakar diagnose penyakit kambing dan domba dengan menggunakan metode *certainty factor*

1. Skenario *Login*

Tabel A.1. 1. Skenario *Login*

Name	<i>Login</i>
Aktor	Pakar
Entry Condition	pakar membuka web sistem
Exit Condition	Pakar telah melakukan <i>Login</i> dan membuka halaman <i>home</i>
Skenario Utama	
<i>Login</i>	
Aksi Aktor	Reaksi Sistem
1. Buka halaman web sistem	2. Menampilkan halaman sistem pakar penentuan penyakit kambing dengan <i>detail</i> tampilan sebagai berikut: <ol style="list-style-type: none"> a. <i>Footer</i> yang berisi informasi Instansi b. <i>header</i> menu (home, konsultasi,

	informasi dan <i>login</i>) c. <i>content</i> berisi keterangan penjelasan sistem pakar
3. Klik menu <i>Login</i>	
	4. Menampilkan halaman <i>login</i> , yang berisi sebagai berikut: a. menampilkan <i>pop up form Login Password dan Username</i> b. Button <i>login</i> c. <i>Footer</i> yang berisi informasi Instansi d. <i>header</i> menu (<i>home, konsultasi, informasi dan login</i>) e. button <i>login</i>
5. Isi form <i>Login</i> a. Username b. Password	
6. Klik <i>Login</i>	
	7. Melakukan proses pemeriksaan kolom dan <i>autentifikasi</i>
	8. Menampilkan pesan "selamat datang admin" dan menampilkan tombol ok
9. Klik ok	
	10. Menampilkan halaman home serta <i>header</i> menu <i>Gejala, Penyakit, Rule, Record, User, home dan log out</i> . Menampilkan tabel <i>record</i> terakhir

	diambil dari data konsultasi, <i>user</i> , hasil_konsultasi berisi: a. Nama <i>user</i> b. Tanggal konsultasi c. Penyakit terbesar yang kemungkinan di derita
Scenario Alternatif “password atau <i>username</i> salah”	
Aksi Aktor	Aksi Sistem
	8.a. Menampilkan pesan <i>Login</i> “Kombinasi <i>Username</i> dan Password salah”.
9.a. Klik <i>button</i> ok	
	10.a. Menampilkan halaman home
Skenario Alternatif “kolom form <i>Login</i> kosong”	
Aksi Aktor	Aksi Sistem
	9.b Menampilkan pesan “please fillout the field pada kolom yang kosong

2. Skenario Mengelola Data Gejala

Tabel A.1. 2 Skenario Mengelola Gejala

Name	Mengelola data Gejala (tambah gejala)
Aktor	Pakar
Entry Condition	Pakar telah melakukan <i>login</i> dan membuka halaman gejala
Exit Condition	Berhasil menambahkan data gejala

Skenario Utama	
“Menambah gejala”	
Aksi Aktor	Reaksi Sistem
1. Memilih menu gejala	
	2. Menampilkan halaman menu gejala, dengan <i>content</i> sebagai berikut: <ol style="list-style-type: none"> a. tabel <i>list</i> gejala yang tersimpan (kode gejala, nama gejala) yang diambil dari <i>database</i> <i>tb_gejala</i>. b. <i>action button</i> (<i>edit</i> dan <i>hapus</i>) dan <i>Tambah gejala</i> c. <i>Header</i> menu <i>home</i>. Gejala, <i>penyakit</i>, <i>rule</i>, <i>user</i>, <i>record</i> <i>penyakit</i> dan <i>logout</i> d. <i>Footer</i> berisi informasi mengenai instansi terkait.
3. Klik tombol tambah	
	4. Menampilkan <i>pop up form</i> tambah gejala, yang meliputi: <ol style="list-style-type: none"> a. <i>IDgejala</i> b. <i>Nama gejala</i> c. <i>Button</i> <i>simpan</i>
5. Mengisi fom gejala <ol style="list-style-type: none"> a. <i>ID Gejala</i> b. <i>Nama gejala</i> 	
6. Klik tombol simpan	
	7. Sistem melakukan pemeriksaan data

	yang akan tersimpan
	8. menampilkan pesan “Berhasil Tambah Data” dan <i>button</i> ok
9. Klik tombol ok	
	10. Menampilkan halaman menu gejala, a. tabel <i>list</i> gejala(kode gejaladan nama gejala) yang baru tersimpan , <i>action button</i> (<i>edit</i> dan <i>hapus</i>) dan tambah gejala b. navigasi menu home. Gejala, penyakit, <i>rule</i> , <i>record</i> penyakit dan <i>logout</i>
Scenario Alternatif “Data Inputan kosong”	
Aksi Aktor	Reaksi Sistem
	8. a Menampilkan pesan “please fill out the field” pada kolom <i>input</i> yang kosong
Skenario Mengelola Gejala “Mengubah Gejala”	
Name	Mengelola data Gejala(Mengubah)
Aktor	Pakar
Entry Condition	Pakar telah melakukan <i>login</i> dan membuka halaman edit gejala
Exit Condition	Pakar berhasil mengubah data gejala
Aksi Aktor	Reaksi Sistem
1. Klik tombol <i>edit</i> pada table	

	<p>2. Menampilkan halaman <i>edit</i> tabel <i>form detail</i> gejala</p> <ol style="list-style-type: none"> idgejala nama gejala <i>action button</i> simpan <i>footer</i> <i>header</i> menu menu home. Gejala, penyakit, <i>rule</i>, <i>record</i> , <i>user</i> dan <i>logout</i>
3. Mengubah gejala nama gejala	
4. Klik simpan	
	5. Menampilkan pesan “berhasil <i>update data</i> ” dan tombol ok
5. Klik tombol ok	
	<p>6. Menampilkan halaman gejala yang meliputi:</p> <ol style="list-style-type: none"> tabel <i>list</i> gejala(kode gejala dan nama gejala) yang tersimpan , <i>action button</i> (<i>edit</i> dan <i>hapus</i>) serta <i>button</i> tambah gejala <i>header</i> menu <i>home</i>. Gejala, penyakit, <i>rule</i>, <i>record</i> penyakit dan <i>logout</i> <i>footer</i>
Skenario	
“Hapus Gejala”	
Name	Mengelola data Gejala(hapus gejala)

Aktor	Pakar
Entry Condition	Pakar telah melakukan <i>login</i> dan membuka halaman gejala
Exit Condition	Pakar berhasil menghapus gejala
Aksi Sitem	Reaksi Sistem
1. Klik tombol hapus	
	2. Melakukan proses pemeriksaan data belum digunakan
	3. Manampilkan Pesan “berhasil hapus data” dan <i>button</i> ok
4. Klik ok	
	5. Menampilkan halaman menu gejala, tabel <i>list</i> gejala(kode gejala dan nama gejala) yang tersimpan , <i>action button</i> (<i>edit</i> dan hapus) dan navigasi menu home. Gejala, penyakit, <i>rule</i> , <i>record</i> penyakit dan <i>logout</i> serta <i>button</i> tambah gejala
Skenario Alternatif	
“Gagal Hapus, Data telah digunakan”	
	3.a. Manampilkan Pesan Gagal hapus data dan <i>button</i> ok
4.a. Klik ok	
	5.a. Menampilkan halaman menu gejala, tabel <i>list</i> gejala(kode gejala dan nama gejala) yang tersimpan , <i>action button</i> (<i>edit</i> dan hapus) dan navigasi menu home. Gejala, penyakit, <i>rule</i> ,

	<i>record</i> penyakit dan <i>logout</i> .
--	--

3. Mengelola Data *User*

A.1. 3 Skenario Mengelola Data *User*

Name	Mengelola data pengguna
Aktor	Admin Pendaftar
Entry Condition	Admin telah melakukan <i>login</i> dan membuka halaman menu user
Exit Condition	Admin berhasil menambahkan data <i>user</i>
Skenario Utama “Menambah anggota”	
Aksi Aktor	Reaksi Sistem
1. Memilih menu user	2. Menampilkan halaman menu anggota, yang meliputi: <ul style="list-style-type: none"> a. tabel pengguna yang berisi nomor anggota dan nama pengguna. b. <i>Button</i> tambah user c. <i>Action button</i> (<i>edit</i>, <i>detail</i> dan <i>hapus</i>) d. <i>Header</i> menu <i>home</i>. Gejala, penyakit, <i>rule</i>, <i>user</i>, <i>record</i> penyakit dan <i>logout</i>

3. Klik tombol tambah <i>user</i>	
	<p>4. Menampilkan <i>pop up form</i> tambah <i>user</i>, yang meliputi:</p> <ul style="list-style-type: none"> a. No_anggota b. Nama c. Alamat d. Nomor Hp/telp e. No identitas f. <i>Username</i> g. Password h. <i>Button</i> simpan i. <i>Footer</i>
<p>5. Mengisi form tambah <i>user</i>:</p> <ul style="list-style-type: none"> a. No_anggota b. Nama c. Alamat d. Nomor Hp/telp e. No identitas f. <i>Username</i> g. <i>Password</i> 	
6. Klik tombol simpan	
	7. Sistem melakukan pemeriksaan data yang akan tersimpan
	8. Menampilkan pesan “Berhasil Tambah Data” dan <i>button ok</i>
9. Klik tombol ok	
	10. Menampilkan halaman menu

	anggota, tabel pengguna yang berisi nomor anggota dan nama pengguna, <i>action button</i> (<i>edit</i> , <i>detail</i> dan <i>hapus</i>) dan navigasi menu <i>home</i> . Gejala, penyakit, <i>rule</i> , <i>record</i> , <i>user</i> dan <i>logout</i>
Scenario Alternatif “Data Input Tidak Lengkap”	
Aksi Aktor	Reaksi Sistem
	8.a. Menampilkan pesan “Gagal Tambah Data”
9.a. Klik <i>button</i> ok	
	10.a. Menampilkan halaman menu anggota, tabel pengguna yang berisi nomor anggota dan nama pengguna, <i>action button</i> (<i>edit</i> , <i>detail</i> dan <i>hapus</i>) dan navigasi menu <i>home</i> . Gejala, penyakit, <i>rule</i> , <i>record</i> dan <i>logout</i>
Name	Mengelola data pengguna
Aktor	Admin
Entry Condition	Pakar telah melakukan <i>login</i> dan membuka halaman edit pengguna
Exit Condition	Admin berhasil mengubah data pengguna
Skenario Utama “Mengubah Data Anggota”	

Aksi Aktor	Reaksi Sistem
1. Klik tombol <i>edit</i> pada table	
	2. Menampilkan halaman <i>form</i> tambah <i>user/pengguna</i> , yang meliputi: <ol style="list-style-type: none"> a. No anggota b. Nama c. Alamat d. Nomor Hp/telp e. No identitas f. <i>Username</i> g. password <i>Button</i> tambah, dan navigasi menu home. Gejala, penyakit, <i>rule</i> , <i>record</i> dan <i>logout</i>
3. Mengubah data pengguna: <ol style="list-style-type: none"> a. No_anggota b. Nama c. Alamat d. Nomor Hp/telp e. No identitas f. <i>Username</i> g. password <i>Button</i> tambah, dan navigasi menu home. Gejala, penyakit, <i>rule</i> , <i>record</i> dan <i>logout</i>	
4. Klik simpan	
	5. Menampilkan pesan berhasil simpan dan tombol ok

6. Klik tombol ok	
	7. Menampilkan halaman menu anggota, tabel pengguna yang berisi nomor anggota dan nama pengguna, <i>action button</i> (<i>edit</i> , <i>detail</i> dan <i>hapus</i>) dan navigasi menu home. Gejala, penyakit, <i>rule</i> , <i>record</i> dan <i>logout</i>
Name	Mengelola data pengguna(hapus pengguna)
Aktor	Admin
Entry Condition	Pakar telah melakukan <i>login</i>
Exit Condition	Pakar berhasil menghapus data pengguna
Skenario Utama	
“Hapus pengguna”	
1. Klik tombol hapus	
	2. Melakukan proses pemeriksaan data belum digunakan
	3. Menampilkan Pesan “berhasil hapus data” dan <i>button</i> ok
4. Klik ok	
	5. Menampilkan halaman menu anggota, tabel pengguna yang berisi nomor anggota dan nama pengguna, <i>action button</i> (<i>edit</i> , <i>detail</i> dan <i>hapus</i>) dan navigasi menu home. Gejala,

	penyakit, <i>rule</i> , <i>record</i> dan <i>logout</i>
Skenario Alternatif	
“gagal hapus, data digunakan”	
	3.a.Menampilkan Pesan “gagal hapus data” dan <i>button</i> ok
4.a.Klik ok	
	5.a.Menampilkan halaman menu anggota, tabel pengguna yang berisi nomor anggota dan nama pengguna, <i>action button</i> (<i>edit</i> , <i>detail</i> dan <i>hapus</i>) dan navigasi menu home. Gejala, penyakit, <i>rule</i> , <i>record</i> dan <i>logout</i>

4. Menampilkan *Record*

Tabel A.1. 4 Skenario Menampilkan *Record*

Name	Melihat <i>Record</i> Kesehatan Ternak
Aktor	Pakar
Entry Condition	Pakar akan melihat <i>record</i> kesehatan ternak dan melihat <i>detail record</i> ternak
Exit Condition	Pakar telah melihat <i>record</i> kesehatan ternak dan melihat <i>detail record</i> ternak
Skenario Utama	
“melihat detail record kesehatna ternak”	
Aksi Aktor	Reaksi Sistem
1. Klik menu <i>record</i>	

	<p>2. Menampilkan halaman <i>record</i> kesehatan ternak dan berisikan table hasil <i>record</i> yang telah disimpan, menampilkan tabel berisi tanggal, no anggota ,nama, id ternak, penyakit, konsultasi dan <i>action button detail (detail)</i> serta navigasi menu gejala, penyakit <i>rule, record</i> penyakit, <i>user</i> dan <i>logout</i></p>
<p>3. Klik button detail</p>	
	<p>4. Menampilkan informasi <i>detail</i> data penyakit yang memiliki persentase terbesar, menampilkan gejala yang dipilih, menampilkan hasil persentase kemungkinan penyakit yang diderita meliputi nama penyakit dan besarnya niali cf serta tanggal konsultasi</p> <ol style="list-style-type: none"> a. Menampilkan <i>detail</i> penyakit no, nama, penyebab, solusi, obat dan pencegahan. b. Gejala pilih c. Tanggal, nomor ternak, penyakit dan persentase d. serta navigasi menu gejala, penyakit <i>rule, record</i> penyakit, <i>user</i> dan <i>logout</i>

5. Menampilkan Informasi penyakit

Tabel A.1. 5 Menampilkan Informasi penyakit

Name	Melihat Informasi <i>detail</i> penyakit
Aktor	User umum
Entry Condition	Pengguna akan melihat <i>detail</i> dari penyakit
Exit Condition	Pengguna telah melihat <i>detail</i> penyakit
Skenario Utama	
Aksi Aktor	Reaksi Sistem
1. Klik menu Informasi Penyakit	2. Menampilkan halaman menu informasi penyakit, meliputi: <ul style="list-style-type: none"> a. <i>list</i> penyakit nomor dan nama penyakit serta <i>action detail</i> b. menampilkan <i>header</i> menu home, konsultasi, informasi penyakit dan <i>login</i> c. <i>Footer</i>
3. Klik detail	4. Menampilkanta halaman dan informasi <i>detail</i> penakit meliputi: <ul style="list-style-type: none"> a. Nama penyakit b. Jenis penyakit c. Kategori penyakit d. Penanganan e. Pengobatan f. Obat g. Pencegahan

	Navigasi home, konsultasi, informasi penyakit dan <i>login</i>
--	--

Tabel A.1. 6 Melihat Detail Informasi

Name	Melihat Informasi <i>detail</i> penyakit
Aktor	User Member
Entry Condition	Pengguna akan melihat <i>detail</i> dari penyakit
Exit Condition	Pengguna telah melihat <i>detail</i> penyakit
Skenario Utama	
Aksi Aktor	Reaksi Sistem
1. Klik menu Informasi Penyakit	
	2. Menampilkan <i>list</i> nomor, nama penyakit dan <i>action detail</i> serta menampilkan <i>list</i> menu home, Konsultasi, <i>Record</i> , informasi dan <i>dropdown</i> pengguna(<i>user</i> profil dan <i>logout</i>)
3. Klik detail	
	4. Menampilkanta halaman dan informasi <i>detail</i> penakit meliputi: <ul style="list-style-type: none"> a. Nama penyakit b. Jenis penyakit c. Kategori penyakit d. Penanganan e. Pengobatan f. Obat g. Pencegahan

	<i>Header</i> home, Konsultasi, <i>Record</i> , informasi dan <i>dropdown</i> pengguna(<i>user</i> profil dan <i>logout</i>)
--	--

6. Skenario *Log out*A.1. 7 Skenario *logout*

Name	Melakukan <i>Logout</i>
Aktor	<i>User Member</i>
Entry Condition	<i>User member telah melakukan login</i>
Exit Condition	Pengguna telah melakukan <i>log out</i> pada sistem
Skenario Utama “<i>Logout</i>”	
Aksi Aktor	Reaksi Sistem
1. Klik log out	2. Menampilkan halaman utama sistem dengna <i>header</i> (home, konsultasi dan <i>login</i>)

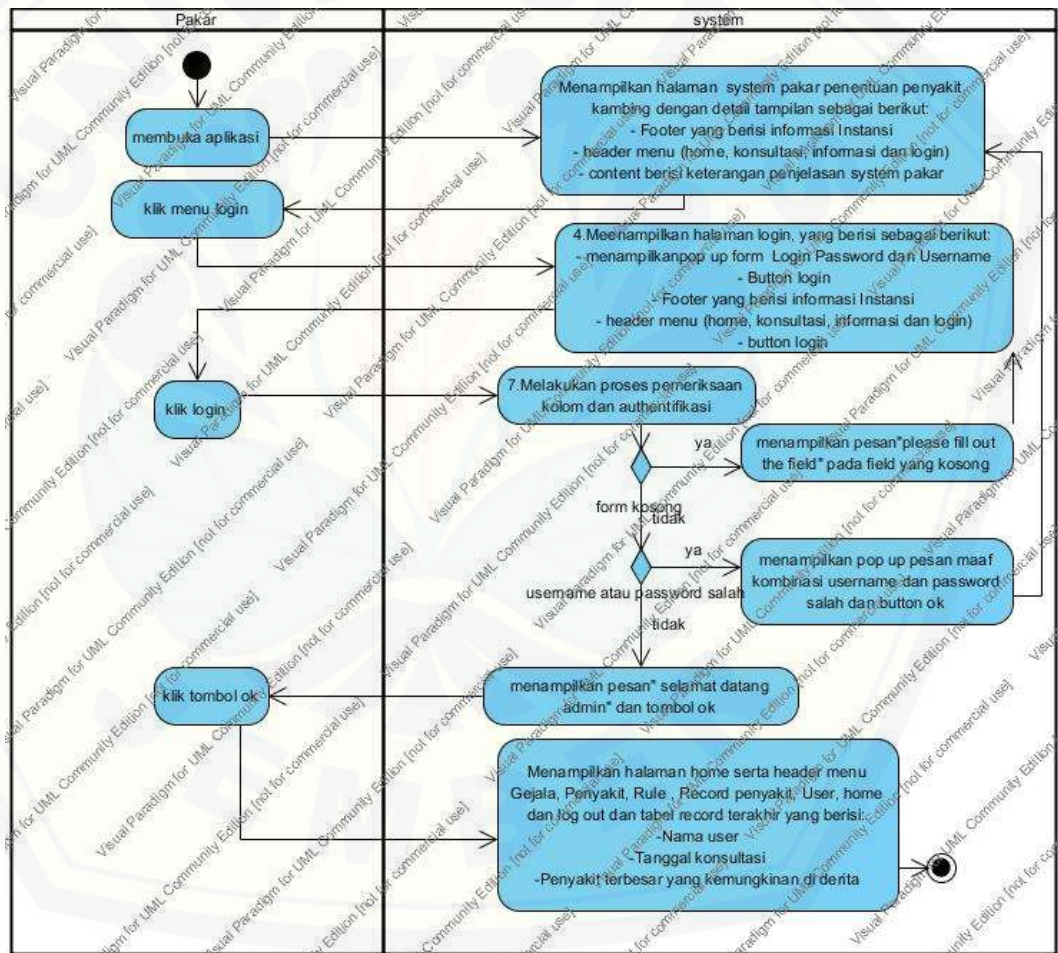
Tabel A.1. 8 Skenario Logout pakar

Name	Melakukan <i>Logout</i>
Aktor	Pakar
Entry Condition	<i>User member telah melakukan login</i>
Exit Condition	Pengguna telah melakukan <i>log out</i> pada sistem
Skenario Utama	

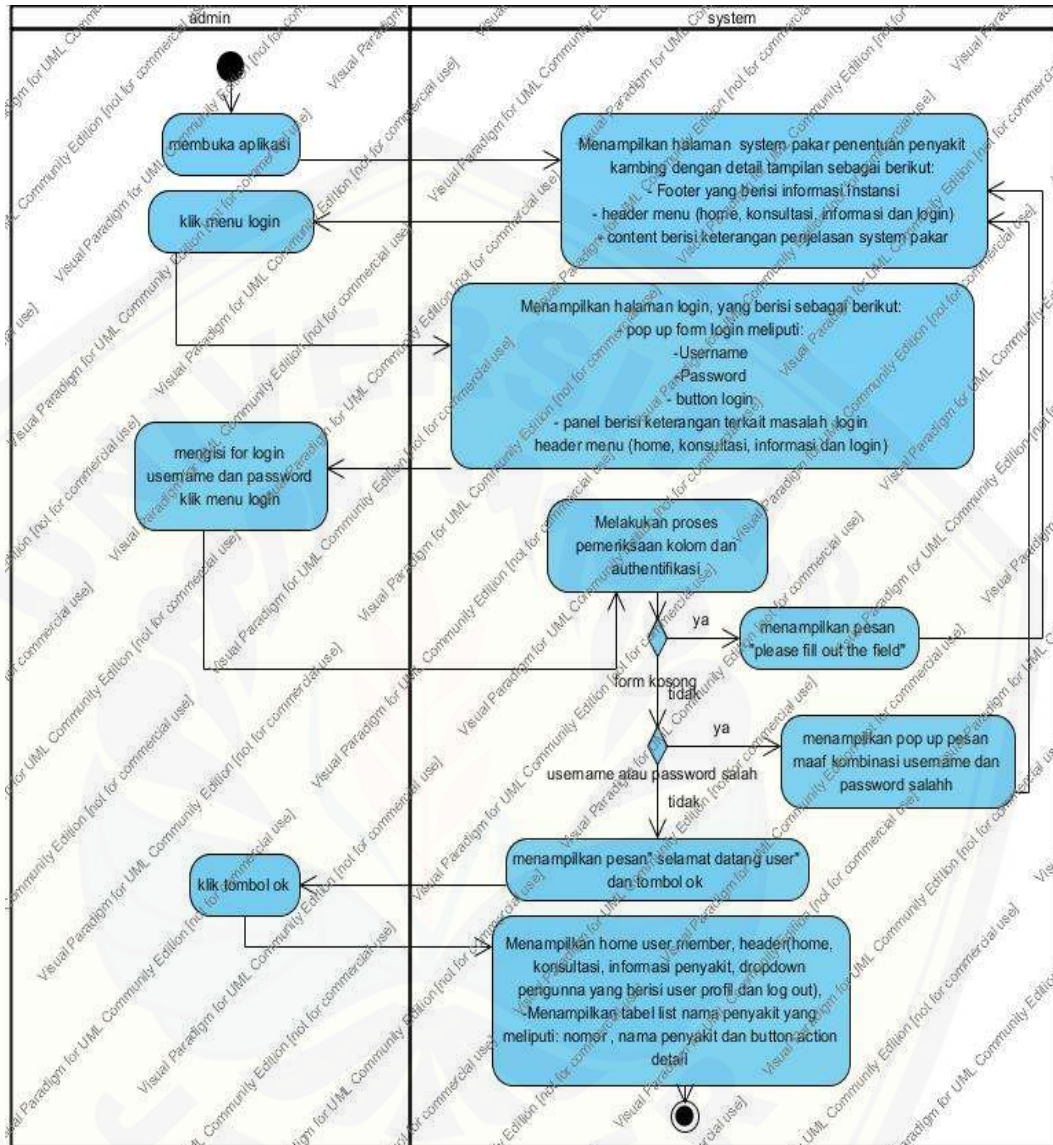
Aksi Aktor	Reaksi Sistem
1. Klik log out	
	2. Menampilkan halaman utama sistem dengan <i>header</i> (home, konsultasi dan <i>login</i>)

A. 2 Activity Sistem

1. Login

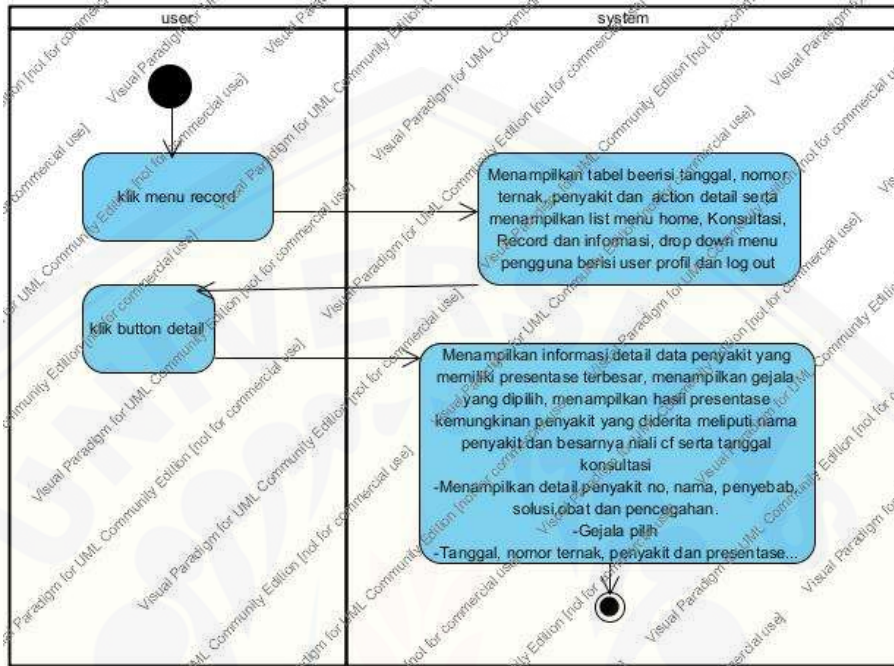


A.2. 1 Gambar Activity Login



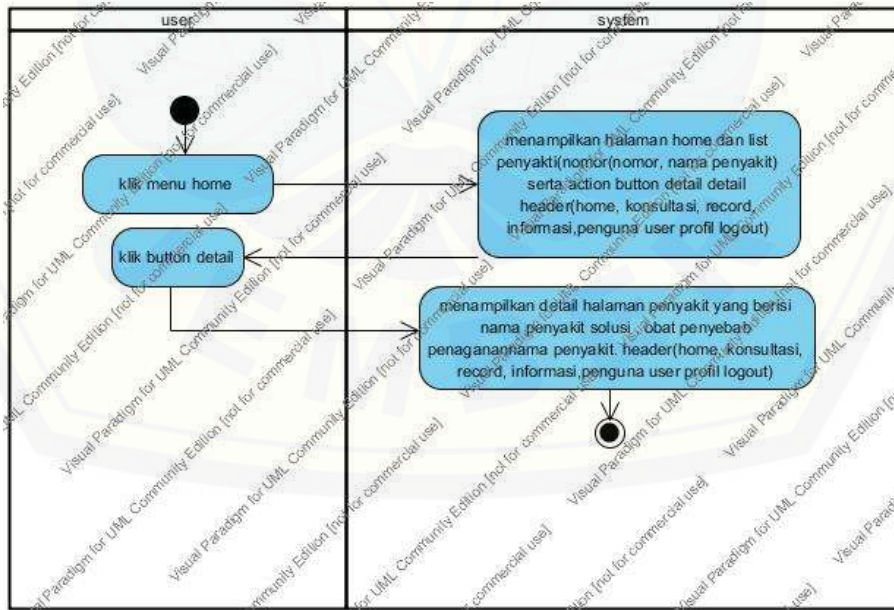
A.2. 2 Gambar Activity Login User Anggota

2. Melihat Record



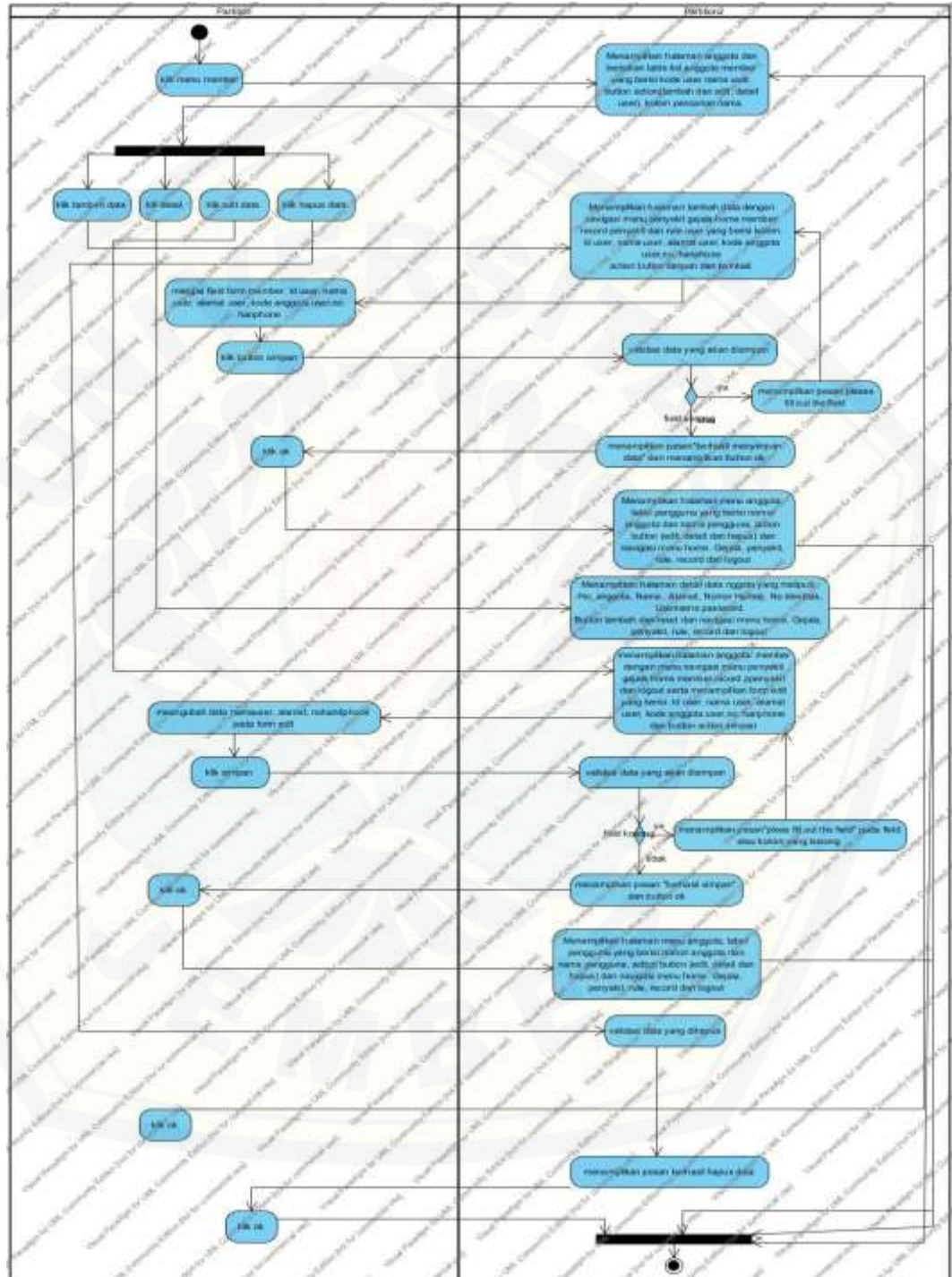
A.2. 3 Activity Diagram Record Konsultasi

3. Melihat Informasi Penyakit



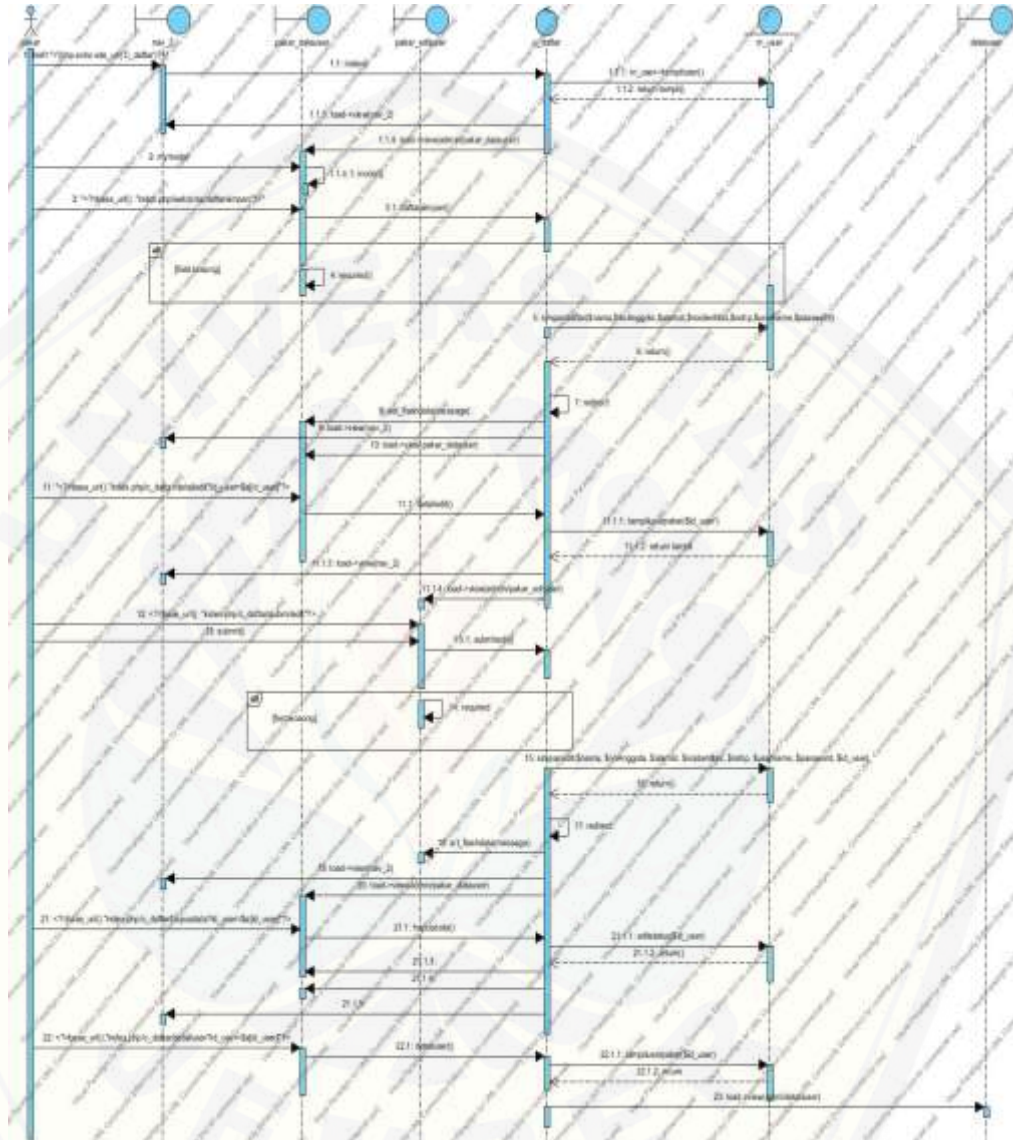
A.2. 4 Activity Diagram Melihat Informasi Penyakit

4. Mengelola User



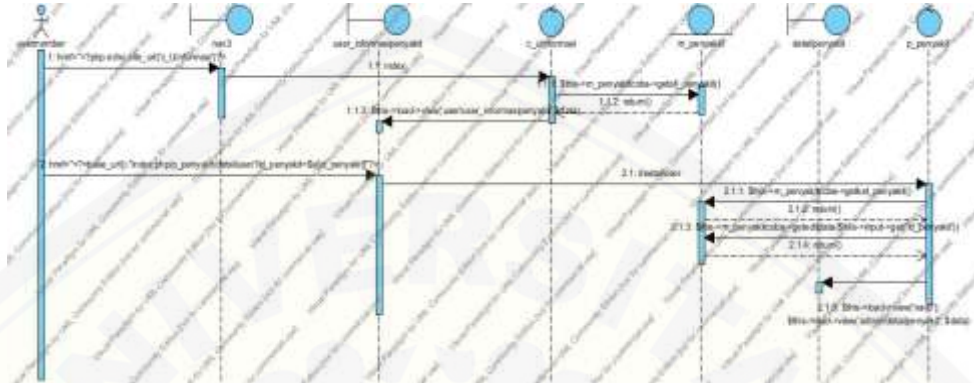
A.2. 5 Activity Diagram Mengelola data User

4. Mengelola *User*



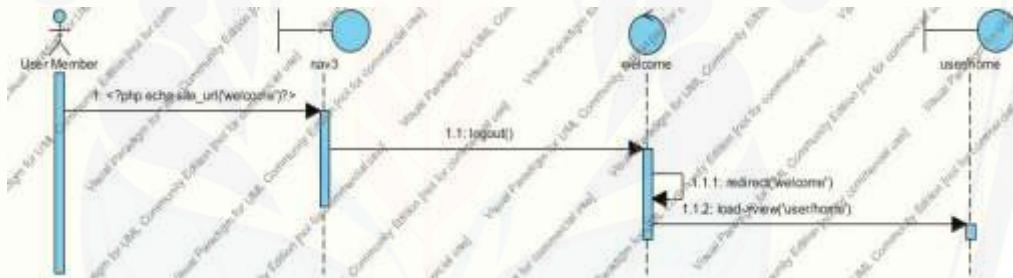
A.3. 5 Sequence Mengelola Data user

5. Melihat Informasi Penyakit



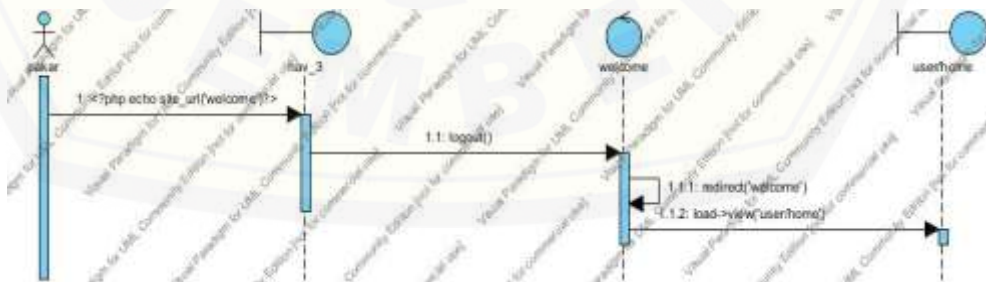
A.3. 6 Sequence Informasi Penyakit

6. Logout User Anggota



A.3. 7 Sequence Logout User Anggota

7. Logout Pakar



A.3. 8 Sequence Logout Pakar

Lampiran B. Kode Program

1. Model/m_login

```

<?php if (!defined('BASEPATH'))
exit('No direct script access allowed');
class M_login extends CI_Model
{

    public function get_login ($username, $password){
        $login=$this->db->query("select * from user_member where username='".$username.'"
and password='".$password.'" and status='aktif'");

        if($login->num_rows(>0){
            foreach($login->result_array() as $row){
                $this->session->set_userdata('id_user',$row['id_user']);
                $this->session->set_userdata('nama',$row['nama']);
                $this->session->set_userdata('noAnggota',$row['noAnggota']);
                $this->session->set_userdata('alamat',$row['alamat']);
                $this->session-
>set_userdata('level',$row['level']);
                $this->session->set_userdata('username', $row['username']);
                $this->session->set_userdata('password', $row['password']);
                if($row['level']== '1'){
                    $this->session->set_userdata('1', true);
                }
                else {
                    $this->session->set_userdata('2', true);
                }
            }
            return true;
        }
        else{
            return false;
        }
    }
}

?>

```

Gambar B. 1 Kode Program Login

2. Model/ m_penyakitcoba

```

?php if (!defined('BASEPATH'))
exit('No direct script access allowed');

```

```
class M_penyakitcoba extends CI_Model
{
    function __construct(){
        parent::__construct();
    }
    public function getall_penyakit(){
        $query= $this->db->query("select * from tb_penyakit");
        return $query;
    }
    public function getkat_penyakit(){
        $kat_penyakit= $this->db->query("select * from kategoripenyakit");
        return $kat_penyakit ;
    }
    public function getkat_penyebab(){
        $kat_penyebab= $this->db->query("select * from kategoripenyebab");
        return ;
    }
    public function hapuspenyakit($id_penyakit){
        $hapus=$this->db->query("delete from tb_penyakit where
id_penyakit='".$id_penyakit.'");
        if($hapus){
            return true;
        }else{
            return false;
        }
    }
    public function p_ambil($id_penyakit){
        $query=$this->db->query("select * from tb_penyakit where
id_penyakit='".$id_penyakit.'");
        return $query;
    }
    public function geteditdata($id_penyakit){
        $ambil=$this->db->query("select * from
tb_penyakit p join kategoripenyakit k on
p.id_jnspenyakit=k.id_jnspenyakit where id_penyakit='".$id_penyakit.'" ");
        return $ambil;
    }
}
```

```

    }
    public function editsubmit($id_penyakit, $nama_penyakit, $id_jnspenyakit,
$Penyebab, $solusi, $obat, $pencegahan){
        $edit=$this->db->query("update tb_penyakit set nama_penyakit=
"". $nama_penyakit.", id_jnspenyakit="."$id_jnspenyakit.",
        Penyebab="."$Penyebab.", solusi="."$solusi.", obat="."$obat.",
pencegahan="."$pencegahan." where id_penyakit="."$id_penyakit." ");
        return $edit;
    }
    public function insertpenyakit($nama_penyakit, $id_jnspenyakit, $Penyebab, $solusi,
$obat, $pencegahan){
        $inputnomor= $this->db->query("SELECT MAX(RIGHT(id_penyakit,2))
as idge from tb_penyakit");
        $Kgejala="";
        if($inputnomor->num_rows() > 0){
            foreach ($inputnomor->result() as $x) {
                $angka= ((int)$x->idge) + 1;
                $Kgejala = sprintf("%01s", $angka);
            }
        }
        else
        {
            $Kgejala = "01";
        }
        $a="P0".$Kgejala;
        $query= $this->db->query(" INSERT INTO `tb_penyakit`
(`id_penyakit`, `nama_penyakit`, `id_jnspenyakit`, `Penyebab`,
`solusi`, `obat`, `pencegahan`)
        values
        (". $a.", "."$nama_penyakit.", "."$id_jnspenyakit.",
"."$Penyebab.", "."$solusi.", "."$obat.", "."$pencegahan."");
        return $query;
    }
}

```

Gambar B. 2 Kode Program Model M_penyakit

3. Model/m_gejala

```
<?php if (!defined('BASEPATH'))
    exit('No direct script access allowed');

class M_gejala extends CI_Model
{
    function __construct() {
        parent::__construct();
    }

    public function getall_gejala(){
        $query= $this->db->query("select * from tb_gejala");
        return $query;
    }

    public function getall_gejala2(){
        $query= $this->db->query("select g.nama_gejala, r.id_gejala from tb_gejala g join gabungan r
        on g.id_gejala=r.id_gejala");
        return $query;
    }

    function tambahGejala($nama_gejala ){
        $inputnomor= $this->db->query("SELECT MAX(RIGHT(id_gejala,2)) as idge from
        tb_gejala");
        $Kgejala="";
        if($inputnomor->num_rows() > 0){
            foreach ($inputnomor->result() as $x) {
                $angka= ((int)$x->idge) + 1;
                $Kgejala = sprintf("%01s", $angka);
            }
        }

        else
        {
```

```
$Kgejala = "01";  
  
}  
  
$a="g".$Kgejala;  
  
$insert= $this->db->query(" INSERT INTO `tb_gejala`(`id_gejala`,`nama_gejala`)  
values (".$a.", " ".$nama_gejala."");  
  
return true;  
  
}  
  
function deletegejala($id_gejala){  
$delete=$this->db->query("delete from `tb_gejala` where `id_gejala`= " ".$id_gejala."");  
if($delete){  
        return true;  
    }else{  
        return false;  
    }  
}  
  
function geteditgejala($id_gejala){  
    $sambbiledit=$this->db->query("select * from tb_gejala where id_gejala  
=" ".$id_gejala."");  
return $sambbiledit;  
}  
  
function editgejala($id_gejala, $nama_gejala){  
    $edit=$this->db->query("update `tb_gejala` set `nama_gejala`= " ".$nama_gejala."  
where `tb_gejala`.`id_gejala` = " ".$id_gejala."");  
return $edit;  
}  
  
}  
  
?>
```

Gambar B. 3 Kode Program Model M_Gejala

4. Model/m_rule

```
<?php if (!defined('BASEPATH'))
exit('No direct script access allowed');

class M_rule extends CI_Model
{
    public function tambah($id_penakit,$id_gejala, $mb, $md ){
        $hasilcf= $mb-$md;
        $query= $this->db->query("INSERT INTO `gabungan` (`id_penakit`,`id_gejala`,`mb`,`md`,`cf`)
VALUES('".$id_penakit."','".$id_gejala."','".$mb."','".$md."','".$hasilcf."')");
        return true;
    }

    public function dnamapenyakit(){
        $nama= $this->db->query("select * from tb_penakit");
        return $nama;
    }

    public function dnamagejala(){
        $nama= $this->db->query("select * from tb_gejala");
        return $nama;
    }

    public function deleterule($id_rule){
        $delete=$this->db->query("DELETE from gabungan where `id_rule`='".$id_rule.'"");
        if($delete){
            return true;
        }else{
            return false;
        }
    }
}
```

```
public function geteditrule($id_rule){
    $ambilrule=$this->db->query("SELECT id_rule, id_penyakit, id_gejala,
mb, md, cf from `gabungan` where `id_rule`='.$id_rule'");
    return $ambilrule;
}

public function geteditrule2($id_rule){
    $ambilrule=$this->db->query("SELECT id_rule, p.id_penyakit,
l.id_gejala, mb, md, cf, l.nama_gejala, p.nama_penyakit from gabungan g join tb_gejala l on
g.id_gejala=l.id_gejala join tb_penyakit p on g.id_penyakit=p.id_penyakit where
`id_rule`='.$id_rule'");
    return $ambilrule;
}

public function simpanedit($id_rule, $id_penyakit, $id_gejala, $mb, $md){
    $hasilcf= $mb-$md;
    $edit=$this->db->query("UPDATE `gabungan` set
id_penyakit='.$id_penyakit.', id_gejala='.$id_gejala.', mb='.$mb.', md='.$md.', cf=$hasilcf
where id_rule='.$id_rule.' ");
    return $edit;
}

public function getid($id_rule){
    $stampil=$this->db->query("select id_rule from gabung where
id_rule=$id_rule");
    return $stampil;
}

public function getall_kambing(){
    $stampil=$this->db->query("select * from jeniskambing");
    return $stampil;
}

public function get_record(){
    $record= $this->db->query("select * from konsultasi k join diagnosa d ");
}
```



```

        public function getrule(){
            $rule=$this->db->query("SELECT * FROM tb_penyakit g JOIN gabungan
gb
            ON g.id_penyakit=gb.id_penyakit
            JOIN tb_gejala p
            on gb.id_gejala= p.id_gejala ");
            return $rule;
        }
    }
?>

```

Gambar B. 4 Kode Program Model M_Rule

5. Model/m_konsultasi

```

<?php if (!defined('BASEPATH'))
exit('No direct script access allowed');

class M_konsultasi extends CI_Model{
    //input
    public function ambil_id(){
        $idkonsul=$this->db->query("SELECT max(id_konsultasi) as id_konsultasi from
konsultasi");
        foreach ($idkonsul->result_array() as $k) {
            $n =$k ['id_konsultasi'];
        }
        $nomor= $n + 1;
        return $nomor;
    }
    public function inputkonsultasi($data){
        $iduser = $this->session->userdata('id_user');
        echo $iduser;
        $submit=$this->db->query("INSERT INTO `konsultasi`
(`id_konsultasi`,`id_jnskambing`,`tanggakonsul`,`nomorkambing`,`id_user`)
VALUES('".$data['nomer'].",".$data['idKambing'].",NOW(),".$data['nomorkambing'].",".$iduse

```

```

r."");

for ($i=0; $i< count($data['total']); $i++) {// input kedetail konsultasi
    $submit2=$this->db->query("INSERT INTO `detailkonsul`(`id_konsultasi`,`id_gejala`)
        VALUES ('".$data['nomer']."','".$data['idgejala'][$i]."");}

}

public function inputkonsultasi2($data,$nilaicf,$gejalapilihhey,$gejalahitung){
    for($x=0; $x<$gejalahitung;$x++){
        $inputkonsul= $this->db->query("INSERT INTO `hasil_diagnosa`
(`id_konsultasi`,`hasilcf`,`id_penyakit`)

VALUES('".$data['nomer']."','".$nilaicf[$x]."',".$gejalapilihhey[$x]."");}
    }

//ambildata
public function ambilkonsulpenyakit($idgejala,$idpenyakit){//select data gabungan

    $scaripenyakit = $this->db->query("SELECT id_penyakit, mb, md,
id_gejala,cf FROM gabungan where id_gejala in('".$idgejala."') and id_penyakit = '".$idpenyakit."'");
    return $scaripenyakit;
}

public function tampilhasildiagnoos()//menampilkan hasil diagnosa berdasarkan
penyakit???
    $stampil= $this->db->query("SELECT * FROM hasil_diagnosa where id_
='".$idpenyakit.'" ");
    return $sambilpenyakit;
}

public function ambilkonsulgejala($idpenyakit){//
    $sambilpenyakit = $this->db->query("SELECT * FROM gabungan where
id_penyakit = '".$idpenyakit.'" ");

```

```
        return $ambilpenyakit;
    }

    public function ambilgejalapilih($sidpenyakit, $gejalap){

        $gejalapilih= $this->db->query("SELECT * FROM gabungan where
id_penyakit ='".$sidpenyakit."' and id_gejala ='".$gejalap."' ");

        return $gejalapilih;
    }

    public function tampilgejalarecord(){

        $recordgejala= $this->db->query("SELECT * FROM konsultasi k join
detailkonsul d on k.id_konsultasi=d.id_konsultasi join tb_gejala g on d.id_gejala=g.id_gejala");

        return $recordgejala;
    }

    public function tampilkanpenyakitgejala($idgejala){

        $ambil= $this->db->query("SELECT g.id_penyakit, p.nama_penyakit FROM
gabungan g join tb_penyakit p on g.id_penyakit=p.id_penyakit WHERE g.id_gejala in ('".$idgejala."')
GROUP by id_penyakit order by id_penyakit ");

        return $ambil;
    }

//tampilkan data
    public function gejalaselect($data){

        for ($i=0; $i< count($data['total']); $i++) { // input kedetail konsultasi

            $pilih= $this->db->query("SELECT * from tb_gejala where
id_gejala ='".$data['idgejala'][$i]."'");}

            return $pilih;

        }

        public function datakonsultampil($nomor){

            $ambil= $this->db->query("select * from konsultasi where
```

```

id_konsultasi=".$nomor."");

        return $ambil;
    }

    public function tampilhasilakhirsemua($nomor){
        $stampil= $this->db->query("SELECT h.hasilcf,h.id_penyakit,
p.nama_penyakit from hasil_diagnosa h join tb_penyakit p on h.id_penyakit = p.id_penyakit where
h.id_konsultasi=".$nomor." and hasilcf > 0");

        return $stampil;
    }

    public function nilaiterbesar($nomor){
        // $stampil= $this->db->query("SELECT max(hasilcf) as terbesar,
p.nama_penyakit, p.Penyebab, p.solusi, p.obat, p.pencegahan from hasil_diagnosa h join tb_penyakit p
on h.id_penyakit= p.id_penyakit where h.id_konsultasi=".$nomor."");

        $stampil= $this->db->query("SELECT hasilcf, p.nama_penyakit,
p.Penyebab, p.solusi, p.obat, p.pencegahan, p.id_penyakit from tb_penyakit p join (select hasilcf,
h.id_penyakit from hasil_diagnosa h where h.id_konsultasi=".$nomor."")b on
p.id_penyakit=b.id_penyakit ORDER BY hasilcf DESC LIMIT 1");

        return $stampil;
    }

    public function ambildatakonsul($nomor){
        $ambil= $this->db->query("select * from konsultasi where
id_konsultasi=".$nomor."");

        return $ambil;
    }

//tampilkantarecord

    public function recorddata(){
        $iduser = $this->session->userdata('id_user');

        $ambil= $this->db->query("SELECT *FROM hasil_diagnosa h join
tb_penyakit t on h.id_penyakit=t.id_penyakit join konsultasi k on h.id_konsultasi=k.id_konsultasi

        WHERE hasilcf = ( select max(hasilcf) FROM
`hasil_diagnosa` AS hasil

        WHERE hasil.id_konsultasi = h.id_konsultasi

        ) and k.id_user=".$iduser." group by h.id_konsultasi");
    }

```

```
        return $ambil;
    }

    public function nilaiterbesarrecord($id_konsultasi){
        $stampil= $this->db->query("SELECT hasilcf, p.nama_penyakit,
p.Penyebab, p.solusi, p.obat, p.pencegahan, p.id_penyakit from tb_penyakit p join (select hasilcf,
h.id_penyakit from hasil_diagnosa h where h.id_konsultasi='".$id_konsultasi."')b on
p.id_penyakit=b.id_penyakit ORDER BY hasilcf DESC LIMIT 1");

        return $stampil;
    }

    public function datagejaladirecord($id_konsultasi){
        // $stampil= $this->db->query("SELECT max(hasilcf) as terbesar,
p.nama_penyakit, p.Penyebab, p.solusi, p.obat, p.pencegahan from hasil_diagnosa h join tb_penyakit p
on h.id_penyakit= p.id_penyakit where h.id_konsultasi='".$id_konsultasi."");

        $stampil= $this->db->query("SELECT * from konsultasi k join
detailkonsul d on k.id_konsultasi= d.id_konsultasi join tb_gejala tg on d.id_gejala=tg.id_gejala where
k.id_konsultasi='".$id_konsultasi."");

        return $stampil;
    }

    public function tampilnamarecord($id_konsultasi){
        $stampil=$this->db->query("SELECT * From konsultasi k join
user_member u on u.id_user=k.id_user where id_konsultasi='".$id_konsultasi."");

        return $stampil;
    }

    public function datarecord2($id_konsultasi){

        $stampil= $this->db->query("SELECT * from konsultasi k join
hasil_diagnosa h on k.id_konsultasi= h.id_konsultasi join tb_penyakit t on h.id_penyakit=t.id_penyakit
where k.id_konsultasi='".$id_konsultasi."");

        return $stampil;
    }

    public function recordsemuauser($id_konsultasi){

        $stampil= $this->db->query("SELECT * from user_member u join
konsultasi k on u.id_user=k.id_user join hasil_diagnosa h on k.id_konsultasi= h.id_konsultasi join
tb_penyakit t on h.id_penyakit=t.id_penyakit join detailkonsul d on k.id_konsultasi=d.id_konsultasi
join tb_gejala g on d.id_gejala=g.id_gejala join jeniskambing j on j.id_jnskambing=k.id_jnskambing
where k.id_konsultasi='".$id_konsultasi."");
    }
}
```

```

        return $ampil;
    }

    public function tampilrecordpakar(){
        $ambil= $this->db->query("SELECT * FROM hasil_diagnosa h join
        tb_penyakit t on h.id_penyakit=t.id_penyakit join konsultasi k on h.id_konsultasi=k.id_konsultasi join
        user_member u on k.id_user=u.id_user

        WHERE hasilcf = ( select max(hasilcf) FROM
        `hasil_diagnosa` AS hasil

        WHERE hasil.id_konsultasi = h.id_konsultasi

        ) group by h.id_konsultasi");

        return $ambil;
    }

    public function tampilrecordpakar2(){
        $ambil= $this->db->query("SELECT * FROM hasil_diagnosa h join
        tb_penyakit t on h.id_penyakit=t.id_penyakit join konsultasi k on h.id_konsultasi=k.id_konsultasi join
        user_member u on k.id_user=u.id_user

        WHERE hasilcf = ( select max(hasilcf) FROM

        WHERE hasil.id_konsultasi = h.id_konsultasi

        ) group by h.id_konsultasi order by tanggalkonsul desc
        limit 10");

        return $ambil;
    }
}
}

```

Gambar B. 5 Kode Program Model Class M_Konsultasi

6. Controller/ welcome

```
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
```

```
class Welcome extends CI_Controller {

    public function __construct(){
        parent::__construct();
        $this->load->model('m_login');
        //$this->cek_session();
    }

    function cek_session(){
        //if ($this->session->userdata('level')!=null) {

            if($this->session->userdata('level') == '1'){
                redirect('admin');
            }elseif($this->session->userdata('level') == '2'){
                redirect('user');
            }

            else{
                redirect('welcome');
            }
        }

    public function index()
    {
        $this->load->view('nav_1');
        $this->load->view('user/home_user2');
    }

    public function index2()
    {
```

```
$this->load->view('nav_1');  
$this->load->view('user/home');  
}  
public function ceklogin(){  
    $username= $this->input->post('username');  
    $password= $this->input->post('password');  
    $cek= $this->m_login->get_login($username, $password);  
    if($cek){  
        if($this->session->userdata('level') == '1'){  
            $this->session->set_flashdata("message", "Selamat  
Datang Admin");  
            redirect('admin');}  
        else if($this->session->userdata('level') == '2'){  
            $this->session->set_flashdata("message", "Selamat  
Datang User");  
            redirect('user');  
        }  
    }  
    else{  
        $this->session->set_flashdata("message", "maaf, kombinasi username dan  
password salah");  
        redirect('welcome/index2'); }  
}  
  
public function logout(){  
    session_destroy();  
    redirect('welcome');  
}
```



```
}
```

Gambar B. 6 Kode Program Controller Welcome

7. Controller/ user

```
<?php
session_start();

class User extends CI_controller {

public function __construct(){
    parent :: __construct();
    $this->load->model('m_user');
    $this->load->model('m_penyakitcoba');
    // $this->user = $this->session->userdata('id_user');

}

public function index (){
    $data['penyakit']= $this->m_penyakitcoba->getall_penyakit();
    $this->load->view('nav3');
    $this->load->view('user/home_user', $data);
}

public function tampiluser2(){
    $data['tampil']=$this->m_user->tampiluser2();
    $this->load->view('nav3');
    $this->load->view('user/user_profil', $data);
}

public function simpanuser(){
    $id_user= $this->session->userdata('id_user');
    $result=$this->m_user->simpanedit(
```

```
$this->input->post('nama'),
$this->input->post('noAnggota'),
$this->input->post('alamat'),
$this->input->post('noidentitas'),
$this->input->post('nohp'),
$this->input->post('username'),
$this->input->post('password'),
$Id_user);

if($result){
    $this->session->set_flashdata('message', 'Berhasil Mengubah Data');
    redirect('User/index');
}else{
    $this->session->set_flashdata('message', 'Gagal mengubah Data');
    redirect('User/index');
}
}
}
?>
```

Gambar B. 7 Kode Program Controller/User

8. Controller/admin

```
<?php
session_start();

class Admin extends CI_controller {
public function __construct(){
    parent :: __construct();
    $this->load->model('m_konsultasi');
    //$this->cek_session();
}
}
```

```

public function index (){

    $data['Rterakhir']= $this->m_konsultasi->tampilrecordpakar2();

    $this->load->view('nav_2');

    $this->load->view('admin/pakar_home',$data);

}

}

?>

```

Gambar B. 8 Kode Program Controller/admin

9. Controller/p_penyakit

```

<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');

class p_penyakit extends CI_Controller {
    public function __construct(){
        parent :: __construct();
        $this->load->model('m_penyakitcoba');

    }

    public function index()
    { $this->load->model('m_penyakitcoba');
      $this->load->view('nav_2');
      $data['wew']= $this->m_penyakitcoba->getkat_penyakit();
      $data['wew1']= $this->m_penyakitcoba->getall_penyakit();
      $this->load->view('admin/pakar_penyakitt',$data);

    }

    function edit(){
        $data['wew']=$this->m_penyakitcoba->getkat_penyakit();
        $data["edit"]=$this->m_penyakitcoba->geteditdata($this->input->get('id_penyakit'));
        $this->load->view('nav_2');
        $this->load->view('admin/edit_penyakit', $data);
    }

    public function submitedit(){
        $result=$this->m_penyakitcoba->editsubmit($this->input->post('id_penyakit',
        $this->input->post('nama_penyakit'),
        $this->input->post('id_jnspenyakit'),
        $this->input->post('Penyebab'),
        $this->input->post('solusi'),
        $this->input->post('obat'),
        $this->input->post('pencegahan'));
        if($result){

```

```
        $this->session->set_flashdata('message', 'Berhasil Mengubah Data');
        redirect('p_penyakit');
    }
    else{
        $this->session->set_flashdata('message', 'Gagal Mengubah Data');
        redirect('p_penyakit');
    }
}
function hapus(){
    $result= $this->m_penyakitcoba->hapuspenyakit($this->input->get('id_penyakit'));
    if($result == true){
        $this->session->set_flashdata('message', 'berhasil hapus data');
        redirect('p_penyakit');
    }
    else{
        $this->session->set_flashdata('message', 'gagal hapus data');
        redirect('p_penyakit');
    }
}
public function insertPenyakit(){
    $result= $this->m_penyakitcoba->insertpenyakit(
        $this->input->post('nama_penyakit'),
        $this->input->post('id_jnspenyakit'),
        $this->input->post('Penyebab'),
        $this->input->post('solusi'),
        $this->input->post('obat'),
        $this->input->post('pencegahan'));

    if($result){
        $this->session->set_flashdata('message','Berhasil');
        redirect('p_penyakit');
    }
    else{
        $this->session->set_flashdata('message','Gagal');
        redirect();
    }
}
public function detailuser(){
    $data['wew']=$this->m_penyakitcoba->getkat_penyakit();
    $data["edit"]=$this->m_penyakitcoba->geteditdata($this->input->get('id_penyakit'));
    $this->load->view('nav3');
    $this->load->view('admin/detailpenyakit', $data);
}
public function detailumum(){
    $data['wew']=$this->m_penyakitcoba->getkat_penyakit();
    $data["edit"]=$this->m_penyakitcoba->geteditdata($this->input->get('id_penyakit'));
    $this->load->view('nav_1');
    $this->load->view('admin/detailpenyakit', $data);
}
public function detailadmin(){
    $data['wew']=$this->m_penyakitcoba->getkat_penyakit();
    $data["edit"]=$this->m_penyakitcoba->geteditdata($this->input->get('id_penyakit'));
    $this->load->view('nav_2');
```

```
$this->load->view('admin/detailpenyakit', $data);  
}  
  
}
```

Gambar B. 9 Kode Program Controller/Penyakit

10. Controller/p_gejala

```
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');  
  
class p_gejala extends CI_Controller {  
  
    public function __construct(){  
        parent::__construct();  
        $this->load->model('m_gejala');  
    }  
  
    public function index()  
    {  
        $data['wek'] = $this->m_gejala->getall_gejala();  
        $this->load->view('nav_2');  
        $this->load->view('admin/pakar_gejala', $data);  
    }  
  
    function edit($id_gejala){  
        $data["edit"] = $this->m_gejala->geteditgejala($id_gejala);  
        $this->load->view('nav_2');  
        $this->load->view('admin/edit_gejala', $data);  
    }  
  
    function submittedit(){  
        $result = $this->m_gejala->editgejala($this->input->post('id_gejala'),  
            $this->input->post('nama_gejala'));  
        if($result){  
            $this->session->set_flashdata('message', 'Berhasil Update Data');  
            redirect('p_gejala');  
        }else{  
            $this->session->set_flashdata('message', 'Gagal Update Data');  
            redirect(base_url().'index.php/'.$this->input->get('page'));  
        }  
    }  
  
    function hapus(){  
        $result = $this->m_gejala->deletegejala($this->input->get('id_gejala'));  
  
        if($result){  
            $this->session->set_flashdata('message', 'Berhasil Hapus Data');  
            redirect('p_gejala');  
        }else{  
            $this->session->set_flashdata('message', 'Gagal Hapus Data');  
            redirect('p_gejala');
```

```

    }

    }

    public function submit(){
        $result= $this->m_gejala->tambahGejala($this->input->post('nama_gejala'));

        if ($result){
            $this->session->set_flashdata('message','Gejala Berhasil Ditambah');
            redirect('p_gejala');}

        else{$this->session->set_flashdata('message','gagal input data');

        redirect('base_url/'.$this->input->get('page'));
        }

    }

}

?>

```

Gambar B. 10 Kode Program Controller/p_gejala

11. Controller/p_rule

```

<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');

class p_rule extends CI_Controller {
    public function __construct(){
        parent :: __construct();
        $this->load->model('m_rule');
        $this->load->model('m_gejala');
        $this->load->model('m_penyakitcoba');

    }

    public function index()
    {
        $this->load->view('nav_2');
        $data['detail']=$this->m_rule->getrule();
        $data['wew']=$this->m_penyakitcoba->getall_penyakit();
        $data['wew1']= $this->m_gejala->getall_gejala();
        $this->load->view('admin/pakar_rule', $data);
    }

    function hapus(){
        $result= $this->m_rule->deleterule($this->input->get('id_rule'));
        if($result == true){
            $this->session->set_flashdata('message', 'berhasil hapus data');
            redirect('p_rule');
        }
    }
}

```

```
        else{
            $this->session->set_flashdata('message', 'gagal hapus data');
            redirect('p_rule');
        }
    }

    public function edittampil(){
        $id_rule= $this->input->get('id_rule');
        echo $id_rule;
        $data['detail']=$this->m_rule->geteditrule2($id_rule);
        $this->load->view('admin/edit_rule', $data);
    }

    public function submittedit(){
        $result= $this->m_rule->simpanedit(
            $this->input->post('id_rule'),
            $this->input->post('id_penyakit'),
            $this->input->post('id_gejala'),
            $this->input->post('mb'),
            $this->input->post('md')
        );

        if($result){
            $this->session->set_flashdata('message','Berhasil Menyimpan');
            redirect('p_rule');
        }else{
            $this->session->set_flashdata('message','Gagal MenyimpanS');
            redirect('p_rule');
        }
    }

    function inputnilai(){

        $result= $this->m_rule->tambah(
            $this->input->post('id_penyakit'),
            $this->input->post('id_gejala'),
            $this->input->post('mb'),
            $this->input->post('md'));

        if($result){
            $this->session->set_flashdata('message','Berhasil menyimpan data');
            redirect('P_rule');
        }else{
            $this->session->set_flashdata('message','Gagal menyimpan data');
            redirect('p_rule');
        }
    }
}
```

```

    }
}

```

Gambar B. 11 Kode Program Controller P_rule

12. Controller/c_userrecord

```

<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');

class C_userrecord extends CI_Controller {

public function __construct(){
parent :: __construct();
$this->load->model('m_konsultasi');
$this->load->model('m_penyakitcoba');
$this->load->library('pagination');
}

public function index()
{

$data['we']= $this->m_konsultasi->recorddata();
$this->load->view('nav3');
$this->load->view('user/user_record', $data);
}

public function detail()
{
//$data['we']= $this->m_konsultasi->recorddata();
$data['tampil']= $this->m_konsultasi->datarecord2($this->input-
>get('id_konsultasi'));
//$data['tampil4']= $this->m_konsultasi->getkat_penyakit();
$data['tampil2']= $this->m_konsultasi-> datagejaladirecord($this->input-
>get('id_konsultasi'));
$data['tampil3']= $this->m_konsultasi->nilaiterbesarrecord($this->input-
>get('id_konsultasi'));
$this->load->view('nav3');
$this->load->view('user/detailrecord', $data);
}

public function recorduser(){
$data['we']= $this->m_konsultasi->tampilrecordpakar();
$this->load->view('nav_2');
$this->load->view('admin/paka_userrecord', $data);
}

public function detailrecordpakar(){
$data['tampil']= $this->m_konsultasi->datarecord2($this->input->get('id_konsultasi'));
$data['tampil2']= $this->m_konsultasi-> datagejaladirecord($this->input-
>get('id_konsultasi'));
$data['tampil3']= $this->m_konsultasi->nilaiterbesarrecord($this->input-
>get('id_konsultasi'));
$data['tampil4']= $this->m_konsultasi->tampilnamarecord($this->input-
>get('id_konsultasi'));
}
}

```



```
$this->load->view('nav_2');
$this->load->view('admin/pakar_detailrecord', $data);

}
public function tambahuser(){
    $result=$this->m_login->simpandaftar(
        $this->input->post('nama'),
        $this->input->post('alamat'),
        $this->input->post('username'),
        $this->input->post('password')
    );
    if($result){
        echo $this->session->set_flashdata('message','Berhasil Tambah Data');
        redirect('welcome');
    }
    else{
        echo $this->session->set_flashdata('message','Gagal Tambah Data');
        redirect('C_daftar');
    }
}
public function edituser(){
}
public function updatestatus(){
}
}
?>
```

13. Controller/c_konsultasi

```
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');

class C_konsultasi extends CI_Controller {

    public function __construct(){
        parent :: __construct();
        $this->load->model('m_gejala');
        $this->load->model('m_rule');
        $this->load->model('m_konsultasi');
        $this->load->model('m_penyakitcoba');
    }

    public function index(){
```

```

        $this->load->view('nav3');
        $data['gejala'] = $this->m_gejala->getall_gejala();
        $data['kambing'] = $this->m_rule->getall_kambing();
        $this->load->view('user/user_konsultasi',$data);
    }
    public function halamanhasil(){
        $this->load->view('nav3');
        $this->load->view('user/halamanhasil');
    }

    public function inputHitungCF(){

        $data['nomer'] = $this->m_konsultasi->ambil_id();
        $data['idKambing'] = $this->input->post('id_jnskambing');
        $data['nomorkambing'] = $this->input->post('nomorkambing');
        $data['idgejala'] = $this->input->post('idgejala');
        $idgejala = $data['idgejala'];
        $data['total'] = $data['idgejala'];
        $nomor = $data['nomer'];
        $tambah = $this->m_konsultasi->inputkonsultasi($data);

        $gejalaambil = "";
        $gejalaambil = implode("", $idgejala);
        $gejalapilih = $this->m_konsultasi->tampilkanpenyakitgejala($gejalaambil)-
>result_array();

        $data1['tampilgejala'] = $this->m_konsultasi->gejalaselect($data);
        for ($i=0; $i < count($gejalapilih); $i++) {
            foreach ($gejalapilih as $gejalapilih2) {
                $gejalapilih2['id_penyakit'];
                $namapenyakit1 [] = $gejalapilih2['nama_penyakit'];
            }
        }
        $gejalahitung = count($gejalapilih);
        for($c=0; $c<$gejalahitung; $c++){
            $pilih = $this->m_konsultasi->
ambilkonsulpenyakit($gejalaambil,$gejalapilih2[$c]);
            for($j=0; $j<count($pilih);$j++){
                foreach ($pilih->result_array() as $gabungan) {
                    $cf[] = $gabungan['cf'];
                    $gejalapilih1 = $gabungan['id_gejala'];
                }
            }
            $nilaimb = 0;
            $nilaicf = 0;
            for($m = 0; $m<count($md); $m++){
                if($nilaicf<0 && $cf[$m]<0){
                    $nilaicf = $nilaicf + ($cf[$m] * (1 + $nilaicf));
                }
                elseif($nilaicf>0 && $cf[$m]>0){

```

```

        $nil62aicf = $nilaicf + ($cf[$m] * (1 - $nilaicf));
    }
    else {
        $nilaicf = ($cf[$m] + $nilaicf) / 1 - (min(abs($nilaicf),
abs($cf[$m])));
    }
}

$nilai= $nilaicf*100;
$nilaicf2 []= $nilai;
$data2['cari']=$namapenyakit1;
$data2['coba']=$nilaicf2;
}

$input= $this->m_konsultasi->inputkonsultasi2($data,$nilaicf2,$gejalapilihyy,$gejalahitung);
$data1['datakonsul']= $this->m_konsultasi->tampilhasilakhirsemua($nomor);
$data1['hasil']= $this->m_konsultasi->datakonsultampil($nomor);
$data1['maks']= $this->m_konsultasi->nilaiterbesar($nomor);
$this->load->view('nav3', $data1);
$this->load->view('user/user_hasilakhir', $data1);
}

}
?>

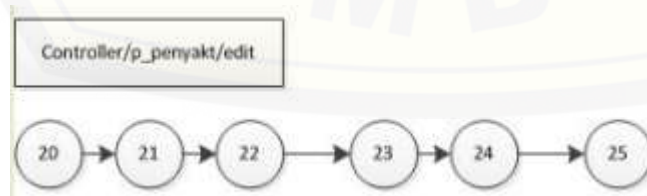
```

Gambar B. 12 Kode Program Controller c_konsultasi

Lampiran C. Pengujian

C. 1. White box

1. Controller/P_penyakit



Gambar C.1. 1 Cyclomatic Complexity P_Penyakit(edit)

Penghitungan *Cyclomatic Complexity* (CC)

edit() : $V(G) = 5 - 6 + 2 = 1$

Basis Set

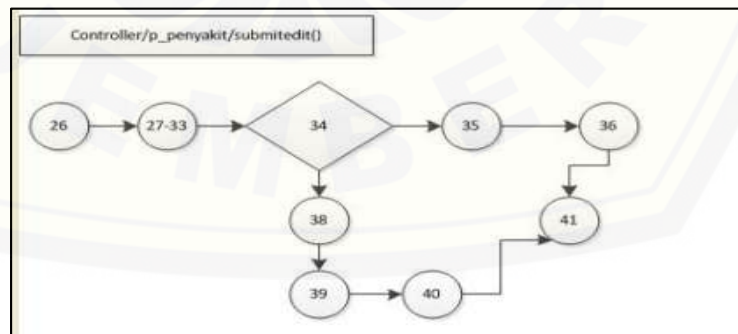
Edit() : 20-21-22-23-24-25 (Alur 1)

Kasus Uji atau *Test Case*

Tabel C.1. 1 Test Case Controller P_penyakit function Edit

Alur 1	
Test Case	Edit
Target yang Diharapkan	Mengambil data kategori penyakit dan data penyakit sesuai dengan id yang dipilih yang terdapat pada model m_penyakit. Ditampilkan pada halaman view/pakar_penyakit
Hasil Pengujian	Benar
Alur / Path	20-21-22-23-24-25

2. Controller p_penyakit function submittededit



Gambar C.1. 2 *Cyclomatic Complexity* Submittededit()

Penghitungan *Cyclomatic Complexity* (CC)

submittedit() : $V(G) = 9 - 9 + 2 = 2$

Basis Set

submittedit() : 26-(27-33)-34-35-36-41 (Alur 1)

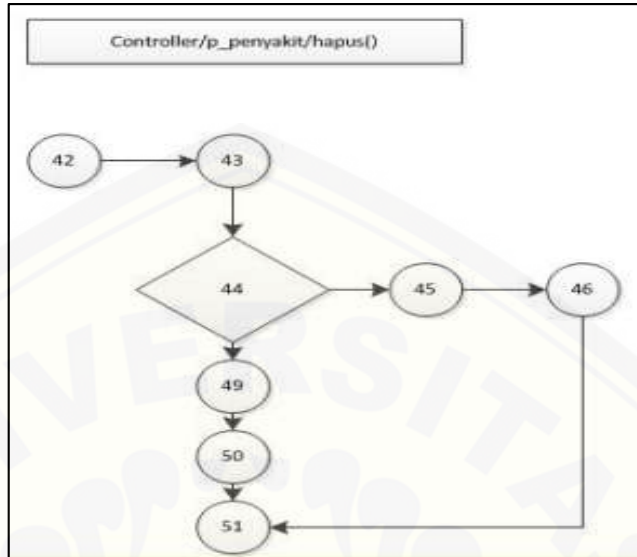
26-(27-33)-34-38-39-40-41(alur 2)

Kasus Uji atau *Test Case*

Tabel C.1. 2 Test Case Controller P_Penyakit Function Submittedit

Alur 1	
Test Case	Submiedit
Target yang Diharapkan	Mengambil <i>inputan</i> pada <i>view</i> yang kemudian disimpan pada <i>database</i> <i>tb_penyakit</i> . Proses <i>input</i> berhasil menampilkan pesan “berhasil mengubah data”
Hasil Pengujian	Benar
Alur / <i>Path</i>	26-(27-33)-34-35-36-41
Alur 2	
Test Case	Submiedit
Target yang Diharapkan	Mengambil <i>inputan</i> pada <i>view</i> yang kemudian disimpan pada <i>database</i> <i>tb_penyakit</i> . Proses <i>input</i> gagal menampilkan pesan “gagal mengubah data”
Hasil Pengujian	Benar
Alur / <i>Path</i>	26-(27-33)-34-38-39-40-41

3. White Box Controller p_penyakit function Hapus



Gambar C.1. 3 Cyclomatic Complexity controller p_penyakit function Hapus

Penghitungan Cyclomatic Complexity (CC)

hapus() : $V(G) = 8 - 8 + 2 = 2$

Basis Set

hapus1() : 42-43-44-45-46-51 (Alur 1)

42-43-44-49-50-51(alur 2)

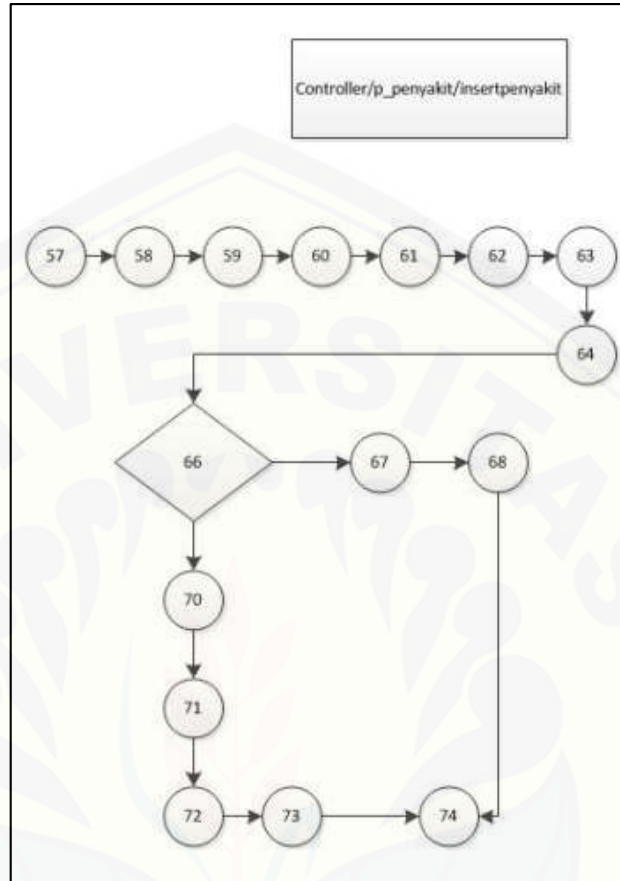
Kasus Uji atau Test Case

Tabel C.1. 3 Test case controller p_penyakit function hapus

Alur 1	
Test Case	Hapus
Target yang Diharapkan	Mengambil id_penyakit dan menghapus data penyakit sesuai id_penyakit pada databse. Menampilkan pesan berhasil hapus data dan menampilkan halaman penyakit
Hasil Pengujian	Benar

Alur / Path	42-43-44-45-46-51 (Alur 1)
Alur 2	
Test Case	Hapus
Target yang Diharapkan	Mengambil id_penyakit dan menghapus data penyakit sesuai id_penyakit pada databse. Menampilkan pesan gagal hapus penyakit data dan menampilkan halaman penyakit
Hasil Pengujian	Benar
Alur / Path	42-43-44-49-50-51 (Alur 2)

4. White box controller penyakit function insertpenyakit



Gambar C.1. 4 Cyclomatic Complexity InsertPenyakit

Penghitungan Cyclomatic Complexity (CC)

insertpenyakit() : $V(G) = 16 - 16 + 2 = 2$

Basis Set

insertpenyakit() : 57-58-59-60-61-62-63-64-66-67-68-74 (Alur 1)

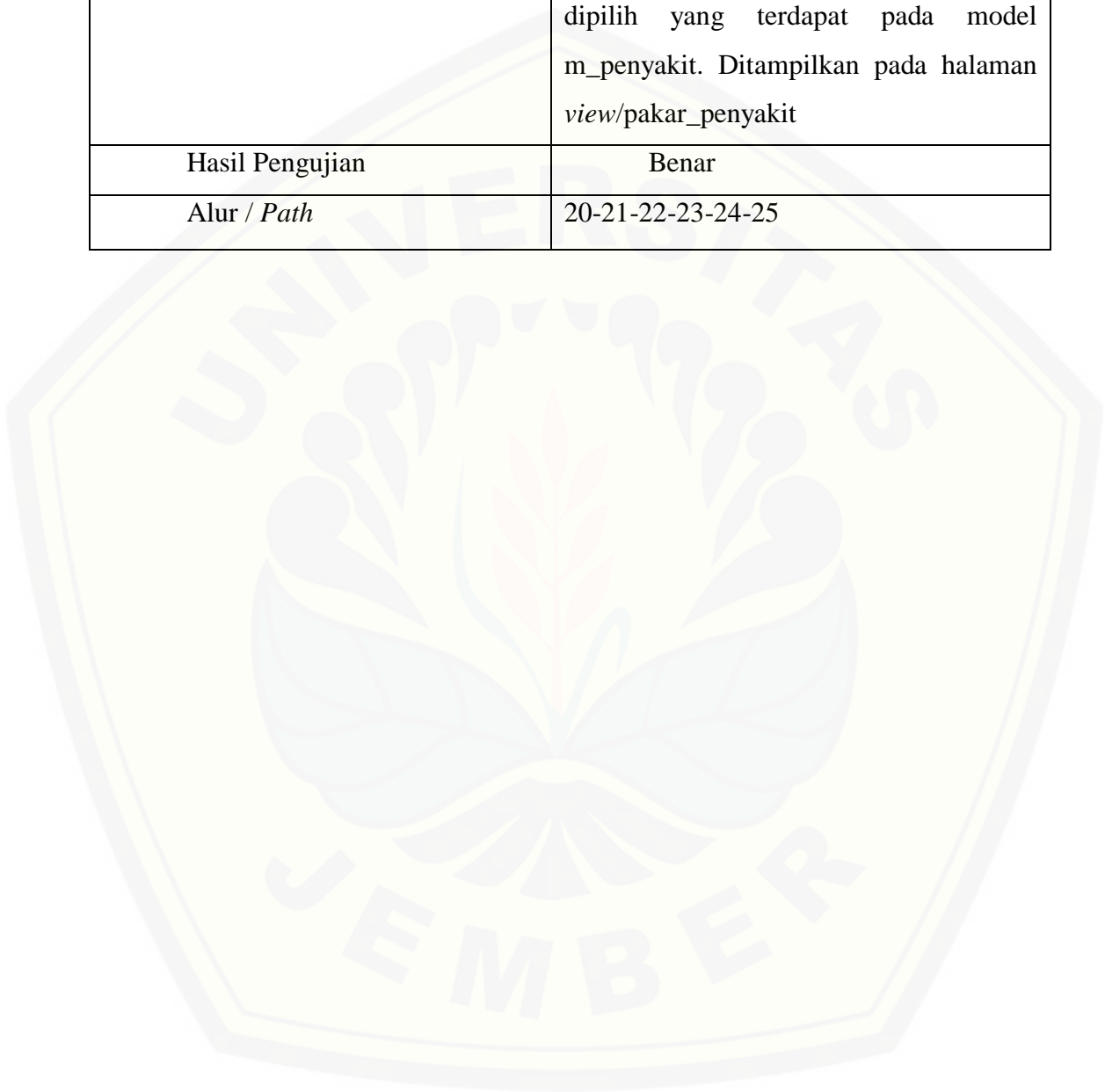
57-58-59-60-61-62-63-64-66-70-71-72-73-74 (alur 2)

Kasus Uji atau *Test Case*

Tabel C.1. 4 Test Case Controller p_penyakit function Insertpenyakit

Alur 1	
Test Case	Insertpenyakit

Target yang Diharapkan	Mengambil data kategori penyakit dan data penyakit sesuai dengan id yang dipilih yang terdapat pada model <code>m_penyakit</code> . Ditampilkan pada halaman <code>view/pakar_penyakit</code>
Hasil Pengujian	Benar
Alur / Path	20-21-22-23-24-25



C. 2. Black Box

Tabel C.2. 1 Tabel Uji *Black Box*

No	Menu	Fungsi	Aksi	Hasil	Ket
1.	<i>Login</i> (Pakar)	Menu <i>login</i> merupakan menu awal yang digunakan untuk autentikasi admin, untuk masuk ke dalam sistem	Mengisi form <i>login</i> berupa <i>username</i> dan <i>password</i> lalu menekan tombol “ <i>Login</i> ”	<i>Login</i> berhasil dan menampilkan halaman home pakar	√
			<i>Username</i> atau <i>Password</i> salah	Menampilkan halaman <i>login</i> admin <i>error</i> , dan pesan “Kombinasi password dan username salah”	√
			Kesalahan pada form tidak diisi atau kosong	Menampilkan pesan “Please fill out the field”	√
2.	Penyakit (Pakar)	Menu Penyakit berfungsi untuk mengelola data admin (tambah, lihat, mengubah dan nonaktif).	Memilih menu penyakit	Menampilkan halaman penyakit dan tabel kode penyakit nama penyakit dan <i>action button</i>	√
			Menekan tombol “tambah penyakit: pada halaman penyakit	Menampilkan form <i>input</i> penyakit meliputi nama penyakit, jenis penyakit, penanganan, solusi, jenis penyakit, obat, pencegahan	√

No	Menu	Fungsi	Aksi	Hasil	Ket
			Menekan tombol “simpan” pada form pop up	Data form <i>input</i> data berhasil disimpan kedalam <i>database</i> tb_penyakit	√
		Mengubah data penyakit	Menekan tombol <i>edit</i> ” pada halaman penyakit	Menampilkan form ubah data admin berupa nama nama penyakit jenis penyakit, penyebab, penaganan, pencegahan dan obat	√
			Menekan tombol “simpan” pada halaman <i>edit</i> penyakit	Berhasil mengubah data penyakit menampilkan pesan”berhasil mengubah data” dan menampilkan halaman penyakit	√
			Menekan tombol hapus	Menampilkan pesan “berhasil hapus data” dan menampilkan halaman penyakit	√
			Gagal hapus data	Gagal hapus data dan menampilkan pesan gagal hapus data.	√

3.	Gejala(pakar)	Menu Gejala berfungsi untuk mengelola data gejala (tambah, lihat, ubah dan nonaktif)	Memilih menu gejala	Menampilkan data gejala berupa no(id_gejala) dan nama gejala	√
			Menekan tombol “tambah gejala” pada halaman gejala	Menampilkan <i>pop up</i> form <i>input</i> gejala berupa nama kategori dan <i>button</i> simpan	√
			Menekan tombol “simpan” pada halaman <i>input</i> gejala	Data form <i>input</i> gejala berhasil masuk ke dalam <i>tb_gejala</i> di <i>database</i> . Menampilkan pesan “berhasil tambah data” dan <i>button</i> ok	√
			Menekan tombol “ok” pada <i>pop up</i> pesan	Menampilkan halaman gejala dengan tabel gejala no dan nama gejala	√
			Menekan tombol <i>edit</i> ” pada halaman gejala	Menampilkan form ubah data gejala berupa nama gejala dan <i>button</i> simpan	√
			Menekan tombol “simpan” pada halaman <i>edit</i> gejala	Berhasil mengubah data gejala menampilkan <i>pop up</i> pesan	√
			Menekan tombol “ok” pada <i>pop up</i> pesan	Menampilkan data gejala berupa nama gejala dan no(id_gejala)	√

			Menekan tombol “hapus” pada halaman gejala	Berhasil menghapus gejala	√
			Menekan tombol hapus gejala (gagal hapus gejala)	Gagal hapus gejala menampilkan <i>pop up</i> pesan gagal hapus gejala	√
4.	Mengelola Aturan / <i>rule</i> (pakar)	Menu Produk berfungsi untuk mengelola data aturan (tambah, <i>edit</i> dan hapus)	Memilih menu <i>rule</i>	Menampilkan data <i>rule</i> berupa id namapenyakit, nama gejala , nilai mb, niali md dan nilai cf	√
			Menekan tombol “tambah” pada halaman data <i>rule</i>	Menampilkan <i>pop up</i> form <i>input</i> data <i>rule</i> berupa id namapenyakit, nama gejala , nilai mb, niali md dan nilai cf	√
			Menekan tombol “simpan” pada halaman <i>rule</i>	Berhasil menambah data pada tabel gabungan di <i>database</i>	√
			Menekan tombol “ok”	Menampilkan data <i>rule</i> berupa id namapenyakit, nama gejala , nilai mb, niali md dan nilai cf.	√
			Menekan tombol “hapus” pada halaman data <i>rule</i>	Berhasil mengubah hapus data <i>rule</i>	√
			Menekan tombol “hapus” pada halaman data <i>rule</i>	Gagal mengubah hapus data <i>rule</i>	√

			Menekan tombol edit	Menampilkan halaman edit rule	√
			Mengisi form dan menekan tombol simpan	Berhasil merubah data dan menampilkan pesan data berhasil disimpan	√
5.	Konsultasi	Menu konsultasi berfungsi untuk memilih gejala yang diderita yang selanjutnya digunakan untuk konsultasi.	Memilih menu konsultasi	Menampilkan form <i>input</i> konsultasi yang berisi check box gejala	√
			Menekan memilih gejala dan klik simpan	Menyimpan data konsultasi(gejala, tanggal konsultasi) dan Menampilkan <i>detail</i> penyakit yang memiliki nilai tertinggi serta kemungkinan penyakit lain yang kemungkinan diderita	√
7.	Konsultasi	Menu konsultasi berfungsi untuk memilih gejala yang diderita yang selanjutnya digunakan untuk konsultasi.	Memilih menu konsultasi	Menampilkan form <i>input</i> konsultasi yang berisi check box gejala	√

			Memilih gejala dan tekan tombol proses	Menampilkan halaman hasil konsultasi dengan 2 nilai tertinggi	√
8.	<i>Record</i> (member)	Menu <i>record</i> berfungsi menampilkan hasil konsultasi <i>user</i> yang tersimpan	Menekan memilih <i>record</i>	Menampilkan data konsultasi tanggal kosultasi dan penyakit yang terbesar	√
			Menekan tombol <i>detail</i>	Menampilkan <i>detail</i> konultasi, berupa data gejala, nama penyakit, penanganan, pengobatan, pencegahan dan daftar <i>list</i> penyakit yang kemungkinan lain diderita	√
9.	<i>Record</i> (member)	Menu <i>record</i> berfungsi menampilkan hasil konsultasi <i>user</i> yang tersimpan	Menekan memilih <i>record</i>	Menampilkan data konsultasi nomor, nama, tanggal kosultasi dan penyakit yang terbesar	√
			Menekan tombol <i>detail</i>	Menampilkan <i>detail</i> konultasi, berupa data gejala, nama penyakit, penanganan, pengobatan, pencegahan dan daftar <i>list</i> penyakit yang kemungkinan lain diderita	√
10.	<i>User</i> (pakar)	Menu <i>user</i> berfungsi	Menekan tombol “tambah”	Menampilkan <i>pop up</i> form	√

		untuk mengelola data <i>user</i> (tambah, hapus dan <i>edit</i>)		<i>input</i> data <i>user</i> no.anggota, alamat, nomorhp, no. identitas, <i>username</i> dan password	
			Menekan tombol “simpan”	Berhasil mengubah menambah data dan menampilkan pesan berhasil simpan	√
			Menekan tombol ok	Menampilkan halaman <i>user</i> berisi id <i>user</i> , nama <i>user</i>	√
			Menekan tombol <i>edit</i>	Menampilkan halaman form <i>input</i> data <i>user</i> no.anggota, alamat, nomorhp, no. identitas, <i>username</i> dan password	√
			Klik tombol simpan	Berhasil mengubah data dan menampilkan pesan berhasil ubah data	√
			Klik ok	Menampilkan halaman <i>user</i> berisi id <i>user</i> , nama <i>user</i>	√
			Klik hapus pesan	Menampilkan pesan berhasil dan mengubah status <i>user</i> menjadi tidak aktif	√
	User Profil	Menu <i>user</i> profil berguna untuk melihat dan melakukan	Menekan menu <i>user</i> profil	Menampilkan halaman profil <i>user</i> sesuai dengan hak akses	√

		perubahan data pribadi user			
			Menekan tombol simpan jika melakukan perubahan data	Menampilkan pesan mengubah data dan dat user telah berubah	√
	<i>Record</i> penyakit(pakar)	Menu <i>record</i> berfungsi menampilkan hasil konsultasi <i>user</i> yang tersimpan	Menekan memilih <i>record</i> penyakit	Menampilkan data konsultasi nomor, nama, tanggal kosultasi dan penyakit yang terbesar	√
			Menekan tombol <i>detail</i>	Menampilkan <i>detail</i> konsultasi, berupa data gejala, nama penyakit, penanganan, pengobatan, pencegahan dan daftar <i>list</i> penyakit yang kemungkinan lain diderita	√
	Informasi (<i>user</i>)	Menampilkan <i>detail</i> penyakit	Memilih menu informasi	Menampilkan halaman informasi dan <i>list</i> nama penyakit	√
			Klik <i>detail</i>	Menampilkan halaman <i>detail</i> penyakit jenis penyakit , penyebab, nama penyakit, pencegahan, penanganan, obat	√

Lampiran D. Hasil Analisis Data

Tabel D. 1 Tabel Pembobotan

Nama Penyakit	Gejala Penyakit	MB	MD
Scabies	Ternak Terlihat menggosok Badan dan menggaruk badan	0.9	0.3
	Nafsu makan ternak menurun	0.4	0.6
	Timbul Cairan Pada Bekas Luka	0.8	0.1
	Terdapat Keropeng pada tubuh ternak	0.8	0.4
	Kulit Menebal hingga berlipat lipatan	0.9	0.15
Pink Eye	Peradangan selaput mata	0.9	0.1
	Warna mata menjadi merah	0.9	0.15
	Bobot hewan menurun	0.6	0.4
	Mata ternak berair	0.9	0.1
	Mata ternak berair	0.9	0.1
ORF	kelopak mata membesar	0.8	0.2
	Muncul Bintik Merah Disekitar bibir atau lesi disekitar bibir	0.9	0.1
	Terdapat benjolan bernanah disekitar mulut	0.7	0.3
	Pembengkakan pada mulut	0.6	0.4
	nafsu makan ternak menurun	0.6	0.4
Mastitis	Keropeng pada bibir dan area sekitar mulut	0.8	0.2
	Terjadi pembengkakan pada ambing ternak	0.9	0.1
	suhu tubuh ternak meningkat	0.8	0.2
	Frekuensi Nafas ternak cepat	0.6	0.4
Myasis	Nafsu makan ternak menurun	0.45	0.5
	Timbul Cairan pada bekas luka	0.6	0.4
	Terdapat belatung pada bekas luka	0.9	0.05
	suhu tubuh ternak meningkat	0.6	0.5
	terjadi peradangan pada luka	0.8	0.2
Cacingan	Timbul cairan pada bekas luka	0.6	0.2
	Bobot hewan menurun drastic dan berulcer membesar	0.7	0.05
	Ternak kurus walaupun banyak makan	1	0.1
	kotoran ternak berbentuk cair dan berbau khusus	0.8	0.2

	Kotoran ternak kering	0.7	0.5
	Gerak ternak tidak lincah dan lemah	0.6	0.4
	terjadi enedema antara rahang	0.6	0.4
	ternak lambat dan malas dalam bergerak	0.6	0.4
	bulu ternak kasar, kusam dan berdiri	0.8	0.2
	mata pucat	0.6	0.4
Kelumpuhan	Nafsu makan ternak menurun	1	0.5
	hewan terlihat cemas	0.8	0.2
	Kembung	0.2	0.8
	Kaki lemah hingga ternak tidak dapat berdiri	1	0
	mata pucat	0.6	0.4
Keracunan	Frekuensi Nafas meningkat	0.8	0.2
	Tubuh ternak gemetar	0.6	0.4
	Pupil mata melebar	0.6	0.4
	keluar liur dan busa pada mulut	0.8	0.2
Diare	Kotoran ternak berbentuk cair	1	0.15
	kotoran ternak disertai lendir	0.8	0.25
Kuku dan Mulut	Kambing lemah tidak lincah	0.7	0.5
	Ternak mengalami Peningkatan salivasi	0.8	0.4
	Bobot hewan menurun	0.6	0.4
	Nafsumakan menurun	0.6	0.5
	Melepuh dibagian sekitar kuku dan batas kuku	0.9	0.1
	suhu tubuh ternak meningkat	0.6	0.4
	Terdapat benjolan bernanah disekitar mulut	0.9	0.2
Kembung	Nafsu makan ternak menurun	0.8	0.5
	Ternak terlihat lemah karena kesakitan	0.8	0.5
	Frekuensi nafas menurun	0.6	0.4
	Perut ternak membengkak dan kembung	0.8	0.2
	Selaput lendir berwarna kebiruan	0.8	0.4
	Pembesaran perut sebelah kiri	0.9	0.05
Foot Root	kaki pincang	0.8	0.2
	terjadi peradangan pada kuku ternak	0.9	0.1
	terdapat nanah pada kuku yang luka	0.9	0.1
	Kulit ternak melepuh disekitar kuku kaki	0.8	0.2
	nafsu makan menurun	0.6	0.4
	penurunan berat bada ternak	0.6	0.4

	Kaki ternak terlihat lemah hingga tidak mampu berdiri	0.7	0.3
--	---	-----	-----

