



**SISTEM PAKAR DIAGNOSIS PENYAKIT PADA ANJING
PELIHARAAN MENGGUNAKAN METODE *CERTAINTY*
FACTOR BERBASIS WEB**

SKRIPSI

oleh

Muliyansyah Suciono

NIM 102410101105

FAKULTAS ILMU KOMPUTER

UNIVERSITAS JEMBER

2018



**SISTEM PAKAR DIAGNOSIS PENYAKIT PADA ANJING
PELIHARAAN MENGGUNAKAN METODE *CERTAINTY*
FACTOR BERBASIS WEB**

SKRIPSI

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

oleh

Muliyansyah Suciono

NIM 102410101105

FAKULTAS ILMU KOMPUTER

UNIVERSITAS JEMBER

2018

PERSEMBAHAN

Skripsi ini saya persembahkan untuk:

1. Allah SWT yang senantiasa memberikan kekuatan dalam mengerjakan skripsi ini;
2. Keluarga tercinta, Ayah Bambang Sutiono dan Ibu Arsusiyah, serta Adik-adik Suryansyah Suciono dan Adhiyansyah Suciono;
3. Prof. Drs. Slamin, M.Comp.Sc., Ph.D. selaku Ketua Fakultas Ilmu Komputer Universitas Jember;
4. Pembimbing I Anang Andrianto, S.T., M.T. dan Pembimbing II Fahrobby Adnan, S.Kom., M.M.S.ISI.;
5. Teman seperjuangan di Program Studi Sistem Informasi Universitas Jember;
6. Almamater Program Studi Sistem Informasi Universitas Jember.

MOTO

“Sayangilah yang ada di bumi, niscaya Yang ada di langit akan menyayangimu.”
(HR. At Thabrani dalam al Mu’jam al Kabir, Shahihul Jaami’ no. 896)



PERNYATAAN

Saya yang bertanda tangan di bawah ini:

Nama : MuliyanSyah Suciono

NIM : 102410101105

menyatakan dengan sesungguhnya bahwa karya ilmiah yang berjudul “Sistem Pakar Diagnosis Penyakit Pada Anjing Peliharaan Menggunakan Metode *Certainty Factor* Berbasis Web” adalah benar-benar hasil karya sendiri, kecuali kutipan yang sudah saya sebutkan sumbernya, belum pernah diajukan pada institusi mana pun, dan bukan karya jiplakan. Saya bertanggung jawab atas keabsahan dan kebenaran isisnya sesuai dengan sikap ilmiah yang harus dijunjung tinggi.

Demikian pernyataan ini saya buat dengan sebenarnya, tanpa ada tekanan dan paksaan dari pihak mana pun serta bersedia mendapat sanksi akademik jika ternyata di kemudian hari pernyataan ini tidak benar.

Jember,
Yang menyatakan,

MuliyanSyah Suciono
NIM 102410101105

SKRIPSI

**SISTEM PAKAR DIAGNOSIS PENYAKIT PADA ANJING
PELIHARAAN MENGGUNAKAN METODE CERTAINTY
FACTOR BERBASIS WEB**

oleh

Muliyansyah Suciono
NIM 102410101105

Pembimbing:

Dosen Pembimbing Utama : Anang Andrianto, S.T., M.T.
NIP 196906151997021002

Dosen Pembimbing Anggota : Fahrobby Adnan, S.Kom., M.M.S.I.
NIP 198706192014041001

PENGESAHAN

Skripsi berjudul “Sistem Pakar Diagnosis Penyakit Pada Anjing Peliharaan Menggunakan Metode *Certainty Factor* Berbasis Web” telah diuji dan disahkan pada:

hari, tanggal :

tempat : Fakultas Ilmu Komputer Universitas Jember

Tim Penguji:

Penguji I,

Penguji II,

Drs. Antonius Cahya P., M.App.Sc., Ph.D.
NIP 196909281993021001

Fajrin Nurman Arifin, S.T., M.Eng.
NIP 198511282015041002

Mengesahkan
Ketua Fakultas,

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

RINGKASAN

Sistem Pakar Diagnosis Penyakit Pada Anjing Peliharaan Menggunakan Metode *Certainty Factor* Berbasis Web; Mulyansyah Suciono; 102410101105; 2018; 157 halaman; Program Studi Sistem Informasi Fakultas Ilmu Komputer Universitas Jember.

Anjing merupakan salah satu jenis hewan yang digemari oleh masyarakat untuk dipelihara. Alasannya karena merawatnya mudah, biaya perawatannya relatif sedikit, dan bisa memberi banyak manfaat kepada pemiliknya. Sama halnya dengan makhluk hidup yang lain, anjing juga dapat terserang penyakit dan sudah menjadi tanggung jawab bagi pemiliknya untuk mencari cara untuk menyembuhkan peliharaannya. Hal yang harus dilakukan pemilik anjing peliharaan adalah mencari informasi atau menemui seorang pakar untuk berkonsultasi mengenai penyakit yang menyerang peliharaannya. Terkadang muncul masalah di mana pemilik anjing peliharaan kesulitan mendapatkan informasi maupun bertemu dengan pakar sehingga kondisi kesehatan anjing memburuk dan terlambat untuk disembuhkan.

Pengembangan sistem pakar diagnosis penyakit pada anjing peliharaan mampu menjawab masalah yang ditemukan karena sistem mampu menganalisis fakta-fakta yang tersedia dan membantu pemilik anjing peliharaan dalam mengambil keputusan mengenai kondisi kesehatan peliharaannya. Sistem bekerja menyerupai seorang pakar dalam menganalisis fakta dan mengambil keputusan untuk memecahkan suatu masalah. Sistem dapat memberikan hasil diagnosis penyakit pada anjing peliharaan berdasarkan analisis pada gejala dalam menyebabkan suatu penyakit, sehingga pemilik anjing peliharaan tidak perlu mengalami kesulitan untuk mencari informasi maupun berkonsultasi dengan pakar. Pengembangan sistem pakar akan menerapkan metode *certainty factor* untuk meningkatkan akurasi hasil diagnosis penyakit, karena metode *certainty factor* menggunakan nilai dari tingkat keyakinan seorang pakar dalam menganalisis gejala-gejala yang menimbulkan suatu penyakit sebagai aturan dalam sistem. Sistem pakar yang menerapkan metode *certainty factor* sangat mudah untuk digunakan, karena pemilik anjing peliharaan hanya perlu memilih pilihan gejala penyakit yang disediakan oleh sistem sesuai

dengan pengamatan mereka terhadap kondisi kesehatan anjing peliharaan mereka. Setelah itu, metode *certainty factor* hanya perlu menghitungnya berdasarkan aturan-aturan yang sudah dibuat sebelumnya. Akhir dari proses perhitungan akan diperoleh tingkat kepastian dari penyakit-penyakit yang mempunyai kemungkinan untuk mengganggu kondisi kesehatan anjing peliharaan. Setelah itu, tingkat kepastian dari kemungkinan-kemungkinan tersebut akan diurutkan dan penyakit yang memiliki kemungkinan terbesar akan menjadi hasil diagnosis dari sistem. Sistem pakar juga akan dikembangkan dengan basis web, sehingga pemilik anjing peliharaan lebih mudah untuk mengakses sistem kapan saja apabila diperlukan, dengan begitu pemilik anjing peliharaan dapat mengetahui penyakit yang menyerang anjing peliharaan mereka tanpa perlu menemui seorang pakar terlebih dahulu.

PRAKATA

Puji syukur ke hadirat Allah SWT, atas segala nikmat dan karunia-Nya sehingga penulis dapat menyelesaikan skripsi yang berjudul “Sistem Pakar Diagnosis Penyakit Pada Anjing Peliharaan Menggunakan Metode *Certainty Factor* Berbasis Web”. Skripsi ini disusun untuk memenuhi salah satu syarat menyelesaikan pendidikan strata satu (S1) pada Program Studi Sistem Informasi Fakultas Ilmu Komputer Universitas Jember.

Pada kesempatan ini penulis ingin menyampaikan ucapan terima kasih kepada pihak-pihak yang telah memberikan bantuan selama penyusunan skripsi:

1. Prof. Drs. Slamir, M.Comp.Sc., Ph.D., selaku Ketua Fakultas Ilmu Komputer Universitas Jember;
2. Anang Andrianto, S.T., M.T., selaku Dosen Pembimbing Utama dan Fahrobby Adnan, S.Kom., M.M.S.I., selaku Dosen Pembimbing Anggota yang telah membimbing selama proses penulisan skripsi;
3. Windi Eka Yulia Retnani, S.Kom., M.T., selaku Dosen Pembimbing Akademik yang telah membimbing selama penulis menjadi mahasiswa;
4. Seluruh Bapak dan Ibu dosen beserta staf karyawan Program Studi Sistem Informasi Fakultas Ilmu Komputer Universitas Jember;
5. Semua pihak yang tidak dapat disebutkan satu-persatu.

Penulis menerima segala kritik dan saran dari semua pihak demi perbaikan skripsi ini. Semoga skripsi ini bermanfaat bagi banyak pihak.

Jember, 17 Januari 2018

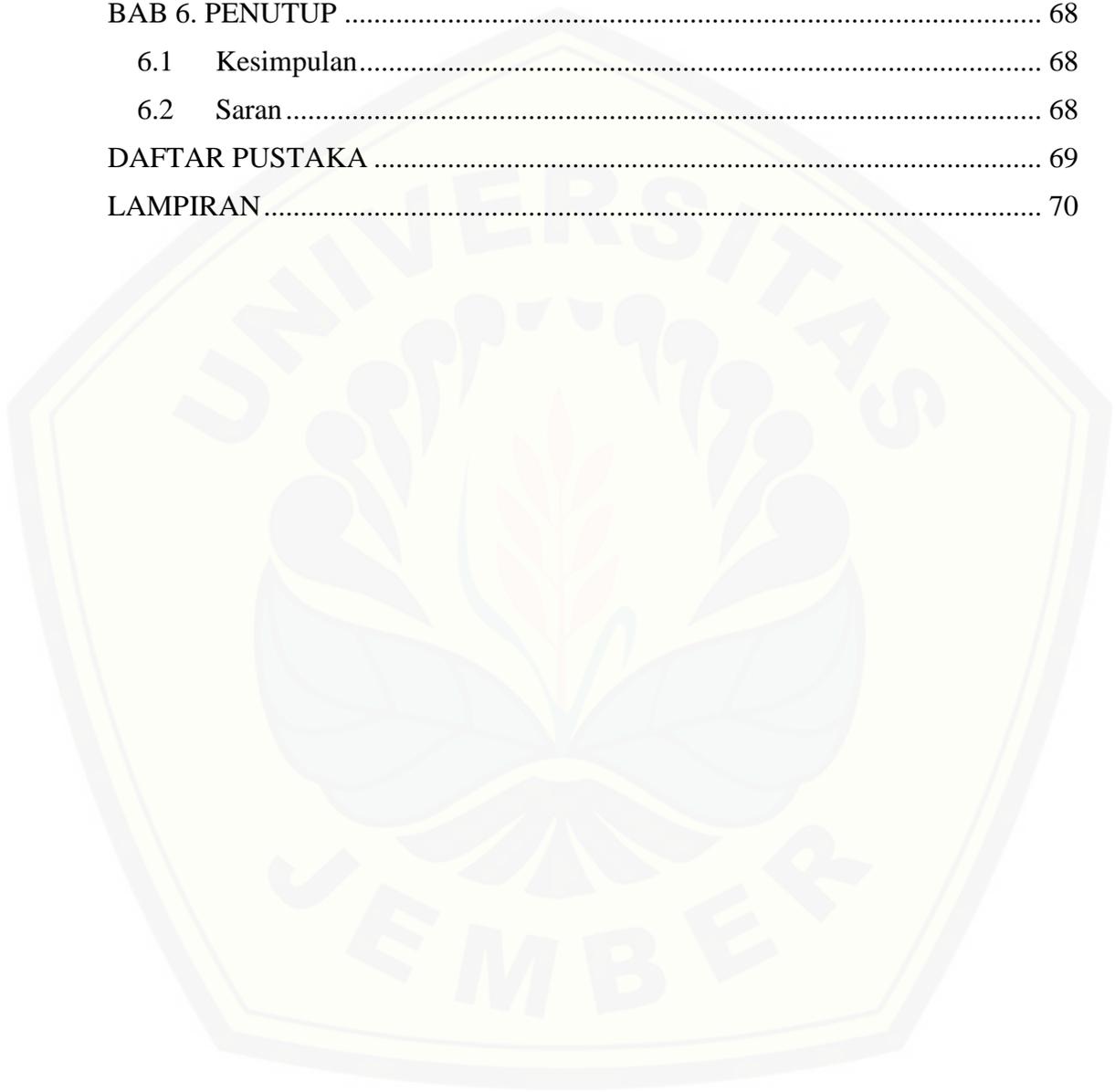
Penulis

DAFTAR ISI

HALAMAN JUDUL.....	ii
HALAMAN PERSEMBAHAN	iii
HALAMAN MOTO	iv
HALAMAN PERNYATAAN	v
HALAMAN PEMBIMBING.....	vi
HALAMAN PENGESAHAN.....	vii
RINGKASAN	viii
PRAKATA.....	x
DAFTAR ISI.....	xi
DAFTAR TABEL.....	xiv
DAFTAR GAMBAR	xv
DAFTAR LAMPIRAN.....	xvi
BAB 1. PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan dan Manfaat.....	3
1.3.1 Tujuan	3
1.3.2 Manfaat	3
1.4 Batasan Masalah.....	3
1.5 Sistematika Penulisan.....	4
BAB 2. TINJAUAN PUSTAKA	5
2.1 Penelitian Sebelumnya	5
2.2 Sistem Pakar	6
2.3 Metode Certainty Factor.....	7
2.4 Diagnosis	9
2.5 Penyakit Pada Anjing Peliharaan	12
2.6 Web	15
BAB 3. METODOLOGI PENELITIAN.....	16

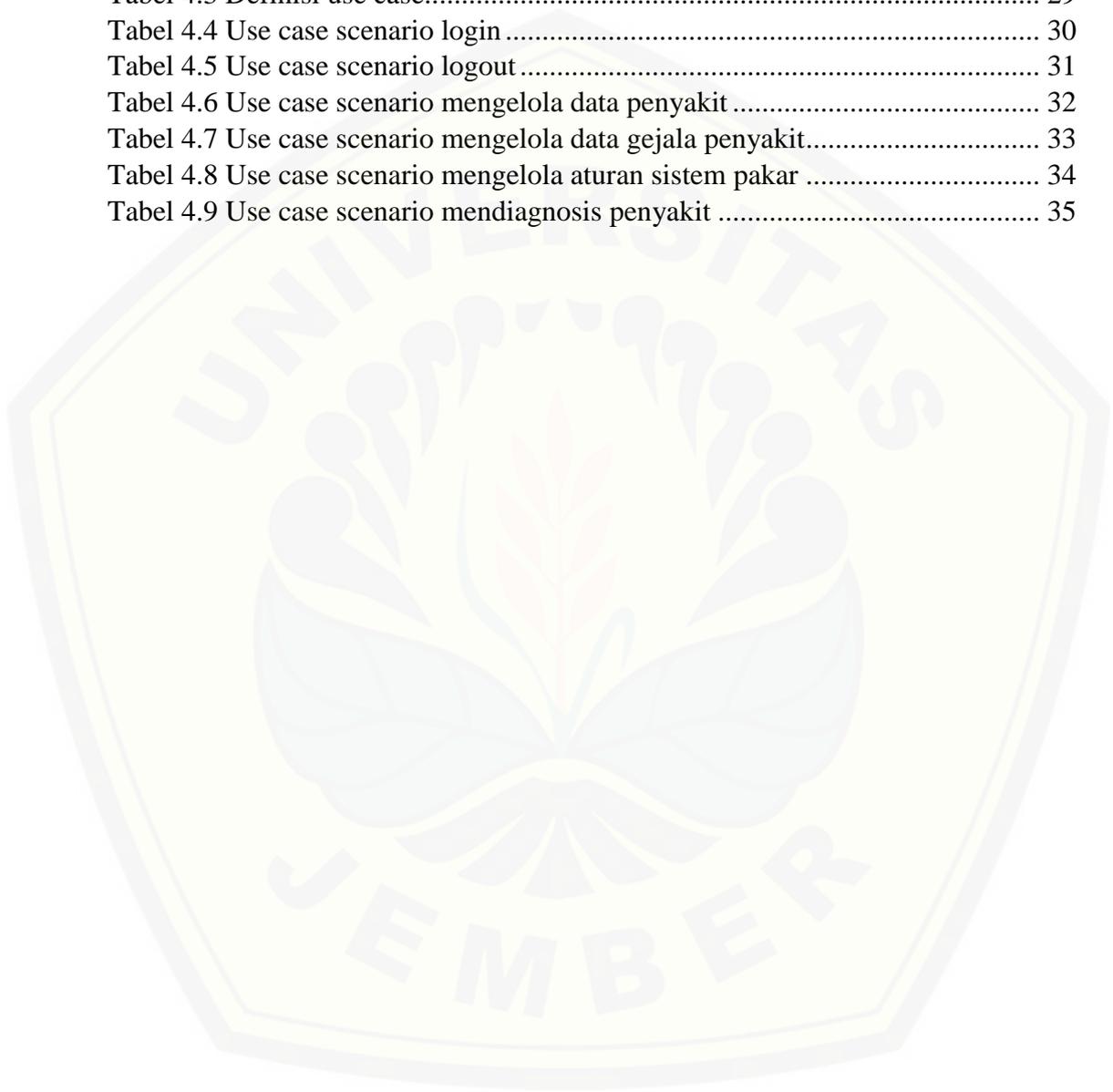
3.1	Jenis Penelitian	16
3.2	Alur Penelitian.....	16
3.2.1	Studi Literatur	17
3.2.2	Pengumpulan Data	17
3.2.3	Pengembangan Sistem	18
3.3	Gambaran Sistem	22
BAB 4. PENGEMBANGAN SISTEM.....		24
4.1	Pengumpulan Kebutuhan Sistem.....	24
4.1.1	Kebutuhan Fungsional	26
4.1.2	Kebutuhan Non-fungsional	26
4.2	Perancangan 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>	29
4.2.4	<i>Activity Diagram</i>	36
4.2.5	<i>Sequence Diagram</i>	42
4.2.6	<i>Class Diagram</i>	47
4.2.7	<i>Entity Relationship Diagram</i>	48
4.3	Mengkodekan Sistem	49
4.4	Pengujian Sistem	49
4.4.1	<i>White Box Testing</i>	50
4.4.2	<i>Black Box Testing</i>	50
4.5	Evaluasi Sistem	50
4.6	Implementasi Sistem	50
BAB 5. HASIL DAN PEMBAHASAN.....		51
5.1	Hasil.....	51
5.1.1	Halaman <i>Login Admin</i>	52
5.1.2	Halaman <i>Dashboard</i>	53
5.1.3	Halaman <i>Kelola Data Penyakit</i>	54
5.1.4	Halaman <i>Kelola Data Gejala</i>	55
5.1.5	Halaman <i>Kelola Data Aturan</i>	56
5.1.6	Halaman <i>Pengunjung</i>	57

5.1.7	Halaman Diagnosis	58
5.2	Pembahasan	59
5.2.1	Penerapan Metode <i>Certainty Factor</i> pada Sistem	59
5.2.2	Perhitungan Metode <i>Certainty Factor</i>	60
BAB 6.	PENUTUP	68
6.1	Kesimpulan.....	68
6.2	Saran	68
DAFTAR PUSTAKA	69
LAMPIRAN	70



DAFTAR TABEL

Tabel 4.1 Data Set Pembobotan Gejala Penyakit.....	25
Tabel 4.2 Definisi aktor.....	29
Tabel 4.3 Definisi use case.....	29
Tabel 4.4 Use case scenario login	30
Tabel 4.5 Use case scenario logout.....	31
Tabel 4.6 Use case scenario mengelola data penyakit	32
Tabel 4.7 Use case scenario mengelola data gejala penyakit.....	33
Tabel 4.8 Use case scenario mengelola aturan sistem pakar	34
Tabel 4.9 Use case scenario mendiagnosis penyakit	35



DAFTAR GAMBAR

Gambar 3.1 Diagram alir tahap penelitian	16
Gambar 3.2 Siklus hidup pengembangan perangkat lunak model prototype.....	19
Gambar 4.1 Business process sistem	27
Gambar 4.2 Use case diagram sistem	28
Gambar 4.3 Activity diagram login	36
Gambar 4.4 Activity diagram logout	37
Gambar 4.5 Activity diagram mengelola data penyakit.....	38
Gambar 4.6 Activity diagram mengelola data gejala penyakit.....	39
Gambar 4.7 Activity diagram mengelola aturan sistem pakar.....	40
Gambar 4.8 Activity diagram mendiagnosis penyakit.....	41
Gambar 4.9 Sequence diagram login	42
Gambar 4.10 Sequence diagram logout	43
Gambar 4.11 Sequence diagram mengelola data penyakit	44
Gambar 4.12 Sequence diagram mengelola data gejala penyakit.....	45
Gambar 4.13 Sequence diagram mengelola aturan sistem pakar.....	46
Gambar 4.14 Sequence diagram mendiagnosis penyakit.....	47
Gambar 4.15 Class diagram sistem.....	48
Gambar 4.16 Entity relationship diagram sistem.....	49
Gambar 5.1 Pembagian hak akses halaman web	51
Gambar 5.2 Halaman login admin	52
Gambar 5.3 Halaman dashboard.....	53
Gambar 5.4 Halaman kelola data penyakit	54
Gambar 5.5 Halaman kelola data gejala	55
Gambar 5.6 Halaman kelola data aturan	56
Gambar 5.7 Halaman pengunjung.....	57
Gambar 5.8 Halaman diagnosis	58

DAFTAR LAMPIRAN

Lampiran A. Use Case Scenario	70
Lampiran B. Hasil Penulisan Kode Program	104
Lampiran C. Hasil White Box Testing Pada Sistem	124
Lampiran D. Hasil Black Box Testing Pada Sistem	137



BAB 1. PENDAHULUAN

Bab Pendahuluan merupakan langkah awal dari penulisan skripsi. Bab ini berisi latar belakang, rumusan masalah, tujuan dan manfaat, batasan masalah, dan sistematika penulisan.

1.1 Latar Belakang

Memiliki hewan peliharaan merupakan hal populer yang diminati oleh masyarakat saat ini. Masyarakat beranggapan bahwa memiliki hewan peliharaan dapat memberikan dampak positif kepada mereka. Pada umumnya hewan yang dipilih menjadi hewan peliharaan adalah hewan yang dapat menghibur, hewan yang memiliki penampilan fisik menarik, dan hewan yang mampu melindungi pemiliknya. Ada banyak hewan yang dapat dijadikan sebagai hewan peliharaan, salah satunya adalah anjing.

Anjing adalah mamalia karnivora yang telah mengalami penjinakan sejak lama sehingga sifat anjing menjadi selaras dengan manusia. Hasil dari penjinakan tersebut, anjing menjadi mudah berkomunikasi dengan manusia melalui tindakan atau pola tertentu. Mudahnya komunikasi manusia dengan anjing membuat manusia dapat melatih anjing dan membantu perkembangan hidup manusia.

Umumnya masyarakat memilih anjing sebagai hewan peliharaan favorit mereka karena merawat anjing peliharaan sangat mudah dan tidak perlu mengeluarkan biaya yang banyak. Akan tetapi masyarakat yang menjadi pemilik anjing peliharaan sering mengabaikan kondisi kesehatan anjing peliharaan mereka, sehingga anjing peliharaan menjadi sakit, bahkan dapat mengakibatkan kematian dan menularkan penyakit tersebut kepada mereka. Untuk menghindari hal tersebut diperlukan informasi mengenai penyakit pada anjing peliharaan yang akurat dari seorang pakar sehingga pemilik anjing peliharaan dapat membuat keputusan mengenai kondisi kesehatan anjing peliharaan mereka. Terkadang muncul masalah di mana pemilik anjing peliharaan kesulitan untuk menemukan seorang pakar yang dapat memberikan informasi mengenai penyakit yang menyerang anjing peliharaan

mereka. Maka dibutuhkan sebuah solusi yang dapat menjawab permasalahan tersebut.

Sistem pakar diagnosis penyakit pada anjing peliharaan dapat menjawab permasalahan tersebut. Sistem dapat bertindak menyerupai seorang pakar dalam menganalisis penyakit pada anjing peliharaan berdasarkan gejala penyakit yang dapat diamati oleh pemiliknya dan menghasilkan keputusan berupa hasil diagnosis penyakit sehingga pemilik anjing peliharaan dapat mengetahui mengenai kondisi kesehatan anjing peliharaan mereka tanpa harus bertemu seorang pakar secara langsung. Penelitian sebelumnya yang berjudul “Sistem Pakar Diagnosis Penyakit Anjing dengan *Forward Chaining*“ (Yulianti & Mewati, 2012) menyarankan untuk memperhitungkan seberapa besar kemungkinan suatu gejala dalam menyebabkan suatu penyakit sehingga hasil diagnosis lebih akurat. Metode *Certainty Factor* akan diterapkan pada sistem yang dikembangkan untuk meningkatkan akurasi hasil diagnosis. Metode *certainty factor* menggunakan nilai keyakinan dari fakta-fakta untuk memperkuat besarnya derajat kepastian. Pada sistem yang dikembangkan, metode *certainty factor* akan menggunakan nilai keyakinan dari kemungkinan suatu gejala dalam menyebabkan suatu penyakit pada anjing peliharaan untuk memberikan hasil diagnosis berupa persentase kepastian dari penyakit pada anjing peliharaan tersebut. Sistem akan dikembangkan pada basis web karena sistem berbasis web mampu menampung banyak informasi dan mudah untuk memperbarui informasi yang ada di dalamnya. Sistem berbasis web juga memiliki kelebihan dapat diakses kapan saja oleh perangkat apa pun selama perangkat tersebut terhubung ke jaringan internet.

1.2 Rumusan Masalah

Berdasarkan uraian latar belakang di atas, dapat dirumuskan beberapa masalah yang dapat diteliti, antara lain:

1. Bagaimana menerapkan metode *certainty factor* dalam mendiagnosis penyakit pada anjing peliharaan?
2. Bagaimana mengembangkan sistem pakar diagnosis penyakit pada anjing peliharaan menggunakan metode *certainty factor* berbasis web?

1.3 Tujuan dan Manfaat

Berdasarkan rumusan masalah di atas, dapat dipaparkan tujuan dan manfaat dari penelitian ini, antara lain:

1.3.1 Tujuan

Tujuan yang ingin dicapai oleh penelitian ini adalah:

1. Menerapkan metode *certainty factor* dalam mendiagnosis penyakit pada anjing peliharaan.
2. Mengembangkan sistem pakar diagnosis penyakit pada anjing peliharaan menggunakan metode *certainty factor* berbasis web.

1.3.2 Manfaat

Manfaat yang ingin dicapai oleh penelitian ini adalah:

1. Mengetahui proses penerapan metode *certainty factor* dalam mendiagnosis penyakit pada anjing peliharaan.
2. Mengetahui proses pengembangan sistem pakar diagnosis penyakit pada anjing peliharaan berbasis web.
3. Menjadi referensi untuk penelitian lebih lanjut.

1.4 Batasan Masalah

Batasan masalah yang diambil dalam penelitian dan pengembangan sistem ini adalah:

1. Sistem hanya memberikan hasil diagnosis penyakit pada anjing peliharaan.
2. Sistem hanya mendiagnosis penyakit berdasarkan gejala pada anjing peliharaan yang dapat diamati oleh pemilik mereka.
3. Sistem hanya memberikan hasil diagnosis penyakit berdasarkan persentase kepastian penyakit terbesar.
4. Sistem hanya dapat berjalan pada aplikasi *browser* yang terhubung ke jaringan internet.

1.5 Sistematika Penulisan

Sistematika penulisan dari penyusunan skripsi ini adalah sebagai berikut:

1. Pendahuluan

Bab ini berisi penjelasan mengenai latar belakang, rumusan masalah, tujuan dan manfaat, batasan masalah, dan sistematika penulisan pada penelitian ini.

2. Tinjauan Pustaka

Bab ini berisi penjelasan mengenai tinjauan pustaka yang menjadi dasar penelitian dan pengembangan sistem ini.

3. Metodologi Penelitian

Bab ini berisi penjelasan mengenai metode-metode yang digunakan pada penelitian dan pengembangan sistem ini.

4. Pengembangan Sistem

Bab ini berisi penjelasan mengenai langkah-langkah dari pengembangan sistem.

5. Hasil dan Pembahasan

Bab ini berisi penjelasan mengenai hasil dan pembahasan dari pengembangan sistem.

6. Penutup

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

BAB 2. TINJAUAN PUSTAKA

Bab ini berisi tentang tinjauan pustaka yang mendasari penelitian. Mencakup dasar teori atau pendapat ahli yang menjadi landasan pikiran peneliti untuk melakukan sebuah penelitian.

2.1 Penelitian Sebelumnya

Penelitian ini didasari oleh penelitian sebelumnya yang berjudul “Sistem Pakar Diagnosis Penyakit Anjing dengan *Forward Chaining*” (Yulianti & Mewati, 2012). Penelitian tersebut membahas penerapan metode *forward chaining* dalam mendiagnosis penyakit pada anjing. Penelitian tersebut menghasilkan kesimpulan bahwa mendiagnosis penyakit pada anjing dapat dilakukan berdasarkan pengamatan gejala pada penyakit. Peneliti mengajukan saran untuk menyempurnakan sistem dengan memperhitungkan seberapa besar kemungkinan terjadinya suatu gejala dalam menyebabkan suatu penyakit sehingga hasil dari diagnosis lebih akurat.

Penelitian lain yang berjudul “Sistem Pakar: Diagnosis Penyakit Unggas dengan Menggunakan Metode *Certainty Factor*” (Rohajawati & Supriyati, 2010). Penelitian tersebut membahas penerapan metode *certainty factor* dalam mendiagnosis penyakit pada unggas. Penelitian tersebut menghasilkan kesimpulan bahwa metode *certainty factor* memberikan hasil lebih akurat dalam mendiagnosis penyakit pada hewan karena metode tersebut menghitung seberapa besar kemungkinan terjadinya suatu gejala dalam menyebabkan suatu penyakit.

Berdasarkan penelitian yang telah dilakukan oleh peneliti sebelumnya maka pengembangan sistem pakar diagnosis penyakit pada anjing peliharaan dengan menerapkan metode *certainty factor* dapat dilakukan. Metode *certainty factor* dapat meningkatkan akurasi hasil diagnosis penyakit pada anjing peliharaan karena metode *certainty factor* memperhitungkan seberapa besar kemungkinan terjadinya suatu gejala dalam menyebabkan suatu penyakit sehingga dapat menyempurnakan sistem sebelumnya yang menerapkan metode *forward chaining* dalam mendiagnosis penyakit pada anjing.

2.2 Sistem Pakar

Sistem pakar adalah aplikasi berbasis komputer yang digunakan untuk menyelesaikan masalah sebagaimana yang dipikirkan oleh pakar (Kusrini, 2008). Ahli lain berpendapat bahwa, Sistem pakar atau *Expert System* biasa disebut juga dengan *Knowledge Based System* yaitu suatu aplikasi computer yang ditujukan untuk membantu pengambilan keputusan atau pemecahan persoalan dalam bidang yang spesifik (Hayadi, 2016). Dari kedua pendapat ahli tersebut dapat disimpulkan bahwa sistem pakar adalah aplikasi berbasis komputer yang memiliki kemampuan menyerupai seorang pakar dalam mengambil keputusan dan memecahkan masalah dalam bidang tertentu.

Sistem pakar memerlukan beberapa komponen dalam pengembangannya, Menurut B. Herawan Hayadi, Komponen itu adalah:

1. Basis Pengetahuan (*Knowledge Base*)

Basis Pengetahuan merupakan inti program sistem pakar karena basis pengetahuan ini merupakan representasi pengetahuan (*Knowledge Representation*) dari seorang pakar (hal. 6).

2. Basis Data

Basis data adalah bagian yang mengandung semua fakta, baik fakta awal pada saat sistem mulai beroperasi maupun fakta yang didapatkan pada saat pengambilan kesimpulan sedang dilaksanakan (hal. 7).

3. Mesin Inferensi

Mesin inferensi adalah bagian yang mengandung mekanisme fungsi berpikir dan pola penalaran sistem yang digunakan oleh seorang pakar. Mekanisme ini akan menganalisa suatu masalah tertentu dan selanjutnya akan mencari jawaban atas kesimpulan yang terbaik. Mesin inferensi memulai pelacakannya dengan mencocokkan kaidah dan dalam basis pengetahuan dengan fakta yang ada dalam basis data. Ada dua teknik inferensi yang ada yaitu pelacakan ke belakang (*Backward Chaining*) yang memulai penalaran dari kesimpulan hipotesa menuju fakta yang mengandung hipotesa tersebut. Dan yang kedua yakni pelacakan ke depan (*Forward Chaining*) yang

merupakan kebalikan dari pelacakan kebelakang yaitu memulai dari sekumpulan data menuju kesimpulan.

Kedua metode inferensi tersebut dipengaruhi oleh tiga macam teknik penelusuran yaitu: *Depth-first search* melakukan penelusuran kaidah secara mendalam dari simpul akan bergerak menurun ke tingkat dalam yang berurutan. *Breadth-first search* bergerak dari simpul akar, simpul yang ada pada setiap tingkat diuji sebelum pindah ke tingkat selanjutnya. *Best-first search* bekerja berdasarkan kombinasi kedua metode sebelumnya (hal. 7).

4. Antar Muka Pemakai (*User Interface*)

Antar muka pemakai adalah bagian penghubung antara program sistem pakar dengan pemakainya. Pada bagian ini akan terjadi dialog antara program dengan pemakai. Program akan mengajukan pertanyaan berbentuk “ya/tidak” (*yes or no question*) atau berbentuk menu pilihan. Melalui jawaban yang diberikan oleh pemakai, sistem pakar akan mengambil kesimpulan yang berupa informasi ataupun anjuran sesuai dengan sifat dari sistem pakar (hal. 8).

2.3 Metode Certainty Factor

Certainty factor mengekspresikan kepercayaan dalam kejadian (atau fakta atau hipotesis) berdasarkan kejadian (atau pada penilaian seorang pakar) (Rohajawati & Supriyati, 2010). Seorang pakar yang digambarkan dalam metode certainty factor harus menunjukkan keyakinan dalam menilai sebuah fakta yang ada untuk meningkatkan hasil kepastian dari fakta. Menurut Daniel dan Virginia Gloria (2010), Certainty Factor memperkenalkan keyakinan dan ketidakkeyakinan yang kemudian diformulakan dalam rumusan dasar sebagai berikut:

$$CF[P, E] = MB[P, E] - MD[P, E] \dots\dots\dots \text{Persamaan 1}$$

Keterangan:

CF = *Certainty Factor* (Faktor Kepastian)

MB = *Measure of Belief* (Tingkat Keyakinan)

MD = *Measure of Disbelief* (Tingkat Ketidakyakinan)

P = *Probability* (Kemungkinan/Hipotesis)

E = *Evidence* (Peristiwa/Fakta)

Berikut adalah deskripsi beberapa kombinasi *Certainty Factor* terhadap berbagai kondisi:

1. *Certainty Factor* untuk kaidah dengan premis tunggal (*single premis rules*):

$$\begin{aligned} CF(H, E) &= CF(E) * CF(rule) \\ &= CF(user) * CF(pakar) \dots\dots\dots \text{Persamaan 2} \end{aligned}$$

2. *Certainty Factor* untuk kaidah dengan premis majemuk (*multiple premis rules*):

$$\begin{aligned} CF(A \text{ AND } B) &= \text{Minimum}(CF(a), CF(b)) \\ &* CF(rule) \dots\dots\dots \text{Persamaan 3} \end{aligned}$$

$$\begin{aligned} CF(A \text{ OR } B) &= \text{Maximum}(CF(a), CF(b)) \\ &* CF(rule) \dots\dots\dots \text{Persamaan 4} \end{aligned}$$

3. *Certainty Factor* untuk kaidah dengan kesimpulan yang serupa (*similarly concluded rules*):

$$CF_{COMBINE}(CF_1, CF_2) = CF_1 + CF_2 * (1 - CF_1) \dots\dots\dots \text{Persamaan 5}$$

Pengembangan Sistem pakar diagnosis penyakit pada anjing peliharaan akan menggunakan nilai keyakinan gejala dari penyakit yang menyerang seekor anjing peliharaan dan pengetahuan pakar terhadap nilai keyakinan gejala dari penyakit pada anjing peliharaan sebagai fakta sehingga sistem membutuhkan Persamaan 2 untuk menghitung faktor kepastian suatu gejala. Sistem juga membutuhkan Persamaan 5 yang didasari oleh Persamaan 2 karena suatu penyakit memungkinkan untuk memiliki gejala lebih dari satu. Dari nilai keyakinan gejala-gejala tersebut maka akan didapatkan faktor kepastian penyakit yang menyerang anjing peliharaan.

2.4 Diagnosis

Diagnosis adalah istilah yang menunjuk pada nama penyakit yang ada pada pasien yang perlu dirumuskan (ditentukan) oleh dokter (Hardjodisastro, 2006). Dalam kasus penyakit pada hewan, diagnosis dapat diartikan pengamatan secara seksama terhadap fakta (gejala dari penyakit) pada hewan untuk menentukan jenis penyakitnya.

Untuk mendiagnosis penyakit, secara klasik akan terdiri dari anamnesis dan pemeriksaan fisik serta pemeriksaan tambahan untuk menunjang diagnosis, misalnya laboratorium, rontgen, EKG (elektro kardiografi), dll. (Triharnoto, 2009). Dokter hewan juga menggunakan langkah di atas untuk mendiagnosis penyakit pada hewan. Berikut adalah penjelasan mengenai langkah-langkah tersebut Menurut Cacang Effendi & Widya Setiawati (2017):

1. Anamnesis

Anamnesis merupakan wawancara terhadap pemilik hewan. Pertanyaan-pertanyaan seputar kebiasaan hewan dan kejadian saat munculnya gejala ini biasanya langsung ditanyakan oleh dokter hewan kepada pemilik hewan (hal. 64).

2. Pemeriksaan Fisik

Pemeriksaan fisik pada hewan dilakukan dengan mengamati fisik hewan secara langsung menggunakan indra penglihatan, peraba, pendengaran, dan penciuman. Untuk lebih jelasnya, tata cara tersebut diuraikan di bawah ini (hal. 64):

- a. Inspeksi

Inspeksi atau pengamatan dilakukan dengan cara melihat hewan secara keseluruhan dari jarak pandang secukupnya sebelum hewan didekati untuk suatu pemeriksaan lebih lanjut.

Bagian yang diinspeksi adalah permukaan luar dari badan hewan dari daerah kepala, leher, badan sisi kiri dan kanan, bagian belakang dan kaki, aspek kulit, aspek rambut, mulut, anus, serta organ kelamin.

Konformitas dan kompakitas dari pertulangan juga diinspeksi (hal. 64).

b. Palpasi atau perabaan

Pemeriksaan permukaan luar dapat dilakukan dengan cara palpasi atau perabaan dengan tangan. Bagian yang dapat dipalpasi yakni, bagian kepala, leher, bagian rongga dada, perut atau abdomen, bagian pangkal pinggul atau pelvis, serta alat gerak atau ekstremitas. Ada pula teknik palpasi profunda, untuk mendeteksi struktur permukaan organ dalam. Secara umum, tujuan palpasi adalah untuk mendeteksi adanya perubahan bentuk, respon khusus, atau rasa nyeri pada bagian yang diraba. Seperti pada bagian leher atau penekanan tulang-tulang rusuk kiri dan kanan secara bersamaan (hal. 64-65).

c. Perkusi atau mengetuk

Prinsip perkusi adalah mengetuk atau memukul alat yang mengeluarkan denting atau gema. Dokter hewan akan mendengarkan pantulan gema yang ditimbulkan oleh alat pleksimeter yang diketuk oleh palu atau jari. Perkusi diarahkan atau diletakkan pada bidang datar di atas daerah yang dipenuhi udara pada bagian bawahnya. Misalnya, bagian sinus-sinus hidung, rongga dada sepertiga bagian atas, rongga dada sepertiga bawah, lambung, serta daerah usus-usus halus (hal. 65).

d. Auskultasi atau mendengar

Tujuan dari auskultasi adalah mendengarkan suara yang ditimbulkan oleh aktivitas organ, baik pada saat sehat maupun pada kasus-kasus tertentu. Auskultasi dapat dilakukan dengan menggunakan stetoskop maupun dengan menempelkan telinga langsung ke bagian tubuh hewan. Suara yang dapat ditangkap pada saat melakukan auskultasi berasal dari gerak paru-paru pada saat inspirasi maupun ekspirasi, suara katup-katup jantung serta suara peristaltik lambung dan usus-usus (hal. 65).

e. Mencium atau membaui

pemeriksaan fisik hewan dengan cara mencium atau membaui bertujuan untuk mengetahui perubahan aroma dari bagian tubuh hewan.

Hasil pemeriksaan ini dapat menuntun dokter hewan menuju suatu diagnosis.

Beberapa penyakit memiliki aroma khas. Misalnya aroma ureum yang dikeluarkan dari mulut seekor kucing dapat menuntun pemeriksaan pada kejadian gagal ginjal kronis stadium akhir. Sementara, bau feses yang sangat anyir pada seekor kucing menandakan adanya kandungan darah dalam feses (hal. 65).

f. Mengukur dan menghitung

Dokter hewan akan mengukur dan menghitung secara kuantitatif menggunakan satuan-satuan yang lazim, misalnya kali per menit dan derajat Celsius. Frekuensi pernapasan, denyut jantung, dan pulsus nadi dapat dihitung dengan satuan kali per menit. Sementara suhu tubuh kucing akan dihitung dengan satuan derajat Celsius. Namun, beberapa negara lain, misalnya Amerika Serikat mengukur satuan Fahrenheit untuk mengukur suhu tubuh kucing (hal. 65).

g. Pungsi pembuktian atau proof punctio

Tindakan ini dilakukan untuk mendapatkan ketegasan tunggal dari beberapa kemungkinan yang didapat dari inspeksi atau palpasi sebelumnya. Diperlukan tindakan ini untuk menyederhanakan gejala klinis yang muncul. Sebagai contoh, jika ada gejala timbunan cairan di rongga abdomen, biasanya dokter hewan akan mengambil sampel cairan tersebut dengan teknik proof punctio. Tujuannya adalah untuk mengetahui jenis cairan tersebut sehingga dapat dilakukan tindakan pengobatan yang tepat sasaran. Dikirimkan ke laboratorium klinik tertentu untuk tujuan pemeriksaan lebih lanjut (hal. 65-66).

3. Pemeriksaan Tambahan/Pemeriksaan Penunjang

Ketika pemeriksaan fisik masih belum menemukan diagnosis spesifik, biasanya dokter hewan akan menanyakan kepada pemilik hewan mengenai keinginan untuk dilakukan pemeriksaan penunjang. Tujuannya adalah untuk meneguhkan diagnosis.

Adapun contoh pemeriksaan penunjang adalah pemeriksaan darah rutin, kimia darah yang termasuk di dalamnya fungsi hati dan fungsi ginjal, radiografi, ultrasonografi (USG), dan lain sebagainya (hal. 66).

Berdasarkan penjelasan di atas, maka pengembangan sistem akan menggunakan langkah diagnosis pengamatan fisik karena pada langkah tersebut memuat banyak fakta berdasarkan gejala penyakit dan pada langkah tersebut memungkinkan seseorang yang bukan pakar untuk mengamati gejala penyakit pada anjing peliharaan sehingga sistem dapat memproses fakta dan memberikan hasil diagnosis dari gejala penyakit.

2.5 Penyakit Pada Anjing Peliharaan

Anjing Peliharaan sama halnya dengan makhluk hidup yang lain yaitu dapat terserang penyakit, baik itu penyakit ringan maupun penyakit berat yang dapat menyebabkan kematian. Berikut adalah beberapa penyakit berbahaya pada anjing beserta gejala klinis atau penularannya yang patut diperhatikan oleh pemilik anjing peliharaan menurut Bobby Santoso dan N.S Budiana (2015):

1. Canine Distemper

Penyebab: virus golongan paramyxo-virus.

Gejala klinis:

- a. Muncul radang di selaput lendir.
- b. Demam tinggi dan menyerang saluran pernapasan, pencernaan, serta susunan syaraf (hal. 147).

2. Infectious Canine Hepatitis

Penyebab: Virus golongan adeno-virus tipe 1 dan 2.

Gejala klinis:

- a. Mula-mula nafsu makan anjing turun, lesu, dan demam dengan suhu di atas 40. Selama 1-6 hari, lalu suhu tubuh normal dan naik kembali. Ia senang mencari tempat yang dingin.
- b. Matanya bengkak, berlanjut selaput mata memerah, keluar cairan encer di mata dan hidung. Anjing sering minum air karena haus dan

cenderung muntah-muntah. Kadang-kadang disertai sakit perut, dan bila diraba ada pembengkakan hati.

- c. Anjing sering batuk, gigi-giginya berkerut menahan sakit, dan gemetar, Terkadang anjing mati mendadak tanpa sebab yang jelas (hal. 147-148).

3. Canine Parvovirus

Penyebab: virus golongan Parvovirus.

Gejala klinis:

- a. Anjing kehilangan nafsu makan dan disertai muntah-muntah.
- b. Anjing mengalami diare.
- c. Biasanya disertai demam tinggi.
- d. Anjing senang tidur-tiduran di sudut ruangan (hal. 148).

4. Leptospirosis

Penyebab: bakteri dari ordo Spirochaetales.

Penularan:

- a. Melalui kotak langsung dengan anjing penderita.
- b. Anjing bisa tertular saat dikawinkan dengan anjing penderita.
- c. Induk yang menderita leptospirosis dapat menginfeksi ke anaknya meski masih dalam kandungan.
- d. Makanan dan minuman yang tercemar mudah menginfeksi tubuh anjing.
- e. Binatang yang menjadi vektor antara lain caplak dan tikus (hal. 148-149).

5. Cacing Pita (Roundworm)

Penyebab: cacing pita *Toxocara canis*.

Gejala klinis:

- a. Perut membesar.
- b. Pucat.
- c. Anemia.
- d. Mencret.
- e. Mati mendadak tanpa gejala yang jelas dengan angka kematian tinggi (hal. 149).

6. Cacing Tambang (Hookworm)

Penyebab: cacing tambang atau gelang *Ancylostoma canium* dan *Uncinaria stenocephala*.

Gejala klinis:

- a. Nafsu makan turun.
- b. Lesu.
- c. Pucat.
- d. Enemia.
- e. Bulu kusam.
- f. Mata berair (hal. 149-150).

7. Cacing Cambuk (Whipworm)

Penyebab: cacing cambuk *Trichuris vulpis*.

Cara penularan:

- a. Anjing secara tidak sengaja menelan telur, lalu masuk ke perut.
- b. Setelah 1 bulan, telur menetas, lalu masuk ke usus halus dan menjadi dewasa selama 10 minggu sampai akhirnya menetap di usus besar selama 6 bulan.
- c. Perkembangan cacing ini lambat sehingga gejala agak sukar dideteksi dengan seksama. Biasanya ditandai tubuh kurus, pucat, dan pertumbuhan terhambat (hal. 150).

8. Cacing Pita (Tepeworm)

Penyebab:

- a. *Echinococcus granulosus*
- b. *Dipylidium caninum*
- c. *Taenia taeniaformis*
- d. *T. Pisiformis*
- e. *T. Hydatigena*
- f. *T. Ovis*

Gejala klinis:

- a. Mula-mula anjing lesu.
- b. Nafsu makan tidak menentu

c. Diare ringan, dan sakit perut (hal. 151).

Penyakit anjing di atas beserta gejala klinis atau cara penularannya akan menjadi data dasar pengembangan sistem pakar diagnosis penyakit pada anjing peliharaan menggunakan metode certainty factor berbasis web.

2.6 Web

Web dalam pengembangan sistem ini berfungsi sebagai wadah berupa halaman aplikasi yang terdiri dari dokumen multimedia untuk membangun sebuah sistem pakar. Menurut M. Rudyanto Arief (2011) Web adalah salah satu aplikasi yang berisikan dokumen-dokumen multimedia (teks, gambar, animasi, video) di dalamnya yang menggunakan protokol HTTP (Hypertext Transfer Protocol) dan untuk mengaksesnya menggunakan perangkat lunak yang disebut browser (hal. 7).

Sebuah web kemudian dihubungkan dengan web lainnya secara terstruktur menjadi sebuah website dan dikirim ke jaringan internet. Website merupakan kumpulan dari halaman web yang sudah dipublikasikan di jaringan internet dan memiliki domain/URL (Uniform Resource Locator) yang dapat diakses semua pengguna internet dengan cara mengetikkan alamatnya (hal. 8)

Untuk mengakses website dari jaringan internet dibutuhkan sebuah perangkat lunak bernama browser yang mampu menerjemahkan website agar dapat dilihat dan dibaca oleh pengguna. Browser adalah aplikasi yang mampu menjalankan dokumen-dokumen web dengan cara diterjemahkan (hal. 8). Beberapa contoh browser yang populer digunakan oleh masyarakat antara lain Google Chrome, Mozilla Firefox, Opera, dan Safari.

Kelebihan dari sistem berbasis web adalah:

1. Sistem dapat diakses dari perangkat apa pun jika terhubung dengan internet dengan cara mengetikkan domain/URL website pada browser.
2. Lebih mudah untuk melakukan pembaruan sistem dan data yang ada di dalamnya.

BAB 3. METODOLOGI PENELITIAN

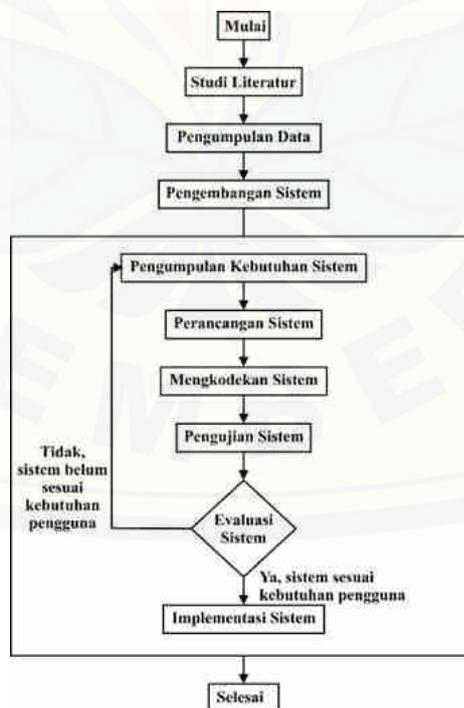
Bab ini berisi metode-metode yang digunakan pada penelitian dan pengembangan Sistem Pakar Diagnosis Penyakit pada Anjing Peliharaan Menggunakan Metode *Certainty Factor* Berbasis Web.

3.1 Jenis Penelitian

Jenis penelitian pada penelitian ini adalah pengembangan sistem. Tujuannya adalah meneliti implementasi metode *certainty factor* pada sistem pakar diagnosis penyakit anjing peliharaan berbasis web.

3.2 Alur Penelitian

Alur penelitian pada penelitian ini berisi tahapan penelitian yang meliputi studi literatur, pengumpulan data, dan pengembangan sistem. Berikut adalah diagram alir tahapan penelitian yang dapat dilihat pada Gambar 3.1.



Gambar 3.1 Diagram alir tahap penelitian

(Sumber: Hasil analisis, 2018)

Gambar 3.1 menggambarkan tahapan penelitian yang meliputi studi literatur, pengumpulan data, dan pengembangan sistem. Model siklus hidup pengembangan perangkat lunak yang digunakan pada Sistem Pakar Diagnosis Penyakit pada Anjing Peliharaan Menggunakan Metode *Certainty Factor* Berbasis Web adalah model *prototype*. Berikut adalah penjelasan dari tahapan penelitian di atas.

3.2.1 Studi Literatur

Studi literatur dilakukan untuk mendapatkan referensi dari peneliti terdahulu dan kasus-kasus yang telah terjadi, tujuannya adalah untuk menjadi acuan dilakukan suatu penelitian. Penelitian ini membutuhkan referensi yang meliputi informasi penyakit pada anjing peliharaan, metode yang digunakan untuk mendiagnosis penyakit pada anjing yaitu metode *certainty factor*, cara menerapkan penelitian ke sistem berbasis web, dan hal-hal yang berhubungan dengan karya tulis ilmiah. Referensi tersebut didapatkan dari literatur berupa jurnal, buku, karya tulis, dan artikel. Hasil dari referensi yang didapat akan menjadi dasar penelitian ini dan menggambarkan pengembangan sistem yang akan dibuat.

3.2.2 Pengumpulan Data

Pengumpulan data dilakukan untuk mengumpulkan data yang menunjang dasar penelitian yang telah dilakukan, sumber data dapat diperoleh dari narasumber yang terkait maupun mengamati obyek secara langsung. Berikut adalah penjelasan mengenai beberapa metode pengumpulan data menurut W. Gulo (2008):

1. Pengamatan (Observasi)

Pengamatan (Observasi) adalah metode pengumpulan data di mana peneliti atau kolaboratornya mencatat informasi sebagaimana yang mereka saksikan selama penelitian. Penyaksian terhadap peristiwa-peristiwa itu bisa dengan melihat, mendengarkan, merasakan, yang kemudian dicatat seobyektif mungkin (hal. 116).

2. Survei

Survei adalah metode pengumpulan data dengan menggunakan instrumen untuk meminta tanggapan dari responden tentang sampel (hal. 118).

3. Wawancara

Wawancara adalah bentuk komunikasi langsung antara peneliti dan responden. Komunikasi berlangsung dalam bentuk tanya-jawab dalam hubungan tatap muka, sehingga gerak dan mimik responden merupakan pola media yang melengkapi kata-kata secara verbal (hal. 119).

4. Kuesioner (Angket)

Kuesioner atau angket hanya berbeda dalam bentuknya. Pada kuesioner, pertanyaan disusun dalam bentuk kalimat tanya, sedangkan pada angket, pertanyaan disusun dalam kalimat pernyataan dengan opsi jawaban yang tersedia. Kalau metode pengamatan dan metode wawancara menempatkan peneliti dalam hubungan langsung dengan responden, maka dalam metode angket hubungan itu dilakukan melalui media, yaitu daftar pertanyaan yang dikirim kepada responden (hal. 122).

5. Metode Dokumenter

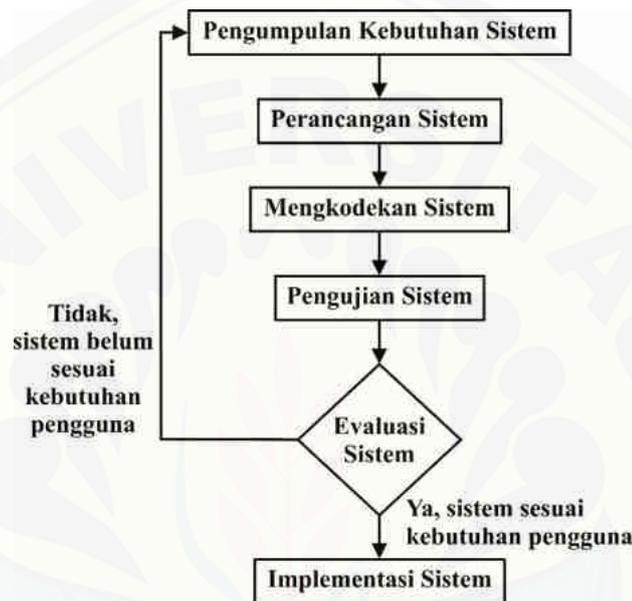
Dokumen adalah catatan tertulis tentang berbagai kegiatan peristiwa pada waktu lalu. (hal. 123).

Penelitian ini akan menggunakan metode pengumpulan data wawancara karena data yang diperoleh dengan metode ini lebih tepat sasaran karena peneliti berkomunikasi langsung dengan responden dalam bentuk tanya jawab. Data yang dikumpulkan harus berasal dari responden yang dapat dipercaya dan ahli dalam bidang/topik yang sedang diteliti. Penelitian ini mewawancarai responden yang memiliki pengetahuan lebih tentang anjing serta penyakit dan gejalanya, karena pertanyaan yang diajukan meliputi informasi penyakit pada anjing dan gejala timbul pada penyakit tersebut. Responden yang dipilih merupakan salah satu dari seorang dokter hewan, seseorang yang telah melakukan studi mendalam mengenai anjing, atau seseorang yang sudah hidup berdampingan dengan anjing dalam waktu yang cukup lama.

3.2.3 Pengembangan Sistem

Pengembangan sistem merupakan tahap untuk menggambarkan sistem yang sedang dikembangkan, dimulai dari mengumpulkan kebutuhan sistem hingga

sistem siap untuk diimplementasikan. Siklus hidup pengembangan perangkat lunak yang akan digunakan pada sistem pakar diagnosis penyakit pada anjing peliharaan menggunakan metode *certainty factor* berbasis web adalah model *prototype*. Siklus hidup pengembangan perangkat lunak model *prototype* dapat dilihat pada Gambar 3.2.



Gambar 3.2 Siklus hidup pengembangan perangkat lunak model *prototype*
(Sumber: Allent Kent & James G. Williams)

Gambar 3.2 menggambarkan tahapan dari siklus hidup pengembangan perangkat lunak model *prototype* yang terdiri dari pengumpulan kebutuhan sistem, perancangan sistem, mengkodekan sistem, pengujian sistem, evaluasi sistem dan implementasi sistem. *Prototyping* adalah membuat model kerja dari aplikasi basis data yang memungkinkan perancang dan pengguna untuk mengevaluasi perkembangan sistem, baik dari segi sistemnya sendiri ataupun dari fungsi yang dimilikinya (Irwansyah & V, 2014). Dengan model kerja tersebut maka pengembangan sistem akan lebih mudah karena hasil dari evaluasi perkembangan sistem oleh pengguna akan memudahkan pengembang dalam menentukan kebutuhan untuk sistem sehingga hasil akhir dari sistem akan sangat dekat sesuai

dengan keinginan pengguna. Berikut adalah tahapan penerapan model *prototype* pada pengembangan sistem pakar diagnosis penyakit pada anjing peliharaan menggunakan metode *certainty factor* berbasis web.

3.2.3.1 Pengumpulan Kebutuhan Sistem

Pengumpulan kebutuhan sistem/Analisis kebutuhan sistem adalah tahap pertama dari siklus hidup pengembangan perangkat lunak model *prototype*. Tahap ini mengumpulkan dan menganalisis kebutuhan dari pengguna sehingga pengembang mengetahui fitur-fitur yang perlu diterapkan ke sistem. Ada 2 kebutuhan yang diperlukan oleh sistem, yaitu kebutuhan fungsional dan kebutuhan non-fungsional. Berikut adalah pengertiannya:

1. Kebutuhan fungsional adalah kebutuhan yang meliputi proses yang berjalan dan data yang diproses di dalam sistem.
2. Kebutuhan non-fungsional adalah kebutuhan yang meliputi performa sistem dan kemudahan pengguna dalam menggunakan sistem.

3.2.3.2 Perancangan Sistem

Perancangan sistem/Desain sistem adalah tahap menggambarkan fitur dan proses yang akan berjalan pada perangkat lunak berdasarkan hasil analisis kebutuhan pada tahap pengumpulan kebutuhan sistem. Pengembangan perangkat lunak ini akan menggunakan metode *Unified Modeling Language* (UML) untuk menggambarkan fitur dan proses yang akan berjalan di dalamnya. Metode UML juga mendukung konsep *Object-Oriented Programming* (OOP) sehingga lebih mudah dalam mengembangkan perangkat lunak. Untuk menggambarkannya akan digunakan model desain *Business Process* dan *UML Diagram* yang terdiri dari *Use Case Diagram*, *Use Case Scenario*, *Activity Diagram*, *Sequence Diagram*, *Class Diagram*, dan *Entity Relationship Diagram*.

3.2.3.3 Mengkodekan Sistem

Mengkodekan Sistem adalah tahap penerapan kode program ke sistem berdasarkan desain yang telah digambarkan pada tahap perancangan sistem.

Pengembangan perangkat lunak ini akan menggunakan PHP (*Page Hyper Text Pre-Process*) sebagai bahasa pemrograman, *CodeIgniter* sebagai PHP *framework* dengan model MVC (*Model-View-Controller*) yang sesuai dengan konsep OOP, dan *MySQL* untuk manajemen basis data SQL (*Structured Query Language*).

3.2.3.4 Pengujian Sistem

Pengujian Sistem adalah tahap menguji perangkat lunak yang telah dikembangkan sebelum digunakan oleh pengguna. Tahap ini menguji apakah kode program berjalan dengan baik sehingga sistem dapat memenuhi kebutuhan pengguna. Pengembangan perangkat lunak ini akan menerapkan metode *White Box Testing* dan *Black Box Testing* sebagai metode pengujian sistem. Berikut adalah penjelasan dari kedua metode tersebut menurut Ade Djohar Maturidi (2014):

1. *White Box Testing*

Adalah meramalkan cara kerja perangkat lunak secara rinci, karena *logikal path* (jalur logika) perangkat lunak akan ditest dengan menyediakan *test case* yang akan mengerjakan kumpulan kondisi dan atau pengulangan secara spesifik (hal. 68). Jadi, *white box testing* menganalisis secara langsung kode program yang terdapat pada sistem. Pengujian *white box* pada pengembangan perangkat lunak ini akan menerapkan metode *Basis Path Testing* yang meliputi Listing Program, Diagram Alir, Perhitungan *Cyclomatic Complexity*, Basis Set, dan *Test Case*,

2. *Black Box Testing*

Bertujuan untuk menunjukkan fungsi PL tentang cara beroperasinya, apakah pemasukan data keluaran telah berjalan sebagaimana yang diharapkan dan apakah informasi yang disimpan secara eksternal selalu dijaga kemutakhirannya (hal. 68). Jadi, *black box testing* menjalankan sistem lalu mengamati apakah fungsionalitas sistem telah sesuai dengan kebutuhan pengguna.

3.2.3.5 Evaluasi Sistem

Evaluasi Sistem adalah tahap di mana pengguna mengambil keputusan untuk menerima perangkat lunak berdasarkan hasil dari tahap pengujian sistem. Jika hasil dari tahap pengujian sistem benar, maka pengembangan sistem dapat dilanjutkan ke tahap implementasi sistem. Tetapi jika hasil dari tahap pengujian sistem salah, maka pengembangan sistem kembali ke tahap pengumpulan kebutuhan sistem.

3.2.3.6 Implementasi Sistem

Implementasi Sistem adalah tahap akhir di mana perangkat lunak telah diterima oleh pengguna dan siap untuk digunakan.

3.3 Gambaran Sistem

Ada 2 pengguna sistem pakar diagnosis penyakit pada anjing peliharaan menggunakan metode *certainty factor* berbasis web, yaitu admin dan pengunjung. Admin adalah seseorang yang bertugas untuk mengelola data penyakit, gejala, dan aturan sistem pakar. Pengunjung adalah pemilik anjing peliharaan yang menggunakan sistem hanya untuk melakukan diagnosis penyakit anjing peliharaannya. Berikut adalah gambaran sistem pakar diagnosis penyakit pada anjing peliharaan menggunakan metode *certainty factor* berbasis web berdasarkan pengguna:

1. Admin

Admin *login* ke sistem berdasarkan *username* dan *password* yang dimilikinya. Setelah *login*, admin dapat mengelola data penyakit, gejala, dan aturan sistem pakar yang meliputi memasukkan data ke dalam *database*. Admin juga dapat mengubah dan menghapus data yang telah ada pada *database*. Setelah admin selesai mengelola data, maka admin dapat keluar dari sistem menggunakan *logout*.

2. Pengunjung (Pemilik Anjing Peliharaan)

Pengunjung tidak perlu sebuah akun untuk dapat menggunakan sistem. Pengunjung hanya perlu pergi ke halaman web sistem pakar diagnosis

penyakit anjing untuk mengetahui penyakit yang menyerang anjing peliharaan mereka. Di halaman web sistem, pengunjung harus memilih *checkbox* daftar gejala penyakit anjing dan mengisi bobot keyakinan gejala penyakit berdasarkan pengamatan terhadap kondisi anjing peliharaannya dengan ketentuan pilihan berupa Sangat Yakin, Yakin, Cukup Yakin, Kurang Yakin, dan Tidak Yakin. Setelah mengisi semua bobot keyakinan sesuai dengan gejalanya, maka pengunjung diharuskan menekan tombol Proses agar sistem dapat memulai perhitungan. Sistem lalu mencocokkan gejala penyakit dari pengunjung dengan daftar penyakit yang ada di dalam sistem. Apabila gejala penyakit yang dipilih oleh pengunjung sesuai dengan penyakit yang ada di sistem, maka sistem akan melakukan perhitungan menggunakan metode *certainty factor* berdasarkan kedua data tersebut. Jika gejala penyakit yang dipilih oleh pengunjung ternyata sesuai dengan lebih dari satu penyakit yang ada di sistem, maka masing-masing data penyakit yang sesuai akan dihitung dengan gejala penyakit yang dipilih oleh pengunjung. Setelah ditemukan nilai keyakinan diagnosis oleh sistem berupa persentase penyakit yang menyerang anjing peliharaan, maka sistem akan menampilkan persentase penyakit tersebut beserta informasinya. Jika data penyakit yang dihitung lebih dari satu, maka sistem akan mengurutkan semua hasil perhitungan dari besar ke kecil lalu mengambil perhitungan terbesar untuk ditampilkan ke pengunjung sebagai hasil diagnosis penyakit.

BAB 4. PENGEMBANGAN SISTEM

Bab ini berisi pembahasan mengenai pengembangan sistem pakar diagnosis penyakit pada anjing peliharaan menggunakan metode *certainty factor* berbasis web. Pengembangan sistem ini akan menggunakan model *prototype* yang terdiri dari pengumpulan kebutuhan sistem, perancangan sistem, mengkodekan sistem, pengujian sistem, evaluasi sistem, dan implementasi sistem.

4.1 Pengumpulan Kebutuhan Sistem

Tahap ini merupakan tahap awal dari pengembangan perangkat lunak menggunakan model *prototype*. Tahap ini mengumpulkan dan menganalisis kebutuhan pengguna berdasarkan data yang dikumpulkan sehingga pengembang mengetahui fitur-fitur yang perlu diterapkan pada sistem. Pengumpulan data pada penelitian ini dilakukan dengan studi literatur dan wawancara. Data yang diperoleh dari pengumpulan data tersebut adalah data penyakit anjing, data gejalanya, dan data bobot keyakinan terjadinya gejala penyakit pada anjing tersebut. Jurnal, buku, karya tulis, dan artikel dapat dijadikan referensi studi literatur untuk memperoleh data tersebut. Wawancara dilakukan kepada seorang responden yang dinilai ahli dalam hal yang berhubungan dengan data yang dibutuhkan. Seorang responden dapat dikatakan ahli apabila memiliki salah satu atau lebih dari kategori berikut:

1. Seorang dokter hewan
2. Seseorang yang pernah melakukan studi atau penelitian mengenai anjing
3. Seseorang yang hidup berdampingan dengan anjing dalam waktu yang cukup lama.

Responden yang dipilih adalah seorang dokter hewan yang hidup berdampingan dengan anjing dalam waktu yang cukup lama bernama drh. Nova Anggraini. Wawancara kepada responden mendapatkan hasil berupa data penyakit pada anjing, data gejalanya, dan data bobot keyakinan terjadinya gejala penyakit. Pengumpulan data tersebut mengambil 4 jenis penyakit anjing, yaitu *Canine distemper*, *Infectious canine hepatitis*, *Canine parvovirus*, *Roundworm*. Data tersebut disusun dalam sebuah data set bernama Data Set Pembobotan Gejala

Penyakit yang terdiri dari kolom Nomor, Penyakit, Gejala, dan Bobot Gejala. Data Set Pembobotan Gejala Penyakit dapat dilihat pada Tabel 4.1.

Tabel 4.1 Data Set Pembobotan Gejala Penyakit

No.	Penyakit	Gejala	Bobot Gejala Penyakit
1.	<i>Canine distemper</i>	Batuk	0.7
		Cuping hidung kering bahkan mengelupas	0.7
		Demam	0.8
		Diare	0.5
		Kedutan	0.6
		Gelisah	0.8
		Kotoran mata kehijauan	0.8
		Lesu	0.8
		Muntah	0.5
		Pilek kehijauan	0.8
2.	<i>Infectious canine hepatitis</i>	Demam	0.8
		Lesu	0.7
		Nyeri di perut	0.65
		Selaput lendir kuning	0.8
		Tidak mau makan	0.8
3.	<i>Canine parvovirus</i>	Demam	0.9
		Diare berdarah	0.8
		Lesu	0.9
		Muntah	0.9
		Pucat	0.8
		Tidak mau makan	0.9
4.	<i>Roundworm</i>	Batuk	0.6
		Diare	0.8
		Kurus	0.7
		Lesu	0.7
		Perut buncit seperti busung lapar	0.60
		Pucat	0.65

Sumber: Hasil analisis dari studi literatur dan wawancara kepada responden (2018).

Berdasarkan kumpulan data tersebut, maka kebutuhan pengguna terhadap sistem dapat dianalisis. Ada 2 kategori kebutuhan yang diperlukan oleh sistem, yaitu kebutuhan fungsional dan kebutuhan non-fungsional. Kebutuhan fungsional adalah kebutuhan yang meliputi proses yang berjalan dan data yang diproses di

dalam sistem, sedangkan kebutuhan non-fungsional adalah kebutuhan yang meliputi performa sistem dan kemudahan pengguna dalam menggunakan sistem.

4.1.1 Kebutuhan Fungsional

Berikut adalah kebutuhan fungsional yang diperlukan oleh sistem:

1. Sistem mampu mengelola data penyakit.
2. Sistem mampu mengelola data gejala penyakit.
3. Sistem mampu mengelola aturan sistem pakar.
4. Sistem mampu memproses perhitungan metode *certainty factor*.

4.1.2 Kebutuhan Non-fungsional

Berikut adalah kebutuhan non-fungsional yang diperlukan oleh sistem:

1. Sistem memiliki tampilan dan bahasa yang mudah dipahami oleh pengguna.
2. Sistem dapat diakses 24 jam dalam sehari.
3. Sistem dapat diakses bersamaan oleh banyak pengguna.
4. Sistem memiliki autentikasi hak akses untuk admin berupa *username* dan *password* demi keamanan sistem.

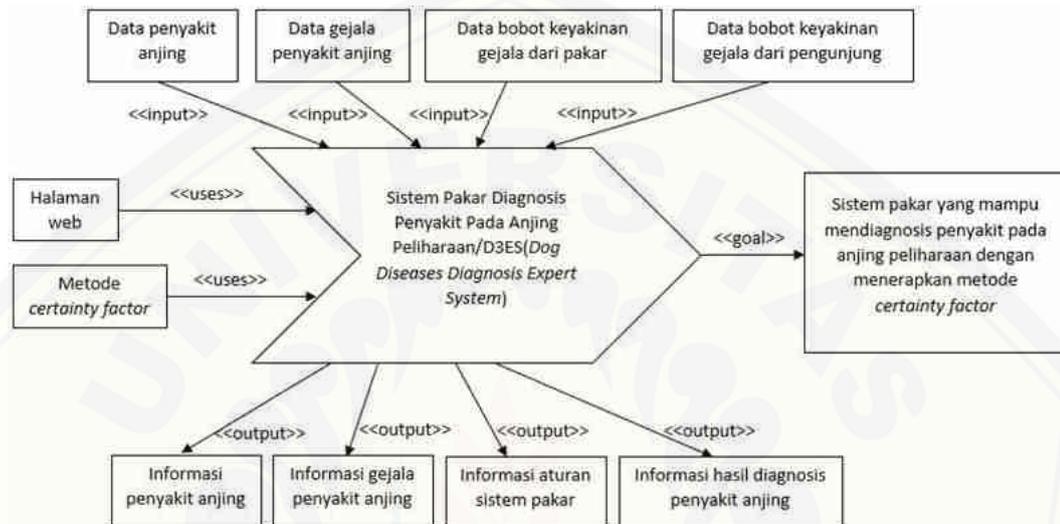
4.2 Perancangan Sistem

Tahap ini merupakan tahap memodelkan atau mendesain sistem yang dikembangkan berdasarkan kumpulan kebutuhan sistem yang dianalisis pada tahap sebelumnya. Perancangan sistem pada pengembangan sistem ini akan menerapkan metode UML untuk memodelkan fitur atau proses yang akan berjalan pada sistem. Perancangan sistem akan menggunakan model desain *Business Process*, *Use Case Diagram*, *Use Case Scenario*, *Activity Diagram*, *Sequence Diagram*, *Class Diagram*, dan *Entity Relationship Diagram*.

4.2.1 *Business Process*

Business process merupakan gambaran umum dari sistem yang dikembangkan. *Business process* memiliki beberapa komponen untuk menggambarkan sistem yang meliputi data masukan ke sistem (*input*), data

keluaran yang dihasilkan dari data masukan yang telah diproses (*output*), media yang digunakan untuk memproses data masukan (*uses*), dan tujuan yang ingin dicapai oleh sistem (*goal*). *Business process* dari sistem yang dikembangkan dapat dilihat pada Gambar 4.1.



Gambar 4.1 *Business process* sistem

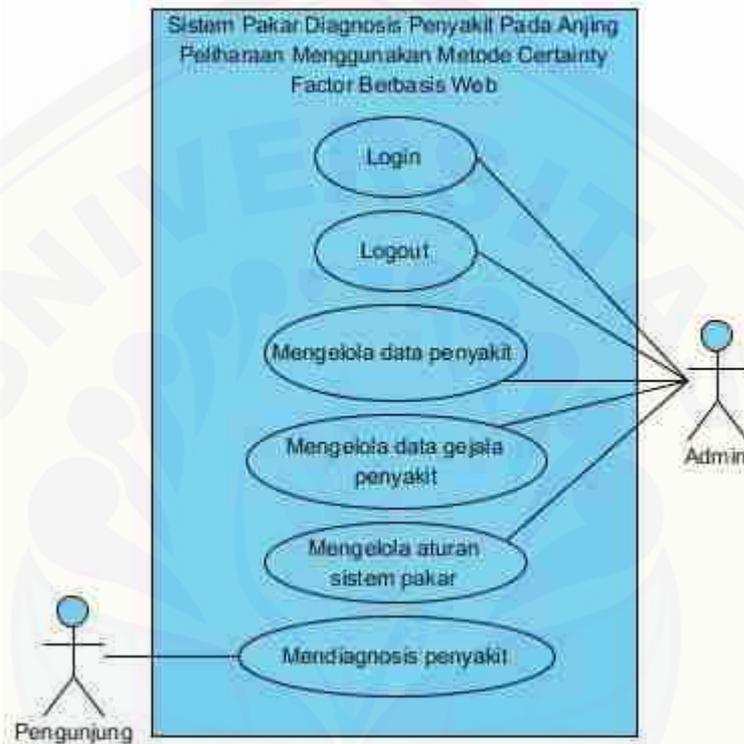
(Sumber: Hasil analisis, 2018)

Gambar 4.1 menggambarkan *business process* sistem pakar diagnosis penyakit pada anjing peliharaan yang terdiri dari 4 *input*, 2 *uses*, 4 *output*, dan 1 *goal*. *Input* ke sistem adalah data penyakit anjing, data gejala penyakit anjing, data bobot keyakinan gejala dari pakar dan data bobot keyakinan gejala dari pengunjung. *Input* tersebut akan diproses dalam sistem berbasis web dan dihitung menggunakan metode *certainty factor*. *Output* dari sistem adalah informasi penyakit anjing, informasi gejala penyakit anjing, informasi aturan sistem pakar, dan informasi hasil diagnosis penyakit anjing. *Goal* dari pengembangan sistem adalah sistem pakar yang mampu mendiagnosis penyakit pada anjing peliharaan dengan menerapkan metode *certainty factor*.

4.2.2 Use Case Diagram

Use case diagram merupakan gambaran interaksi antara pengguna dan sistem. Melalui *use case diagram* dapat diketahui fitur sistem apa saja yang dapat

digunakan oleh pengguna sesuai dengan hak aksesnya masing-masing. Komponen yang terdapat pada *use case diagram* adalah aktor dan *use case* yang terdapat pada sistem. *Use case diagram* dari sistem yang dikembangkan dapat dilihat pada Gambar 4.2.



Gambar 4.2 *Use case diagram* sistem
(Sumber: Hasil analisis, 2018)

Gambar 4.2 menggambarkan *use case diagram* sistem pakar diagnosis penyakit pada anjing peliharaan yang terdiri dari 2 aktor yaitu admin dan pengunjung, serta 6 *use case* yaitu *login*, *logout*, mengelola data penyakit, mengelola data gejala penyakit, mengelola aturan sistem pakar, dan mendiagnosis penyakit. Aktor admin dapat mengakses fitur *login*, *logout*, mengelola data penyakit, mengelola data gejala penyakit, dan mengelola aturan sistem pakar. Sedangkan aktor pengunjung hanya dapat mengakses fitur mendiagnosis penyakit. Definisi aktor dapat dilihat pada Tabel 4.2, sedangkan definisi dari *use case* dapat dilihat pada Tabel 4.3.

Tabel 4.2 Definisi aktor

No.	Aktor	Definisi
1	Admin	Admin adalah aktor yang memiliki hak akses untuk mengelola (melihat, memasukkan, mengubah, dan menghapus) data penyakit, data gejala penyakit, dan aturan sistem pakar, sehingga admin memerlukan hak akses <i>login</i> dan <i>logout</i> pada sistem demi keamanan data.
2	Pengunjung	Pengunjung adalah aktor yang tidak memiliki hak akses untuk mengelola data pada sistem sehingga tidak memerlukan hak akses <i>login</i> dan <i>logout</i> pada sistem. Pengunjung hanya dapat mendiagnosis penyakit anjing peliharaannya.

Sumber: Hasil analisis (2018).

Tabel 4.3 Definisi *use case*

No.	<i>Use case</i>	Definisi
1.	<i>Login</i>	Fitur masuk ke sistem sebagai admin untuk mengelola data di dalam sistem. Fitur hanya dapat diakses oleh admin.
2.	<i>Logout</i>	Fitur keluar dari sistem sebagai admin setelah admin menyelesaikan pengelolaan data di dalam sistem. Fitur hanya dapat diakses oleh admin.
3.	Mengelola data penyakit	Fitur untuk mengelola data penyakit meliputi melihat data penyakit, memasukkan data penyakit baru, mengubah dan menghapus data penyakit yang telah ada pada <i>database</i> sistem. Fitur hanya dapat diakses oleh admin.
4.	Mengelola data gejala penyakit	Fitur untuk mengelola data gejala penyakit meliputi melihat data gejala penyakit, memasukkan data gejala penyakit baru, mengubah dan menghapus data gejala penyakit yang telah ada pada <i>database</i> sistem. Fitur hanya dapat diakses oleh admin.
5.	Mengelola aturan sistem pakar	Fitur untuk mengelola aturan sistem pakar meliputi melihat data aturan memasukkan data aturan baru, mengubah dan menghapus data aturan yang telah ada pada <i>database</i> sistem.. Fitur merepresentasikan pengetahuan pakar mengenai hubungan antara penyakit anjing dan gejalanya menjadi aturan yang dapat dibaca dan diproses oleh sistem. Admin membuat aturan dan memasukkan data bobot keyakinan gejala dari pakar berdasarkan gejala dan penyakitnya. Fitur hanya dapat diakses oleh admin.
6.	Mendiagnosis penyakit	Fitur untuk mendiagnosis penyakit yang menerapkan metode <i>certainty factor</i> berdasarkan data yang dimasukkan oleh pengunjung dan data yang berada pada <i>database</i> sistem. Data yang dimasukkan oleh pengunjung adalah pilihan gejala penyakit dan bobot keyakinan gejalanya. Hasil diagnosis akan ditampilkan oleh sistem sebagai jawaban dari permasalahan yang diajukan oleh pengunjung. Hasil diagnosis hanya digunakan dalam sekali proses sehingga tidak perlu disimpan pada <i>database</i> sistem. Fitur hanya dapat diakses oleh pengunjung.

Sumber: Hasil analisis (2018).

4.2.3 Use Case Scenario

Use case scenario merupakan penjelasan secara rinci mengenai skenario/kemungkinan alur yang terjadi pada *use case*. Komponen yang terdapat

pada *use case scenario* adalah, nama *use case*, aktor, *entry condition*, *exit condition*, skenario normal, dan skenario alternatif(bila diperlukan). Berdasarkan *use case diagram*, sistem pakar diagnosis penyakit pada anjing peliharaan memiliki 6 *use case* yaitu *login*, *logout*, mengelola data penyakit, mengelola data gejala penyakit, mengelola aturan sistem pakar, dan mendiagnosis penyakit.

4.2.3.1 Use Case Scenario Login

Penjelasan secara rinci mengenai kemungkinan skenario dari aksi aktor dan reaksi sistem pada *use case login* dapat dilihat pada Tabel 4.4 dan untuk lebih lengkapnya dapat dilihat pada Lampiran A nomor 1.

Tabel 4.4 Use case scenario login

Nama Use Case	<i>Login</i>
Aktor	Admin
Entry Condition	Aktor memiliki <i>username</i> dan <i>password</i> dan ingin masuk ke dalam sistem
Exit Condition	Aktor mendapatkan status <i>login</i> dan berada pada Halaman <i>Dashboard</i>
Aksi Aktor	Reaksi Sistem
SKENARIO NORMAL	
1. Aktor mengakses Halaman <i>Login</i> Admin	2. Sistem menampilkan Halaman <i>Login</i> Admin, berisi <i>Form Login</i> yang terdiri dari: - <i>Textfield Username</i> - <i>Textfield Password</i> - Tombol <i>Login</i>

Sumber: Hasil analisis (2018).

4.2.3.2 Use Case Scenario Logout

Penjelasan secara rinci mengenai kemungkinan skenario dari aksi aktor dan reaksi sistem pada *use case logout* dapat dilihat pada Tabel 4.5 dan untuk lebih lengkapnya dapat dilihat pada Lampiran A nomor 2.

Tabel 4.5 Use case scenario logout

Nama Use Case	<i>Logout</i>
Aktor	Admin
Entry Condition	Aktor memiliki status <i>login</i> dan ingin keluar dari sistem
Exit Condition	Aktor kehilangan status <i>login</i> dan berada pada Halaman <i>Login Admin</i>
Aksi Aktor	Reaksi Sistem
SKENARIO NORMAL	
<i>Logout dari Halaman Dashboard</i>	
1. Aktor berada pada Halaman <i>Dashboard</i>	
2. Aktor menekan Tombol <i>Logout</i> yang berada pada bagian kanan atas Halaman <i>Dashboard</i>	
	3. Sistem memproses permintaan <i>logout</i> aktor
	4. Sistem menghilangkan status <i>login</i> aktor dan menampilkan Halaman <i>Login Admin</i> , berisi <i>Form Login</i> yang terdiri dari: <ul style="list-style-type: none"> - <i>Text field Username</i> - <i>Text field Password</i> - Tombol <i>Login</i>

Sumber: Hasil analisis (2018).

4.2.3.3 Use Case Scenario Mengelola Data Penyakit

Penjelasan secara rinci mengenai kemungkinan skenario dari aksi aktor dan reaksi sistem pada *use case mengelola data penyakit* dapat dilihat pada Tabel 4.6 dan untuk lebih lengkapnya dapat dilihat pada Lampiran A nomor 3.

Tabel 4.6 *Use case scenario* mengelola data penyakit

Nama Use Case	Mengelola Data Penyakit
Aktor	Admin
Entry Condition	Aktor memiliki status <i>login</i> dan ingin mengelola data penyakit
Exit Condition	Aktor berhasil mengelola data penyakit
Aksi Aktor	Reaksi Sistem
SKENARIO NORMAL	
1. Aktor berada pada Halaman <i>Dashboard</i>	
2. Aktor memilih Menu Penyakit	
	3. Sistem menampilkan <i>Dropdown</i> Submenu, berisi: - Submenu Kelola Data
4. Aktor memilih Submenu Kelola Data	
	5. Sistem menampilkan Halaman Kelola Data Penyakit, berisi: - Tombol Masukkan Data - Tombol Ubah Data - Tombol Hapus Data - Tombol Kembali - Tombol <i>Logout</i>

Sumber: Hasil analisis (2018).

4.2.3.4 Use Case Scenario Mengelola Data Gejala Penyakit

Penjelasan secara rinci mengenai kemungkinan skenario dari aksi aktor dan reaksi sistem pada *use case data gejala penyakit* dapat dilihat pada Tabel 4.7 dan untuk lebih lengkapnya dapat dilihat pada Lampiran A nomor 4.

Tabel 4.7 Use case scenario mengelola data gejala penyakit

Nama Use Case	Mengelola Data Gejala Penyakit
Aktor	Admin
Entry Condition	Aktor memiliki status <i>login</i> dan ingin mengelola data gejala
Exit Condition	Aktor berhasil mengelola data gejala
Aksi Aktor	Reaksi Sistem
SKENARIO NORMAL	
1. Aktor berada pada Halaman <i>Dashboard</i>	
2. Aktor memilih Menu Gejala	
	3. Sistem menampilkan <i>Dropdown</i> Submenu, berisi: - Submenu Kelola Data
4. Aktor memilih Submenu Kelola Data	
	5. Sistem menampilkan Halaman Kelola Data Gejala, berisi: - Tombol Masukkan Data - Tombol Ubah Data - Tombol Hapus Data - Tombol Kembali - Tombol <i>Logout</i> - Tabel Data Gejala yang terdiri dari Kolom No dan Gejala

Sumber: Hasil analisis (2018).

4.2.3.5 Use Case Scenario Mengelola Aturan Sistem pakar

Penjelasan secara rinci mengenai kemungkinan skenario dari aksi aktor dan reaksi sistem pada *use case* mengelola aturan sistem pakar dapat dilihat pada Tabel 4.8 dan untuk lebih lengkapnya dapat dilihat pada Lampiran A nomor 5.

Tabel 4.8 *Use case scenario* mengelola aturan sistem pakar

Nama Use Case	Mengelola Aturan Sistem Pakar
Aktor	Admin
Entry Condition	Aktor memiliki status <i>login</i> dan ingin mengelola data aturan
Exit Condition	Aktor berhasil mengelola data aturan
Aksi Aktor	Reaksi Sistem
SKENARIO NORMAL	
1. Aktor berada pada Halaman <i>Dashboard</i>	
2. Aktor memilih Menu Aturan	
	3. Sistem menampilkan <i>Dropdown</i> Submenu, berisi: - Submenu Kelola Data
4. Aktor memilih Submenu Kelola Data	
	5. Sistem menampilkan Halaman Kelola Data Aturan, berisi: - Tombol Masukkan Data - Tombol Ubah Data - Tombol Hapus Data - Tombol Kembali - Tombol <i>Logout</i> - Tabel Data Aturan yang terdiri dari Kolom No, Penyakit, Gejala, dan Bobot Keyakinan

Sumber: Hasil analisis (2018).

4.2.3.6 Use Case Scenario Mendiagnosis Penyakit

Penjelasan secara rinci mengenai kemungkinan skenario dari aksi aktor dan reaksi sistem pada *use case mendiagnosis penyakit* dapat dilihat pada Tabel 4.9 dan untuk lebih lengkapnya dapat dilihat pada Lampiran A nomor 6.

Tabel 4.9 Use case scenario mendiagnosis penyakit

Nama Use Case	Mendiagnosis Penyakit
Aktor	Pengunjung
Entry Condition	Aktor ingin mengetahui kondisi kesehatan anjing peliharaannya
Exit Condition	Aktor mendapatkan hasil diagnosis penyakit anjing peliharaannya
Aksi Aktor	Reaksi Sistem
SKENARIO NORMAL	
1. Aktor mengakses Halaman Pengunjung	2. Sistem menampilkan halaman Pengunjung, berisi: - Menu Home - Menu Diagnosis
3. Aktor memilih Menu Diagnosis	4. Sistem menampilkan halaman Diagnosis, berisi: - Checkbox Gejala Penyakit - Tombol Selanjutnya
5. Aktor memilih <i>Checkbox</i> Gejala Penyakit	
6. Aktor menekan Tombol Selanjutnya	

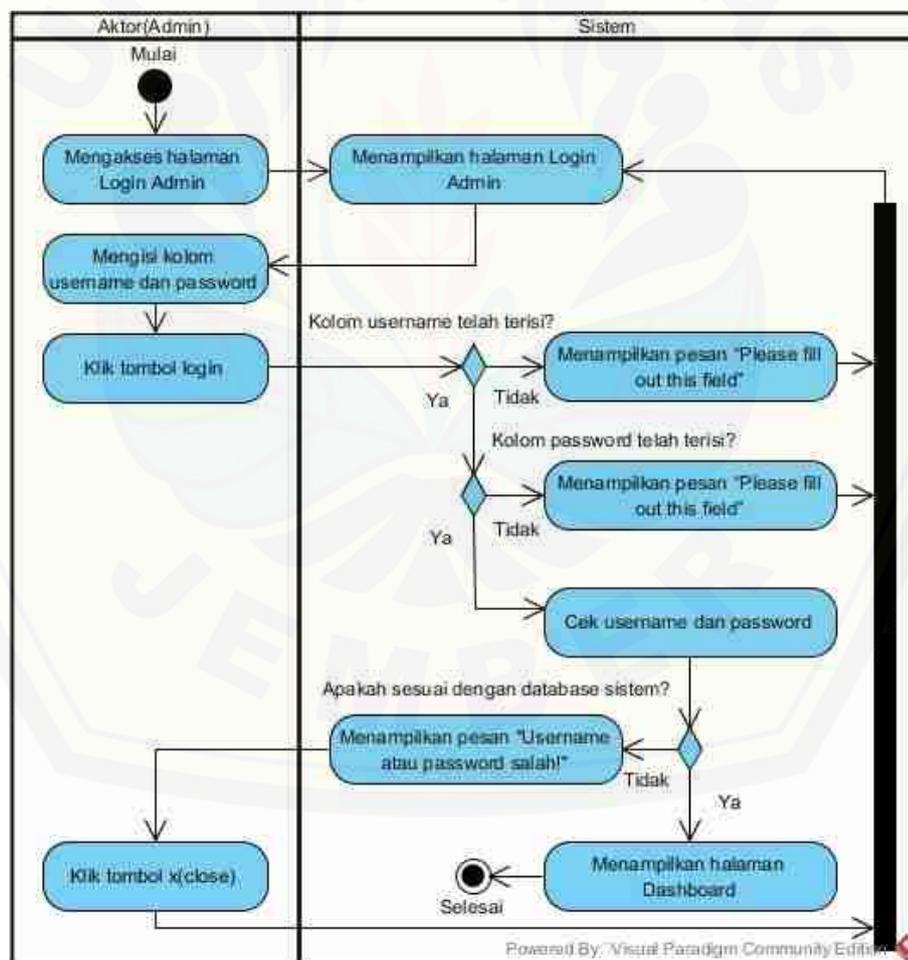
Sumber: Hasil analisis (2018).

4.2.4 Activity Diagram

Activity Diagram merupakan gambaran aktivitas yang terjadi di dalam sistem. Menampilkan langkah-langkah proses kerja sistem. Berdasarkan *use case diagram*, sistem pakar diagnosis penyakit pada anjing peliharaan memiliki 6 *use case* yaitu *login*, *logout*, mengelola data penyakit, mengelola data gejala penyakit, mengelola aturan sistem pakar, dan mendiagnosis penyakit.

4.2.4.1 Activity Diagram Login

Gambaran mengenai langkah-langkah proses *login* pada sistem dapat dilihat pada Gambar 4.3.



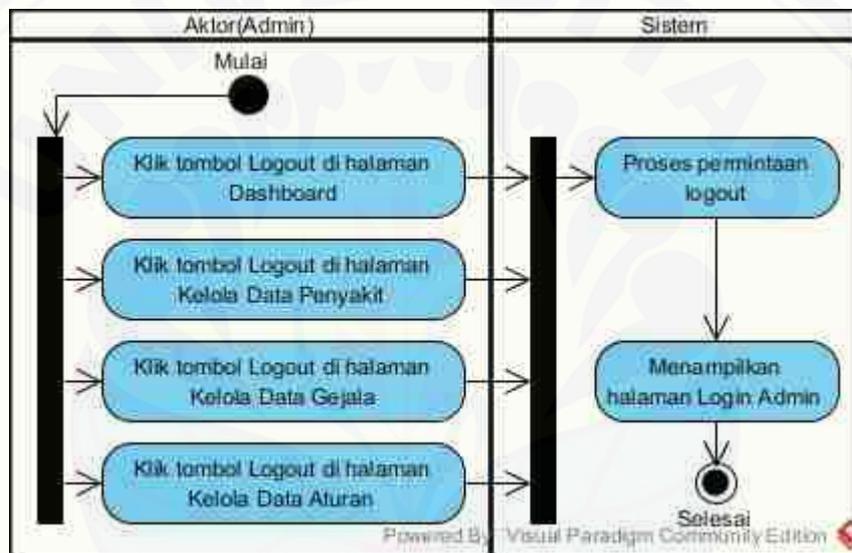
Gambar 4.3 Activity diagram login

(Sumber: Hasil analisis, 2018)

Gambar 4.3 menggambarkan *activity diagram login* sistem pakar diagnosis penyakit pada anjing peliharaan yang terdiri dari langkah-langkah yang diawali dengan aktor mengakses halaman *Login Admin* sampai sistem menampilkan halaman *Dashboard*.

4.2.4.2 Activity Diagram Logout

Gambaran mengenai langkah-langkah proses *logout* pada sistem dapat dilihat pada Gambar 4.4.



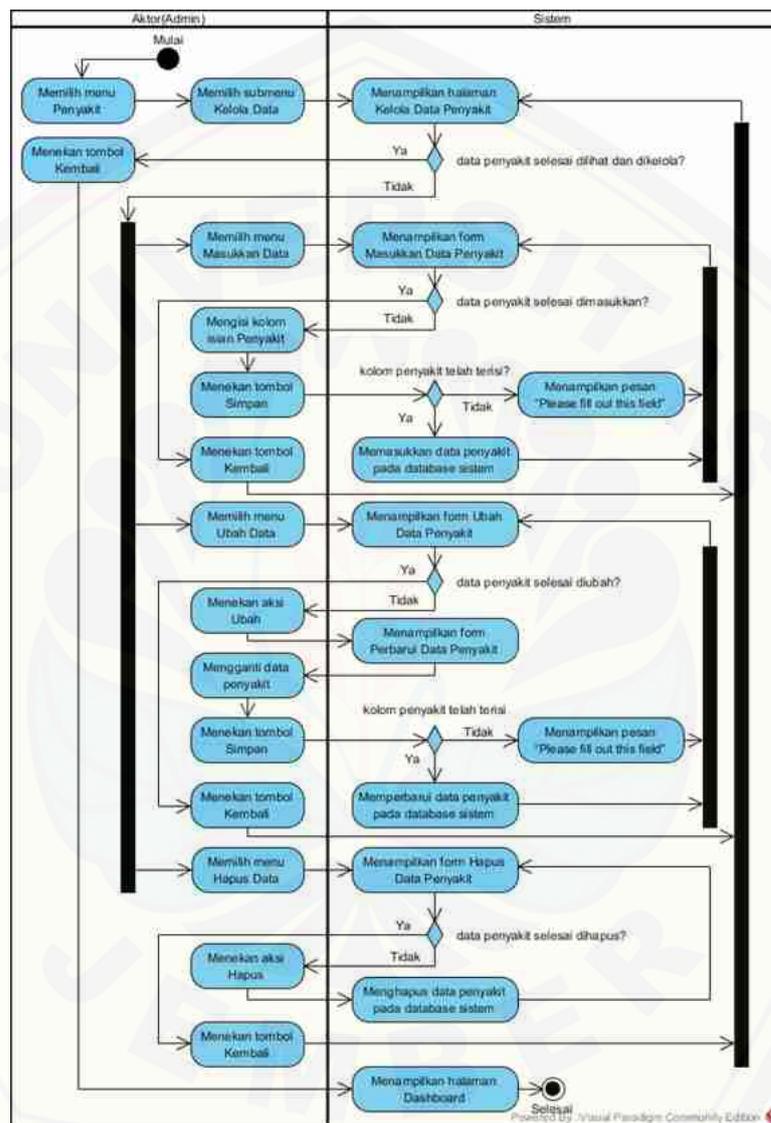
Gambar 4.4 Activity diagram logout

(Sumber: Hasil analisis, 2018)

Gambar 4.4 menggambarkan *activity diagram logout* sistem pakar diagnosis penyakit pada anjing peliharaan yang terdiri dari langkah-langkah yang diawali dengan aktor menekan tombol *Logout* di halaman *Dashboard*, halaman *Kelola Data Penyakit*, halaman *Kelola Data Gejala*, atau halaman *Kelola Data Aturan* sampai sistem menampilkan halaman *Login Admin*.

4.2.4.3 Activity Diagram Mengelola Data Penyakit

Gambaran mengenai langkah-langkah proses mengelola data penyakit pada sistem dapat dilihat pada Gambar 4.5.



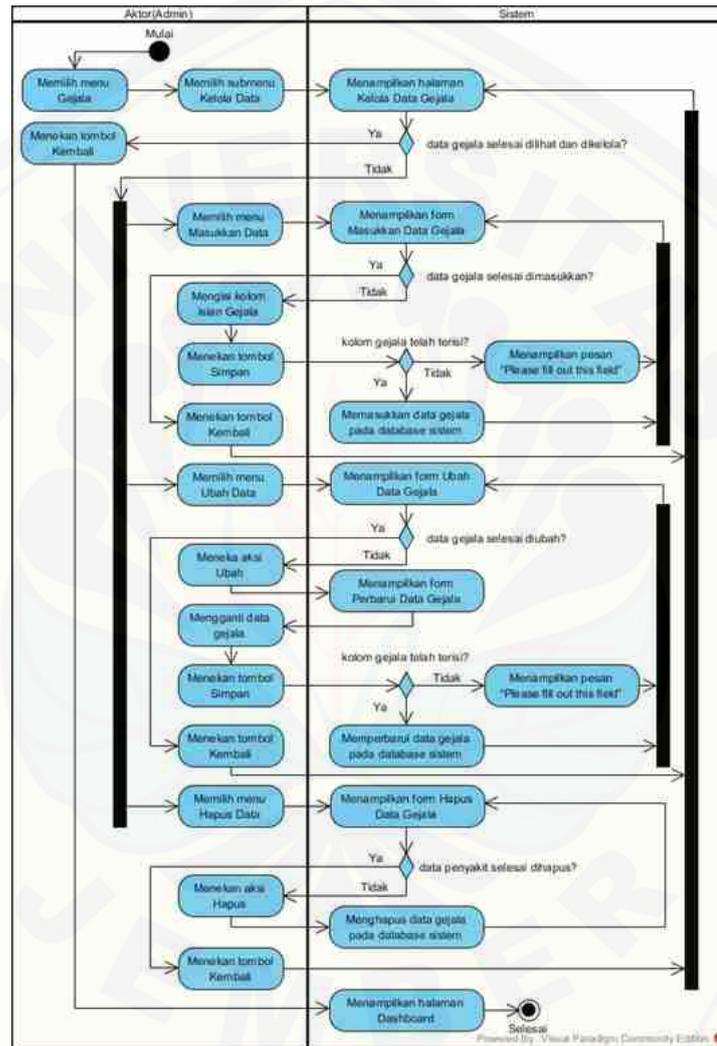
Gambar 4.5 Activity diagram mengelola data penyakit

(Sumber: Hasil analisis, 2018)

Gambar 4.5 menggambarkan *activity diagram* mengelola data penyakit sistem pakar diagnosis penyakit pada anjing peliharaan yang terdiri dari langkah-langkah yang diawali dengan aktor memilih menu Penyakit sampai sistem menampilkan halaman *Dashboard*.

4.2.4.4 Activity Diagram Mengelola Data Gejala Penyakit

Gambaran mengenai langkah-langkah proses mengelola data gejala penyakit pada sistem dapat dilihat pada Gambar 4.6.



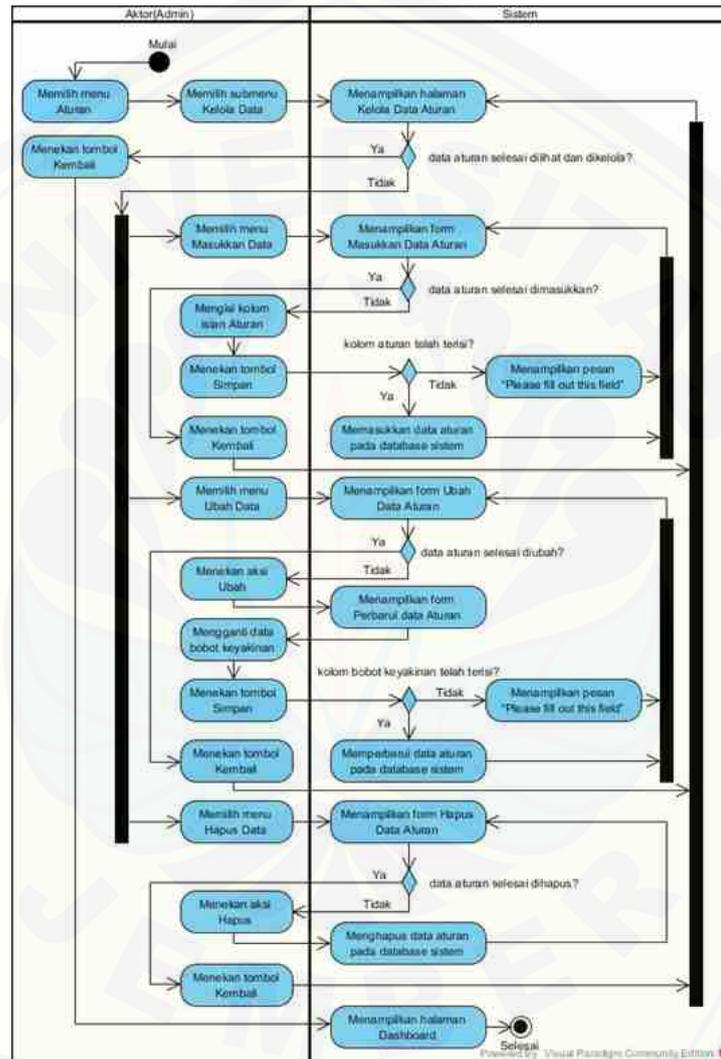
Gambar 4.6 Activity diagram mengelola data gejala penyakit

(Sumber: Hasil analisis, 2018)

Gambar 4.6 menggambarkan *activity diagram* mengelola data gejala penyakit sistem pakar diagnosis penyakit pada anjing peliharaan yang terdiri dari langkah-langkah yang diawali dengan aktor memilih menu Gejala sampai sistem menampilkan halaman *Dashboard*.

4.2.4.5 Activity Diagram Mengelola Aturan Sistem Pakar

Gambaran mengenai langkah-langkah proses mengelola aturan sistem pakar pada sistem dapat dilihat pada Gambar 4.7.



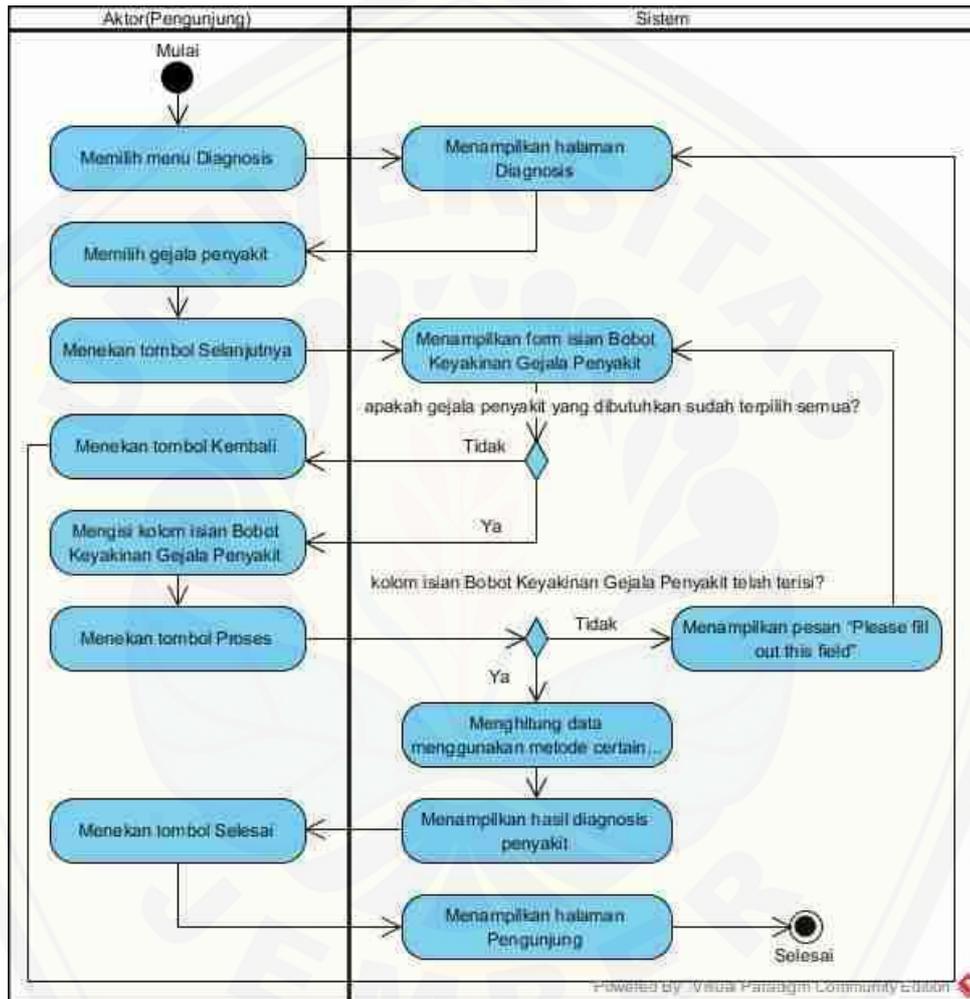
Gambar 4.7 Activity diagram mengelola aturan sistem pakar

(Sumber: Hasil analisis, 2018)

Gambar 4.7 menggambarkan *activity diagram* mengelola aturan sistem pakar diagnosis penyakit pada anjing peliharaan yang terdiri dari langkah-langkah yang diawali dengan aktor memilih menu Aturan sampai sistem menampilkan halaman *Dashboard*.

4.2.4.6 Activity Diagram Mendiagnosis Penyakit

Gambaran mengenai langkah-langkah proses mendiagnosis penyakit pada sistem dapat dilihat pada Gambar 4.8.



Gambar 4.8 Activity diagram mendiagnosis penyakit

(Sumber: Hasil analisis, 2018)

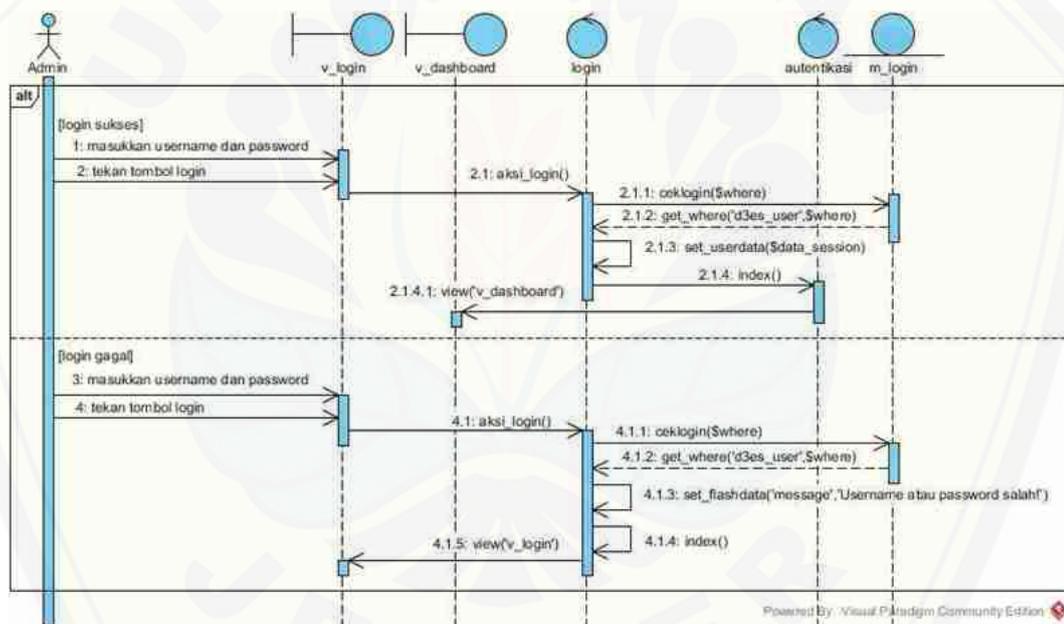
Gambar 4.8 menggambarkan *activity diagram* mendiagnosis penyakit sistem pakar diagnosis penyakit pada anjing peliharaan yang terdiri dari langkah-langkah yang diawali dengan aktor mengakses halaman Pengunjung sampai sistem selesai mendiagnosis penyakit dan menampilkan kembali halaman pengunjung.

4.2.5 Sequence Diagram

Sequence Diagram merupakan gambaran interaksi dan pesan-pesan yang dikirim antar objek dalam sebuah proses di sistem. Berdasarkan *use case diagram*, sistem pakar diagnosis penyakit pada anjing peliharaan memiliki 6 *use case*/proses yaitu *login*, *logout*, mengelola data penyakit, mengelola data gejala penyakit, mengelola aturan sistem pakar, dan mendiagnosis penyakit.

4.2.5.1 Sequence Diagram Login

Gambaran mengenai interaksi dan pesan-pesan yang dikirim antar objek dalam proses *login* dapat dilihat pada Gambar 4.9.



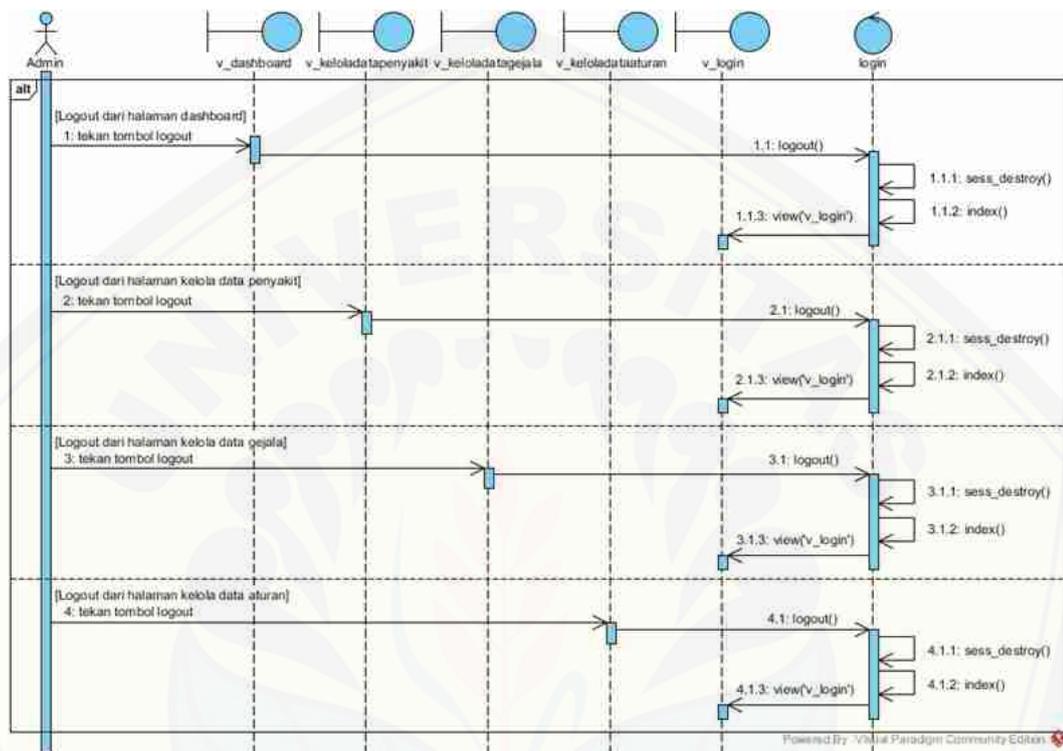
Gambar 4.9 Sequence diagram login

(Sumber: Hasil analisis, 2018)

Gambar 4.9 menggambarkan *sequence diagram login* sistem pakar diagnosis penyakit pada anjing peliharaan dengan komponen 1 aktor, 2 kelas *view*, 2 kelas *controller*, dan 1 kelas *model*. Aktor adalah Admin, kelas *view* terdiri dari *v_login* dan *v_dashboard*, kelas *controller* terdiri dari *login* dan *autentikasi*, dan kelas *model* terdiri dari *m_login*.

4.2.5.2 Sequence Diagram Logout

Gambaran mengenai interaksi dan pesan-pesan yang dikirim antar objek dalam proses *logout* dapat dilihat pada Gambar 4.10.



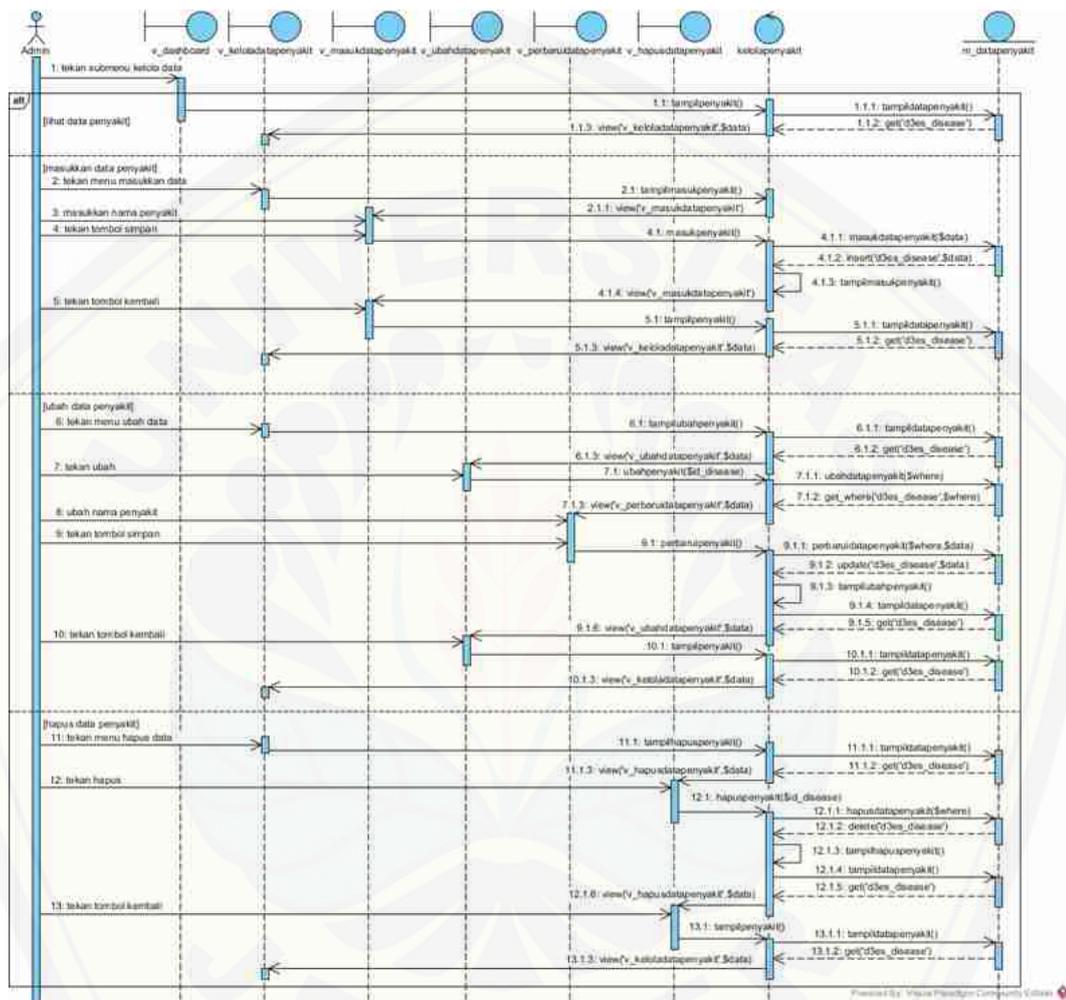
Gambar 4.10 Sequence diagram logout

(Sumber: Hasil analisis, 2018)

Gambar 4.10 menggambarkan *sequence diagram logout* sistem pakar diagnosis penyakit pada anjing peliharaan dengan komponen 1 aktor, 4 kelas *view*, dan 1 kelas *controller*. Aktor adalah Admin, kelas *view* terdiri dari *v_dashboard*, *v_keloladatapenyakit*, *v_keloladatagejala*, *v_keloladataaturan*, dan *v_login*, kelas *controller* terdiri dari *login*.

4.2.5.3 Sequence Diagram Mengelola Data Penyakit

Gambaran mengenai interaksi dan pesan-pesan yang dikirim antar objek dalam proses mengelola data penyakit dapat dilihat pada Gambar 4.11.



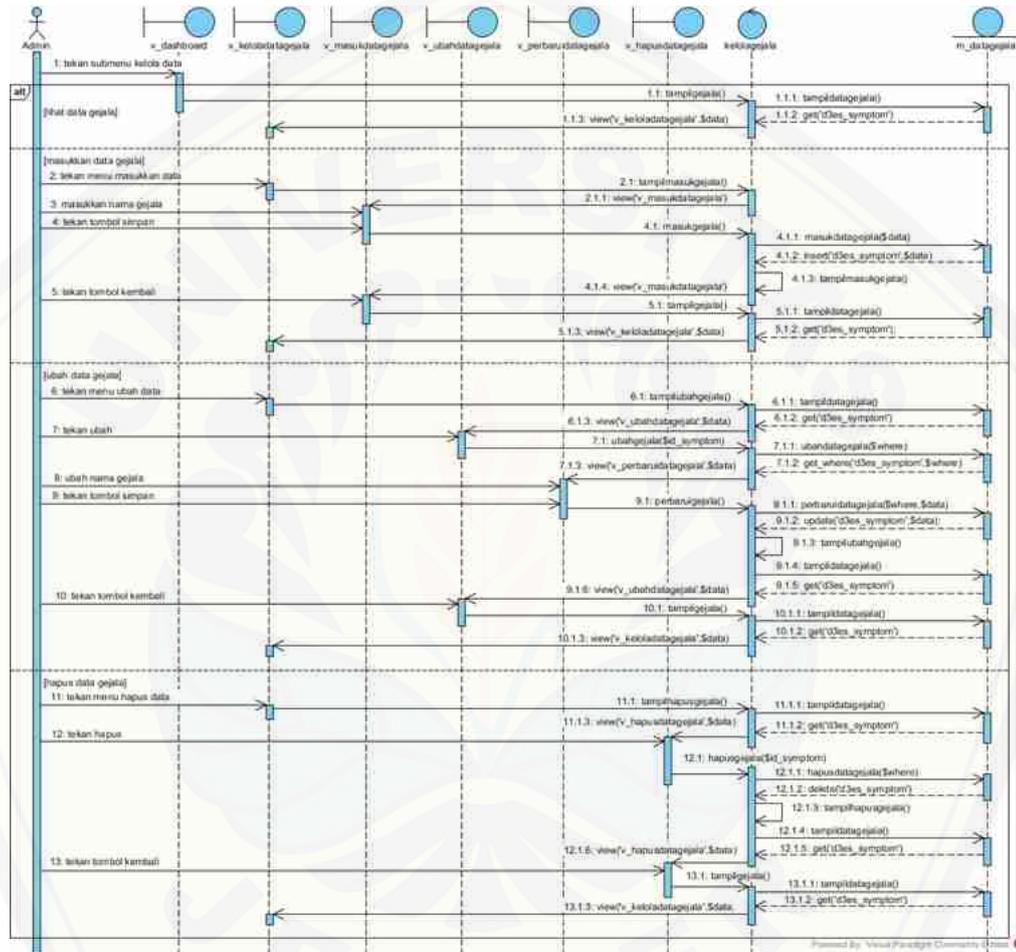
Gambar 4.11 Sequence diagram mengelola data penyakit

(Sumber: Hasil analisis, 2018)

Gambar 4.11 menggambarkan *sequence diagram* mengelola data penyakit sistem pakar diagnosis penyakit pada anjing peliharaan dengan komponen 1 aktor, 6 kelas *view*, 1 kelas *controller*, dan 1 kelas *model*. Aktor adalah Admin, kelas *view* terdiri dari *v_dashboard*, *v_keloladatapenyakit*, *v_masukdatapenyakit*, *v_ubahdatapenyakit*, *v_perbaruidatapenyakit*, dan *v_hapusdatapenyakit*, kelas *controller* terdiri dari *kelolapenyakit*, dan kelas *model* terdiri dari *m_datapenyakit*.

4.2.5.4 Sequence Diagram Mengelola data Gejala Penyakit

Gambaran mengenai interaksi dan pesan-pesan yang dikirim antar objek dalam proses mengelola data gejala penyakit dapat dilihat pada Gambar 4.12.



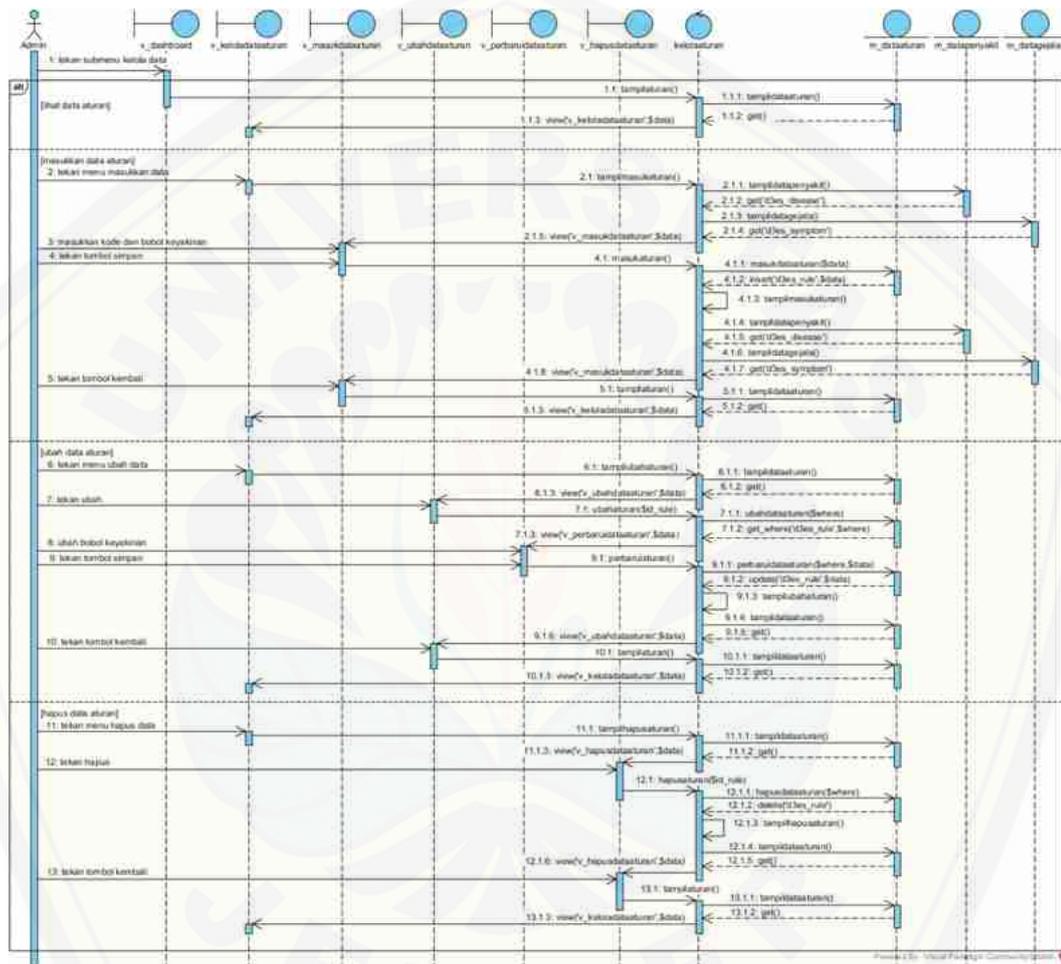
Gambar 4.12 Sequence diagram mengelola data gejala penyakit

(Sumber: Hasil analisis, 2018)

Gambar 4.12 menggambarkan *sequence diagram* mengelola data gejala penyakit sistem pakar diagnosis penyakit pada anjing peliharaan dengan komponen 1 aktor, 6 kelas *view*, 1 kelas *controller*, dan 1 kelas *model*. Aktor adalah Admin, kelas *view* terdiri dari *v_dashboard*, *v_kelolagejala*, *v_masukgejala*, *v_ubahgejala*, *v_perbaruagejala*, dan *v_hapusgejala*, kelas *controller* terdiri dari *kelolagejala*, dan kelas *model* terdiri dari *m_datagejala*.

4.2.5.5 Sequence Diagram Mengelola Aturan Sistem Pakar

Gambaran mengenai interaksi dan pesan-pesan yang dikirim antar objek dalam proses mengelola aturan sistem pakar dapat dilihat pada Gambar 4.13.

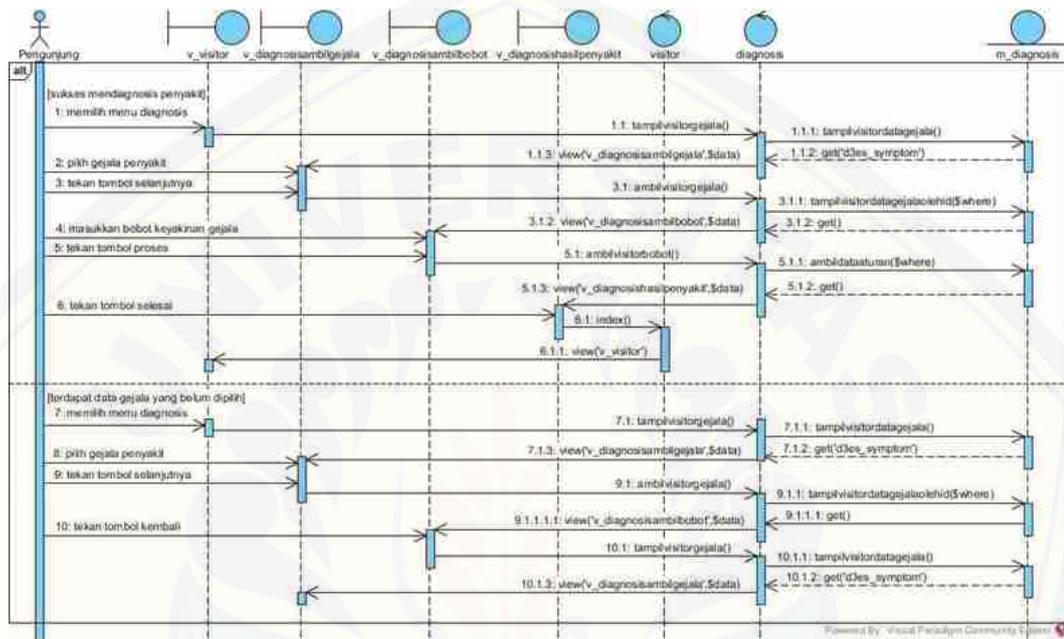


Gambar 4.13 Sequence diagram mengelola aturan sistem pakar
(Sumber: Hasil analisis, 2018)

Gambar 4.13 menggambarkan *sequence diagram* mengelola aturan sistem pakar diagnosis penyakit pada anjing peliharaan dengan komponen 1 aktor, 6 kelas *view*, 1 kelas *controller*, dan 3 kelas *model*. Aktor adalah Admin, kelas *view* terdiri dari *v_dashboard*, *v_kelolaaturan*, *v_masukaturan*, *v_ubahaturan*, *v_perbaruaturan*, dan *v_hapusaturan*, kelas *controller* terdiri dari *kelola aturan*, dan kelas *model* terdiri dari *m_dataaturan*, *m_datapenyakit*, dan *m_datagejala*.

4.2.5.6 Sequence Diagram Mendiagnosis Penyakit

Gambaran mengenai interaksi dan pesan-pesan yang dikirim antar objek dalam proses mendiagnosis dapat dilihat pada Gambar 4.14.



Gambar 4.14 Sequence diagram mendiagnosis penyakit

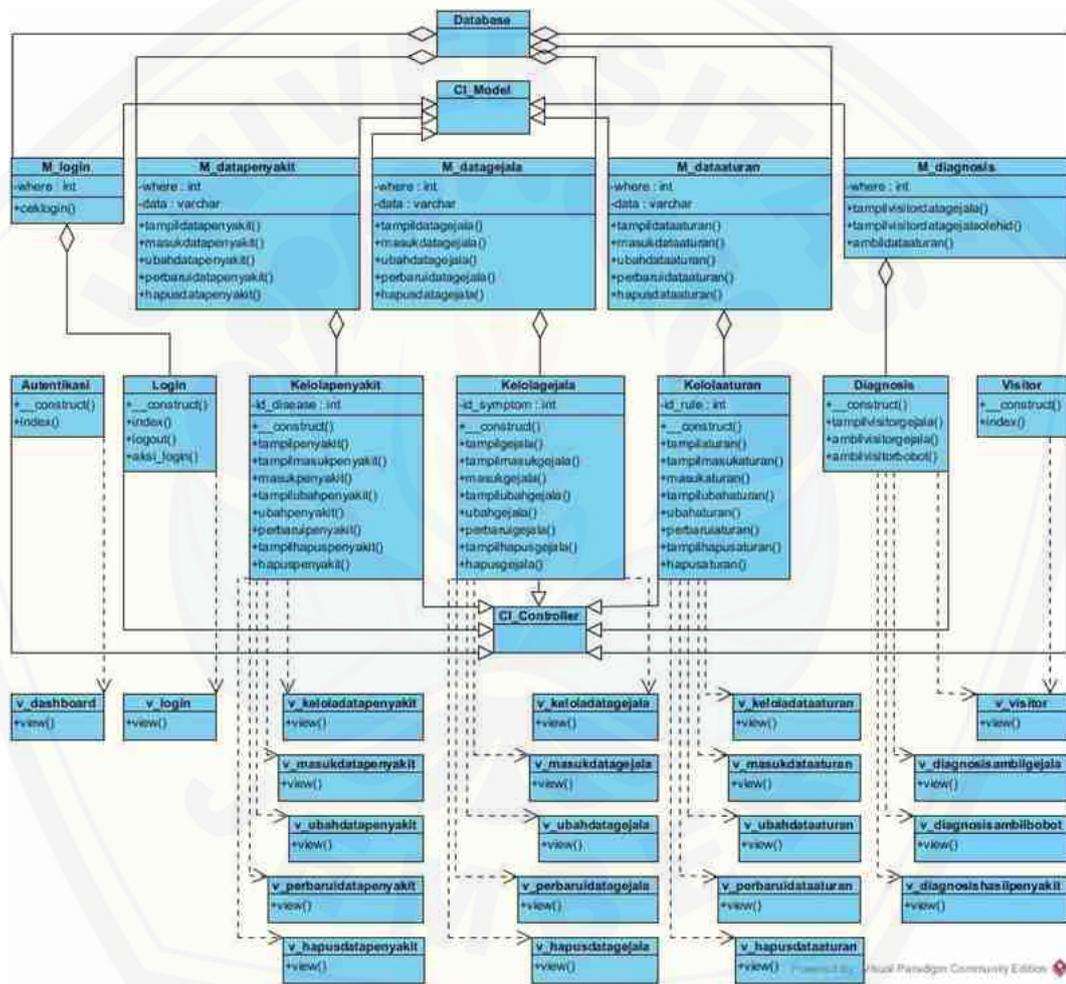
(Sumber: Hasil analisis, 2018)

Gambar 4.14 menggambarkan *sequence diagram* mendiagnosis penyakit sistem pakar diagnosis penyakit pada anjing peliharaan dengan komponen 1 aktor, 4 kelas *view*, 1 kelas *controller*, dan 1 kelas *model*. Aktor adalah Pengunjung, kelas *view* terdiri dari *v_visitor*, *v_diagnosisambilgejala*, *v_diagnosisambilbobot*, dan *v_diagnosishasilpenyakit*, kelas *controller* terdiri dari *diagnosis* dan *visitor*, dan kelas *model* terdiri *m_diagnosis*.

4.2.6 Class Diagram

Class Diagram merupakan gambaran struktur kelas pada sistem dan hubungannya. Melalui *class diagram* dapat diketahui obyek yang terdapat pada sebuah kelas dan strukturnya. *Class diagram* mempermudah pengembangan sistem

dalam pembuatan kelas dan obyek sistem. Sistem yang dikembangkan menggunakan konsep OOP akan memiliki kelas 3 kelas utama yaitu, *model*, *view*, dan *controller*. Berdasarkan *sequence diagram*, sistem pakar diagnosis penyakit pada anjing peliharaan memiliki 21 *subclass view*, 5 *subclass model* dan 7 *subclass controller*. *Class diagram* dari sistem yang dikembangkan dapat dilihat pada Gambar 4.15.



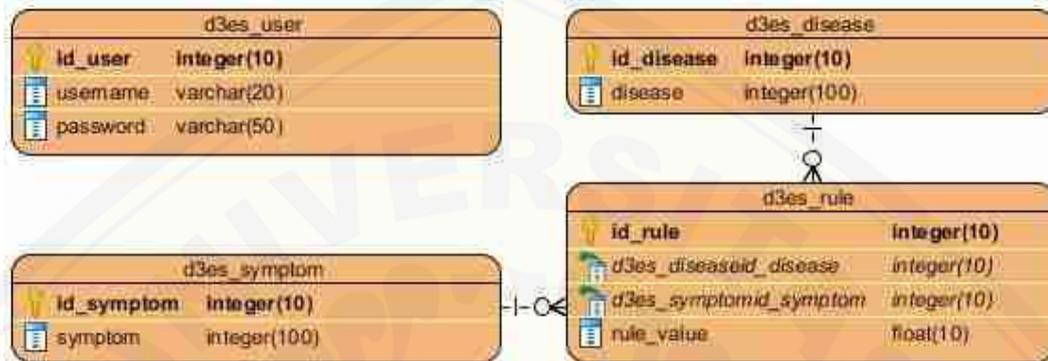
Gambar 4.15 Class diagram sistem

(Sumber: Hasil analisis, 2018)

4.2.7 Entity Relationship Diagram

Entity Relationship Diagram merupakan gambaran hubungan antara entitas yang terdapat pada sistem. Melalui *entity relationship diagram* dapat diketahui

entitas yang diperlukan oleh sistem dan jenis hubungannya. *Entity relationship diagram* mempermudah pengembangan sistem dalam pembuatan *database* sistem. *Entity relationship diagram* dari sistem yang dikembangkan dapat dilihat pada Gambar 4.16.



Gambar 4.16 *Entity relationship diagram* sistem

(Sumber: Hasil analisis, 2018)

Gambar 4.16 menggambarkan *entity relationship diagram* sistem pakar diagnosis penyakit pada anjing peliharaan yang terdiri dari 4 entitas yaitu *d3es_user*, *d3es_disease*, *d3es_syptom*, dan *d3es_rule*.

4.3 Mengkodekan Sistem

Mengkodekan Sistem adalah tahap penerapan kode program ke sistem berdasarkan desain yang telah digambarkan pada tahap perancangan sistem. Pengembangan sistem akan menggunakan PHP sebagai bahasa pemrograman, *CodeIgniter* sebagai PHP *framework* dengan model MVC yang sesuai dengan konsep OOP, dan *MySQL* untuk manajemen basis data SQL. Penulisan kode program dapat dilihat pada Lampiran B.

4.4 Pengujian Sistem

Pengujian Sistem adalah tahap menguji perangkat lunak yang telah dikembangkan sebelum digunakan oleh pengguna. Tahap ini menguji apakah kode program berjalan dengan baik sehingga sistem dapat memenuhi kebutuhan

pengguna. Fitur sistem yang akan diuji pada tahap ini adalah fitur *login*, *logout*, mengelola data penyakit, mengelola data gejala penyakit, mengelola data aturan sistem pakar, dan mendiagnosis penyakit. Pengembangan perangkat lunak ini akan menerapkan metode *White Box Testing* dan *Black Box Testing* sebagai metode pengujian sistem.

4.4.1 *White Box Testing*

White box testing adalah pengujian dengan cara menganalisis secara langsung kode program yang terdapat pada sistem. Pengujian *white box* pada pengembangan perangkat lunak ini akan menerapkan metode *Basis Path Testing* yang meliputi Listing Program, Diagram Alir, Perhitungan *Cyclomatic Complexity*, Basis Set, dan *Test Case*, Hasil *white box testing* sistem dapat dilihat pada Lampiran C.

4.4.2 *Black Box Testing*

Black box testing adalah pengujian dengan cara menjalankan sistem lalu mengamati apakah fungsionalitas sistem telah sesuai dengan kebutuhan pengguna. Pengujian *black box* akan dilakukan pada setiap fitur sistem. Hasil *black box testing* sistem dapat dilihat pada Lampiran D.

4.5 Evaluasi Sistem

Evaluasi Sistem adalah tahap di mana pengguna mengambil keputusan untuk menerima perangkat lunak berdasarkan hasil dari tahap pengujian sistem. Jika hasil dari tahap pengujian sistem benar, maka pengembangan sistem dapat dilanjutkan ke tahap implementasi sistem. Tetapi jika hasil dari tahap pengujian sistem salah, maka pengembangan sistem kembali ke tahap pengumpulan kebutuhan sistem.

4.6 Implementasi Sistem

Implementasi Sistem adalah tahap akhir di mana perangkat lunak telah diterima oleh pengguna dan siap untuk digunakan. Hasil dari implementasi sistem dapat dilihat pada Bab 5.

BAB 6. PENUTUP

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

6.1 Kesimpulan

Kesimpulan yang didapatkan dari penelitian yang telah dilakukan adalah:

1. Sistem pakar diagnosis penyakit pada anjing peliharaan berbasis web menerapkan metode *certainty factor* untuk memperoleh tingkat akurasi yang tinggi dari hasil diagnosis penyakit pada anjing peliharaan.
2. Sistem pakar diagnosis penyakit pada anjing peliharaan menggunakan metode *certainty factor* berbasis web memiliki fitur login, logout, mengelola data penyakit, mengelola data gejala penyakit, mengelola aturan sistem pakar dan mendiagnosis penyakit yang sesuai dengan kebutuhan sistem.
3. Proses perhitungan metode *certainty factor* oleh sistem yang menggunakan tambahan kaidah dengan premis tunggal, dan proses perhitungan metode *certainty factor* tanpa menggunakan kaidah dengan premis tunggal sama-sama memiliki hasil yang valid dan tingkat akurasi yang baik walaupun proses perhitungan metode *certainty factor* tanpa menggunakan kaidah dengan premis tunggal memiliki akurasi sedikit lebih tinggi daripada proses perhitungan metode *certainty factor* oleh sistem.

6.2 Saran

Saran yang diajukan oleh peneliti dengan harapan sistem dapat dikembangkan lebih baik pada penelitian selanjutnya adalah:

1. Sistem perlu dikembangkan lebih luas dengan model pengembangan tingkat lanjut dan menambahkan fitur yang mendukung fitur utama.
2. Ruang lingkup penelitian perlu diperluas lagi sehingga hasil dari penelitian lebih lengkap.

DAFTAR PUSTAKA

- Arief, M. R. (2011). *Pemrograman Web Dinamis Menggunakan PHP Dan MySQL*. Yogyakarta: Andi Publisher.
- Daniel, & Gloria, V. (2010). Implementasi Sistem Pakar Untuk Mendiagnosis Penyakit Dengan Gejala Demam Menggunakan Metode Certainty Factor. *Jurnal Informatika*, 6(1).
- Effendi, C., & Setiawati, W. (2017). *Solusi Permasalahan Kucing*. Cibubur: Swadaya.
- Gulo, W. (2008). *Metodologi Penelitian*. Jakarta: Grasindo.
- Hardjodisastro, D. (2006). *Menuju Seni Ilmu Kedokteran: Bagaimana Dokter Berpikir, Bekerja, Dan Menampilkan Diri*. Jakarta: Gramedia Pustaka Utama.
- Hayadi, B. H. (2016). *Sistem Pakar*. Yogyakarta: Deepublish.
- Irwansyah, E., & V, J. (2014). *Pengantar Teknologi Informasi*. Yogyakarta: Deepublish.
- Kent, A., & Williams, J. G. (1989). *Encyclopedia of Computer Science and Technology: Volume 21 - Supplement 6: ADA and Distributed Systems to Visual Languages*. CRC Press.
- Kusrini. (2008). *APLIKASI SISTEM PAKAR Menentukan Faktor Kepastian Pengguna dengan Metode Kuantifikasi Pertanyaan*. Yogyakarta: ANDI.
- Maturidi, A. D. (2014). *Metode Penelitian Teknik Informatika*. Yogyakarta: Deepublish.
- Rohajawati, S., & Supriyati, R. (2010). Sistem Pakar: Diagnosis Penyakit Unggas Dengan Metode Certainty Factor. *CommIT*, 4(1).
- Santoso, B., & Budiana, N. (2015). *Anjing*. Jakarta: Agriflo.
- Triharnoto. (2009). *The Doctor: Catatan Hati Seorang Dokter*. Yogyakarta: Galangpress Group.
- Yulianti, & Mewati, A. (2012). Sistem Pakar Diagnosis Penyakit Anjing dengan Forward Chaining. *Jurnal Informatika*, 8(2).

LAMPIRAN

Lampiran A. Use Case Scenario

1. Use Case Scenario Login

Nama Use Case	<i>Login</i>
Aktor	Admin
Entry Condition	Aktor memiliki <i>username</i> dan <i>password</i> dan ingin masuk ke dalam sistem
Exit Condition	Aktor mendapatkan status <i>login</i> dan berada pada Halaman <i>Dashboard</i>
Aksi Aktor	Reaksi Sistem
SKENARIO NORMAL	

<p>1a. Aktor mengakses Halaman <i>Login Admin</i></p>	<p>2a. Sistem menampilkan Halaman <i>Login Admin</i>, berisi <i>Form Login</i> yang terdiri dari:</p> <ul style="list-style-type: none"> - <i>Textfield Username</i> - <i>Textfield Password</i> - <i>Tombol Login</i>
<p>3a. Aktor mengisi <i>Textfield Username</i> dan <i>Textfield Password</i> sesuai dengan <i>username</i> dan <i>password</i> miliknya</p>	
<p>4a. Aktor menekan <i>Tombol Login</i></p>	<p>5a. Sistem melakukan proses validasi <i>username</i> dan <i>password</i></p>
	<p>6a. Apabila <i>username</i> dan <i>password</i> yang diisi sudah benar, sistem</p>

memberikan Status *Login* dan menampilkan Halaman

Dashboard, berisi:

- Menu Penyakit
- Menu Gejala
- Menu Aturan
- Tombol *Logout*

SKENARIO ALTERNATIF

Aktor menekan Tombol *Login* ketika mengisi *username* dan *password* yang tidak sesuai dengan miliknya pada *Form Login*

3a. Aktor mengisi *Textfield Username* dan *Textfield Password* tetapi tidak sesuai dengan *username* dan *password* miliknya

4a. Aktor menekan Tombol *Login*

5a. Sistem melakukan proses validasi *username* dan *password*

6a. Sistem menampilkan pesan peringatan berisi tulisan "*Username* atau *password* salah!" di atas Tombol *Login*

7a. Aktor menekan Tombol *x(Close)* yang berada di sisi kanan pesan peringatan

8a. Sistem menampilkan kembali Halaman *Login Admin*, berisi *Form Login* yang terdiri dari:

- *Textfield Username*
- *Textfield Password*
- Tombol *Login*

SKENARIO ALTERNATIF**Aktor menekan Tombol *Login* tanpa mengisi *Textfield Username***

3b. Aktor tidak mengisi *Textfield*

Username

4b. Aktor menekan Tombol *Login*

5b. Sistem menampilkan pesan peringatan berisi tulisan “*Please fill out this field*” di bawah *Textfield Username*

6b. Sistem menampilkan kembali Halaman *Login Admin*, berisi *Form Login* yang terdiri dari:

- *Textfield Username*
- *Textfield Password*
- Tombol *Login*

SKENARIO ALTERNATIF**Aktor menekan Tombol *Login* tanpa mengisi *Textfield Password***

3c. Aktor mengisi *Textfield Username*

tetapi tidak mengisi *Textfield*

Password

4c. Aktor menekan Tombol *Login*

5c. Sistem menampilkan pesan peringatan berisi tulisan “*Please fill out this field*” di bawah *Textfield Password*

6c. Sistem menampilkan kembali Halaman *Login Admin*, berisi *Form Login* yang terdiri dari:

- *Textfield Username*
- *Textfield Password*

- Tombol *Login*

Sumber: Hasil analisis (2018).

2. Use Case Scenario Logout

Nama Use Case	<i>Logout</i>
Aktor	Admin
Entry Condition	Aktor memiliki status <i>login</i> dan ingin keluar dari sistem
Exit Condition	Aktor kehilangan status <i>login</i> dan berada pada Halaman <i>Login</i> Admin
Aksi Aktor	Reaksi Sistem
SKENARIO NORMAL	
<i>Logout dari Halaman Dashboard</i>	
1. Aktor berada pada Halaman <i>Dashboard</i>	
2. Aktor menekan Tombol <i>Logout</i> yang berada pada bagian kanan atas Halaman <i>Dashboard</i>	
	3. Sistem memproses permintaan <i>logout</i> aktor
	4. Sistem menghilangkan status <i>login</i> aktor dan menampilkan Halaman <i>Login</i> Admin, berisi <i>Form Login</i> yang terdiri dari: - <i>Text field Username</i> - <i>Text field Password</i> - Tombol <i>Login</i>
SKENARIO NORMAL	
<i>Logout dari Halaman Kelola Data Penyakit</i>	
5. Aktor berada pada Halaman Kelola Data Penyakit	

6. Aktor menekan Tombol *Logout* yang berada pada bagian kanan atas Halaman Kelola Data Penyakit

7. Sistem memproses permintaan *logout* aktor

8. Sistem menghilangkan status *login* aktor dan menampilkan Halaman *Login* Admin, berisi *Form Login* yang terdiri dari:

- *Text field Username*
- *Text field Password*
- Tombol *Login*

SKENARIO NORMAL

Logout dari halaman Kelola Data Gejala

9. Aktor berada pada Halaman Kelola Data Gejala

10. Aktor menekan Tombol *Logout* yang berada pada bagian kanan atas Halaman Kelola Data Gejala

11. Sistem memproses permintaan *logout* aktor

12. Sistem menghilangkan status *login* aktor dan menampilkan Halaman *Login* Admin, berisi *Form Login* yang terdiri dari:

- *Text field Username*
- *Text field Password*
- Tombol *Login*

SKENARIO NORMAL

Logout dari halaman Kelola Data Aturan

13. Aktor berada pada Halaman Kelola Data Aturan	
14. Aktor menekan Tombol <i>Logout</i> yang berada pada bagian kanan atas Halaman Kelola Data Aturan	
	15. Sistem memproses permintaan <i>logout</i> aktor
	16. Sistem menghilangkan status <i>login</i> aktor dan menampilkan Halaman <i>Login</i> Admin, berisi <i>Form Login</i> yang terdiri dari: <ul style="list-style-type: none"> - <i>Text field Username</i> - <i>Text field Password</i> - Tombol <i>Login</i>

Sumber: Hasil analisis (2018).

3. Use Case Scenario Mengelola Data Penyakit

Nama Use Case	Mengelola Data Penyakit
Aktor	Admin
Entry Condition	Aktor memiliki status <i>login</i> dan ingin mengelola data penyakit
Exit Condition	Aktor berhasil mengelola data penyakit
Aksi Aktor	Reaksi Sistem
SKENARIO NORMAL	

1. Aktor berada pada Halaman <i>Dashboard</i>	
2. Aktor memilih Menu Penyakit	
	3. Sistem menampilkan <i>Dropdown</i> Submenu, berisi:

- Submenu Kelola Data

4. Aktor memilih Submenu Kelola Data

5. Sistem menampilkan Halaman Kelola Data Penyakit, berisi:

- Tombol Masukkan Data
- Tombol Ubah Data
- Tombol Hapus Data
- Tombol Kembali
- Tombol *Logout*
- Tabel Data Penyakit yang terdiri dari Kolom No dan Penyakit

SKENARIO ALTERNATIF

Aktor selesai melihat dan mengelola data penyakit

4a. Aktor memilih Submenu Kelola Data

5a. Sistem menampilkan Halaman Kelola Data Penyakit, berisi:

- Tombol Masukkan Data
- Tombol Ubah Data
- Tombol Hapus Data
- Tombol Kembali
- Tombol *Logout*
- Tabel Data Penyakit yang terdiri dari Kolom No dan Penyakit

6a. Aktor menekan Tombol Kembali

7a. Sistem menampilkan Halaman *Dashboard*, berisi:

- Menu Penyakit
 - Menu Gejala
 - Menu Aturan
-

- Tombol *Logout*

SKENARIO NORMAL

Aktor memasukkan data penyakit

6. Jika aktor ingin memasukkan data penyakit, maka aktor menekan Tombol Masukkan Data

7. Sistem menampilkan *Form* Masukkan Data Penyakit, berisi:

- *Textfield* Penyakit
- Tombol Simpan
- Tombol Kembali

8. Aktor mengisi *Textfield* Penyakit dengan data penyakit

9. Aktor menekan Tombol Simpan

10. Sistem melakukan proses memasukkan data penyakit pada *database* sistem berdasarkan isi dari *Textfield* Penyakit pada *Form* Masukkan Data Penyakit

11. Sistem menampilkan kembali *Form* Masukkan Data Penyakit, berisi:

- *Textfield* Penyakit
- Tombol Simpan
- Tombol Kembali

SKENARIO ALTERNATIF

Aktor ingin memasukkan data penyakit tetapi membatalkannya

8a. Aktor mengisi *Textfield* Penyakit dengan data penyakit

9a. Aktor menekan Tombol Kembali

10a. Sistem menampilkan kembali

Halaman Kelola Data Penyakit,
berisi:

- Tombol Masukkan Data
- Tombol Ubah Data
- Tombol Hapus Data
- Tombol Kembali
- Tombol *Logout*
- Tabel Data Penyakit yang terdiri dari Kolom No dan Penyakit

SKENARIO ALTERNATIF

Aktor menekan Tombol Simpan tetapi *Textfield* Penyakit kosong

8b. Aktor tidak mengisi *Textfield*
Penyakit

9b. Aktor menekan Tombol Simpan

10b. Sistem menampilkan pesan peringatan berisi tulisan "*Please fill out this field*" di bawah *Textfield* Penyakit

11b. Sistem menampilkan kembali *Form* Masukkan Data Penyakit,
berisi:

- *Textfield* Penyakit
- Tombol Simpan
- Tombol Kembali

SKENARIO NORMAL

Aktor mengubah data penyakit

12. Jika aktor ingin mengubah data penyakit, maka aktor menekan Tombol Ubah Data

13. Sistem menampilkan *Form* Ubah Data Penyakit, berisi:

- Tabel Ubah Data Penyakit yang terdiri dari Kolom No, Penyakit, dan Aksi
- Tombol Kembali

14. Aktor menekan aksi Ubah pada Kolom Aksi sesuai dengan data penyakit yang ingin diubah

15. Sistem menampilkan *Form* Perbarui Data Penyakit, berisi:

- *Textfield* Penyakit yang berisi data penyakit yang dipilih sebelumnya
- Tombol Simpan
- Tombol Kembali

16. Aktor mengganti data penyakit pada *Textfield* Penyakit

17. Aktor menekan Tombol Simpan

18. Sistem melakukan proses memperbarui data penyakit pada *database* sistem berdasarkan isi dari *Textfield* Penyakit di *Form* Perbarui Data Penyakit

19. Sistem menampilkan kembali *Form* Ubah Data Penyakit, berisi:

-
- Tabel Ubah Data Penyakit yang terdiri dari Kolom No, Penyakit, dan Aksi
 - Tombol Kembali

SKENARIO ALTERNATIF**Aktor ingin memperbarui data penyakit tetapi membatalkannya**

16a. Aktor mengganti data penyakit pada

Textfield Penyakit

17a. Aktor menekan Tombol Kembali

18a. Sistem menampilkan kembali Halaman Kelola Data Penyakit, berisi:

- Tombol Masukkan Data
- Tombol Ubah Data
- Tombol Hapus Data
- Tombol Kembali
- Tombol *Logout*
- Tabel Data Penyakit yang terdiri dari Kolom No dan Penyakit

SKENARIO ALTERNATIF**Aktor menekan Tombol Simpan tetapi *Textfield* Penyakit kosong**

16b. Aktor menghilangkan data penyakit yang berada pada *Textfield* Penyakit

17b. Aktor menekan Tombol Simpan

18b. Sistem menampilkan pesan peringatan berisi tulisan "*Please fill out this field*" di bawah *Textfield* Penyakit

19b. Sistem menampilkan kembali

Form Perbarui Data Penyakit,
berisi:

- *Textfield* Penyakit yang berisi data penyakit yang dipilih sebelumnya
- Tombol Simpan
- Tombol Kembali

SKENARIO NORMAL

Aktor menghapus data penyakit

20. Jika aktor ingin menghapus data penyakit, maka aktor menekan Tombol Hapus Data

21. Sistem menampilkan *Form*

Hapus Data Penyakit, berisi:

- Tabel Hapus Data Penyakit yang terdiri dari Kolom No, Penyakit, dan Aksi
- Tombol Kembali

22. Aktor menekan aksi Hapus pada Kolom Aksi sesuai dengan data penyakit yang ingin dihapus

23. Sistem melakukan proses menghapus data penyakit pada *database* sistem berdasarkan pilihan di *Form* Hapus Data Penyakit

24. Sistem menampilkan kembali *Form* Hapus Data Penyakit, berisi:

-
- Tabel Hapus Data Penyakit yang terdiri dari Kolom No, Penyakit, dan Aksi
 - Tombol Kembali

SKENARIO ALTERNATIF

Aktor ingin menghapus data penyakit tetapi membatalkannya

22a. Aktor menekan Tombol Kembali

- 23a. Sistem menampilkan kembali Halaman Kelola Data Penyakit, berisi:
- Tombol Masukkan Data
 - Tombol Ubah Data
 - Tombol Hapus Data
 - Tombol Kembali
 - Tombol *Logout*
 - Tabel Data Penyakit yang terdiri dari Kolom No dan Penyakit

Sumber: Hasil analisis (2018).

4. Use Case Scenario Mengelola Data Gejala Penyakit

Nama Use Case	Mengelola Data Gejala Penyakit
Aktor	Admin
Entry Condition	Aktor memiliki status <i>login</i> dan ingin mengelola data gejala
Exit Condition	Aktor berhasil mengelola data gejala
Aksi Aktor	Reaksi Sistem
SKENARIO NORMAL	

1. Aktor berada pada Halaman *Dashboard*
-

2. Aktor memilih Menu Gejala

3. Sistem menampilkan *Dropdown*

Submenu, berisi:

- Submenu Kelola Data
-

4. Aktor memilih Submenu Kelola Data

5. Sistem menampilkan Halaman Kelola Data Gejala, berisi:

- Tombol Masukkan Data
 - Tombol Ubah Data
 - Tombol Hapus Data
 - Tombol Kembali
 - Tombol *Logout*
 - Tabel Data Gejala yang terdiri dari Kolom No dan Gejala
-

SKENARIO ALTERNATIF

Aktor selesai melihat dan mengelola data gejala

4a. Aktor memilih Submenu Kelola Data

5a. Sistem menampilkan Halaman Kelola Data Gejala, berisi:

- Tombol Masukkan Data
 - Tombol Ubah Data
 - Tombol Hapus Data
 - Tombol Kembali
 - Tombol *Logout*
 - Tabel Data Gejala yang terdiri dari Kolom No dan Gejala
-

6a. Aktor menekan Tombol Kembali

8a. Sistem menampilkan Halaman *Dashboard*, berisi:

- Menu Penyakit
-

-
- Menu Gejala
 - Menu Aturan
 - Tombol *Logout*

SKENARIO NORMAL**Aktor memasukkan data gejala**

6. Jika aktor ingin memasukkan data gejala, maka aktor menekan Tombol Masukkan Data

7. Sistem menampilkan *Form* Masukkan Data Gejala, berisi:
 - *Textfield* Gejala
 - Tombol Simpan
 - Tombol Kembali

8. Aktor mengisi *Textfield* Gejala dengan data gejala

9. Aktor menekan Tombol Simpan

10. Sistem melakukan proses memasukkan data gejala pada *database* sistem berdasarkan isi dari *Textfield* Gejala pada *Form* Masukkan Data Gejala

11. Sistem menampilkan kembali *Form* Masukkan Data Gejala, berisi:
 - *Textfield* Gejala
 - Tombol Simpan
 - Tombol Kembali

SKENARIO ALTERNATIF**Aktor ingin memasukkan data gejala tetapi membatalkannya**

8a. Aktor mengisi *Textfield* Gejala dengan data gejala

9a. Aktor menekan Tombol Kembali

10a. Sistem menampilkan kembali Halaman Kelola Data Gejala, berisi:

- Tombol Masukkan Data
- Tombol Ubah Data
- Tombol Hapus Data
- Tombol Kembali
- Tombol *Logout*
- Tabel Data Gejala yang terdiri dari Kolom No dan Gejala

SKENARIO ALTERNATIF

Aktor menekan Tombol Simpan tetapi *Textfield* Gejala kosong

8b. Aktor tidak mengisi *Textfield* Gejala

9b. Aktor menekan Tombol Simpan

10b. Sistem menampilkan pesan peringatan berisi tulisan "*Please fill out this field*" di bawah *Textfield* Gejala

11b. Sistem menampilkan kembali *Form* Masukkan Data Gejala, berisi:

- *Textfield* Gejala
- Tombol Simpan
- Tombol Kembali

SKENARIO NORMAL

Aktor mengubah data gejala

12. Jika aktor ingin mengubah data gejala, maka aktor menekan Tombol Ubah Data

13. Sistem menampilkan *Form* Ubah Data Gejala, berisi:

- Tabel Ubah Data Gejala yang terdiri dari Kolom No, Gejala, dan Aksi
- Tombol Kembali

14. Aktor menekan aksi Ubah pada Kolom Aksi sesuai dengan data gejala yang ingin diubah

15. Sistem menampilkan *Form* Perbarui Data Gejala, berisi:

- *Textfield* Gejala yang berisi data gejala yang dipilih sebelumnya
- Tombol Simpan
- Tombol Kembali

16. Aktor mengganti data gejala pada *Textfield* Gejala

17. Aktor menekan Tombol Simpan

18. Sistem melakukan proses memperbarui data gejala pada *database* sistem berdasarkan isi dari *Textfield* Gejala di *Form* Perbarui Data Gejala

19. Sistem menampilkan kembali *Form* Ubah Data Gejala, berisi:

-
- Tabel Ubah Data Gejala yang terdiri dari Kolom No, Gejala, dan Aksi
 - Tombol Kembali

SKENARIO ALTERNATIF**Aktor ingin memperbarui data gejala tetapi membatalkannya**

16a. Aktor mengganti data gejala pada

Textfield Gejala

17a. Aktor menekan Tombol Kembali

18a. Sistem menampilkan kembali Halaman Kelola Data Gejala, berisi:

- Tombol Masukkan Data
- Tombol Ubah Data
- Tombol Hapus Data
- Tombol Kembali
- Tombol *Logout*
- Tabel Data Gejala yang terdiri dari Kolom No dan Gejala

SKENARIO ALTERNATIF**Aktor menekan Tombol Simpan tetapi *Textfield* Gejala kosong**

16b. Aktor menghilangkan data gejala yang berada pada *Textfied* Gejala

17b. Aktor menekan Tombol Simpan

18b. Sistem menampilkan pesan peringatan berisi tulisan "*Please fill out this field*" di bawah *Textfield* Gejala

19b. Sistem menampilkan kembali

Form Perbarui Data Gejala,
berisi:

- *Textfield* Gejala yang berisi data gejala yang dipilih sebelumnya
- Tombol Simpan
- Tombol Kembali

SKENARIO NORMAL

Aktor menghapus data gejala

20. Jika aktor ingin menghapus data gejala, maka aktor menekan Tombol Hapus Data

21. Sistem menampilkan *Form*

Hapus Data Gejala, berisi:

- Tabel Hapus Data Gejala yang terdiri dari Kolom No, Gejala, dan Aksi
- Tombol Kembali

22. Aktor menekan aksi Hapus pada Kolom Aksi sesuai dengan data gejala yang ingin dihapus

23. Sistem melakukan proses menghapus data gejala pada *database* sistem berdasarkan pilihan di *Form* Hapus Data Gejala

24. Sistem menampilkan kembali *Form* Hapus Data Gejala, berisi:

- Tabel Hapus Data Gejala yang terdiri dari Kolom No, Gejala, dan Aksi
- Tombol Kembali

SKENARIO ALTERNATIF
Aktor ingin menghapus data gejala tetapi membatalkannya

22a. Aktor menekan Tombol Kembali	
	23a. Sistem menampilkan kembali Halaman Kelola Data Gejala, berisi: <ul style="list-style-type: none"> - Tombol Masukkan Data - Tombol Ubah Data - Tombol Hapus Data - Tombol Kembali - Tombol <i>Logout</i> - Tabel Data Gejala yang terdiri dari Kolom No dan Gejala

Sumber: Hasil analisis (2018).

5. Use Case Scenario Mengelola Aturan Sistem Pakar

Nama Use Case	Mengelola Aturan Sistem Pakar
Aktor	Admin
Entry Condition	Aktor memiliki status <i>login</i> dan ingin mengelola data aturan
Exit Condition	Aktor berhasil mengelola data aturan
Aksi Aktor	Reaksi Sistem

SKENARIO NORMAL

1. Aktor berada pada Halaman *Dashboard*
2. Aktor memilih Menu Aturan

3. Sistem menampilkan *Dropdown*

Submenu, berisi:

- Submenu Kelola Data

4. Aktor memilih Submenu Kelola Data

5. Sistem menampilkan Halaman

Kelola Data Aturan, berisi:

- Tombol Masukkan Data

- Tombol Ubah Data

- Tombol Hapus Data

- Tombol Kembali

- Tombol *Logout*

- Tabel Data Aturan yang terdiri dari Kolom No, Penyakit, Gejala, dan Bobot Keyakinan

SKENARIO ALTERNATIF

Aktor selesai melihat dan mengelola data aturan

4a. Aktor memilih Submenu Kelola Data

5a. Sistem menampilkan Halaman Kelola Data Aturan, berisi:

- Tombol Masukkan Data

- Tombol Ubah Data

- Tombol Hapus Data

- Tombol Kembali

- Tombol *Logout*

- Tabel Data Aturan yang terdiri dari Kolom No, Penyakit, Gejala, dan Bobot Keyakinan

6a. Aktor menekan Tombol Kembali

9a. Sistem menampilkan Halaman

Dashboard, berisi:

-
- Menu Penyakit
 - Menu Gejala
 - Menu Aturan
 - Tombol *Logout*

SKENARIO NORMAL

Aktor memasukkan data aturan

6. Jika aktor ingin memasukkan data aturan, maka aktor menekan Tombol Masukkan Data

-
7. Sistem menampilkan *Form* Masukkan Data Aturan, berisi:
- *Textfield* Kode Penyakit
 - *Textfield* Kode Gejala
 - *Textfield* Bobot Keyakinan Gejala Penyakit
 - Tabel Kode Penyakit yang terdiri dari Kolom Kode dan Penyakit
 - Tabel Kode Gejala yang terdiri dari Kolom Kode dan Gejala
 - Tombol Simpan
 - Tombol Kembali

8. Aktor mengisi *Textfield* Kode Penyakit dengan data kode penyakit, *Textfield* Kode Gejala dengan data kode gejala, dan *Textfield* Bobot Keyakinan Gejala Penyakit dengan data bobot keyakinan gejala dari pakar

9. Aktor menekan Tombol Simpan

10. Sistem melakukan proses memasukkan data aturan pada *database* sistem berdasarkan isi dari *Textfield* Kode Penyakit, *Textfield* Kode Gejala, dan *Textfield* Bobot Keyakinan Gejala Penyakit pada *Form* Masukkan Data Aturan

11. Sistem menampilkan kembali *Form* Masukkan Data Aturan, berisi:

- *Textfield* Kode Penyakit
- *Textfield* Kode Gejala
- *Textfield* Bobot Keyakinan Gejala Penyakit
- Tabel Kode Penyakit yang terdiri dari Kolom Kode dan Penyakit
- Tabel Kode Gejala yang terdiri dari Kolom Kode dan Gejala
- Tombol Simpan
- Tombol Kembali

SKENARIO ALTERNATIF

Aktor ingin memasukkan data aturan tetapi membatalkannya

8a. Aktor mengisi *Textfield* Kode Penyakit dengan data kode penyakit, *Textfield* Kode Gejala dengan data kode gejala, dan *Textfield* Bobot Keyakinan Gejala Penyakit dengan

9a. Aktor menekan Tombol Kembali

10a. Sistem menampilkan kembali

Halaman Kelola Data Aturan,
berisi:

- Tombol Masukkan Data
- Tombol Ubah Data
- Tombol Hapus Data
- Tombol Kembali
- Tombol *Logout*
- Tabel Data Aturan yang terdiri dari Kolom No, Penyakit, Gejala, dan Bobot Keyakinan

SKENARIO ALTERNATIF

Aktor menekan Tombol Simpan tetapi *Textfield* Kode Penyakit kosong

8b. Aktor tidak mengisi *Textfield* Kode
Penyakit

9b. Aktor menekan Tombol Simpan

10b. Sistem menampilkan pesan peringatan berisi tulisan "*Please fill out this field*" di bawah *Textfield* Kode Penyakit

11b. Sistem menampilkan kembali *Form* Masukkan Data Aturan, berisi:

- *Textfield* Kode Penyakit
 - *Textfield* Kode Gejala
 - *Textfield* Bobot Keyakinan
Gejala Penyakit
 - Tabel Kode Penyakit yang terdiri dari Kolom Kode dan Penyakit
-

-
- Tabel Kode Gejala yang terdiri dari Kolom Kode dan Gejala
 - Tombol Simpan
 - Tombol Kembali

SKENARIO ALTERNATIF

Aktor menekan Tombol Simpan tetapi *Textfield* Kode Gejala kosong

8c. Aktor tidak mengisi *Textfield* Kode Gejala

9c. Aktor menekan Tombol Simpan

10c. Sistem menampilkan pesan peringatan berisi tulisan "*Please fill out this field*" di bawah *Textfield* Kode Gejala

11c. Sistem menampilkan kembali *Form* Masukkan Data Aturan, berisi:

- *Textfield* Kode Penyakit
- *Textfield* Kode Gejala
- *Textfield* Bobot Keyakinan Gejala Penyakit
- Tabel Kode Penyakit yang terdiri dari Kolom Kode dan Penyakit
- Tabel Kode Gejala yang terdiri dari Kolom Kode dan Gejala
- Tombol Simpan
- Tombol Kembali

SKENARIO ALTERNATIF

Aktor menekan Tombol Simpan tetapi *Textfield* Bobot Keyakinan Gejala Penyakit kosong

8d. Aktor tidak mengisi *Textfield* Bobot

Keyakinan Gejala Penyakit

9d. Aktor menekan Tombol Simpan

10d. Sistem menampilkan pesan peringatan berisi tulisan "*Please fill out this field*" di bawah *Textfield* Bobot Keyakinan Gejala Penyakit

11d. Sistem menampilkan kembali *Form* Masukkan Data Aturan, berisi:

- *Textfield* Kode Penyakit
- *Textfield* Kode Gejala
- *Textfield* Bobot Keyakinan Gejala Penyakit
- Tabel Kode Penyakit yang terdiri dari Kolom Kode dan Penyakit
- Tabel Kode Gejala yang terdiri dari Kolom Kode dan Gejala
- Tombol Simpan
- Tombol Kembali

SKENARIO NORMAL

Aktor mengubah data bobot keyakinan gejala penyakit

12. Jika aktor ingin mengubah data bobot keyakinan gejala penyakit, maka aktor menekan Tombol Ubah Data

13. Sistem menampilkan *Form* Ubah Data Aturan berisi:

	<ul style="list-style-type: none">- Tabel Ubah Data Aturan yang terdiri dari Kolom No, Penyakit, Gejala, Bobot Keyakinan, dan Aksi- Tombol Kembali
14. Aktor menekan aksi Ubah pada Kolom Aksi sesuai dengan data bobot keyakinan gejala penyakit yang ingin diubah	
	15. Sistem menampilkan <i>Form</i> Perbarui Data Aturan, berisi: <ul style="list-style-type: none">- <i>Textfield</i> Bobot Keyakinan Gejala Penyakit yang berisi data bobot keyakinan gejala penyakit yang dipilih sebelumnya- Tombol Simpan- Tombol Kembali
16. Aktor mengganti data bobot keyakinan gejala penyakit pada <i>Textfield</i> Bobot Keyakinan Gejala Penyakit	
17. Aktor menekan Tombol Simpan	
	18. Sistem melakukan proses memperbarui data bobot keyakinan gejala penyakit pada <i>database</i> sistem berdasarkan isi dari <i>Textfield</i> Bobot Keyakinan Gejala Penyakit di <i>Form</i> Perbarui Data Aturan

19. Sistem menampilkan kembali

Form Ubah Data Aturan, berisi:

- Tabel Ubah Data Aturan yang terdiri dari Kolom No, Penyakit, Gejala, Bobot Keyakinan, dan Aksi
- Tombol Kembali

SKENARIO ALTERNATIF

Aktor ingin memperbarui data bobot keyakinan gejala penyakit tetapi membatalkannya

16a. Aktor mengganti data bobot keyakinan gejala penyakit pada *Textfield* Bobot keyakinan gejala penyakit

17a. Aktor menekan Tombol Kembali

18a. Sistem menampilkan kembali

Halaman Kelola Data Aturan, berisi:

- Tombol Masukkan Data
- Tombol Ubah Data
- Tombol Hapus Data
- Tombol Kembali
- Tombol *Logout*
- Tabel Data Aturan yang terdiri dari Kolom No, Penyakit, Gejala, dan Bobot Keyakinan

SKENARIO ALTERNATIF

Aktor menekan Tombol Simpan tetapi *Textfield* Bobot Keyakinan Gejala Penyakit kosong

16b. Aktor menghilangkan data bobot keyakinan gejala penyakit yang berada pada *Textfield* Bobot Keyakinan gejala Penyakit

17b. Aktor menekan Tombol Simpan

18b.Sistem menampilkan pesan peringatan berisi tulisan "*Please fill out this field*" di bawah *Textfield* Bobot Keyakinan Gejala Penyakit

19b.Sistem menampilkan kembali *Form* Perbarui Data Aturan, berisi:

- *Textfield* Bobot Keyakinan Gejala Penyakit yang berisi data bobot keyakinan gejala penyakit yang dipilih sebelumnya
 - Tombol Simpan
 - Tombol Kembali
-

SKENARIO NORMAL

Aktor menghapus data aturan

20. Jika aktor ingin menghapus data aturan, maka aktor menekan Tombol Hapus Data

21. Sistem menampilkan *Form* Hapus Data Aturan, berisi:

- Tabel Hapus Data Aturan yang terdiri dari Kolom No,
-

Penyakit, Gejala, Bobot
Keyakinan, dan Aksi
- Tombol Kembali

22. Aktor menekan aksi Hapus pada
Kolom Aksi sesuai dengan data
aturan yang ingin dihapus

23. Sistem melakukan proses
menghapus data aturan pada
database sistem berdasarkan
pilihan di *Form* Hapus Data
Aturan

24. Sistem menampilkan kembali
Form Hapus Data Aturan, berisi:
- Tabel Hapus Data Aturan yang
terdiri dari Kolom No,
Penyakit, Gejala, Bobot
Keyakinan, dan Aksi
- Tombol Kembali

SKENARIO ALTERNATIF

Aktor ingin menghapus data aturan tetapi membatalkannya

22a. Aktor menekan Tombol Kembali

23a. Sistem menampilkan kembali
Halaman Kelola Data Aturan,
berisi:
- Tombol Masukkan Data
- Tombol Ubah Data
- Tombol Hapus Data
- Tombol Kembali
- Tombol *Logout*

-
- Tabel Data Aturan yang terdiri dari Kolom No, Penyakit, Gejala, dan Bobot Keyakinan
-

Sumber: Hasil analisis (2018).

6. Use Case Scenario Mendiagnosis Penyakit

Nama Use Case	Mendiagnosis Penyakit
Aktor	Pengunjung
Entry Condition	Aktor ingin mengetahui kondisi kesehatan anjing peliharaannya
Exit Condition	Aktor mendapatkan hasil diagnosis penyakit anjing peliharaannya
Aksi Aktor	Reaksi Sistem
SKENARIO NORMAL	

1. Aktor mengakses Halaman Pengunjung
-

2. Sistem menampilkan halaman Pengunjung, berisi:
 - Menu Home
 - Menu Diagnosis
-

3. Aktor memilih Menu Diagnosis
-

4. Sistem menampilkan halaman Diagnosis, berisi:
 - Checkbox Gejala Penyakit
 - Tombol Selanjutnya
-

5. Aktor memilih *Checkbox* Gejala Penyakit
-

6. Aktor menekan Tombol Selanjutnya
-

7. Sistem menampilkan:
-

-
- Tabel yang terdiri dari Kolom Gejala yang berisi data gejala penyakit yang dipilih sebelumnya, dan Kolom Bobot Keyakinan Gejala yang berisi *Textfield* Bobot Keyakinan Gejala Penyakit sesuai gejala penyakit yang dipilih
 - Bagan Keterangan yang berisi panduan cara mengisi bobot keyakinan gejala
 - Tombol Proses
 - Tombol Kembali
-

8. Aktor mengisi *Textfield* Bobot Keyakinan Gejala sesuai gejalanya masing-masing

9. Aktor menekan Tombol Proses

10. Sistem mengambil data dari *Textfield* Bobot Keyakinan Gejala dari pengunjung dan data bobot keyakinan gejala penyakit dari sistem lalu menghitungnya menggunakan metode *certainty factor*

11. Sistem menampilkan:

- Hasil Diagnosis Penyakit
- Tombol Selesai

12. Aktor menekan Tombol Selesai

13. Sistem menampilkan kembali Halaman Pengunjung, berisi

-
- Menu Home
 - Menu Diagnosis

SKENARIO ALTERNATIF**Terdapat gejala penyakit yang belum dipilih oleh Aktor**

5a. Aktor memilih *Checkbox* Gejala Penyakit

6a. Aktor menekan Tombol Selanjutnya

7a. Sistem menampilkan:

- Tabel yang terdiri dari Kolom Gejala yang berisi data gejala penyakit yang dipilih sebelumnya, dan Kolom Bobot Keyakinan Gejala yang berisi *Textfield* Bobot Keyakinan Gejala Penyakit sesuai gejala penyakit yang dipilih
- Bagan Keterangan yang berisi panduan cara mengisi bobot keyakinan gejala
- Tombol Proses
- Tombol Kembali

8a. Aktor menekan Tombol Kembali

9a. Sistem menampilkan kembali

- Halaman Diagnosis, berisi:
- *Checkbox* Gejala Penyakit
 - Tombol Selanjutnya

SKENARIO ALTERNATIF

Aktor menekan Tombol Proses tetapi *Textfield* Bobot Keyakinan Gejala kosong

8a. Aktor tidak mengisi *Textfield* Bobot

Keyakinan Gejala

9a. Aktor menekan Tombol Proses

10a. Sistem menampilkan pesan peringatan berisi tulisan “*Please fill out this field*” di bawah *Textfield* Bobot Keyakinan Gejala

11a. Sistem menampilkan kembali:

- Tabel yang terdiri dari Kolom Gejala yang berisi data gejala penyakit yang dipilih sebelumnya, dan Kolom Bobot Keyakinan Gejala yang berisi *Textfield* Bobot Keyakinan Gejala Penyakit sesuai gejala penyakit yang dipilih
- Bagan Keterangan yang berisi panduan cara mengisi bobot keyakinan gejala
- Tombol Proses
- Tombol Kembali

Sumber: Hasil analisis (2018).

Lampiran B. Hasil Penulisan Kode Program

1. Hasil Kode Program Fitur Login

a. Screenshot kode program fitur login pada kelas view(v_login)

```

27 <div class="card">
28 <div class="body">
29 <form action="{php echo base_url('index.php/login/aksi_login'); ?}" id="sign_in" method="post">
30 <div class="msg">Silahkan isi data berikut untuk login ke sistem</div>
31 <div class="input-group">
32 <input class="input-group-addon">
33 <i class="material-icons">person</i>
34 </input>
35 <div class="form-line">
36 <input type="text" class="form-control" name="username" placeholder="Username" required autofocus>
37 </div>
38 </div>
39 <div class="input-group">
40 <input class="input-group-addon">
41 <i class="material-icons">lock</i>
42 </input>
43 <div class="form-line">
44 <input type="password" class="form-control" name="password" placeholder="Password" required>
45 </div>
46 </div>
47 <?php
48 if (($this->session->flashdata('message')) != null)
49 {
50 echo <div class="alert alert-'. $this->session->flashdata('type') .'" role="alert">;
51 echo <a href="login" class="close" data-dismiss="alert" aria-label="close">&times;</a>;
52 echo $this->session->flashdata('message') . ' ? $this->session->flashdata('message') : '';
53 echo </div>;
54 }
55 }>
56 <div class="row">
57 <div class="col-xs-8 p-t-5">
58 </div>
59 <div class="col-xs-4">
60 <button class="btn btn-block bg-indigo waves-effect" type="submit" value="Login">Login</button>
61 </div>
62 </div>
63 </form>
64 </div>
65 </div>
66 </div>

```

(Sumber: Hasil analisis, 2018)

b. Screenshot kode program fitur login pada kelas controller(login)

```

19
20 function aksi_login(){
21 $username = $this->input->post('username');
22 $password = $this->input->post('password');
23 $where = array(
24 'username' => $username,
25 'password' => md5($password)
26 );
27 $cek = $this->m_login->ceklogin($where)->num_rows();
28 if($cek > 0){
29 $data_session = array(
30 'nama' => $username,
31 'status' => "login"
32 );
33 $this->session->set_userdata($data_session);
34 redirect(base_url("index.php/autentikasi"));
35 }else{
36 $this->session->set_flashdata('message','Username atau password salah!');
37 $this->session->set_flashdata('type','danger');
38 redirect('login');
39 }
40 }
41

```

(Sumber: Hasil analisis, 2018)

c. Screenshot kode program fitur *login* pada kelas *model(m_login)*

```

2
3 class M_login extends CI_Model{
4
5     function ceklogin($where){
6         return $this->db->get_where('d3es_user',$where);
7     }
8

```

(Sumber: Hasil analisis, 2018)

d. Screenshot kode program fitur *login* pada kelas *controller*(autentikasi)

```

5
6     function __construct(){
7         parent::__construct();
8         if($this->session->userdata('status') != "login"){
9             redirect(base_url("index.php/login"));
10        }
11    }
12
13    function index(){
14        $this->load->view('v_dashboard');
15    }
16

```

(Sumber: Hasil analisis, 2018)

2. Hasil Kode Program Fitur *Logout*

a. Screenshot kode program fitur *logout* pada kelas *view*(v_dashboard)

```

35 <div class="collapse navbar-collapse" id="navbar-collapse">
36 <ul class="nav navbar-nav navbar-right">
37 <li class="Logout">
38 <a href="php echo base_url('index.php/login/logout'); ?" class="btn bg-indigo waves-effect">Logout</a>
39 </li>
40 </ul>
41 </div>

```

(Sumber: Hasil analisis, 2018)

b. Screenshot kode program fitur *logout* pada kelas *view*(v_keloladatapenyakit)

```

37 <div class="collapse navbar-collapse" id="navbar-collapse">
38 <ul class="nav navbar-nav navbar-right">
39 <li class="Logout">
40 <a href="php echo base_url('index.php/login/logout'); ?" class="btn bg-indigo waves-effect">Logout</a>
41 </li>
42 </ul>
43 </div>

```

(Sumber: Hasil analisis, 2018)

c. Screenshot kode program fitur *logout* pada kelas *view*(v_keloladatagejala)

```

37 <div class="collapse navbar-collapse" id="navbar-collapse">
38 <ul class="nav navbar-nav navbar-right">
39 <li class="Logout">
40 <a href="php echo base_url('index.php/login/logout'); ?" class="btn bg-indigo waves-effect">Logout</a>
41 </li>
42 </ul>
43 </div>

```

(Sumber: Hasil analisis, 2018)

d. Screenshot kode program fitur *logout* pada kelas *view(v_keloladaturan)*

```

37 <div class="collapse navbar-collapse" id="navbar-collapse">
38 <ul class="nav navbar-nav navbar-right">
39 <li class="logout">
40 <a href="{?php echo base_url('index.php/login/logout'); ?}" class="btn bg-indigo waves-effect">Logout</a>
41 </li>
42 </ul>
43 </div>

```

(Sumber: Hasil analisis, 2018)

e. Screenshot kode program fitur *logout* pada kelas *controller(login)*

```

18
19
20 function index(){
21     $this->load->view('v_login');
22 }
23
24
25 function logout(){
26     $this->session->sess_destroy();
27     redirect(base_url('index.php/login'));
28 }
29

```

(Sumber: Hasil analisis, 2018)

3. Hasil Kode Program Fitur Kelola Data Penyakit

a. Screenshot kode program fitur kelola data penyakit pada kelas *view(v_dashboard)*

```

72 </li>
73 <a href="javascript:void(0);" class="menu-toggle">
74 <i class="material-icons">local_hospital</i>
75 <span>Penyakit</span>
76 </a>
77 <ul class="ml-menu">
78 <li>
79 <a href="{?php echo base_url('index.php/kelolapenyakit/tampilpenyakit'); ?}">Kelola Data</a>
80 </li>
81 </ul>
82 </li>

```

(Sumber: Hasil analisis, 2018)

b. Screenshot kode program fitur kelola data penyakit pada kelas *view(v_keloladatapenyakit)*

```

127 <div class="block-header">
128 <h2>KELOLA DATA PENYAKIT</h2>
129 </div>
130 <div class="row">
131 <div class="row clearfix">
132 <div class="col-lg-12 col-md-12 col-sm-12 col-xs-12">
133 <div class="card">
134 <div class="header">
135 <h3>Menu Kelola Data Penyakit</h3>
136 </div>
137 <div class="body">
138 <a href="{?php echo base_url('index.php/kelolapenyakit/tampilmasipenyakit'); ?}" class="btn bg-indigo waves-effect">Masukkan Data</a>
139 <a href="{?php echo base_url('index.php/kelolapenyakit/tampilubahpenyakit'); ?}" class="btn bg-green waves-effect">Ubah Data</a>
140 <a href="{?php echo base_url('index.php/kelolapenyakit/tampilhapuspenyakit'); ?}" class="btn bg-red waves-effect">Hapus Data</a>
141 </div>
142 </div>
143 </div>
144 </div>
145 </div>

```

```

146 <!-- Tabel Data Penyakit -->
147 <div class="row clearfix">
148 <div class="col-lg-12 col-md-12 col-sm-12 col-xs-12">
149 <div class="card">
150 <div class="header">
151 <h2>Tabel Data Penyakit</h2>
152 </div>
153 <div class="body">
154 <div class="table-responsive">
155 <table class="table table-bordered table-striped table-hover dataTable">
156 <thead>
157 <tr>
158 <th>No</th>
159 <th>Penyakit</th>
160 </tr>
161 </thead>
162 <tbody>
163 <tr>
164 <td><?php echo $tp->id disease ?</td>
165 <td><?php echo $tp->disease ?</td>
166 </tr>
167 </tbody>
168 </table>
169 <?php } ?>
170 </div>
171 </div>
172 </div>
173 </div>
174 </div>
175 </div>
176 <!-- SEHIDE Tabel Data Penyakit -->
177 <!-- Tombol Kembali -->
178 <a href="<?php echo base_url('index.php/autentikasi'); ?>" class="btn bg-indigo waves-effect right">Kembali</a>
179 <!-- akhir Tombol Kembali -->

```

(Sumber: Hasil analisis, 2018)

- c. Screenshot kode program fitur kelola data penyakit pada kelas `view(v_masukdatapenyakit)`

```

130 <!-- Masukan Data Penyakit -->
131 <div class="row clearfix">
132 <div class="col-lg-12 col-md-12 col-sm-12 col-xs-12">
133 <div class="card">
134 <div class="header">
135 <h2>Masukkan Data Penyakit</h2>
136 </div>
137 <div class="body">
138 <form action="<?php echo base_url('index.php/kelolapenyakit/masukpenyakit'); ?>" method="post">
139 <label for="disease">Penyakit</label>
140 <div class="form-group">
141 <div class="form-line">
142 <input type="text" class="form-control" name="disease" placeholder="Tulis Nama Penyakit Disini" required>
143 </div>
144 </div>
145 <button type="submit" class="btn btn-primary m-t-15 waves-effect">Simpan</button>
146 </form>
147 </div>
148 </div>
149 </div>
150 </div>
151 <!-- SEHIDE Masukan Data Penyakit -->
152 <!-- Tombol Kembali -->
153 <a href="<?php echo base_url('index.php/kelolapenyakit/tampilpenyakit'); ?>" class="btn bg-indigo waves-effect right">Kembali</a>
154 <!-- akhir Tombol Kembali -->

```

(Sumber: Hasil analisis, 2018)

- d. Screenshot kode program fitur kelola data penyakit pada kelas `view(v_ubahdatapenyakit)`

```
130 <!-- Ubah Data Penyakit -->
131 <div class="row clearfix">
132 <div class="col-lg-12 col-md-12 col-sm-12 col-xs-12">
133 <div class="card">
134 <div class="header">
135 <h2>Ubah Data Penyakit</h2>
136 </div>
137 <div class="body">
138 <div class="table-responsive">
139 <table class="table table-bordered table-striped table-hover dataTable">
140 <thead>
141 <tr>
142 <th>No</th>
143 <th>Penyakit</th>
144 <th>Aksi</th>
145 </tr>
146 </thead>
147 <tbody>
148 <?php foreach($tampilubahpenyakit as $stup){ ?>
149 <tr>
150 <td><?php echo $stup->id_disease ?></td>
151 <td><?php echo $stup->disease ?></td>
152 <td><?php echo anchor('kelolapenyakit/ubahpenyakit/' . $stup->id_disease, 'Ubah'); ?></td>
153 </tr>
154 </tbody>
155 </table>
156 </div>
157 </div>
158 </div>
159 </div>
160 </div>
161 </div>
162 <!-- END Ubah Data Penyakit -->
163 <!-- Tombol Kembali -->
164 <a href="<?php echo base_url('index.php/kelolapenyakit/tampilpenyakit'); ?>" class="btn bg-indigo waves-effect right">Kembali</a>
165 <!-- END Tombol Kembali -->
```

(Sumber: Hasil analisis, 2018)

- e. Screenshot kode program fitur kelola data penyakit pada kelas `view(v_perbaruidatapenyakit)`

```
130 <!-- Perbarui Data Penyakit -->
131 <div class="row clearfix">
132 <div class="col-lg-12 col-md-12 col-sm-12 col-xs-12">
133 <div class="card">
134 <div class="header">
135 <h2>Perbarui Data Penyakit</h2>
136 </div>
137 <div class="body">
138 <?php foreach($ubahpenyakit as $sup){ ?>
139 <form action="<?php echo base_url('index.php/kelolapenyakit/perbaruidatapenyakit'); ?>" method="post">
140 <input type="hidden" name="id_disease" value="<?php echo $sup->id_disease ?>">
141 <input type="text" class="form-control" name="disease" value="<?php echo $sup->disease ?>" required>
142 </div>
143 <div class="form-group">
144 <input type="submit" class="btn btn-primary w-t-15 waves-effect">Simpan</div>
145 </form>
146 </div>
147 </div>
148 </div>
149 </div>
150 </div>
151 </div>
152 </div>
153 </div>
154 </div>
155 <!-- END Perbarui Data Penyakit -->
156 <!-- Tombol Kembali -->
157 <a href="<?php echo base_url('index.php/kelolapenyakit/tampilpenyakit'); ?>" class="btn bg-indigo waves-effect right">Kembali</a>
158 <!-- END Tombol Kembali -->
```

(Sumber: Hasil analisis, 2018)

- f. Screenshot kode program fitur kelola data penyakit pada kelas *view(v_hapusdatapenyakit)*

```

130 <!-- Hapus Data Penyakit -->
131 <div class="row clearfix">
132 <div class="col-lg-12 col-md-12 col-sm-12 col-xs-12">
133 <div class="card">
134 <div class="header">
135 <h2>Hapus Data Penyakit</h2>
136 </div>
137 <div class="body">
138 <div class="table-responsive">
139 <table class="table table-bordered table-striped table-hover dataTable">
140 <thead>
141 <tr>
142 <th>No</th>
143 <th>Penyakit</th>
144 <th>Aksi</th>
145 </tr>
146 </thead>
147 <tbody>
148 <?php foreach($tampilhapuspenyakit as $thp) { ?>
149 <tr>
150 <td><?php echo $thp->id_disease ?></td>
151 <td><?php echo $thp->disease ?></td>
152 <td><?php echo anchor('kelolapenyakit/hapuspenyakit/' . $thp->id_disease, 'Hapus'); ?></td>
153 </tr>
154 </tbody>
155 </table>
156 </div>
157 </div>
158 </div>
159 </div>
160 </div>
161 <!-- #END# Hapus Data Penyakit -->
162 <!-- Tombol Kembali -->
163 <a href="<?php echo base_url('index.php/kelolapenyakit/tampilpenyakit'); ?>" class="btn bg-indigo waves-effect right">Kembali</a>
164 <!-- #END# Tombol Kembali -->

```

(Sumber: Hasil analisis, 2018)

- g. Screenshot kode program fitur kelola data penyakit pada kelas *controller(kelolapenyakit)*

```

3
4 class Kelolapenyakit extends CI_Controller {
5
6     function __construct(){
7         parent::__construct();
8         $this->load->model('m_datapenyakit');
9     }
10
11     function tampilpenyakit(){
12         $data['tampilpenyakit'] = $this->m_datapenyakit->tampildatapenyakit()->result();
13         $this->load->view('v_keloladatapenyakit',$data);
14     }
15
16     function tampilmasukpenyakit(){
17         $this->load->view('v_masukdatapenyakit');
18     }
19
20     function masukpenyakit(){
21         $disease = $this->input->post('disease');
22
23         $data = array(
24             'disease' => $disease
25         );
26         $this->m_datapenyakit->masukdatapenyakit($data);
27         redirect('kelolapenyakit/tampilmasukpenyakit');
28     }
29
30     function tampilubahpenyakit(){
31         $data['tampilubahpenyakit'] = $this->m_datapenyakit->tampildatapenyakit()->result();
32         $this->load->view('v_ubahdatapenyakit',$data);
33     }
34
35     function ubahpenyakit($id_disease){
36         $where = array('id_disease' => $id_disease);
37         $data['ubahpenyakit'] = $this->m_datapenyakit->ubahdatapenyakit($where)->result();
38         $this->load->view('v_perbaruidatapenyakit',$data);
39     }
40

```

```

48
41 function perbaruiPenyakit(){
42     $id_disease = $this->input->post('id_disease');
43     $disease = $this->input->post('disease');
44
45     $data = array(
46         'disease' => $disease
47     );
48
49     $where = array(
50         'id_disease' => $id_disease
51     );
52
53     $this->m_datapenyakit->perbaruidatapenyakit($where,$data);
54     redirect('kelolapenyakit/tampilubahpenyakit');
55 }
56
57 function tampilhapuspenyakit(){
58     $data['tampilhapuspenyakit'] = $this->m_datapenyakit->tampildatapenyakit()->result();
59     $this->load->view('v_hapusdatapenyakit',$data);
60 }
61
62 function hapuspenyakit($id_disease){
63     $where = array('id_disease' => $id_disease);
64     $this->m_datapenyakit->hapusdatapenyakit($where);
65     redirect('kelolapenyakit/tampilhapuspenyakit');
66 }
67

```

(Sumber: Hasil analisis, 2018)

- h. Screenshot kode program fitur kelola data penyakit pada kelas *model(m_datapenyakit)*

```

2
3 class M_datapenyakit extends CI_Model{
4
5     function tampildatapenyakit(){
6         return $this->db->get('d3es_disease');
7     }
8
9     function masukdatapenyakit($data){
10        $this->db->insert('d3es_disease',$data);
11    }
12
13    function ubahdatapenyakit($where){
14        return $this->db->get_where('d3es_disease',$where);
15    }
16
17    function perbaruidatapenyakit($where,$data){
18        $this->db->where($where);
19        $this->db->update('d3es_disease',$data);
20    }
21
22    function hapusdatapenyakit($where){
23        $this->db->where($where);
24        $this->db->delete('d3es_disease');
25    }
26

```

(Sumber: Hasil analisis, 2018)

4. Hasil Kode Program Fitur Kelola Data Gejala

a. Screenshot kode program fitur kelola data gejala pada kelas `view(v_dashboard)`

```
83
84
85
86
87
88
89
90
91
92
93
<!--
< a href="javascript:void(0);" class="menu-toggle"
< class="material-icons">add_box</i>
<span>Gejala</span>
</a>
<li class="ml-menu">
<!--
< a href="{?php echo base_url('index.php/kelolagejala/tampilgejala')} ?>">Kelola Data</a>
</li>
</li>
-->
```

(Sumber: Hasil analisis, 2018)

b. Screenshot kode program fitur kelola data gejala pada kelas `view(v_keloladatagejala)`

```
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
<!-- block-header -->
<h2>KELOLA DATA GEJALA</h2>
</div>
<!-- Menu Kelola Data Gejala -->
<div class="row clearfix">
<div class="col-lg-12 col-md-12 col-sm-12 col-xs-12">
<div class="card">
<div class="header">
<h3>Menu Kelola Data Gejala</h3>
</div>
<div class="body">
< a href="{?php echo base_url('index.php/kelolagejala/tampilmasugejala')} ?>" class="btn bg-indigo waves-effect">Masukkan Data</a>
< a href="{?php echo base_url('index.php/kelolagejala/tampilubahgejala')} ?>" class="btn bg-green waves-effect">Ubah Data</a>
< a href="{?php echo base_url('index.php/kelolagejala/tampilhapusgejala')} ?>" class="btn bg-red waves-effect">Hapus Data</a>
</div>
</div>
</div>
</div>
</div>
<!-- ENDE Menu Kelola Data Gejala -->
```

```
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
<!-- Tabel Data Gejala -->
<div class="row clearfix">
<div class="col-lg-12 col-md-12 col-sm-12 col-xs-12">
<div class="card">
<div class="header">
<h3>Tabel Data Gejala</h3>
</div>
<div class="body">
<div class="table-responsive">
<table class="table table-bordered table-striped table-hover dataTable">
<thead>
<tr>
<th>No</th>
<th>Gejala</th>
</tr>
</thead>
<tbody>
<?php foreach($tampilgejala as $stg){ ?>
<tr>
<td>{?php echo $stg->id_symptom ?}</td>
<td>{?php echo $stg->symptom ?}</td>
</tr>
</tbody>
<?php } ?>
</table>
</div>
</div>
</div>
</div>
</div>
<!-- ENDE Tabel Data Gejala -->
<!-- Tombol Kembali -->
< a href="{?php echo base_url('index.php/autentikasi')} ?>" class="btn bg-indigo waves-effect right">Kembali</a>
<!-- ENDE Tombol Kembali -->
```

(Sumber: Hasil analisis, 2018)

- c. Screenshot kode program fitur kelola data gejala pada kelas `view(v_masukdatagejala)`

```

130 <!-- Masukan Data Gejala -->
131 <div class="row clearfix">
132 <div class="col-lg-12 col-md-12 col-sm-12 col-xs-12">
133 <div class="card">
134 <div class="header">
135 <h2>Masukkan Data Gejala</h2>
136 </div>
137 <div class="body">
138 <form action="{php echo base_url('index.php/kelolagejala/masukgejala'); }" method="post">
139 <table border="1" class="table">
140 <tr>
141 <td class="form-line">
142 <input type="text" class="form-control" name="sympom" placeholder="Tulis Nama Gejala Disini" required>
143 </td>
144 </tr>
145 <tr>
146 <td colspan="2" style="text-align: center;>
147 <button type="submit" class="btn btn-primary m-t-15 waves-effect">Simpan</button>
148 </td>
149 </tr>
150 </table>
151 </div>
152 </div>
153 <div class="text-align: center;">
154 <button href="{php echo base_url('index.php/kelolagejala/tampilgejala'); }" class="btn bg-indigo waves-effect right">Kembali</button>
155 </div>
156 </div>
157 </div>
158 </div>
159 </div>
160 </div>
161 </div>
162 </div>
163 </div>
164 </div>
165 </div>

```

(Sumber: Hasil analisis, 2018)

- d. Screenshot kode program fitur kelola data gejala pada kelas `view(v_ubahdatagejala)`

```

130 <!-- Ubah Data Gejala -->
131 <div class="row clearfix">
132 <div class="col-lg-12 col-md-12 col-sm-12 col-xs-12">
133 <div class="card">
134 <div class="header">
135 <h2>Ubah Data Gejala</h2>
136 </div>
137 <div class="body">
138 <div class="table-responsive">
139 <table class="table table-bordered table-striped table-hover dataTable">
140 <thead>
141 <tr>
142 <th>No</th>
143 <th>Gejala</th>
144 <th>Aksi</th>
145 </tr>
146 </thead>
147 <tbody>
148 <tr>
149 <td>
150 <input type="text" value="{php echo $tug->id_sympom }">
151 <input type="text" value="{php echo $tug->sympom }">
152 <input type="button" value="{php echo anchor('kelolagejala/ubahgejala/' . $tug->id_sympom, 'Ubah') }">
153 </td>
154 </tr>
155 </tbody>
156 </table>
157 </div>
158 </div>
159 </div>
160 </div>
161 </div>
162 </div>
163 </div>
164 </div>
165 </div>

```

(Sumber: Hasil analisis, 2018)

- e. Screenshot kode program fitur kelola data gejala pada kelas `view(v_perbaruidatagejala)`

```
130 <!-- Perbarui Data Gejala -->
131 <div class="row clearfix">
132 <div class="col-lg-12 col-md-12 col-sm-12 col-xs-12">
133 <div class="card">
134 <div class="header">
135 <h2>Perbarui Data Gejala</h2>
136 </div>
137 <div class="body">
138 <?php foreach($ubahgejala as $ug){ ?>
139 <form action="{php echo base_url('index.php/kelolagejala/perbaruigejala'); }" method="post">
140 <label for="symptom">Gejala</label>
141 <div class="form-group">
142 <input type="hidden" name="id_symptom" value="{php echo $ug->id_symptom }">
143 <input type="text" class="form-control" name="symptom" value="{php echo $ug->symptom }" required>
144 </div>
145 </div>
146 <button type="submit" class="btn btn-primary w-15 waves-effect">Simpan</button>
147 </form>
148 <?php } ?>
149 </div>
150 </div>
151 </div>
152 <div class="row">
153 <div class="col">
154 <!-- SIMPAN Perbarui Data Gejala -->
155 <!-- Tombol Kembali -->
156 <a href="{php echo base_url('index.php/kelolagejala/tampilgejala'); }" class="btn bg-indigo waves-effect right">Kembali</a>
157 <!-- akhir Tombol Kembali -->
```

(Sumber: Hasil analisis, 2018)

- f. Screenshot kode program fitur kelola data gejala pada kelas `view(v_hapusdatagejala)`

```
130 <!-- Hapus Data Gejala -->
131 <div class="row clearfix">
132 <div class="col-lg-12 col-md-12 col-sm-12 col-xs-12">
133 <div class="card">
134 <div class="header">
135 <h2>Hapus Data Gejala</h2>
136 </div>
137 <div class="body">
138 <div class="table-responsive">
139 <table class="table table-bordered table-striped table-hover dataTable">
140 <thead>
141 <tr>
142 <th>No</th>
143 <th>Gejala</th>
144 <th>Aksi</th>
145 </tr>
146 </thead>
147 <tbody>
148 <?php foreach($tampilhapusgejala as $thg){ ?>
149 <tr>
150 <td>{php echo $thg->id_symptom }</td>
151 <td>{php echo $thg->symptom }</td>
152 <td>{php echo anchor('kelolagejala/hapusgejala/' . $thg->id_symptom, 'Hapus'); }</td>
153 </tr>
154 </tbody>
155 <?php } ?>
156 </table>
157 </div>
158 </div>
159 </div>
160 </div>
161 </div>
162 <div class="row">
163 <div class="col">
164 <!-- akhir Hapus Data Gejala -->
165 <!-- Tombol Kembali -->
166 <a href="{php echo base_url('index.php/kelolagejala/tampilgejala'); }" class="btn bg-indigo waves-effect right">Kembali</a>
167 <!-- akhir Tombol Kembali -->
```

(Sumber: Hasil analisis, 2018)

- g. Screenshot kode program fitur kelola data gejala pada kelas *controller(kelolagejala)*

```
4 class Kelolagejala extends CI_Controller {
5
6     function __construct(){
7         parent::__construct();
8         $this->load->model('m_datagejala');
9     }
10
11    function tampilgejala(){
12        $data['tampilgejala'] = $this->m_datagejala->tampildatagejala()->result();
13        $this->load->view('v_keloladatagejala',$data);
14    }
15
16    function tampilmasukgejala(){
17        $this->load->view('v_masukdatagejala');
18    }
19
20    function masukgejala(){
21        $symptom = $this->input->post('symptom');
22
23        $data = array(
24            'symptom' => $symptom
25        );
26        $this->m_datagejala->masukdatagejala($data);
27        redirect('kelolagejala/tampilmasukgejala');
28    }
29
30    function tampilubahgejala(){
31        $data['tampilubahgejala'] = $this->m_datagejala->tampildatagejala()->result();
32        $this->load->view('v_ubahdatagejala',$data);
33    }
34
35    function ubahgejala($id_symptom){
36        $where = array('id_symptom' => $id_symptom);
37        $data['ubahgejala'] = $this->m_datagejala->ubahdatagejala($where)->result();
38        $this->load->view('v_perbaruiddatagejala',$data);
39    }
40
41    function perbaruigejala(){
42        $id_symptom = $this->input->post('id_symptom');
43        $symptom = $this->input->post('symptom');
44
45        $data = array(
46            'symptom' => $symptom
47        );
48
49        $where = array(
50            'id_symptom' => $id_symptom
51        );
52
53        $this->m_datagejala->perbaruiddatagejala($where,$data);
54        redirect('kelolagejala/tampilubahgejala');
55    }
56
57    function tampilhapusgejala(){
58        $data['tampilhapusgejala'] = $this->m_datagejala->tampildatagejala()->result();
59        $this->load->view('v_hapusdatagejala',$data);
60    }
61
62    function hapusgejala($id_symptom){
63        $where = array('id_symptom' => $id_symptom);
64        $this->m_datagejala->hapusdatagejala($where);
65        redirect('kelolagejala/tampilhapusgejala');
66    }
67
```

(Sumber: Hasil analisis, 2018)

- h. *Screenshot* kode program fitur kelola data gejala pada kelas *model(m_datagejala)*

```
3 class M_datagejala extends CI_Model{
4
5     function tampildatagejala(){
6         return $this->db->get('d3es_symptom');
7     }
8
9     function masukdatagejala($data){
10        $this->db->insert('d3es_symptom',$data);
11    }
12
13    function ubahdatagejala($where){
14        return $this->db->get_where('d3es_symptom',$where);
15    }
16
17    function perbaruidatagejala($where,$data){
18        $this->db->where($where);
19        $this->db->update('d3es_symptom',$data);
20    }
21
22    function hapusdatagejala($where){
23        $this->db->where($where);
24        $this->db->delete('d3es_symptom');
25    }
26
```

(Sumber: Hasil analisis, 2018)

5. Hasil Kode Program Fitur Kelola Data Aturan

- a. *Screenshot* kode program fitur kelola data aturan pada kelas *view(v_dashboard)*

```
54 </li>
55 <li href="javascript:void(0);" class="menu-toggle">
56     <i class="material-icons">library_books</i>
57     <span>Aturan</span>
58 </li>
59 <li class="ml-menu">
60     <li>
61         <a href="<?php echo base_url('index.php/kelolaaturan/tampilaturan'); ?>">Kelola Data</a>
62     </li>
63 </li>
64 </li>
```

(Sumber: Hasil analisis, 2018)

b. *Screenshot* code program fitur kelola data aturan pada kelas `view(v_keloladataaturan)`

```

127 <div class="black-header">
128 <h2>KELOLA DATA ATURAN</h2>
129 </div>
130 <!-- Menu Kelola Data Aturan -->
131 <div class="row clearfix">
132 <div class="col-lg-12 col-md-12 col-sm-12 col-xs-12">
133 <div class="card">
134 <div class="header">
135 <h3>Menu Kelola Data Aturan</h3>
136 </div>
137 <div class="body">
138 <a href="{php echo base_url('index.php/kelolaaturan/tampilmasukaturan')}"; ?>" class="btn bg-indigo waves-effect">Masukkan Data</a>
139 <a href="{php echo base_url('index.php/kelolaaturan/tampilubahaturan')}"; ?>" class="btn bg-green waves-effect">Ubah Data</a>
140 <a href="{php echo base_url('index.php/kelolaaturan/tampilhapusaturan')}"; ?>" class="btn bg-red waves-effect">Hapus Data</a>
141 </div>
142 </div>
143 </div>
144 </div>
145 <!-- @END Menu Kelola Data Aturan -->

```

```

146 <!-- Tabel Data Aturan -->
147 <div class="row clearfix">
148 <div class="col-lg-12 col-md-12 col-sm-12 col-xs-12">
149 <div class="card">
150 <div class="header">
151 <h3>Tabel Data Aturan</h3>
152 </div>
153 <div class="body">
154 <div class="table-responsive">
155 <table class="table table-bordered table-striped table-hover dataTable">
156 <thead>
157 <tr>
158 <th>No</th>
159 <th>Penyakit</th>
160 <th>Gejala</th>
161 <th>Bobot Keyakinan</th>
162 </tr>
163 </thead>
164 <tbody>
165 <tr>
166 <td>{php echo $ta->id_rule ?}</td>
167 <td>{php echo $ta->disease ?}</td>
168 <td>{php echo $ta->symptom ?}</td>
169 <td>{php echo $ta->rule_value ?}</td>
170 </tr>
171 </tbody>
172 </table>
173 <div class="text-align: right">
174 <div class="text-align: right">
175 <div class="text-align: right">
176 <div class="text-align: right">
177 <div class="text-align: right">
178 <div class="text-align: right">
179 <div class="text-align: right">
180 <div class="text-align: right">
181 <div class="text-align: right">
182 <div class="text-align: right">
183 <div class="text-align: right">

```

(Sumber: Hasil analisis, 2018)

c. *Screenshot* code program fitur kelola data aturan pada kelas `view(v_masukdataaturan)`

```

130 <!-- Masukan Data Aturan -->
131 <div class="row clearfix">
132 <div class="col-lg-12 col-md-12 col-sm-12 col-xs-12">
133 <div class="card">
134 <div class="header">
135 <h3>Masukkan Data Aturan</h3>
136 </div>
137 <div class="body">
138 <form action="{php echo base_url('index.php/kelolaaturan/masukkanaturan')}"; ?>" method="post">
139 <div class="form-group">
140 <div class="form-line">
141 <input type="text" class="form-control" name="id_disease" placeholder="Tulis Kode Penyakit Disini" required>
142 </div>
143 </div>
144 <div class="form-group">
145 <div class="form-line">
146 <input type="text" class="form-control" name="id_symptom" placeholder="Tulis Kode Gejala Disini" required>
147 </div>
148 </div>
149 <div class="form-group">
150 <div class="form-line">
151 <input type="text" class="form-control" name="rule_value" placeholder="Tulis Bobot Keyakinan Gejala Penyakit Disini" required>
152 </div>
153 </div>
154 <div class="text-align: right">
155 <input type="submit" class="btn btn-primary m-t-15 waves-effect">Simpan</div>
156 </div>
157 </div>
158 </div>
159 </div>
160 </div>

```

```
161 <div class="row clearfix">
162 <div class="col-lg-12 col-md-12 col-sm-12 col-xs-12">
163 <div class="card">
164 <div class="header">
165 <h2>Daftar Kode Penyakit</h2>
166 </div>
167 <div class="body">
168 <div>
169 <table class="table">
170 <thead>
171 <tr>
172 <th>Kode</th>
173 <th>Penyakit</th>
174 </tr>
175 </thead>
176 <?php foreach($stampilpenyakit as $tp){ ?>
177 <tbody>
178 <tr>
179 <td><?php echo $tp->id_disease ?></td>
180 <td><?php echo $tp->disease ?></td>
181 </tr>
182 </tbody>
183 <?php } ?>
184 </table>
185 </div>
186 </div>
187 </div>
188 </div>
189 </div>
```

```
190 <div class="row clearfix">
191 <div class="col-lg-12 col-md-12 col-sm-12 col-xs-12">
192 <div class="card">
193 <div class="header">
194 <h2>Daftar Kode Gejala</h2>
195 </div>
196 <div class="body">
197 <div>
198 <table class="table">
199 <thead>
200 <tr>
201 <th>Kode</th>
202 <th>Gejala</th>
203 </tr>
204 </thead>
205 <?php foreach($stampilgejala as $tg){ ?>
206 <tbody>
207 <tr>
208 <td><?php echo $tg->id_symptom ?></td>
209 <td><?php echo $tg->symptom ?></td>
210 </tr>
211 </tbody>
212 <?php } ?>
213 </table>
214 </div>
215 </div>
216 </div>
217 </div>
218 </div>
219 <!-- tombol Masukkan Data Gejala -->
220 <!-- tombol kembali -->
221 <a href="<?php echo base_url('index.php/kejoalasturan/tampilaturan'); ?>" class="btn bg-indigo waves-effect right">Kembali</a>
222 <!-- tombol kembali -->
```

(Sumber: Hasil analisis, 2018)

- d. Screenshot kode program fitur kelola data aturan pada kelas `view(v_ubahdataaturan)`

```

130 <!-- Ubah Data Aturan -->
131 <div class="row clearfix">
132 <div class="col-lg-12 col-md-12 col-sm-12 col-xs-12">
133 <div class="card">
134 <div class="header">
135 <h2>Ubah Data Aturan</h2>
136 </div>
137 <div class="body">
138 <div class="table-responsive">
139 <table class="table table-bordered table-striped table-hover dataTable">
140 <thead>
141 <tr>
142 <th></th>
143 <th>Penyakit</th>
144 <th>Gejala</th>
145 <th>Robot Keyakinan</th>
146 <th>Aksi</th>
147 </tr>
148 </thead>
149 <tbody>
150 <tr>
151 <td></td>
152 <td><input type="text" value="{ $tua->id_rule }"></td>
153 <td><input type="text" value="{ $tua->disease }"></td>
154 <td><input type="text" value="{ $tua->symptom }"></td>
155 <td><input type="text" value="{ $tua->rule_value }"></td>
156 <td><input type="button" value="Ubah" /></td>
157 </tr>
158 </tbody>
159 </table>
160 </div>
161 </div>
162 </div>
163 </div>
164 </div>
165 </div>
166 <!-- END Ubah Data Aturan -->
167 <!-- Tombol Kembali -->
168 <a href="{ $php echo base_url('index.php/kelolaaturan/tampilaturan') }" class="btn bg-indigo waves-effect right">Kembali</a>
169 </div>
170 </div>
171 </div>

```

(Sumber: Hasil analisis, 2018)

- e. Screenshot kode program fitur kelola data aturan pada kelas `view(v_perbaruidataaturan)`

```

130 <!-- Perbarui Data Aturan -->
131 <div class="row clearfix">
132 <div class="col-lg-12 col-md-12 col-sm-12 col-xs-12">
133 <div class="card">
134 <div class="header">
135 <h2>Perbarui Data Aturan</h2>
136 </div>
137 <div class="body">
138 <div class="form-group">
139 <input type="text" value="{ $tua->id_rule }" />
140 <input type="text" value="{ $tua->disease }" />
141 <input type="text" value="{ $tua->symptom }" />
142 <input type="text" value="{ $tua->rule_value }" />
143 <input type="button" value="Simpan" />
144 </div>
145 </div>
146 </div>
147 </div>
148 </div>
149 </div>
150 </div>
151 </div>
152 </div>
153 </div>
154 <!-- END Perbarui Data Aturan -->
155 <!-- Tombol Kembali -->
156 <a href="{ $php echo base_url('index.php/kelolaaturan/tampilaturan') }" class="btn bg-indigo waves-effect right">Kembali</a>
157 </div>
158 </div>

```

(Sumber: Hasil analisis, 2018)


```

38     function tampilubahaturan(){
39         $data['tampilubahaturan'] = $this->dataaturan->tampildataaturan()->result();
40         $this->load->view('v_ubahdataaturan',$data);
41     }
42
43     function ubahaturan($id_rule){
44         $where = array('id_rule' => $id_rule);
45         $data['ubahaturan'] = $this->dataaturan->ubahdataaturan($where)->result();
46         $this->load->view('v_perbaruidataaturan',$data);
47     }
48
49     function perbaruiaturan(){
50         $id_rule = $this->input->post('id_rule');
51         $rule_value = $this->input->post('rule_value');
52
53         $data = array(
54             'rule_value' => $rule_value
55         );
56
57         $where = array(
58             'id_rule' => $id_rule
59         );
60
61         $this->dataaturan->perbaruidataaturan($where,$data);
62         redirect('kelolaaturan/tampilubahaturan');
63     }
64
65     function tampilhapusaturan(){
66         $data['tampilhapusaturan'] = $this->dataaturan->tampildataaturan()->result();
67         $this->load->view('v_hapusdataaturan',$data);
68     }
69
70     function hapusaturan($id_rule){
71         $where = array('id_rule' => $id_rule);
72         $this->dataaturan->hapusdataaturan($where);
73         redirect('kelolaaturan/tampilhapusaturan');
74     }
75

```

(Sumber: Hasil analisis, 2018)

- h. Screenshot kode program fitur kelola data aturan pada kelas *model(m_dataaturan)*

```

1  class M_dataaturan extends CI_Model{
2
3      function tampildataaturan(){
4          $this->db->select('dtes_rule.id_rule, dtes_disease.disease, dtes_symptom.symptom, dtes_rule.rule_value, dtes_rule.id_disease, dtes_rule.id_symptom')
5              ->from('dtes_rule')
6              ->join('dtes_disease', 'dtes_rule.id_disease = dtes_disease.id_disease')
7              ->join('dtes_symptom', 'dtes_rule.id_symptom = dtes_symptom.id_symptom');
8          return $this->db->get();
9      }
10
11     function masukdataaturan($data){
12         $this->db->insert('dtes_rule',$data);
13     }
14
15     function ubahdataaturan($where){
16         return $this->db->get_where('dtes_rule',$where);
17     }
18
19     function perbaruidataaturan($where,$data){
20         $this->db->where($where);
21         $this->db->update('dtes_rule',$data);
22     }
23
24     function hapusdataaturan($where){
25         $this->db->where($where);
26         $this->db->delete('dtes_rule');
27     }
28 }

```

(Sumber: Hasil analisis, 2018)

- i. *Screenshot* kode program fitur kelola data aturan pada kelas *model(m_datapenyakit)*

```

5     function tampildatapenyakit(){
6         return $this->db->get('d3es_disease');
7     }
8

```

(Sumber: Hasil analisis, 2018)

- j. *Screenshot* kode program fitur kelola data aturan pada kelas *model(m_datagejala)*

```

5     function tampildatagejala(){
6         return $this->db->get('d3es_symptom');
7     }
8

```

(Sumber: Hasil analisis, 2018)

6. Hasil Kode Program Fitur Mendiagnosis Penyakit

- a. *Screenshot* kode program fitur mendiagnosis penyakit pada kelas *view(v_visitor)*

```

22     <li>
23         <a class="diagnosis" href="{?php echo base_url('index.php/diagnosis/tampilvisitorgejala'); }?">Diagnosis</a>
24     </li>

```

(Sumber: Hasil analisis, 2018)

- b. *Screenshot* kode program fitur mendiagnosis penyakit pada kelas *view(v_diagnosisambilgejala)*

```

53     <div class="body">
54         <!--ambil gejala penyakit dari visitor-->
55         <form action="{?php echo base_url('index.php/diagnosis/ambilvisitorgejala'); }" method="post">
56             <div class="card-inside-title">Pilih gejala penyakit sesuai dengan informasi yang anda temukan dari pengamatan pada anjing peliharaan
57             kesayangan anda</div>
58             <div>
59                 <div>
60                     <div>
61                         <div>
62                             <div>
63                                 <div>
64                                     <div>
65                                         <div>
66                                             <div>
67                                                 <div>
68                                                     <div>
69                                                         <div>
70                                                             <div>
71                                                                 <div>
72

```

(Sumber: Hasil analisis, 2018)

- e. Screenshot kode program fitur mendiagnosis penyakit pada kelas *controller(diagnosis)*

```

4  class Diagnosis extends CI_Controller{
5
6      function __construct(){
7          parent::__construct();
8          $this->load->model('m_diagnosis');
9      }
10
11     function tampilvisitorgejala(){
12         $data['tampilvisitorgejala'] = $this->m_diagnosis->tampilvisitorstagejala()->result();
13         $this->load->view('v_diagnosisambilgejala',$data);
14     }
15
16     function ambilvisitorgejala(){
17
18         $where = "" . implode("", $ _POST['id_symptom']) ."";
19
20         $data['ambilvisitorgejala'] = $this->m_diagnosis->tampilvisitorstagejalaolehid($where)->result();
21         $this->load->view('v_diagnosisambilgejala',$data);
22     }
23
24     function ambilvisitorbobot(){
25
26         $where = "" . implode("", $ _POST['id_symptom']) ."";
27         $data['ambilvisitorbobot'] = $this->m_diagnosis->ambildataaturan($where)->result();
28         $this->load->view('v_diagnosishasilpenyakit',$data);
29     }
30
31 }
32
33 }

```

(Sumber: Hasil analisis, 2018)

- f. Screenshot kode program fitur mendiagnosis penyakit pada kelas *model(m_diagnosis)*

```

1  class M_diagnosis extends CI_Model{
2
3      function tampilvisitorstagejala(){
4          return $this->db->get('d3es_symptom');
5      }
6
7      function tampilvisitorstagejalaolehid($where){
8
9          $id = $where;
10         $this->db->select("");
11         $this->db->from('d3es_symptom');
12         $this->db->where_in('id_symptom',$id, TRUE);
13         return $this->db->get();
14     }
15
16     function ambildataaturan($where){
17
18         $id = $where;
19         $this->db->select('d3es_rule.id_rule, d3es_disease.disease, d3es_symptom.symptom, d3es_rule.rule_value, d3es_rule.id_disease, d3es_rule.id_symptom')
20         ->from('d3es_rule')
21         ->join('d3es_disease', 'd3es_rule.id_disease = d3es_disease.id_disease')
22         ->join('d3es_symptom', 'd3es_rule.id_symptom = d3es_symptom.id_symptom')
23         ->where_in('d3es_rule.id_symptom',$id, TRUE)
24         ->order_by('d3es_disease.disease', 'ASC');
25         return $this->db->get();
26     }
27
28 }
29
30 }
31
32 }

```

(Sumber: Hasil analisis, 2018)

Lampiran C. Hasil *White Box Testing* Pada Sistem1. Hasil *White Box Testing* Pada Fitur Login

a. Listing Program

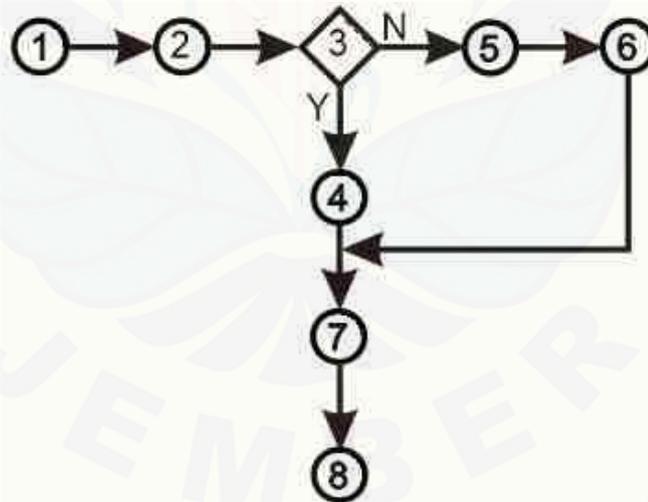
```

30 function aksi_login() { 1
31     $username = $this->input->post('username');
32     $password = $this->input->post('password');
33     $where = array(
34         'username' => $username,
35         'password' => md5($password) 2
36     );
37     $cek = $this->m_login->ceklogin($where)->num_rows();
38     if($cek > 0) { 3
39         $data_session = array(
40             'nama' => $username,
41             'status' => "login" 4
42         );
43         $this->session->set_userdata($data_session);
44         redirect(base_url("index.php/autentikasi"));
45     } else 5
46         $this->session->set_flashdata('message', 'Username atau password salah!');
47         $this->session->set_flashdata('type', 'danger');
48         redirect('login'); 6
49     }
50 } 7 8

```

(Sumber: Hasil analisis, 2018)

b. Diagram Alir



(Sumber: Hasil analisis, 2018)

c. Perhitungan *Cyclomatic Complexity*

$$V(G) = E - N + 2 = 8 - 8 + 2 = 2$$

d. Basis Set

Jalur 1 : 1 - 2 - 3 - 4 - 7 - 8

Jalur 2 : 1 - 2 - 3 - 5 - 6 - 7 - 8

e. *Test Case*

Test Case fuction aksi_login()	
Jalur 1	
<i>Test case</i>	Pengguna mengisi kolom isian <i>username</i> dan <i>password</i> pada <i>form login</i> dengan benar, kemudian menekan tombol <i>Login</i>
Target yang diharapkan	Pengguna mendapatkan status login dan masuk ke sistem
Hasil Pengujian	Benar
<i>Path/Jalur</i>	1-2-3-4-7-8
File	login.php
Jalur 2	
<i>Test case</i>	Pengguna mengisi kolom isian <i>username</i> dan <i>password</i> pada <i>form login</i> dengan salah, kemudian menekan tombol <i>Login</i>
Target yang diharapkan	Menampilkan pesan peringatan “Username atau password salah!”
Hasil Pengujian	Benar
<i>Path/Jalur</i>	1-2-3-5-6-7-8
File	login.php

Sumber: Hasil analisis (2018).

2. Hasil *White Box Testing* Pada Fitur *Logout*a. *Listing Program*

```

15 function logout() { ①
16     $this->session->sess_destroy(); ②
17     redirect(base_url('index.php/login'));
18 } ③

```

(Sumber: Hasil analisis, 2018)

b. Diagram Alir



(Sumber: Hasil analisis, 2018)

c. Perhitungan *Cyclomatic Complexity*

$$V(G) = E - N + 2 = 2 - 3 + 2 = 1$$

d. Basis Set

Jalur 1 : 1 – 2 – 3

e. *Test Case*

<i>Test Case</i> fuction aksi_logout()	
Jalur 1	
<i>Test case</i>	Pengguna menekan tombol <i>Logout</i>
Target yang diharapkan	Pengguna kehilangan status <i>login</i> dan keluar dari sistem
Hasil Pengujian	Benar
<i>Path</i> /Jalur	1-2-3
File	login.php

Sumber: Hasil analisis (2018).

3. Hasil *White Box Testing* Pada Fitur Mengelola Data Penyakit

a. Fitur Memasukkan Data Penyakit

1) *Listing* Program

```

20 function masukpenyakit() (1)
21     $disease = $this->input->post('disease');
22
23     $data = array(
24         'disease' => $disease (2)
25     );
26     $this->m_datapenyakit->masukdatapenyakit($data);
27     redirect('kelolapenyakit/tampilmasukpenyakit'); (3)
28
    
```

(Sumber: Hasil analisis, 2018)

2) Diagram Alir



(Sumber: Hasil analisis, 2018)

3) Perhitungan *Cyclomatic Complexity*

$$V(G) = E - N + 2 = 2 - 3 + 2 = 1$$

4) Basis Set

Jalur 1 : 1 – 2 – 3

5) *Test Case*

<i>Test Case fuction masukpenyakit()</i>	
Jalur 1	
<i>Test case</i>	Pengguna mengisi kolom isian masukkan data penyakit, kemudian menekan tombol Simpan
Target yang diharapkan	Data penyakit tersimpan ke <i>database</i>
Hasil Pengujian	Benar
<i>Path/Jalur</i>	1-2-3
File	kelolapenyakit.php

Sumber: Hasil analisis (2018).

b. Fitur Mengubah Data Penyakit

1) *Listing Program*

```

41 function perbaruipenyakit() { ①
42     $id_disease = $this->input->post('id_disease');
43     $disease = $this->input->post('disease');
44
45     $data = array(
46         'disease' => $disease
47     );
48
49     $where = array(
50         'id_disease' => $id_disease
51     );
52
53     $this->m_datapenyakit->perbaruidatapenyakit($where, $data);
54     redirect('kelolapenyakit/tampilubahpenyakit');
55 } ③

```

(Sumber: Hasil analisis, 2018)

2) Diagram Alir



(Sumber: Hasil analisis, 2018)

3) Perhitungan *Cyclomatic Complexity*

$$V(G) = E - N + 2 = 2 - 3 + 2 = 1$$

4) Basis Set

Jalur 1 : 1 – 2 – 3

5) *Test Case*

<i>Test Case fuction perbaruipenyakit()</i>	
Jalur 1	
<i>Test case</i>	Pengguna mengganti data yang berada pada kolom isian perbarui data penyakit, kemudian menekan tombol Simpan
Target yang diharapkan	Memperbarui data penyakit di <i>database</i>
Hasil Pengujian	Benar
<i>Path/Jalur</i>	1-2-3
File	kelolapenyakit.php

Sumber: Hasil analisis (2018).

c. Fitur Menghapus Data Penyakit

1) *Listing Program*

```

62 function hapuspenyakit($id_disease) (1)
63     $where = array('id_disease' => $id_disease); (2)
64     $this->m_datapenyakit->hapusdatapenyakit($where);
65     redirect('kelolapenyakit/tampilhapuspenyakit');
66 ) (3)

```

(Sumber: Hasil analisis, 2018)

2) Diagram Alir



(Sumber: Hasil analisis, 2018)

3) Perhitungan *Cyclomatic Complexity*

$$V(G) = E - N + 2 = 2 - 3 + 2 = 1$$

4) Basis Set

Jalur 1 : 1 – 2 – 3

5) Test Case

<i>Test Case fuction hapuspenyakit()</i>	
Jalur 1	
<i>Test case</i>	Pengguna menekan aksi Hapus pada kolom Aksi sesuai dengan data penyakit yang ingin dihapus
Target yang diharapkan	Menghapus data penyakit di <i>database</i>
Hasil Pengujian	Benar
<i>Path/Jalur</i>	1-2-3
File	kelolapenyakit.php

Sumber: Hasil analisis (2018).

4. Hasil *White Box Testing* Pada Fitur Mengelola Data Gejala Penyakit

a. Fitur Memasukkan Data Gejala

1) Listing Program

```

20 function masukgejala() { ①
21     $symptom = $this->input->post('symptom');
22
23     $data = array(
24         'symptom' => $symptom ②
25     );
26     $this->m_datagejala->masukdatagejala($data);
27     redirect('kelolagejala/tampilmasukgejala'); ③
28 }

```

(Sumber: Hasil analisis, 2018)

2) Diagram Alir



(Sumber: Hasil analisis, 2018)

3) Perhitungan *Cyclomatic Complexity*

$$V(G) = E - N + 2 = 2 - 3 + 2 = 1$$

4) Basis Set

Jalur 1 : 1 – 2 – 3

5) Test Case

<i>Test Case fuction masukgejala()</i>	
Jalur 1	
<i>Test case</i>	Pengguna mengisi kolom isian masukkan data gejala, kemudian menekan tombol Simpan
Target yang diharapkan	Data gejala tersimpan ke <i>database</i>
Hasil Pengujian	Benar
<i>Path/Jalur</i>	1-2-3
File	kelolagejala.php

Sumber: Hasil analisis (2018).

b. Fitur Mengubah Data Gejala

1) Listing Program

```

41 function perbaruigejala() { ①
42     $id_symptom = $this->input->post('id_symptom');
43     $symptom = $this->input->post('symptom');
44
45     $data = array(
46         'symptom' => $symptom
47     );
48
49     $where = array( ②
50         'id_symptom' => $id_symptom
51     );
52
53     $this->m_datagejala->perbaruidatagejala($where,$data);
54     redirect('kelolagejala/tampilubahgejala'); ③
55 }

```

(Sumber: Hasil analisis, 2018)

2) Diagram Alir



(Sumber: Hasil analisis, 2018)

3) Perhitungan *Cyclomatic Complexity*

$$V(G) = E - N + 2 = 2 - 3 + 2 = 1$$

4) Basis Set

Jalur 1 : 1 – 2 – 3

5) *Test Case*

<i>Test Case fuction perbaruigejala()</i>	
Jalur 1	
<i>Test case</i>	Pengguna mengganti data yang berada pada kolom isian perbarui data gejala, kemudian menekan tombol Simpan
Target yang diharapkan	Memperbarui data gejala di <i>database</i>
Hasil Pengujian	Benar
<i>Path/Jalur</i>	1-2-3
File	kelolagejala.php

Sumber: Hasil analisis (2018).

c. Fitur Menghapus Data Gejala

1) *Listing Program*

```

62 function hapusgejala($id_symptom) {
63     $where = array('id_symptom' => $id_symptom);
64     $this->m_datagejala->hapusdatagejala($where);
65     redirect('kelolagejala/tampilhapusgejala');
66 }

```

(Sumber: Hasil analisis, 2018)

2) Diagram Alir



(Sumber: Hasil analisis, 2018)

3) Perhitungan *Cyclomatic Complexity*

$$V(G) = E - N + 2 = 2 - 3 + 2 = 1$$

4) Basis Set

Jalur 1 : 1 – 2 – 3

5) Test Case

<i>Test Case fuction hapusgejala()</i>	
Jalur 1	
<i>Test case</i>	Pengguna menekan aksi Hapus pada kolom Aksi sesuai dengan data gejala yang ingin dihapus
Target yang diharapkan	Menghapus data gejala di <i>database</i>
Hasil Pengujian	Benar
<i>Path/Jalur</i>	1-2-3
File	kelolagejala.php

Sumber: Hasil analisis (2018).

5. Hasil *White Box Testing* Pada Fitur Mengelola Aturan Sistem Pakar

a. Fitur Memasukkan Data Aturan

1) *Listing Program*

```

24 function masukaturan() { ①
25     $id_disease = $this->input->post('id_disease');
26     $id_symptom = $this->input->post('id_symptom');
27     $rule_value = $this->input->post('rule_value');
28
29     $data = array(
30         'id_disease' => $id_disease,
31         'id_symptom' => $id_symptom,
32         'rule_value' => $rule_value
33     );
34     $this->dataaturan->masukdataaturan($data);
35     redirect('kelolaaturan/campilmasukaturan');
36 } ③

```

(Sumber: Hasil analisis, 2018)

2) Diagram Alir



(Sumber: Hasil analisis, 2018)

3) Perhitungan *Cyclomatic Complexity*

$$V(G) = E - N + 2 = 2 - 3 + 2 = 1$$

4) Basis Set

Jalur 1 : 1 – 2 – 3

5) Test Case

<i>Test Case fuction masukaturan()</i>	
Jalur 1	
<i>Test case</i>	Pengguna mengisi kolom isian masukkan data aturan, kemudian menekan tombol Simpan
Target yang diharapkan	Data aturan tersimpan ke <i>database</i>
Hasil Pengujian	Benar
<i>Path/Jalur</i>	1-2-3
File	kelolaaturan.php

Sumber: Hasil analisis (2018).

b. Fitur Mengubah Data Aturan

1) Listing Program

```

49 function perbaruiaaturan() { ①
50     $id_rule = $this->input->post('id_rule');
51     $rule_value = $this->input->post('rule_value');
52
53     $data = array(
54         'rule_value' => $rule_value
55     );
56
57     $where = array(
58         'id_rule' => $id_rule
59     );
60
61     $this->dataaturan->perbaruidataaturan($where, $data);
62     redirect('kelolaaturan/campilubahaturan');
63 }

```

(Sumber: Hasil analisis, 2018)

2) Diagram Alir



(Sumber: Hasil analisis, 2018)

3) Perhitungan *Cyclomatic Complexity*

$$V(G) = E - N + 2 = 2 - 3 + 2 = 1$$

4) Basis Set

Jalur 1 : 1 – 2 – 3

5) *Test Case*

Test Case fuction perbaruiaaturan()	
Jalur 1	
<i>Test case</i>	Pengguna mengganti data yang berada pada kolom isian perbarui data aturan, kemudian menekan tombol Simpan
Target yang diharapkan	Memperbarui data aturan di <i>database</i>
Hasil Pengujian	Benar
<i>Path/Jalur</i>	1-2-3
File	kelolaaturan.php

Sumber: Hasil analisis (2018).

c. Fitur Menghapus Data Aturan

1) *Listing Program*

```

70 function hapusaturan($id_rule){ 1
71     $where = array('id_rule' => $id_rule);
72     $this->dataaturan->hapusdataaturan($where); 2
73     redirect('kelolaaturan/tampilhapusaturan');
74 } 3
  
```

(Sumber: Hasil analisis, 2018)

2) Diagram Alir



(Sumber: Hasil analisis, 2018)

3) Perhitungan *Cyclomatic Complexity*

$$V(G) = E - N + 2 = 2 - 3 + 2 = 1$$

4) Basis Set

Jalur 1 : 1 – 2 – 3

5) *Test Case*

Test Case fuction hapusaturan()

Jalur 1	
Test case	Pengguna menekan aksi Hapus pada kolom Aksi sesuai dengan data aturan yang ingin dihapus
Target yang diharapkan	Menghapus data aturan di <i>database</i>
Hasil Pengujian	Benar
Path/Jalur	1-2-3
File	kelolaaturan.php

Sumber: Hasil analisis (2018).

6. Hasil *White Box Testing* Pada Fitur Mendiagnosis Penyakit

a. Fitur mengambil pilihan gejala dari pengunjung

1) Listing Program

```

14 function ambilvisitorgejala() ①
15
16     $where = implode(" AND ", $_POST['tid_gejala']);
17
18     $data['ambilvisitorgejala'] = $this->model_diagnosa->ambilvisitordatagejalaolehid($where)->result(); ②
19
20     $this->load->view('diagnosa/ambilgejala', $data);
21
22
23

```

(Sumber: Hasil analisis, 2018)

2) Diagram Alir



(Sumber: Hasil analisis, 2018)

3) Perhitungan *Cyclomatic Complexity*

$$V(G) = E - N + 2 = 2 - 3 + 2 = 1$$

4) Basis Set

Jalur 1 : 1 – 2 – 3

5) Test Case

Test Case fuction <i>ambilvisitorgejala()</i>	
Jalur 1	
Test case	Pengguna menekan tombol Selanjutnyas
Target yang diharapkan	Mengambil pilihan gejala dari pemilik anjing peliharaan

Hasil Pengujian	Benar
Path/Jalur	1-2-3
File	diagnosis.php

Sumber: Hasil analisis (2018).

b. Fitur mengambil bobot dari pengunjung

1) Listing Program

```

25 function ambilvisitorbobot() { ①
26
27     $where = implode(" AND ", $_POST['id_symptom']);
28     $data['ambilvisitorbobot'] = $this->m_diagnosis->ambildataaturan($where)->result(); ②
29     $this->load->view('v_diagnosishasilpenyakit', $data);
30
31 } ③
    
```

(Sumber: Hasil analisis, 2018)

2) Diagram Alir



(Sumber: Hasil analisis, 2018)

3) Perhitungan *Cyclomatic Complexity*

$$V(G) = E - N + 2 = 2 - 3 + 2 = 1$$

4) Basis Set

Jalur 1 : 1 - 2 - 3

5) *Test Case*

<i>Test Case fuction ambilvisitorbobot()</i>	
Jalur 1	
<i>Test case</i>	Pengguna menekan tombol Proses
Target yang diharapkan	Memproses perhitungan metode <i>certainty factor</i> menggunakan data yang telah diambil
Hasil Pengujian	Benar
Path/Jalur	1-2-3
File	diagnosis.php

Sumber: Hasil analisis (2018).

Lampiran D. Hasil *Black Box Testing* Pada Sistem

No.	Aksi	Hasil	Ket.
Fitur Login			
1.	Mengisi kolom isian dengan <i>username</i> dan <i>password</i> yang dimiliki, kemudian menekan tombol <i>Login</i>	Masuk ke sistem dan menampilkan halaman <i>Dahsboard</i>	Berhasil [✓] Gagal []
2.	Mengisi kolom isian dengan <i>username</i> dan <i>password</i> yang salah, kemudian menekan tombol <i>Login</i>	Menampilkan pesan peringatan " <i>Username atau password salah!</i> "	Berhasil [✓] Gagal []
3.	Tidak mengisi kolom isian, kemudian menekan tombol <i>Login</i>	Menampilkan pesan peringatan " <i>Please fill out this field</i> "	Berhasil [✓] Gagal []
Fitur Logout			
1.	Menekan tombol <i>Logout</i> pada halaman <i>Dashboard</i>	Keluar dari sistem dan menampilkan halaman <i>Login Admin</i>	Berhasil [✓] Gagal []
2.	Menekan tombol <i>Logout</i> pada halaman <i>Kelola Data Penyakit</i>	Keluar dari sistem dan menampilkan halaman <i>Login Admin</i>	Berhasil [✓] Gagal []
3.	Menekan tombol <i>Logout</i> pada halaman <i>Kelola Data Gejala</i>	Keluar dari sistem dan menampilkan halaman <i>Login Admin</i>	Berhasil [✓] Gagal []
4.	Menekan tombol <i>Logout</i> pada halaman <i>Kelola Data Aturan</i>	Keluar dari sistem dan menampilkan halaman <i>Login Admin</i>	Berhasil [✓] Gagal []
Fitur Mengelola Data Penyakit			
1.	Memilih submenu <i>Kelola Data</i> pada menu <i>Penyakit</i>	Menampilkan halaman <i>Kelola Data Penyakit</i> berisi <i>Menu Kelola Data Penyakit</i> dan <i>Tabel Data penyakit</i>	Berhasil [✓] Gagal []
2.	Memilih menu <i>Masukkan Data</i>	Menampilkan <i>form</i> <i>Masukkan Data Penyakit</i>	Berhasil [✓] Gagal []

3.	Mengisi kolom isian <i>form</i> Masukkan Data Penyakit dengan data penyakit, kemudian menekan tombol Simpan	Menyimpan data penyakit dan menampilkan kembali <i>form</i> Masukkan Data Penyakit	Berhasil [✓] Gagal []
4.	Tidak mengisi kolom isian <i>form</i> Masukkan Data Penyakit, kemudian menekan tombol Simpan	Menampilkan pesan peringatan "Please fill out this field"	Berhasil [✓] Gagal []
5.	Menekan tombol Kembali pada <i>form</i> Masukkan Data Penyakit	Menampilkan kembali halaman Kelola Data Penyakit	Berhasil [✓] Gagal []
6.	Memilih menu Ubah Data	Menampilkan <i>form</i> Ubah Data Penyakit	Berhasil [✓] Gagal []
7.	Memilih aksi Ubah sesuai dengan data penyakit yang ingin diubah	Menampilkan form Perbarui Data Penyakit	Berhasil [✓] Gagal []
8.	Mengganti data yang ada di kolom isian <i>form</i> Perbarui Data Penyakit dengan data yang baru, kemudian menekan tombol Simpan	Memperbarui data penyakit dan menampilkan kembali <i>form</i> Perbarui Data Penyakit	Berhasil [✓] Gagal []
9.	Tidak mengisi kolom isian <i>form</i> Perbarui Data Penyakit, kemudian menekan tombol Simpan	Menampilkan pesan peringatan "Please fill out this field"	Berhasil [✓] Gagal []
10.	Menekan tombol Kembali pada <i>form</i> Perbarui Data Penyakit	Menampilkan kembali halaman Kelola Data Penyakit	Berhasil [✓] Gagal []
11.	Memilih menu Hapus Data	Menampilkan <i>form</i> Hapus Data Penyakit	Berhasil [✓] Gagal []
12.	Memilih aksi Hapus sesuai dengan data penyakit yang ingin dihapus	Menghapus data penyakit dan menampilkan kembali <i>form</i> Hapus Data Penyakit	Berhasil [✓] Gagal []
13.	Menekan tombol Kembali pada <i>form</i> Hapus Data Penyakit	Menampilkan kembali halaman Kelola Data Penyakit	Berhasil [✓] Gagal []

Mengelola Data Gejala			
1.	Memilih submenu Kelola Data pada menu Gejala	Menampilkan halaman Kelola Data Gejala berisi Menu Kelola Data Gejala dan Tabel Data gejala	Berhasil [✓] Gagal []
2.	Memilih menu Masukkan Data	Menampilkan <i>form</i> Masukkan Data Gejala	Berhasil [✓] Gagal []
3.	Mengisi kolom isian <i>form</i> Masukkan Data Gejala dengan data gejala, kemudian menekan tombol Simpan	Menyimpan data gejala dan menampilkan kembali <i>form</i> Masukkan Data Gejala	Berhasil [✓] Gagal []
4.	Tidak mengisi kolom isian <i>form</i> Masukkan Data Gejala, kemudian menekan tombol Simpan	Menampilkan pesan peringatan "Please fill out this field"	Berhasil [✓] Gagal []
5.	Menekan tombol Kembali pada <i>form</i> Masukkan Data Gejala	Menampilkan kembali halaman Kelola Data Gejala	Berhasil [✓] Gagal []
6.	Memilih menu Ubah Data	Menampilkan <i>form</i> Ubah Data Gejala	Berhasil [✓] Gagal []
7.	Memilih aksi Ubah sesuai dengan data gejala yang ingin diubah	Menampilkan <i>form</i> Perbarui Data Gejala	Berhasil [✓] Gagal []
8.	Mengganti data yang ada di kolom isian <i>form</i> Perbarui Data Gejala dengan data yang baru, kemudian menekan tombol Simpan	Memperbarui data gejala dan menampilkan kembali <i>form</i> Perbarui Data Gejala	Berhasil [✓] Gagal []
9.	Tidak mengisi kolom isian <i>form</i> Perbarui Data Gejala, kemudian menekan tombol Simpan	Menampilkan pesan peringatan "Please fill out this field"	Berhasil [✓] Gagal []
10.	Menekan tombol Kembali pada <i>form</i> Perbarui Data Gejala	Menampilkan kembali halaman Kelola Data Gejala	Berhasil [✓] Gagal []
11.	Memilih menu Hapus Data	Menampilkan <i>form</i> Hapus Data Gejala	Berhasil [✓] Gagal

			[]
12.	Memilih aksi Hapus sesuai dengan data gejala yang ingin dihapus	Menghapus data gejala dan menampilkan kembali <i>form</i> Hapus Data Gejala	Berhasil [✓] Gagal []
13.	Menekan tombol Kembali pada <i>form</i> Hapus Data Gejala	Menampilkan kembali halaman Kelola Data Gejala	Berhasil [✓] Gagal []
Mengelola Aturan Sistem Pakar			
1.	Memilih submenu kelola data pada menu aturan	Menampilkan halaman kelola data aturan berisi menu kelola data aturan dan tabel data aturan	Berhasil [✓] Gagal []
2.	Memilih menu masukkan data	Menampilkan <i>form</i> masukkan data aturan beserta tabel penyakit dan gejala	Berhasil [✓] Gagal []
3.	Mengisi kolom isian <i>form</i> masukkan data aturan dengan data aturan kemudian menekan tombol simpan	Menyimpan data aturan dan menampilkan kembali <i>form</i> masukkan data aturan	Berhasil [✓] Gagal []
4.	Tidak mengisi kolom isian <i>form</i> masukkan data aturan, kemudian menekan tombol simpan	Menampilkan pesan peringatan "Please fill out this field"	Berhasil [✓] Gagal []
5.	Menekan tombol kembali pada <i>form</i> masukkan data aturan	Menampilkan kembali halaman kelola data aturan	Berhasil [✓] Gagal []
6.	Memilih menu ubah data	Menampilkan <i>form</i> ubah data aturan	Berhasil [✓] Gagal []
7.	Memilih aksi ubah sesuai dengan data aturan yang ingin diubah	Menampilkan form perbarui data aturan	Berhasil [✓] Gagal []
8.	Mengganti data yang ada di kolom isian <i>form</i> perbarui data aturan dengan data yang baru, kemudian menekan tombol simpan	Memperbarui data aturan dan menampilkan kembali <i>form</i> perbarui data aturan	Berhasil [✓] Gagal []
9.	Tidak mengisi kolom isian <i>form</i> perbarui data aturan, kemudian menekan tombol simpan	Menampilkan pesan peringatan "Please fill out this field"	Berhasil [✓]

			Gagal []
10.	Menekan tombol kembali pada <i>form</i> perbarui data aturan	Menampilkan kembali halaman kelola data aturan	Berhasil [✓] Gagal []
11.	Memilih menu hapus data	Menampilkan <i>form</i> hapus data aturan	Berhasil [✓] Gagal []
12.	Memilih aksi hapus sesuai dengan data aturan yang ingin dihapus	Menghapus data aturan dan menampilkan kembali <i>form</i> hapus data aturan	Berhasil [✓] Gagal []
13.	Menekan tombol kembali pada <i>form</i> hapus data aturan	Menampilkan kembali halaman kelola data aturan	Berhasil [✓] Gagal []
Mendiagnosis Penyakit			
1.	Memilih menu diagnosis pada halaman pengunjung	Menampilkan halaman diagnosis berisi pilihan gejala penyakit berupa <i>checkbox</i>	Berhasil [✓] Gagal []
2.	Memilih gejala penyakit, kemudian menekan tombol proses	Menampilkan <i>form</i> isian bobot keyakinan gejala penyakit sesuai dengan gejala yang dipilih	Berhasil [✓] Gagal []
3.	Menekan tombol kembali	Menampilkan kembali halaman diagnosis yang berisi pilihan gejala penyakit berupa <i>checkbox</i>	Berhasil [✓] Gagal []
4.	Mengisi kolom isian bobot keyakinan gejala penyakit, kemudian menekan tombol proses	Menampilkan hasil diagnosis penyakit	Berhasil [✓] Gagal []
5.	Tidak mengisi kolom isian bobot keyakinan gejala penyakit, kemudian menekan tombol proses	Menampilkan pesan peringatan "Please fill out this field"	Berhasil [✓] Gagal []
6.	Menekan tombol selesai setelah sistem menampilkan hasil diagnosis	Menampilkan kembali halaman pengunjung	Berhasil [✓] Gagal []