

**SISTEM PAKAR DIAGNOSA PENYAKIT SAPI
MENGUNAKAN METODE *FORWARD*
CHAINING BERBASIS *WEB***

SKRIPSI



**Oleh:
Ria Saraswati
140210088**

**PROGRAM STUDI TEKNIK INFORMATIKA
UNIVERSITAS PUTERA BATAM
2018**

**SISTEM PAKAR DIAGNOSA PENYAKIT SAPI
MENGUNAKAN METODE *FORWARD*
CHAINING BERBASIS *WEB***

SKRIPSI
**Untuk memenuhi salah satu syarat
guna memperoleh gelar Sarjana**



Oleh
Ria Saraswati
140210088

**PROGRAM STUDI TEKNIK INFORMATIKA
UNIVERSITAS PUTERA BATAM
2018**

PERNYATAAN

Dengan ini saya menyatakan bahwa:

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

Batam, 05 februari 2018
Yang membuat pernyataan,

Materai 6000

RIA SARASWATI
140210088

**SISTEM PAKAR DIAGNOSA PENYAKIT SAPI
MENGUNAKAN METODE *FORWARD*
CHAINING BERBASIS *WEB***

Oleh
Ria Saraswati
140210088

SKRIPSI

**Untuk memenuhi salah satu syarat
guna memperoleh gelar Sarjana**

**Telah disetujui oleh Pembimbing pada tanggal
Seperti tertera di bawah ini**

Batam, 05 februari 2018

Anggia Dasa Putri, S.Kom., M.Kom.
Pembimbing

ABSTRAK

Sapi merupakan hewan ternak yang memiliki potensi ekonomi cukup tinggi. Sapi juga banyak manfaat bagi masyarakat dari segi daging, susu, kulit, dan kotorannya. Tetapi fakta dilapangan (peternakan) menunjukkan bahwa pengetahuan peternak masih sangat kurang. Dalam industri peternakan sangat dibutuhkan pengetahuan akan penyakit yang sering menyerang hewan ternak sapi. Faktor yang menyebabkan kematian pada sapi adalah penyebaran penyakit menular yang dapat merusak kesehatan yang berkelanjutan. Oleh karena itu sistem pakar sangat dibutuhkan dalam menganalisa suatu jenis penyakit yang terdapat pada hewan ternak sapi sehingga dapat dengan mudah mengetahui jenis penyakit yang sedang terjangkit pada sapi tersebut maka tidak perlu berhadapan langsung dengan dokter hewan. Oleh karena itu dibutuhkan sebuah sistem pakar yang berfungsi untuk membantu mendiagnosa penyakit pada hewan ternak sapi yang menggunakan metode *forward chaining*. Sistem pakar ini dapat membantu para peternak sapi dalam mengidentifikasi awal penyakit untuk mengambil keputusan. Hasil dari sistem pakar ini adalah sebuah program yang dapat digunakan untuk mendiagnosa penyakit dengan gejala-gejala yang sudah di input oleh *user*. Dari data tersebut akan dilakukan pengujian pada sistem. Yang nantinya akan menghasilkan kesimpulan awal (diagnosa) yang dapat dijadikan pedoman bagi peternak sapi dalam mengenali penyakit yang ada. Desain sistem dilakukan menggunakan bantuan aplikasi *starUML*. Sistem pakar dibuat menggunakan bahasa pemrograman *PHP* dan *database MySQL* sehingga menghasilkan sebuah sistem pakar untuk mendiagnosa penyakit sapi menggunakan metode *forward chaining* berbasis *Web* dapat digunakan untuk membantu peternak sebagai sumber pengetahuan yang berhubungan dengan penyakit sapi seperti gejala, penyebab serta solusi atau cara untuk mengatasinya.

Kata kunci: sistem pakar, penyakit sapi, *forward chaining*.

ABSTRACT

The cow is the livestock that have economic potential is quite high. The cows are also many benefits to society in terms of meat, milk, skin, and his dung. But the real facts (farm) rancher suggests that knowledge is still very lacking. In the ranching industry badly needed knowledge of the diseases that commonly attack farm animals cows. Factors that cause death in cattle is the spread of an infectious disease that can damage the health. Therefore the expert system is urgently needed in analyzing a type of disease that is present on the farm animals cows so that they can easily find out the type of disease being contracted at the cow, then there needs to be a direct line of sight with the vet. Therefore needed an expert system that serves to help diagnose disease in farm animals cows using the method of forward chaining. Expert system may help the breeder cow in identifying early to take a decision. The results of this expert system is a program that can be used to diagnose the disease with symptoms that are already in the input by the user. From these data the test will be performed on the system. That would later produce preliminary conclusions (diagnosis) which can be used as a guideline for cattle farmers in identifying existing disease. System design is done using the help application starUML. Expert system created using the programming language PHP and the MySQL database to produce an expert system for diagnosing diseases of cattle using the method of forward chaining can be used for Web-based help breeders as a source knowledge associated with the cow disease as the symptoms, causes and solutions or ways to deal with it.

Keywords: *expert system, diseases of cattle, forward chaining.*

KATA PENGANTAR

Puji dan syukur saya panjatkan kepada Allah SWT yang telah melimpahkan segala rahmat dan karuniaNya, sehingga penulis dapat menyelesaikan laporan tugas akhir yang merupakan salah satu persyaratan untuk menyelesaikan program studi strata satu (S1) pada program studi Teknik Informatika di Universitas Putera Batam.

Penulis menyadari bahwa skripsi ini masih jauh dari sempurna. Karena itu, kritik dan saran senantiasa penulis terima dengan senang hati.

Dengan segala keterbatasan, penulis menyadari pula bahwa skripsi ini takkan terwujud tanpa bantuan, bimbingan, dan dorongan dari berbagai pihak. Untuk itu, dengan segala kerendahan hati, penulis menyampaikan ucapan terimakasih kepada:

1. Rektor Universitas Putera Batam.
2. Ketua Program Studi Bapak Andi Maslan, S.T., M.SI.
3. Ibu Anggia Dasa Putri, S.Kom., M.Kom. selaku pembimbing Skripsi pada Program Studi Teknik Informatika di Universitas Putera Batam.
4. Dosen dan Staff Universitas Putera Batam.
5. Bapak Drh. Jusak Wira Hardja, M.Si. selaku narasumber yang rela meluangkan banyak waktu untuk mendukung penelitian ini.
6. Kedua orang tua saya yang telah mendukung dan memberi do'a sampai skripsi ini selesai.
7. Rani Manalu, Yayuk Ana kamalin, Nita Sari Purba yang telah mendukung dan memberi motivasi sampai skripsi ini selesai.

Semoga Allah SWT membalas kebaikan dan selalu mencurahkan hidayah serta taufikNya, Amin.

Batam, 05 februari 2018

Penulis

DAFTAR ISI

PERNYATAAN.....	i
ABSTRAK	iii
ABSTRACT	iv
KATA PENGANTAR	v
DAFTAR ISI.....	vi
DAFTAR GAMBAR	viii
DAFTAR TABEL.....	x
BAB I PENDAHULUAN.....	1
1.1. Latar Belakang	1
1.2. Identifikasi Masalah	4
1.3. Pembatasan Masalah	4
1.4. Perumusan Masalah	5
1.5. Tujuan Penelitian	6
1.6. Manfaat Penelitian	6
1.6.1. Manfaat Teoritis	6
1.6.2. Manfaat Praktis	7
BAB II KAJIAN PUSTAKA.....	8
2.1. Teori Dasar.....	8
2.1.1. Kecerdasan buatan	8
2.1.2. Sistem Pakar.....	11
2.1.1.1. Ciri-Ciri Sistem Pakar	12
2.1.1.2. Keuntungan Sistem Pakar	12
2.1.1.3. Kelemahan Sistem Pakar.....	13
2.1.1.4. Bentuk Sistem Pakar	14
2.1.1.5. Struktur Sistem Pakar.....	14
2.1.3. Representasi Pengetahuan.....	18
2.1.4. <i>Database</i> (Basis Data)	25
2.1.5. <i>Forward Chaining</i>	26
2.1.6. Validasi Sistem	27
2.2. Penyakit Sapi.....	28
2.2.1. Penyakit <i>Milk fever</i> (<i>parturient-hypocalcemia</i>).....	28
2.2.2. Penyakit <i>Grass Tetany</i> (<i>Hypomagnesemia</i>).....	31
2.2.3. Penyakit <i>Asetonemia</i> (<i>Ketosis</i>).....	33
2.3. <i>Software</i> Pendukung	36
2.3.1. UML.....	36
2.3.1.1. <i>Use case Diagram</i>	37
2.3.1.2. <i>Activity Diagram</i>	38
2.3.1.3. <i>Sequence Diagram</i>	39
2.3.1.4. <i>Class Diagram</i>	42
2.3.1.4. Diagram UML.....	44

2.3.2. <i>star</i> UML.....	45
2.2.4. PHP	46
2.2.5. <i>HTML (Hyper Text Markup Language)</i>	47
2.2.6. XAMPP	48
2.2.7. Notepad++.....	49
2.2.8. MySQL.....	49
2.4. Penelitian Terdahulu	53
2.5. Kerangka Berpikir.....	55
BAB III METODE PENELITIAN	57
2.3. Desain Penelitian.....	57
3.2. Pengumpulan data	60
3.2.2. Wawancara	60
3.2.3. Studi Literatur	60
3.3. Operasional Variabel.....	61
3.4. Metode Perancangan Sistem	62
3.4.1. Desain Basis Pengetahuan.....	62
3.4.2. Desain UML (<i>Unified Modeling Language</i>).....	69
3.4.2.1. <i>Use case Diagram</i>	69
3.4.2.2. <i>Activity Diagram</i>	70
3.4.2.3. <i>Sequence Diagram</i>	80
3.4.2.4. <i>Class Diagram</i>	88
3.4.3. Desain Database	90
3.4.4. Desain Antarmuka.....	91
3.5. Lokasi dan Jadwal Penelitian	98
3.5.1. Lokasi Penelitian.....	98
3.5.2. Jadwal Penelitian.....	98
BAB IV HASIL DAN PEMBAHASAN	100
4.1. Hasil Penelitian	100
4.1.1. Implementasi Sistem	100
4.2. Pembahasan.....	110
4.2.1 Pengujian Validasi Sistem.....	111
4.2.2. Pengujian Dengan Pakar	115
BAB V SIMPULAN DAN SARAN	117
5.1 Simpulan	117
5.2 Saran.....	118
DAFTAR PUSTAKA	
DAFTAR RIWAYAT HIDUP	
LAMPIRAN	

DAFTAR GAMBAR

	Halaman
Gambar 2.1 Struktur Sistem Pakar	14
Gambar 2.2 Pohon Keputusan.....	21
Gambar 2.3 Alternatif Pohon Keputusan	23
Gambar 2.4 Diagram Forward Chaining.....	26
Gambar 2.5 penyakit Milk Fever (parturient hypocalcemia).....	29
Gambar 2.6 Penyakit Grass Tetany (Hypomagnesemia)	31
Gambar 2.7 Penyakit Asetonemia (Ketosis)	33
<i>Gambar 2.8 Logo UML</i>	36
Gambar 2.9 Diagram UML	44
Gambar 2.10 Logo PHP	46
Gambar 2.11 Logo HTML	47
Gambar 2.12 Logo XAMPP.....	49
Gambar 2.13 Logo Notepad++	49
Gambar 2.14 <i>Logo MySQL</i>	50
Gambar 2.15 Kerangka Berfikir.....	56
Gambar 3.1 Desain Penelitian.....	57
Gambar 3.2 Pohon Keputusan.....	68
Gambar 3.3 Use case Diagram.....	69
Gambar 3.4 Activity Diagram Login	70
Gambar 3.5 Activity Diagram Mengelola Menu Penyakit	72
Gambar 3.6 Activity Diagram Mengelola Gejala	73
Gambar 3.7 Activity Diagram Mengelola Menu Relasi	75
Gambar 3.8 Activity Diagram Mengelola Menu Lihat Relasi.....	76
Gambar 3.9 Activity Diagram Logout	77
Gambar 3.10 Activity Diagram Melihat Menu Beranda.....	78
Gambar 3.11 Activity Diagram Menu Diagnosis	79
Gambar 3.12 Sequence Diagram Login Admin	80
Gambar 3.13 Sequence Diagram Mengelola Menu Penyakit	81
Gambar 3.14 Sequence Diagram Mengelola Gejala	83
Gambar 3.15 Sequence Diagram Mengelola Menu Relasi	84
Gambar 3.16 Sequence diagram Mengelola Menu Lihat Relasi.....	85
Gambar 3.17 Sequence diagram logout admin	86
Gambar 3.18 Sequence diagram Melihat Menu Beranda	87
Gambar 3.19 Sequence Diagram Menu Diagnosis	87
Gambar 3.20 Class Diagram Admin	88
Gambar 3.21 Class Diagram User.....	89
Gambar 3.22 Desain Physical Data Model	90

Gambar 3.23 Halaman Beranda	91
Gambar 3.24 Halaman Diagnosis	92
Gambar 3.25 Halaman Diagnosis Pertanyaan.....	93
Gambar 3.26 Halaman Diagnosis Solusi	93
Gambar 3.27 Halaman Admin	94
Gambar 3.28 Halaman Home admin.....	94
Gambar 3.29 Halaman Menu Penyakit	95
Gambar 3.30 Halaman Tambah Penyakit	95
Gambar 3.31 Halaman Menu Gejala.....	96
Gambar 3.32 Halaman Tambah Gejala.....	96
Gambar 3.33 Halaman Menu Relasi	97
Gambar 3.34 Rancangan Halaman Menu Konsultasi	97
Gambar 4.1 beranda	101
Gambar 4.2 diagnosa.....	101
Gambar 4.3 Isi identitas user.....	102
Gambar 4.4 User memilih gejala	103
Gambar 4.5 hasil diagnosa penyakit sapi.....	104
Gambar 4.6 Sign in (admin).....	105
Gambar 4.7 Home Admin	106
Gambar 4.8 Daftar Penyakit.....	107
Gambar 4.9 Daftar Gejala	108
Gambar 4.10 Input Relasi	108
Gambar 4.11 Menampilkan Relasi Gejala per Penyakit	109
Gambar 4.12 Menu Logout.....	110

DAFTAR TABEL

	Halaman
Tabel 2.1 keputusan	20
Tabel 2.2 Alternatif Tabel Keputusan	22
Tabel 2.3 Variabel, Penyebab Dan Indikator	35
Tabel 2.4 Simbol- Simbol Pada Use case Diagram	37
Tabel 2.5 Activity diagram	38
Tabel 2.6 Sequence Diagram	40
Tabel 2.7 Class Diagram	43
Tabel 3.1 Variabel Dan Indikator	61
Tabel 3.2 Tabel Jenis Penyakit, penyebab dan Solusi	62
Tabel 3.3 Tabel Gejala Penyakit	64
Tabel 3.4 Tabel Data Aturan	65
Tabel 3.5 keputusan	67
Tabel 4.1 Pengujian Menu Beranda	111
Tabel 4.2 Pengujian Menu Diagnosa	112
Tabel 4.3 Pengujian Hasil Diagnosa	112
Tabel 4.4 Pengujian Menu Admin	112
Tabel 4.5 Pengujian Menu Home	113
Tabel 4.6 Pengujian Menu Penyakit	113
Tabel 4.7 Pengujian Menu Gejala	113
Tabel 4.8 Pengujian Menu Relasi	114
Tabel 4.9 Pengujian Menu Lihat Relasi	114
Tabel 4.10 Pengujian Menu Logout	114
Tabel 4.11 Pengujian Diagnosa Pakar Dan Pengujian Diagnosa Sistem	116

BAB I PENDAHULUAN

1.1 Latar Belakang

Sapi adalah salah satu komoditi peternakan yang menjadi andalan sumber protein hewani. Karena hewan ini berkembangbiak dengan cara melahirkan disebut juga sebagai hewan mamalia. Sapi merupakan hewan ternak yang banyak memiliki potensi ekonomi dan sangat banyak manfaatnya bagi manusia dari segi daging, susu, kulit, tenaga, dan kotorannya. Seperti daging nya bisa di konsumsi dan di olah menjadi berbagai macam makanan, susu nya memiliki sumber protein yang baik bagi manusia, kulit nya bisa dijadikan sebagai bahan produk seperti sepatu, sandal, dan tas, tenaga nya bisa membantu meringankan pekerjaan para petani sawah pada saat pembajakan dan kotorannya bisa di jadikan pupuk kandang untuk kesuburan tanah.

Dengan adanya sapi, kebutuhan masyarakat untuk mengonsumsi daging sapi dapat terpenuhi. Tetapi, berbagai masalah dalam beternak sapi untuk mencukupi kebutuhan masyarakat. Sehingga pemerintah masih perlu mengimpor sapi dari luar negeri. Salah satunya peternak sapi di kota Batam adalah peternakan Pak Ali di Tembesi Bengkel RT 004 RW 001.

Adapun beberapa permasalahan yang dirasakan oleh beberapa peternak sapi di Indonesia diantaranya yaitu masih banyak sapi yang berpenyakit. Sapi yang berpenyakit memiliki dampak negatif terhadap manusia. Penyakit pada

hewan ternak sapi dapat menular dengan cepat dan berakibat fatal yaitu kematian.

Penyakit tersebut dapat timbul disebabkan oleh pakan, bakteri, kuman, virus, jamur, dan parasit. Untuk mencegah agar penyakit sapi tidak menular maka peternak sapi harus mengetahui terlebih dahulu mengenai penyakit – penyakit pada hewan ternak sapi sehingga dapat dilakukan pencegahan dan pengobatan terhadap hewan ternak sapi sedini mungkin. (Nur, Ikhsan, Ariadi, Rosyid, & Ridwan, 2017)

Beragam faktor yang mempengaruhi kesehatan sapi. Namun dari beragam faktor tersebut lingkungan adalah penulara faktor yang paling banyak membuat ternak sapi terserang penyakit. Ada beberapa jenis – jenis penyakit sapi tersebut seperti Gangguan Keseimbangan (Metabolik), penyakit *Antrax*, penyakit Surra, penyakit Kuku busuk, penyakit Mulut, penyakit Demam (BEF), penyakit Para Tuberculosis, penyakit Kudis (*Scabies*), penyakit Rabies, penyakit Ngorok (*Septischaema Epizootica*), penyakit Tetanus, penyakit Keluron. (Nur et al., 2017)

Metabolik atau gangguan metabolisme merupakan salah satu penyakit yang sering ditemukan pada sapi perah selama fase awal laktasi pada saat produksi susu tertinggi. Ada 3 jenis penyakit metabolik yaitu *Milk Fever (Parturient Hypocalcemia)*, *Grass Tetany (Hypomagnesemia)*, *Asetonemia (Ketosis)*. (Subronto, 2007)

Perkembangan teknologi informasi pada saat ini banyak mempengaruhi di berbagai bidang termasuk di bidang peternakan, permasalahan tersebut dapat di atasi dengan sistem berbasis komputer yang menyimpan pengetahuan ahli. Sistem pakar adalah suatu sistem yang dirancang untuk dapat menirukan keahlian

seorang pakar dalam menjawab pertanyaan dan memecahkan masalah. Sistem pakar akan memberikan pemecahan suatu masalah yang didapat dari dialog dengan pengguna. Dengan bantuan sistem pakar seseorang yang bukan pakar/ahli dapat menjawab pertanyaan, menyelesaikan masalah serta mengambil keputusan yang biasanya dilakukan oleh seorang pakar. (Sutojo, S Mulyanto, Edy Suhartono, 2011, p. 13)

Forward Chaining adalah teknik pencarian yang dimulai dengan fakta yang diketahui, kemudian mencocokkan fakta-fakta tersebut dengan bagian *IF* dari *rules IF-THEN*. Bila ada fakta yang cocok dengan bagian *IF*, maka *rule* tersebut dieksekusi, maka sebuah fakta baru (bagian *THEN*) ditambahkan kedalam database. Setiap kali pencocokan, dimulai dari *rule* teratas. Setiap *rule* hanya boleh dieksekusi sekali saja. Proses pencocokan berhenti bila tidak ada lagi *rule* yang bisa dieksekusi. (Sutojo, S Mulyanto, Edy Suhartono, 2011, p. 171)

Website merupakan fasilitas internet yang menghubungkan dokumen dalam lingkup lokal maupun jarak jauh. Dokumen pada *Website* disebut dengan *Web page* dan link dalam *Website* memungkinkan pengguna bisa berpindah dari satu page ke page lain (hyper text), baik diantar page yang disimpan dalam server yang sama maupun server diseluruh dunia. Page diakses dan dibaca melalui browser seperti Netscape Navigator, Internet Explorer, Mozilla Firefox, Google Chrome dan aplikasi Browser lainnya.

Dari penjelasan di atas, penelitian ini akan membuat sistem pakar yang dapat mendeteksi perubahan abnormal dalam proses metabolisme pada hewan ternak sapi. Dengan judul “**SISTEM PAKAR DIAGNOSA PENYAKIT SAPI MENGGUNAKAN METODE *FORWARD CHAINING* BERBASIS *WEB***”.

1.2 Identifikasi Masalah

Berdasarkan latar belakang yang telah disebutkan dapat diidentifikasi permasalahan nya sebagai berikut :

1. Kurangnya informasi pada peternak tentang jenis – jenis penyakit sapi sehingga kualitas dan kuantitas sapi masih terkendala untuk memenuhi kebutuhan masyarakat yang mengonsumsi daging sapi.
2. Minimnya informasi penanganan bagi para peternak terhadap sapi ternak untuk mencegah sapi terserang penyakit. Dan penanganan pertama terhadap penyakit yang menyerang sapi.
3. Kesulitan bagi peternak untuk mendiagnosa penyakit sapi yang menyebabkan keterlambatan atas penanggulangan penyakit dan sehingga terjadi kematian.

1.3 Pembatasan Masalah

Berdasarkan pembahasan dari identifikasi masalah tersebut maka balasan dari penelitian adalah :

1. Penelitian ini mengangkat masalah penyakit pada sapi ternak. Jenis penyakit ternak sapi yang akan di bahas pada penelitian ini adalah *Milk fever*

(*parturient-hypocalcemia*), dan *hypomagnesemia (grass tetany)*, Penyakit *Asetonemia (Ketosis)*.

2. Penyakit yang akan di bahas adalah penyakit yang disebabkan karena gangguan keseimbangan metabolik.
3. Penelitian ini membantu diagnosa penyakit sapi menggunakan metode *forward chaining*.
4. Studi kasus pada penelitian ternak sapi ini di peternakan Pak Ali yang bertempat di Tembesi Batam. Wawancara Penelitian ini dilakukan dengan Pak Ali sebagai peternak dan Drh. Jusak Wira Hardja, M.Si. Yang juga sebagai pakar pada penelitian ini, yang berlokasi di Ruko Glass Centre no.14-15, Lubuk Baja, Kota Batam, Kepri.
5. Tools yang digunakan database dan server lokal XAMPP dan bahasa program HTML, PHP, MySQL, *StarUML* dan notepad++.
6. Penelitian ini bertujuan untuk mendiagnosa awal penyakit *Milk fever (parturient-hypocalcemia)*, *hypomagnesemia (grass tetany)*, dan *asetonemia (ketosis)*.

1.4 Perumusan Masalah

Berdasarkan latar belakang di atas, dapat disimpulkan beberapa permasalahan yaitu sebagai berikut :

1. Bagaimana bentuk gejala, penyebab dan penyakit yang menyerang ternak sapi ?
2. Bagaimana sistem pakar metode *forward chaining* memberikan solusi mendiagnosa penyakit yang menyerang sapi ?

3. Bagaimana membangun sistem pakar berbasis *Web* untuk mendiagnosa penyakit ternak sapi ?

1.5 Tujuan Penelitian

Dari rumusan masalah di atas tujuan yang ingin di raih untuk pembuatan aplikasi ini adalah sebagai berikut :

1. Untuk mengetahui gejala – gejala yang berhubungan dengan penyakit ternak sapi.
2. Untuk mengetahui cara penanganan ternak sapi yang disebabkan karena gangguan keseimbangan (Metabolik).
3. Untuk merancang sistem pakar mendiagnosa penyakit pada ternak sapi menggunakan metode *forward chaining* berbasis *Web*.

1.6 Manfaat Penelitian

1.6.1 Manfaat Teoritis

Terdapat beberapa manfaat teoritis dari penelitian ini :

1. Dengan adanya penelitian ini bisa dijadikan referensi oleh penelitian selanjutnya yang berkaitan dengan penyakit ternak sapi dan pengembangan sistem pakar.
2. Menambah wawasan bagi peneliti tentang ternak sapi dan bagaimana cara beternak yang baik untuk menghasilkan kualitas seperti yang diinginkan.
3. Penelitian ini bertujuan untuk mendapatkan gelar sarjana.

1.6.2 Manfaat Praktis

Terdapat beberapa manfaat praktis dari penelitian ini :

1. Diharapkan dengan penelitian ini, dokter mempunyai media interaktif untuk memberi informasi tentang penyakit sapi.
2. Memberi pengetahuan kepada masyarakat agar bisa meminimalisirkan penyakit pada ternak sapi dan mengetahui gejala – gejala yang timbul.
3. Membantu peternak meningkatkan produktifitas sapi dan bisa memenuhi kebutuhan masyarakat baik dari segi daging maupun susu.

BAB II KAJIAN PUSTAKA

2.1 Teori Dasar

Sistem pakar adalah suatu sistem yang dirancang untuk dapat menirukan keahlian seorang pakar dalam menjawab pertanyaan dan memecahkan suatu masalah. sistem pakar akan memberikan pemecahan suatu masalah yang didapat dari dialog dengan pengguna. dengan bantuan sistem pakar seseorang yang bukan pakar/ahli dapat menjawab pertanyaan, menyelesaikan masalah serta mengambil keputusan yang biasanya dilakukan oleh seorang pakar. (Sutojo, S Mulyanto, Edy Suhartono, 2011)

2.1.1 Kecerdasan buatan

(Sutojo, S Mulyanto, Edy Suhartono, 2011) Dimulai dari sejarah berbagai literatur mengenai kecerdasan buatan atau *artificial intelligence* menyebutkan bahwa ide ini diawali pada awal abad 17 ketika *Rene descartes* mengemukakan bahwa tubuh hewan bukanlah apa – apa melainkan hanya mesin – mesin yang rumit. Kemudian *Blaise Pascal* menciptakan mesin penghitung digital mekanis pertama pada 1642. Selanjutnya pada abad ke 19, *Charles Babbage* dan ada *Lovelace* bekerja pada mesin penghitung mekanis yang dapat diprogram. Perkembangan terus berlanjut, *Bertrand Russell* dan *Alfred North Whitehead* menerbitkan *Principia Mathematica* yang merombak logika formal. *Warren McCulloch* dan *Walter Pitts* menerbitkan “*Kalkulus Logis*” yang tetap

ada dalam aktivitas” pada 1943 yang meletakkan pondasi awal untuk jaringan syaraf.

Tahun 1950 – an adalah periode usaha aktif dalam AI. Program AI yang pertama ditulis pada 1951 untuk menjalankan mesin *Farranti Mark I di University Of Manchester (UK)*. Sebuah program permainan naskah yang ditulis oleh Christopher Strachey dan program permainan catur yang ditulis oleh *Diet Rich Prinz*. *John McCarthy* membuat istilah “ Kecerdasan Buatan” pada konferensi pertama pada tahun 1956 , selain itu dia juga menemukan bahasa pemrograman Lisp. *Alan Turing* memperkenalkan “Turing Test” sebagai sebuah cara untuk mengoprasionalkan test perilaku cerdas. *Joseph Weizenbaum* membangun ELIZA, sebuah Chatterbot yang menerapkan psikoterapi Rogerian.

Selama tahun 1960-an sampai 1970-an , *Joel Moses* mendemonstrasikan kekuatan pertimbangan simbolis untuk mengintegrasikan masalah didalam program Macsyma, program berbasis pengetahuan yang sukses pertama kali dibidang matematika. *Marvin Minsky* dan *Seymour Papert* menerbitkan Perceptrons, yang mendemonstrasikan batas jaringan syaraf sederhana dan *Alain Colmerauer* mengembangkan bahasa komputer prolog. *Ted Shortliffe* mendemonstrasikan kekuatan sistem berbasis atauran untuk representasi pengetahuan dan inferensi dalam diagnosa dan terapi medis yang di yakni sebagai sistem pakar pertama. *Hans Moravec* mengembangkan kendaraan terkendali komputer pertama untuk mengatasi jalan yang mempunyai rintangan secara mandiri. Adapun bagian dari Artificial Intelligence ini adalah sebagai berikut :

1. Sistem Pakar (*Expert System*)

Sistem pakar adalah suatu sistem yang dirancang untuk dapat menirukan keahlian seorang pakar dalam menjawab pertanyaan dan memecahkan masalah. Sistem pakar akan memberikan pemecahan suatu masalah yang didapat dari dialog dengan pengguna. Dengan bantuan sistem pakar seseorang yang bukan pakar/ahli dapat menjawab pertanyaan, menyelesaikan masalah serta mengambil keputusan yang biasanya dilakukan oleh seorang pakar. (Sutojo, S Mulyanto, Edy Suhartono, 2011, p. 13).

2. Fuzzy Logic

Fuzzy adalah metodologi sistem kontrol pemecahan masalah, yang cocok untuk di implementasikan pada sistem, mulai dari sistem yang sederhana, sistem kecil, embended system, jaringan PC, multi-channel atau workstation berbasis akuisisi data, dan sistem kontrol. Metodologi ini dapat diterapkan pada perangkat keras, perangkat lunak, atau kombinasi keduanya. Dalam logika klasik dinyatakan bahwa segala sesuatu bersifat biner, yang artinya adalah hanya mempunyai dua kemungkinan, “Ya atau Tidak”, “Benar atau Salah”, “Baik atau Buruk”. Oleh karena itu, semua ini dapat mempunyai nilai keanggotaan 0 atau 1. Akan tetapi dalam logika fuzzy memungkinkan nilai keanggotaan berada di antara 0 dan 1. Artinya bisa saja suatu keadaan mempunyai dua nilai “Ya dan Tidak”, “Benar dan Salah”, “Baik dan Buruk” secara bersamaan, namun besar nilainya tergantung pada bobot keanggotaan yang dimilikinya. (Sutojo, S Mulyanto, Edy Suhartono, 2011)

3. Jaringan saraf tiruan

Jaringan saraf tiruan adalah paradigma pengolahan informasi yang terinspirasi oleh sistem saraf secara biologis, seperti proses informasi pada otak manusia. Elemen kunci dari paradigma ini adalah struktur dari sistem pengolahan informasi yang terdiri dari sejumlah besar elemen pemrosesan yang saling berhubungan (*neuron*), bekerja serentak untuk menyelesaikan masalah tertentu. (Sutojo, S Mulyanto, Edy Suhartono, 2011)

3.1.2. Sistem Pakar

Sistem pakar berasal dari istilah *knowledge-based expert system*. Yaitu suatu sistem yang dirancang untuk dapat menirukan keahlian seorang pakar dalam menjawab pertanyaan dan memecahkan masalah. Sistem pakar akan memberikan pemecahan suatu masalah yang didapat dari dialog dengan pengguna. Dengan bantuan sistem pakar seseorang yang bukan pakar/ahli dapat menjawab pertanyaan, menyelesaikan masalah serta mengambil keputusan yang biasanya dilakukan oleh seorang pakar. (Sutojo, S Mulyanto, Edy Suhartono, 2011a, p. 13). Sistem pakar didefinisikan sebagai sistem komputer yang mempunyai kemampuan meniru pengetahuan dan keterampilan pada seorang pakar tertentu. Secara umum sistem pakar dibangun atas tiga komponen utama, yaitu :

1. *Knowledge base* (basis pengetahuan) merupakan bagian yang ada didalam sebuah sistem pakar yang berfungsi untuk menyimpan pengetahuan. *Knowledge base* yang di simpan oleh sistem pakar

berbeda antara satu dengan yang lain tergantung dari bidang kepakaran yang dibuat.

2. *Working memory* berfungsi untuk menyimpan fakta – fakta yang di dapat dalam proses konsultasi yang dilakukan dengan seorang pakar. *Working memory* akan menyimpan segala informasi yang user masukan berdasarkan hasil konsultasi user dengan seorang pakar.
3. *Inference engine* mempunyai tugas dalam hal mencari fakta – fakta yang terdapat di dalam *working memory* dengan fakta – fakta tertentu yang ada didalam *knowledge base*.

2.1.1.1. Ciri-Ciri Sistem Pakar

Adapun ciri – ciri yang dimiliki sistem pakar, yaitu:

1. Memiliki informasi yang handal.
2. Mudah di modifikasi.
3. Dapat digunakan dalam berbagai jenis komputer.
4. Memiliki kemampuan untuk belajar beradaptasi.

2.1.1.2. Keuntungan Sistem Pakar

Secara garis besar, banyak manfaat yang diambil dengan adanya sistem pakar, antara lain:

1. Memungkinkan orang awam bisa mengerjakan pekerjaan para ahli.
2. Bisa melakukan proses secara berulang secara otomatis .
3. Menyimpan pengetahuan dan keahlian para pakar.
4. Meningkatkan output dan produktivitas.
5. Meningkatkan kualitas.

6. Mampu mengambil dan melestarikan keahlian para pakar.
7. Mampu beroperasi dalam lingkungan yang berbahaya.
8. Memiliki kemampuan untuk mengakses pengetahuan.
9. Memiliki rehabilitas.
10. Meningkatkan kapabilitas sistem pakar komputer.
11. Memiliki kemampuan untuk bekerja dengan informasi yang tidak lengkap dan mengandung ketidak pastian.
12. Sebagai media pelengkap dalam pelatihan.
13. Meningkatkan kapabilitas dalam penyelesaian masalah.
14. Menghemat waktu dalam pengambilan keputusan.

2.1.1.3. Kelemahan Sistem Pakar

Disamping mempunyai kelebihan sistem pakar juga mempunyai kelemahan, kelemahan yang dimiliki oleh sistem pakar, sebagai berikut:

1. Biaya yang diperlukan untuk membuat dan memeliharaanya sangat mahal.
2. Sulit dikembangkan, hal ini tentu saja erat kaitannya dengan ketersediaan pakar di bidangnya.
3. Sistem pakar tidak 100% bernilai benar.

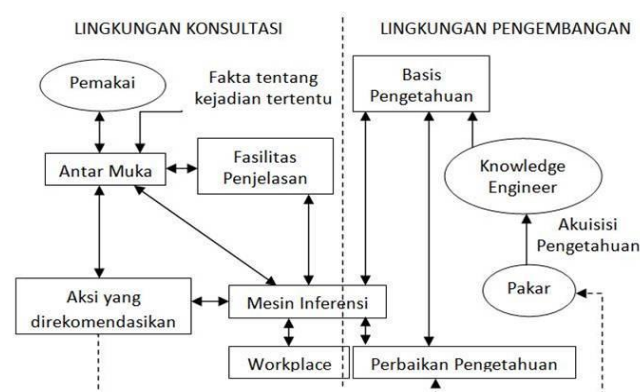
2.1.1.4. Bentuk Sistem Pakar

Ada 4 bentuk sistem pakar yaitu:

1. Berdiri sendiri: sistem pakar jenis ini merupakan software yang berdiri sendiri tidak tergabung dengan software yang lainnya.
2. Tergabung: sistem pakar jenis ini merupakan bagian program yang terkandung di dalam suatu algoritma (konvensional), atau merupakan program dimana didalamnya memanggil algoritma subrutin lain (konvensional).
3. Menghubungkan ke software lain: bentuk ini merupakan sistem pakar yang menghubungkan kesuatu paket program tertentu, misalnya dengan DBMS.
4. Sistem mengabdikan: sistem pakar merupakan bagian dari komputer khusus yang dihubungkan dengan suatu fungsi tertentu, misalnya digunakan untuk membantu menganalisis data radar.

2.1.1.5. Struktur Sistem Pakar

Sistem pakar terdiri dari 2 bagian pokok, yaitu lingkungan pengembangan dan lingkungan konsultasi



Gambar 2.1 Struktur Sistem Pakar

Sumber: (Sutojo, S Mulyanto, Edy Suhartono, 2011)

Berdasarkan gambar diatas, berikut ini adalah penjelasan komponen komponen yang ada pada sistem pakar.(Sutojo, S Mulyanto, Edy Suhartono, 2011 p. 167)

a. Akuisisi Pengetahuan

Subsistem ini digunakan untuk memasukkan pengetahuan dari seorang pakar dengan cara merekayasa pengetahuan agar bisa diproses oleh komputer dan menaruhnya ke dalam basis pengetahuan dengan format tertentu (dalam bentuk representasi pengetahuan).

b. Basis pengetahuan (*Knowledge Base*)

Basis pengetahuan mengandung pengetahuan yang diperlukan untuk memahami, memformulasikan, dan menyelesaikan masalah.

c. Mesin Inferensi (*Inference Engine*)

Mesin inferensi adalah sebuah program yang berfungsi untuk memandu proses penalaran terhadap suatu kondisi berdasarkan pada basis pengetahuan yang ada, memanipulasi dan mengarahkan kaidah, model, an fakta yan disimpan dalam basis pengetahuan untuk mencapai solusi atau kesimpulan.

d. Daerah Kerja (*Blackboard*)

Untuk merekam hasil sementara yang akan dijadikan sebagai keputusan dan untuk menjelaskan sebuah masalah yang akan terjadi, sistem pakar membutuhkan *Blackboard*, yaitu area pada memori yang berfungsi sebagai basis data.

e. Antarmuka Pengguna (*User Interface*)

Digunakan sebagai media komunikasi antara pengguna dan sistem pakar.

Komunikasi ini paling bagus bila disajikan dalam bahasa alami (*natural language*) dan dilengkapi dengan grafik, menu, dan formulir elektronik.

f. Subsistem Penjelasan (*Explanation Subsystem / Justifer*)

Berfungsi memberi penjelasan kepada pengguna, bagaimana suatu kesimpulan dapat diambil. Kemampuan seperti ini sangat penting bagi pengguna untuk mengetahui proses pemindahan keahlian pakar maupun dalam pemecahan masalah.

g. Sistem Perbaikan Pengetahuan (*Knowledge Refining System*)

Kemampuan memperbaiki pengetahuan (*Knowledge Refining System*) dari seorang pakar diperlakukan untuk menganalisis pengetahuan, belajar dari kesalahan masa lalu, kemudia memperbaiki pengetahuannya sehingga dapat dipakai pada masa mendatang.

h. Pengguna (*User*)

Pada umumnya pengguna sistem pakar bukanlah seorang pakar (*non-expert*) yang membutuhkan solusi, saran, atau pelatihan (*training*) dari berbagai permasalahan yang ada.

Untuk membuat sistem pakar yang efektif harus dipilih representasi pengetahuan yang tepat. Pemilihan representasi pengetahuan yang tepat akan membuat sistem pakar dapat mengakses basis pengetahuan tersebut untuk keperluan pembuatan keputusan. (Sutojo, S Mulyanto, Edy Suhartono, 2011)

1. Jaringan Semantik (*Semantic Nets*)

Jaringan semantik merupakan representasi pengetahuan yang digunakan untuk menggambarkan data dan informasi, yang menunjukkan hubungan antara berbagai objek. Objek bisa berupa benda fisik seperti mobil, rumah, orang, atau konsep berupa pikiran, kejadian, atau tindakan. Jaringan semantik merupakan grafik, yang terdiri dari simpul-simpul (*nodes*), yang mempresentasikan objek, dan busur-busur (*arcs*) yang menunjukkan relasi antara objek-objek tersebut. Jaringan semantik merupakan alat efektif untuk mempresentasikan pemetaan data agar tidak terjadi duplikasi data.

2. Bingkai (*Frame*)

Frame pertama kali ditemukan oleh Marvin Minsky pada 1970. Menurut Minsky, *frame* dapat dipandang sebagai struktur data yang digunakan untuk merepresentasikan pengetahuan yang didasarkan pada karakteristik yang sudah dikenal, yang merupakan pengalaman-pengalaman. *Frame* merupakan kumpulan pengetahuan tentang suatu objek tertentu, peristiwa, lokasi, situasi, dan lain-lain. *Frame* terdiri dari dua elemen dasar, yaitu slot dan subslot. Slot merupakan kumpulan atribut atau properti yang menjelaskan objek yang direpresentasikan oleh *frame*. Sedangkan subslot menjelaskan pengetahuan atau prosedur dari atribut pada slot. Slot dalam *frame* berisi beberapa informasi sebagai berikut:

- a. Informasi identifikasi *frame*.
- b. Hubungan *frame* dengan *frame* yang lain.
- c. Menggambarkan persyaratan yang dibutuhkan *frame*.
- d. Informasi prosedural untuk menggunakan struktur yang digambarkan.

- e. Informasi *default frame*.
- f. Informasi baru.

Sedangkan, subplot terdiri dari beberapa bentuk antara lain:

- a. *Value*: nilai dari suatu atribut.
- b. *Default*: nilai yang digunakan jika slot kosong atau tidak dideskripsikan pada instansiasi *frame*.
- c. *Range*: jenis informasi yang muncul pada slot.
- d. *If added*: berisi informasi tindakan yang akan dikerjakan jika nilai slot diisi.
- e. *If needed*: subplot ini digunakan pada kasus dimana tidak ada *value* pada slot.
- f. *Other*: slot dapat berisi *frame*, *rule*, jaringan semantik ataupun tipe lain dari informasi.

3.1.3. Representasi Pengetahuan

Pemrosesan yang dilakukan oleh sistem pakar merupakan pemrosesan pengetahuan, bukan pemrosesan data seperti yang dikerjakan dengan pemrograman secara konvensional yang kebanyakan dilakukan oleh sistem informasi. Pengetahuan (*knowledge*) adalah pemahaman secara praktis maupun teoritis terhadap suatu obyek atau domain tertentu (Hartati & Iswanti, 2008 p 18). Berikut ini adalah contoh struktur kaidah *IF-THEN* yang menghubungkan objek. (Hartati & Iswanti, 2008:25)

1. *IF* premis *THEN* konklusi
2. *IF* masukan *THEN* keluaran
3. *IF* kondisi *THEN* tindakan
4. *IF* antesenden *THEN* konsekuen
5. *IF* data *THEN* hasil
6. *IF* tindakan *THEN* tujuan
7. *IF* aksi *THEN* reaksi
8. *IF* gejala *THEN* diagnosa

Premis mengacu pada fakta yang harus benar sebelum konklusi tertentu dapat diperoleh. Masukan mengacu pada data yang harus tersedia sebelum keluaran dapat diperoleh. Kondisi mengacu pada keadaan yang harus berlaku sebelum tindakan dapat diambil. Antesenden mengacu situasi yang terjadi sebelum kosekuensi dapat diamati. Data mengacu pada informasi yang harus tersedia sehingga sebuah hasil dapat diperoleh. Tindakan menace pada kegiatan yang harus dilakukan sebelum hasil dapat diharapkan. Aksi mengacu pada kegiatan yang menyebabkan munculnya efek dari tindakan tersebut. Sebab mengacu pada keadaan tertentu yang menimbulkan akibat tertentu. Gejala mengacu pada keadaan yang menyebabkan adanya kerusakan atau keadaan tertentu yang mendorong adanya pemeriksaan.

Sebelum sampai pada bentuk kaidah produksi, terdapat langkah-langkah yang harus ditempuh dari pengetahuan yang didapatkan dalam domain tertentu. Langkah-langkah tersebut adalah menyajikan pengetahuan yang berhasil

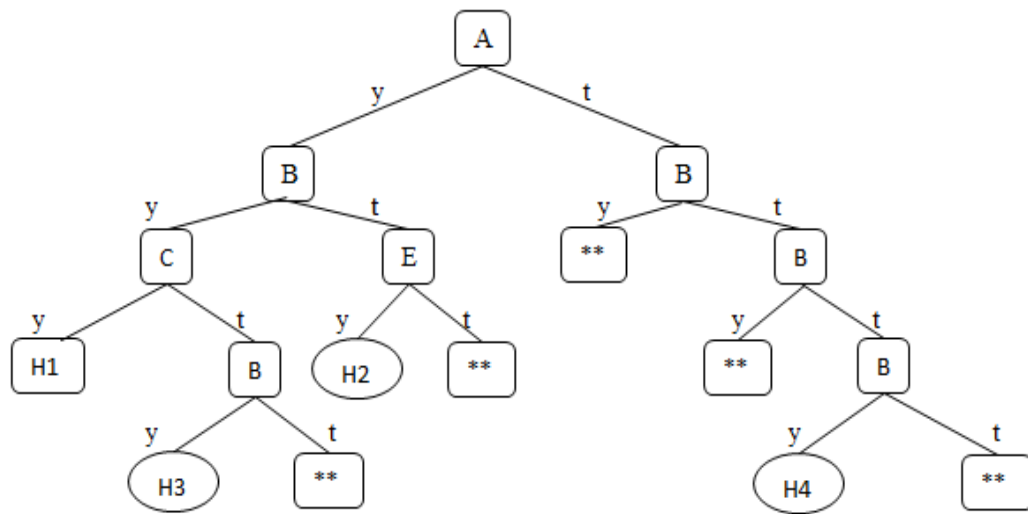
didapatkan dalam bentuk tabel keputusan (*decision table*) kemudian dari tabel keputusan dibuat pohon keputusan (*decision tree*).

Tabel keputusan merupakan suatu cara untuk mendokumentasikan pengetahuan. Tabel keputusan merupakan matrik kondisi yang dipertimbangkan dalam pendeskripsian kaidah. Meskipun kaidah secara langsung dapat dihasilkan dari table keputusan tetapi untuk menghasilkan kaidah yang efesien terdapat suatu langkah yang harus ditempuh yaitu membuat pohon keputusan terlebih dahulu. Dari pohon keputusan dapat diketahui atribut (kondisi) yang dapat direduksi sehingga menghasilkan kaidah yang efisien dan optimal. Berikut ini variasi contoh terkait dengan penyajian dalam bentuk tabel keputusan dan pohon keputusan.

Tabel 2.1 keputusan

Hipotesa <i>Evidence</i>	Hipotesa 1	Hipotesa 2	Hipotesa 3	Hipotesa 4
<i>Evidence A</i>	Ya	Ya	Ya	Tidak
<i>Evidence B</i>	Ya	Tidak	Ya	Ya
<i>Evidence C</i>	Ya	Tidak	Tidak	Ya
<i>Evidence D</i>	Tidak	Tidak	Tidak	Ya
<i>Evidence E</i>	Tidak	Ya	Ya	Tidak

Sumber: (Hartati & Iswanti, 2008)



Gambar 2.2 Pohon Keputusan
Sumber: (Hartati & Iswanti, 2008)

Keterangan:

A = *evidence* A, H1 = hipotesa 1, y = ya

B = *evidence* B, H2 = hipotesa 2, t = tidak

C = *evidence* C, H3 = hipotesa 3, ** = tidak menghasilkan hipotesa tertentu

D = *evidence* D, H4 = hipotesa 4

Dari gambar 2.2 dapat diketahui bahwa hipotesa H1 terpenuhi jika memenuhi *evidence* A, B, dan C. Hipotesa H2 terpenuhi jika memiliki *evidence* A dan *evidence* E. Hipotesa H3 akan terpenuhi jika memiliki *evidence* A, B, dan E. Hipotesa H4 akan dihasilkan jika memenuhi *evidence* B, C, dan D. Notasi “y” mengandung arti memenuhi *node (evidence)* di atasnya, notasi “t” artinya tidak memenuhi.

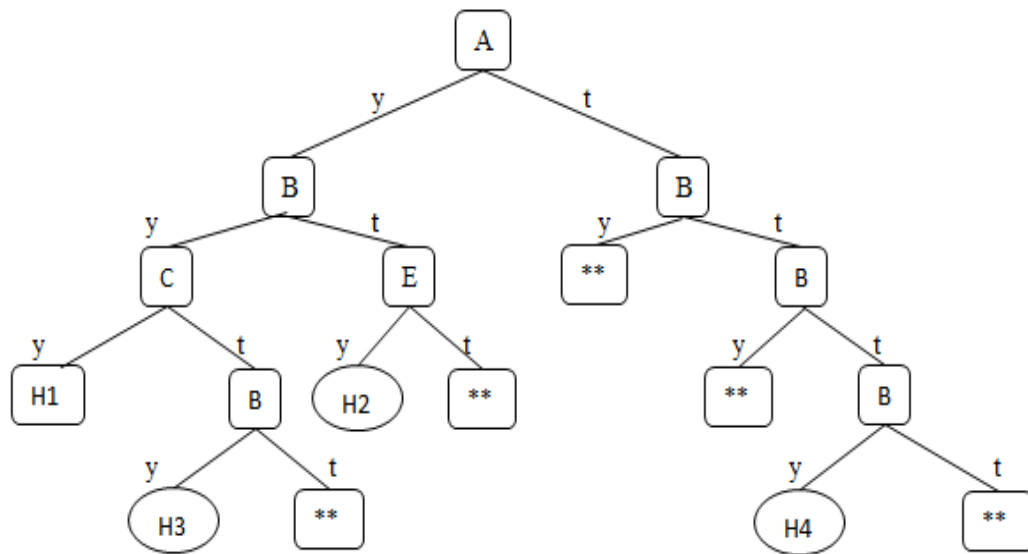
Dalam implementasi sistem pakar terutama dalam sesi konsultasi pada, *node-node* yang mewakili *evidence* biasanya akan menjadi pertanyaan yang diajukan oleh sistem. Dengan melihat pohon keputusan pada gambar 2.2 permasalahan dapat saja terjadi pada awal sesi konsultasi yaitu pada saat sistem pakar menanyakan “apakah memiliki *evidence* A?”. Permasalahannya adalah apapun jawaban pengguna baik “ya” atau “tidak” maka sistem akan menanyakan *evidence* B. Ini berarti jawaban pengguna tidak akan mempengaruhi sistem. Salah satu cara untuk mengatasi hal ini adalah dengan mengubah urutan pada tabel keputusan seperti terlihat pada tabel 2.2.

Tabel 2.2 Alternatif Tabel Keputusan

Hipotesa <i>Evidence</i>	Hipotesa 1	Hipotesa 2	Hipotesa 3	Hipotesa 4
<i>Evidence A</i>	Ya	Ya	ya	Tidak
<i>Evidence D</i>	Tidak	Tidak	tidak	ya
<i>Evidence B</i>	Ya	Tidak	ya	Ya
<i>Evidence C</i>	Ya	Tidak	tidak	Ya
<i>Evidence E</i>	Tidak	Ya	ya	Tidak

Sumber: (Hartati & Iswanti, 2008)

Berdasarkan tabel 2.2 dapat dihasilkan pohon keputusan sebagai berikut:



Gambar 2.3 Alternatif Pohon Keputusan
Sumber: (Hartati & Iswanti, 2008a)

Keterangan:

A = *evidence* A, H1 = hipotesa 1, y = ya

B = *evidence* B, H2 = hipotesa 2, t = tidak

C = *evidence* C, H3 = hipotesa 3, ** = tidak menghasilkan hipotesa tertentu

D = *evidence* D, H4 = hipotesa 4

Dilihat dari gambar 2.3, masing-masing *node* yang mewakili *evidence* tertentu untuk kondisi “y” dan “t” sudah tidak mengarah pada *evidence* yang sama. Hal ini berarti jawaban pengguna yang berbeda akan mengarah pada pertanyaan yang berbeda pula.

Proses akuisisi pengetahuan dalam sistem pakar bukanlah hal yang mudah dan sederhana. Pengetahuan yang berhasil didapatkan dan disajikan dalam salah satu bentuk representasi pengetahuan misalnya disajikan dalam bentuk tabel keputusan dan dapat diubah susunannya, secara teori sudah benar tetapi pada domain pengetahuan tertentu, perubahan seperti itu terkadang tidak diperkenankan. Dengan kata lain, pengetahuan yang akan disajikan dalam bentuk representasi tertentu harus mengikuti kaidah-kaidah kepakaran dengan domainnya dan tidak boleh bertentangan.

4. Logika Predikat

Logika pertama kali dikembangkan oleh Aridtooteles merupakan logika formal. Logika formal adalah logika yang berhubungan dengan bentuk (*syntax*) pernyataan bukan pada arti (*semantic*) dari pernyataan. Logika formal berkonsentrasi pada penalarannya itu sendiri tanpa terganggu oleh obyek yang sedang dinalar. Pada alogika simbolis dikenal aksioma yaitu simbol-simbol yang merepresentasikan obyek dan kelas serta operasi aljabar untuk memanipulasi simbol-simbol tersebut. Logika personal, bisa juga disebut kulkulus proposional, adalah logika simbolis yang memanipulasi proposisi.

Logika propositional hanya mampu menangani pernyataan yang komplit dan tidak biasa menganalisa struktur internal sebuah pernyataan. Sehingga untuk menganalisa kasus yang lebih umum dikembangkan logika predikat yang dapat menganalisis struktur internal kalimat (*sentence*). Bentuk paling sederhana dari logika predikat adalah logika derajat pertama (*first order logic*). Logika derajat pertama tersebut terbentuk dengan menambahkan fungsi atau analisis lain pada

kulkulus predikat. Fungsi merupakan logika yang menghasilkan nilai (*value*). Logika predikat selain digunakan untuk menentukan kebenaran (*truthfulness*) tau kesalahan (*falsity*) sebuah pernyataan, juga dapat digunakan untuk merepresentasikan pernyataan tentang obyek tertentu.

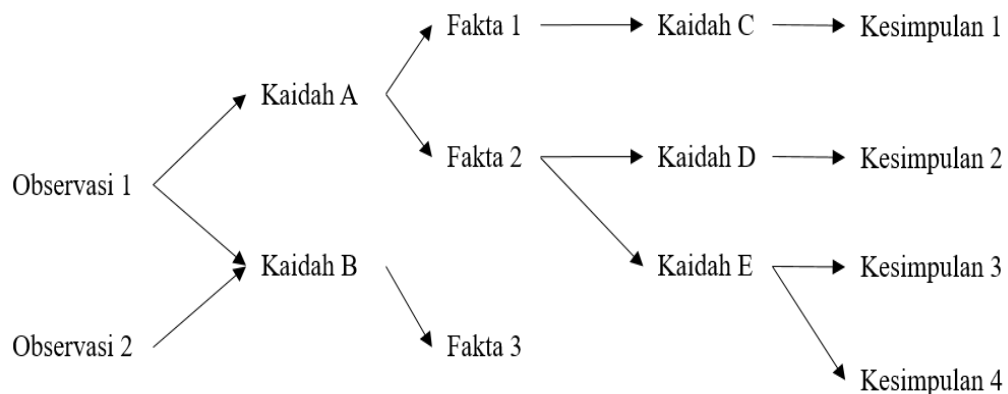
2.1.4. Database (Basis Data)

Menurut (A.S. & Shalahuddin, 2011, p. 44) Sistem basis data adalah sistem terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah diolah atau informasi dan membuat informasi tersedia saat dibutuhkan. Basis data adalah media untuk menyimpan data agar dapat diakses dengan mudah dan cepat. Kebutuhan basis data meliputi memasukkan, menyimpan, dan mengambil data serta membuat laporan berdasarkan data yang telah disimpan. Salah satu bentuk basis data yang dibutuhkan dalam sebuah sistem yaitu *Database Management System (DBMS)*. *DBMS* adalah suatu sistem aplikasi yang digunakan untuk menyimpan, mengelola, dan menampilkan data. Suatu sistem aplikasi disebut *DBMS* jika memenuhi syarat minimal sebagai berikut:

1. Menyediakan fasilitas untuk mengelola akses data.
2. Mampu menangani integritas data.
3. Mampu menangani akses data yang dilakukan.
4. Mampu menangani backup data.

2.1.5. *Forward Chaining*

Forward Chaining adalah teknik pencarian yang dimulai dengan fakta yang diketahui, kemudian mencocokkan fakta-fakta tersebut dengan bagian *IF* dari *rules IF-THEN*. Bila ada fakta yang cocok dengan bagian *IF*, maka *rule* tersebut dieksekusi, maka sebuah fakta baru (bagian *THEN*) ditambahkan kedalam database. Setiap kali pencocokan, dimulai dari *rule* teratas. Setiap *rule* hanya boleh dieksekusi sekali saja. Proses pencocokan berhenti bila tidak ada lagi *rule* yang bisa dieksekusi. (Sutojo, S Mulyanto, Edy Suhartono, 2011, p. 171)



Gambar 2.4 Diagram *Forward Chaining*
 Sumber: (Sutojo, S Mulyanto, Edy Suhartono, 2011)

Kelebihan:

1. Kelebihan utama dari *Forward Chaining* yaitu metode ini akan bekerja dengan baik ketika problem bermula dari mengumpulkan dan menyatukan informasi lalu kemudian mencari kesimpulan apa yang dapat diambil dari informasi tersebut.

2. Metode ini mampu menyediakan banyak sekali informasi dari hanya jumlah kecil data.

Kelemahan:

1. Kelemahan utama metode ini yaitu kemungkinan tidak adanya cara untuk mengenali dimana beberapa fakta lebih penting dari fakta lainnya.
2. Sistem bisa saja menanyakan pertanyaan yang tidak berhubungan. walaupun jawaban dari pertanyaan tersebut penting. Namun hal ini akan membingungkan user untuk menjawab pada subjek yang tidak berhubungan.

2.1.6. Validasi Sistem

Validasi mengacu pada sekumpulan aktifitas yang berbeda yang menjamin bahwa sistem atau perangkat lunak yang dibangun telah sesuai dengan yang diharapkan. Beberapa pendekatan dalam melakukan pengujian untuk validasi sistem antara lain. (Sutojo, S Mulyanto, Edy Suhartono, 2011)

1. *Black-Box Testing* (pengujian kotak hitam)

Pendekatan ini dilakukan dengan menguji sistem atau perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program. Tujuannya untuk mengetahui apakah fungsi-fungsi, masukan, dan keluaran dari sistem atau perangkat lunak sesuai dengan spesifikasi yang dibutuhkan. Pengujian dilakukan dengan membuat kasus uji yang bersifat mencoba semua fungsi dengan menggunakan sistem atau perangkat lunak apakah sesuai dengan spesifikasi yang

dibutuhkan. Kasus uji yang dibuat untuk melakukan *black-box testing* harus dibuat dengan kasus benar dan kasus salah.

2. *White-Box Testing* (pengujian kotak putih)

Pendekatan ini dilakukan dengan menguji sistem atau perangkat lunak dari segi desain dan kode program apakah mampu menghasilkan fungsi-fungsi, masukan, dan keluaran yang sesuai dengan spesifikasi yang dibutuhkan. *White-box testing* dilakukan dengan memeriksa logika dari kode program. Pembuatan kasus uji dapat mengikuti standar pengujian dari standar pemrograman yang ada.

2.2 Penyakit Sapi

Dalam penelitian ini, ada dua variabel yang akan mendukung jalannya penelitian. Variabel yang pertama adalah penyakit *Milk fever (parturient-hypocalcemia)*, dan variabel kedua penyakit *Hypomagnesemia (grass tetany)*, ketiga Penyakit *Asetonemia (Ketosis)*. Dibawah ini adalah penjelasan dari variabel yang akan dibahas untuk memenuhi kebutuhan sistem pakar. Berikut jenis-jenis penyakit yang terdapat pada ternak sapi.

2.2.1. Penyakit *Milk fever (parturient-hypocalcemia)*

Milk fever secara teknis disebut sebagai *parturient hypocalcemia* atau *parturient paresis* yang berarti penurunan kadar kalsium darah pada saat melahirkan. Penyakit ini biasanya disertai dengan penurunan suhu tubuh menjadi subnormal. Kejadian penyakit berlangsung secara akut yang di ikuti dengan penurunan kadar kalsium darah (*hipokalsemia*).



Gambar 2.5 penyakit *Milk Fever (parturient hypocalcemia)*
Sumber : Data Penelitian (2018)

Gejala paresis muncul seiring dengan penurunan kadar kalsium darah dan diikuti dengan *comatose* (pingsan) (Neves, Leno, Stokol, Overton, & McArt, 2017). Umumnya penyakit ini muncul dalam 3 hari setelah melahirkan.

1. Penyebab penyakit

Banyak faktor yang dapat mempengaruhi timbulnya gejala klinis *Milk fever* pada sapi perah, seperti:

- a. Tingkat produksi susu. Tingkat produksi susu yang sangat tinggi sering mengalami *parturient hypocalcemia*.
- b. Umur. Bertambahnya umur seekor hewan akan menurunkan tingkat metabolisme umum. Sapi yang lebih tua mengalami penurunan pergantian mineral tulang dan begitu pula kapasitas penyerapan kalsium oleh lambung. Sapi yang lebih tua akan mengalami resiko tinggi terhadap *parturient hypocalcemia*.
- c. Asupan (*intake*) diet kalsium sebelum melahirkan. Sapi yang mendapat diet kalsium yang berlebihan akan lebih peka di banding yang menerima diet kalsium yang rendah.

d. Statis saluran pencernaan. Proses laktasi pada sapi perah bergantung pada kondisi fungsional saluran pencernaan dan bila terjadi gangguan dapat menimbulkan *hypocalcemia* klinis.

e. Keseimbangan diet. Keseimbangan diet merupakan faktor predisposisi *parturient hypocalcemia* sebagai contoh status asam atau basa (pH) dari total diet.

2. Gejala yang ditimbulkan

Rendahnya kadar kalsium darah dapat menimbulkan *hipersensitivitas* pada membran syaraf serta otot dan kemudian terjadi *hipereksibilitas* dan *grass tetany*. Namun pada stadium akhir *milk fever* akan terjadi paralisis otot. Sapi perah yang menderita *milk fever* umumnya melalui dua stadium yaitu :

- a. Stadium pertama ditandai dengan berbaring pada sternal (*sternal recumbency*). Dan kepala yang mengarah kebelakang.
- b. Stadium kedua melibatkan kolaps dan koma.

3. Pencegahan dan pengobatannya

- a. Mengembalikan kadar kalsium darah pada kondisi normal.
- b. Jangan memberikan rumput mengandung kalsium yang rendah.
- c. Jangan memberikan rumput mengandung magnesium yang rendah.
- d. Selama melahirkan biasanya terjadi periode statis lambung dan akan menurunkan kemampuan sapi mengabsorpsi kalsium.

2.2.2. Penyakit *Grass Tetany* (*Hypomagnesemia*)

Grass tetany disebut juga *hipomagnesemia* akut. Penyakit ini bersifat fatal pada ruminansia yang berlangsung secara cepat tanpa gejala klinis sebelum kematian hewan pada lokasi dimana ditemukan terjatuh.



Gambar 2.6 Penyakit *Grass Tetany* (*Hypomagnesemia*)
Sumber : Data Penelitian (2018)

Grass tetany sering di temukan pada hewan yang di kembalikan di lapangan sapi yang sedang laktasi paling peka terhadap penyakit. Penyakit ini adalah *hypomagnesemia* dimana terjadi penurunan kadar magnesium didalam darah. (Neves et al., 2017)

1. Penyebab penyakit

Grass tetany disebabkan karena menurunnya (*defisiensi*) kadar magnesium dalam darah sapi yang dikembalikan di lapangan. Sapi yang di kandangkan dan diberi *ransum konsentrat* serta rumput kering jarang mengalami *grass tetany*. Masa laktasi dapat meningkatkan kebutuhan magnesium sehingga sapi laktasi menjadi beresiko untuk mengalami *grass tetany*. Masalah utama adalah tidak cukupnya asupan magnesium dari pakan yang disertai dengan beberapa faktor lainnya dimana terbatas nya penyerapan magnesium dari usus. Secara umum, beberapa faktor yang perlu diperhatikan pada penyakit *grass tetany* adalah :

- a. Rumput muda mengandung magnesium yang lebih rendah dibandingkan dengan rumput tua atau kering.
- b. Rumput dengan daun tunggal dan ramping mengandung magnesium lebih rendah dari pada rumput dengan daun jamak (*multiple*) dan lebar.
- c. Pemupukan rumput dengan nitrogen dan kalium dapat menghambat penyerapan magnesium oleh tanaman dan mencegah penyerapan magnesium oleh mukosa usus.
- d. Diet kalsium yang berlebihan dapat menghambat absorpsi magnesium, dan oleh sebab itu pengapuran lahan rumput dapat meningkatkan potensi tetany pada sapi yang di kembalikan.
- e. Pakan yang mengandung protein tinggi yang meningkatkan kandungan anomia dalam rumen dapat menghambat penyerapan magnesium disebabkan terjadinya presipitasi *magnesium amonium forsfat*.
- f. Beberapa jenis pengikat mineral (*magnesium*) terdapat di dalam saluran pencernaan hewan seperti asam *ketobutirat* atau asam *transakonitat*.

4. Gejala yang ditimbulkan

- a. Penyakit ini berlangsung secara akut (sangat cepat) sapi terlihat menderita sangat parah atau mati.
- b. Sapi terlihat berlari – lari secara liar dan jatuh dengan gejala konvulsi.
- c. *Hyperthermia* kontraksi otot yang parah berkaitan dengan tetany serta detak jantung yang tidak teratur dan berbunyi.
- d. Sapi akan mengalami hipersensitif.

- e. Sapi akan terlihat galak serta sulit untuk dipegang gejala ini berlangsung beberapa hari sampai memasuki stadium konvulsi.

5. Pencegahan dan pengobatannya

- a. Memberikan teraoi magnesium.
- b. Segera berikan larutan magnesium sulfat 25% sebanyak 400 ml secara oral.
- c. Hindari pemberian secara intravenus karena dapat mempercepat timbulnya serangan jantung yang bersifat fatal.
- d. Bila terlihat gejala spasmus, maka berikan sedativa untuk menenangkan hewan dan mengurangi resiko serangan jantung.
- e. Berikan kombinasi preparat magnesium dan garam kalsium secara perlahan melalui intravena.
- f. Memberikan suplemen magnesium pada masa – masa kritis seperti pada musim hujan dan penggembalaan pada rumput basah.

2.2.3. Penyakit *Asetonemia (Ketosis)*

Asetonemia juga di sebut *ketosis* atau *hypoglisemia ketosis* yang sering ditemukan pada sapi perah pada masa laktasi. *Ketosis* disebabkan karena ketidak seimbangan antara input dan output energi metabolisme.



Gambar 2.7 Penyakit *Asetonemia (Ketosis)*
Sumber : Data Penelitian (2018)

Penyakit ini merupakan gangguan metabolisme yang terjadi pada awal laktasi dan berkaitan erat dengan *hypoglisemia*, *ketonaemia*, *ketonuria*, hilang nafsu makan, kehilangan berat badan dan inkoordinasi. (Neves et al., 2017)

1. Penyebab penyakit

Mekanisme asetonemia diawal dengan gangguan metabolisme karbohidrat untuk menghasilkan energi. Pakan yang mengandung karbohidrat tinggi dipecah oleh mikro organisme rumen menjadi asam propionat dan kemudian dibawa kedalam hati untuk digunakan menghasilkan glukosa. Sapi pada masa awal laktasi umumnya tidak mampu mengkonsumsi cukup energi dari diet pakannya untuk mencukupi kebutuhan energi dalam menghasilkan susu, oleh karena itu, tubuh mendapatkan energi dari cadangan lemak. Lemak dipecahkan menjadi asam lemak kemudian melalui aliran darah dibawa ke jaringan hati.

2. Gejala Yang ditimbulkan

Gejala awal dari *asetonemia* yaitu:

- a. penurunan produksi susu dan bau aseton dari pernapasan, susu dan urin sapi penderita.
- b. Sapi tidak mau mengkonsumsi konsentrat meskipun tetap mengkonsumsi rumput kering atau silase.
- c. Bulu terlihat kusam dan sapi mengalami *lethargy*(lemah).
- d. Berhentinya ruminasi dan penurunan produksi susu secara drastis.
- e. Aseton dapat mempengaruhi otak dimana sapi menjadi liar.
- f. Keluar air liur dari mulut.

g. Menjilati benda sekitar dan berdiri dengan menjulurkan kepala ke atas.

3. Pengobatan dan pencegahan

Pengobatan dan pencegahan penyakit dapat dilakukan melalui tiga tahapan pengobatan yaitu:

- a. Penyuntikan obat untuk meningkatkan kadar glukosa dan mempercepat metabolisme hati. Obat yang digunakan umumnya *anabolik steroid* atau kelompok *glukokortikoid*.
- b. Obat yang diberikan secara oral untuk meningkatkan kadar gula darah dan memperbaiki metabolisme seperti *sodium propionat* dan *gliserol*.
- c. Perawatan individual secara terpisah pada hewan yang sakit dengan memberikan pakan khusus seperti *molases*.

Tabel 2.3 Variabel, Penyebab Dan Indikator

Variabel	Penyebab	Indikator
PENYAKIT SAPI	1. tingkat produksi susu 2. umur 3. asupan 4. saluran pencernaan 5. keseimbangan diet.	1. Penyakit <i>Milk Fever</i> (<i>Parturient hypocalcemia</i>)
	1. Menurunnya kadar magnesium.	2. Penyakit <i>Grass Tetany</i> (<i>Hypomagnesemia</i>)
	2. Gangguan metabolisme karbohidrat. 3. energi tidak seimbang	3. Penyakit <i>Asetonemia</i> (<i>Ketosis</i>)

Sumber: Data Penelitian (2018)

2.3 Software Pendukung

Software merupakan istilah umum untuk data yang diformat dan disimpan secara digital, termasuk program komputer. Dokumentasinya dan berbagai informasi yang bisa di baca dan ditulis oleh komputer. Definisi *software* lain ialah intruksi dalam bahasa (formal) pemrograman , disusun oleh programmer untuk dikerjakan komputer.

2.3.1. UML

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

UML (Unified Modeling Language) adalah notasi yang lengkap untuk membuat visualisai model suatu sistem sistem berisi informasi dan fungsi, tetapi secara normal digunakan untuk memodelkan sistem komputer didalam pemodelan obyek guna menyajikan sitem yang berorientasi objek pada orang lain. (A.S & Shalahuddin, 2011, p. 113)



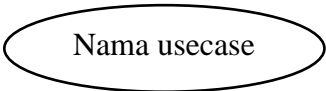


Gambar 2.8 Logo UML
Sumber : Data Penelitian (2017)

2.3.1.1. Use case Diagram

Menurut (A.S & Shalahuddin, 2011, p.130) Diagram *Use case* merupakan pemodelan untuk kelakuan (behavior) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara garis besar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Ada 2 hal utama pada *use case* antaranya:

1. Aktor merupakan orang yang berinteraksi pada sistem informasi baik pengguna maupun pembuat sistem informasi tersebut.
2. *Use case* adalah fungsional yang disediakan sistem sebagai tempat bertukar pesan.

Tabel 2.4 Simbol- Simbol Pada *Use case* Diagram



Simbol	Deskripsi
Use Case 	Fungsionalitas yang disediakan sistem sebagai unit – unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja diawal <i>frase</i> nama <i>use case</i>
 Nama aktor	Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda diawal <i>frase</i> nama aktor.
Asosiasi / <i>association</i> 	Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.
Extensi / <i>Extend</i>	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang





<pre><<Extend>> -----></pre>	ditambah dapat berdiri sendiri walau tanpa <i>use case</i> tambahan. Itu biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan.
<pre>Generalisasi/generalization ────────────────────></pre>	Hubungan generalisasi dan spesialisasi (umum-khusus) antara 2 buah <i>use case</i> dimana fungsi satu adalah fungsi yang lebih umum dari lainnya.
<pre>Menggunakan / include / uses <<include>> ----- > <<uses>> ────────────────────></pre>	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsi atau sebagai syarat dijalankan <i>use case</i> ini.

2.3.1.2. Activity Diagram

Menurut (A.S & Shalahuddin, 2011, p. 134) Diagram aktivitas atau activity diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Yang perlu diketahui bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem.

Tabel 2.5 Activity diagram




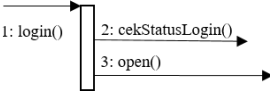

Simbol	Deskripsi
Status awal 	Status awal aktivitas sistem, sebuah diagram aktifitas memiliki sebuah status awal
Aktifitas 	Aktifitas yang dilakukan sistem, aktifitas biasanya diawali dengan kata kerja
Percabangan/ <i>decision</i>	Asosiasi percabangan dimana jika

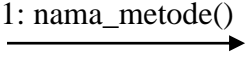
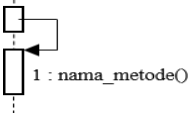
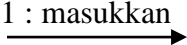
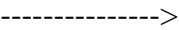
	ada pilihan aktifitas lebih dari satu
Penggabungan/ <i>join</i> 	Asosiasi penggabungan dimana lebih dari satu aktifitas digabungkan menjadi satu
Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktifitas memiliki sebuah status akhir
<i>Swimlane</i> 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktifitas yang terjadi atau

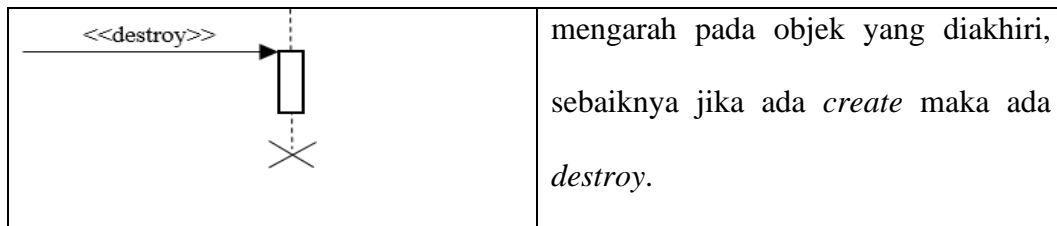
2.3.1.3. *Sequence Diagram*

Menurut (A.S&Shalahuddin, 2011, p.137) Diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterim antar objek.

Tabel 2.6 *Sequence Diagram*

Simbol	Deskripsi
<p data-bbox="512 421 592 450">Aktor</p>  <p data-bbox="300 719 368 748">Atau</p> <div data-bbox="327 853 539 958" style="border: 1px solid black; padding: 5px; width: fit-content; margin: 10px auto;"> <p data-bbox="360 869 505 898"><u>Nama aktor</u></p> </div> <p data-bbox="300 1003 539 1032">Tanpa waktu aktif</p>	<p data-bbox="831 421 1337 1032">Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, walaupun simbol aktor itu berbentuk gambar orang, tetapi belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda diawal frase nama aktor.</p>
<p data-bbox="300 1081 560 1111">Garis hidup/ <i>lifeline</i></p> 	<p data-bbox="831 1081 1294 1111">Menyatakan kehidupan suatu objek</p>
<p data-bbox="300 1303 384 1332">Objek</p> <div data-bbox="331 1384 678 1480" style="border: 1px solid black; padding: 5px; width: fit-content; margin: 10px auto;"> <p data-bbox="352 1400 657 1429"><u>Nama objek : nama kelas</u></p> </div>	<p data-bbox="831 1303 1305 1406">Menyatakan objek yang berinteraksi pesan.</p>
<p data-bbox="300 1525 459 1554">Waktu aktif</p> 	<p data-bbox="831 1525 1273 1554">Menyatakan objek dalam keadaan</p>  <p data-bbox="831 1733 1198 1762">aktif dan berinteraksi pesan.</p>
<p data-bbox="300 1807 523 1836">Pesan tipe <i>create</i></p> <p data-bbox="300 1881 459 1910"><<create>></p> 	<p data-bbox="831 1807 1326 1986">Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat.</p>

<p>Pesan tipe <i>call</i></p> <p>1: nama_metode() </p>	<p>Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau diri sendiri.</p>  <p>Arah panah mengarah pada objek yang memiliki operasi/metode, karena ini memanggil operasi/metode maka operasi/metode yang dipanggil harus ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi.</p>
<p>Pesan tipe <i>send</i></p> <p>1 : masukkan </p>	<p>Menyatakan bahwa suatu objek mengirim data/masukan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.</p>
<p>Pesan tipe <i>return</i></p> <p>1 : keluaran </p>	<p>Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian.</p>
<p>Pesan tipe <i>destroy</i></p>	<p>Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah</p>



Sumber: (A.S. & Shalahuddin, 2011)

2.3.1.4. Class Diagram

Menurut (A.S.&Shalahuddin, 2011, p. 122) Diagram kelas atau *class diagram* menggambarkan struktur dari segi pendefinisian kelas-kelas yang dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dari metode atau operasi. Atribut merupakan variabel- vareiablel yang dimiliki oleh suatu kelas sedangkan operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas.

Kelas-kelas yang ada pada struktur sistem harus dapat melakukan fungsi-fungsi sesuai dengan kebutuhan sistem. Susunan struktur kelas yang baik pada diagram kelas sebaiknya memiliki jenis-jenis kelas berikut:

1. Kelas main

Kelas yang memiliki fungsi awal dieksekusiketika sistem dijalankan.

2. Kelas yang menangani tampilan sistem

Kelas yang mendefinisikan dan mengatur tampilan ke pemakai.

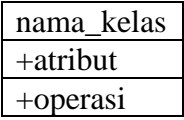




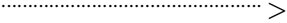

3. Kelas yang diambil dari pendefinisian *use case*

Kelas yang menangani fungsi-fungsi yang harus ada diambil dari pendefinisian *use case*.

4. Kelas yang diambil dari pendefinisian data

Kelas yang digunakan untuk memegang atau membungkus data menjadi sebuah kesatuan yang diambil maupun akan disimpan kebasis data.

Tabel 2.7 *Class Diagram*

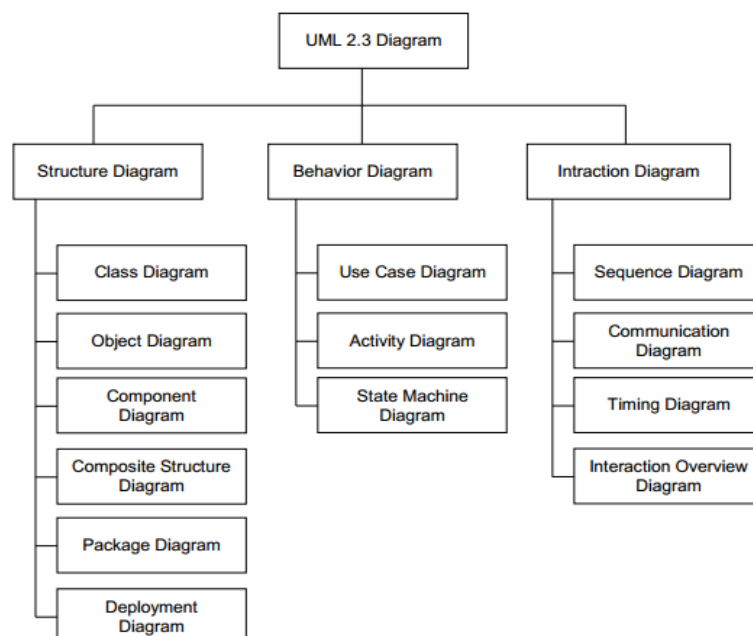
Simbol	Deskripsi
Kelas 	Kelas pada struktur sistem
Antarmuka / <i>interface</i>  nama_interface	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek
Asosiasi / <i>association</i> 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
Asosiasi berarah / <i>directed association</i> 	Relasi antara kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
Generalisasi 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus)
Kebergantungan / <i>defedency</i> 	Relasi anata kelas dengan makna kebergantungan antar kelas
Agregasi / <i>aggregation</i> 	Relasi antar kelas dengan makna semua-bagian (<i>whole-part</i>)

Sumber:(A.S. & Shalahuddin, 2011)

2.3.1.4. Diagram UML

Menurut (A.S. & Shalahuddin, 2011, p. 120) UML 2.3 memiliki 13 macam diagram yang dikelompokkan dalam 3 kategori. Dibawah ini adalah pembagian dari kategori diagram UML:

1. *Strukture diagram*: kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan.
2. *Behavior diagram*: kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada sebuah sistem.
3. *Interaction diagram*: kumpulan diagram – diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun interaksi antar subsistem pada suatu sistem.



Gambar 2.9 Diagram UML
Sumber : Data Penelitian (2018)

2.3.2. *starUML*

starUML adalah *platform* pemodelan yang mendukung UML (*Unified Modeling Language*). *StarUML* mempunyai sebelas macam diagram yang berbeda yang juga mendukung pendekatan MDA (*Model Driven Architecture*) dengan mendukung konsep UML *profile*. *StarUML* dapat memaksimalkan produktifitas dan kualitas dari suatu *software project*.

Fitur – fitur yang ada didalam *starUML* adalah :

1. dukungan terhadap UML:
 - a. *usecase diagram*.
 - b. *class diagram*.
 - c. *activity diagram*.
 - d. *sequence diagram*.
2. dukungan terhadap beberapa bahasa pemrograman:
 - a. *Java profile, code generator, dan reverse engineering*.
 - b. *C++ profile, code generator, dan reserve engineering*.
 - c. *C# profile, code generator, dan reserve engineering*.
 - d. *Microsoft office documen generator*.
 - e. *Microsoft word document template and generation*.
 - f. *Microsoft excel document template and generation*.
 - g. *Microsoft powerpoint document template and generation*.
3. *Customizable code generation*.
4. Mendukung teknologi MDA.
5. Diagram yang dapat diperluas (menentukan sendiri jenis diagram).

6. *Extensibility*.
7. Kompatible yang tinggi.
8. *Editing*.
9. *User interface*

2.2.4. PHP

PHP merupakan singkatan dari “*Hypertext Preprocessor*”. Pada awalnya PHP merupakan kependekan dari *Personal Home Page* (situs personal) dan pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1995 dan pada saat itu PHP masih bernama FI (*Form Interpreter*), yang wujudnya berupa sekumpulan script yang digunakan untuk mengelola data form dari *Web*. Selanjutnya Rasmus merilis kode sumber tersebut untuk umum. PHP adalah sebuah bahasa scripting yang terpasang pada HTML. (Saputra, 2012)

PHP digunakan untuk membuat tampilan *Web* menjadi lebih dinamis, dengan PHP Anda bisa menampilkan atau menjalankan beberapa file dalam satu file dengan cara di-*include* atau *require*. PHP itu sendiri sudah dapat berinteraksi dengan database walaupun dengan kelengkapan yang berbeda. (Saputra, 2012)



Gambar 2.10 Logo
Sumber : <http://php.net>

2.2.5. HTML (*Hyper Text Markup Language*)



Gambar 2.11 Logo HTML
Sumber: Data Penelitian (2018)

HTML merupakan singkatan dari *Hyper Text Markup Language*. HTML biasa disebut bahasa paling dasar dan penting yang digunakan untuk menampilkan dan mengelola tampilan pada halaman *Website*.

HTML versi 5 yang paling marak dibicarakan didunia maya, HTML 5 layaknya sebuah html biasa yang sering kita gunakan dalam membangun aplikasi *Web*, hanya saja html 5 ini memiliki keunggulan disbanding versi terdahulunya. HTML 5 juga mampu menyederhanakan kode-kode html terdahulu menjadi lebih ringkas. Hal yang paling mencolok adalah tersedianya fitur baru seperti elemen multimedia, misalnya `<audio>` dan `<video>`, yang tak lain adalah fungsi untuk memutar audio dan juga video. Pada versi HTML sebelumnya, jika anda ingin memutar perangkat multimedia, haruslah menggunakan perintah `<embed>`. Dengan adanya penambahan fitur dan tag khusus yang ada di HTML 5, akan membuat semuanya menjadi sangat mudah. Dokumen html memiliki sebuah struktur yang harus kita ikuti aturan pembuatanya. Kita akan mengenal beberapa elemen-elemen wajib yang ada pada file html apabila kita ingin membangun sesuatu pondasi kerangka wenside, elemen tersebut diantaranya:

1. Elemen HTML `<html>` merupakan tag dasar apabila kita ingin melalui suatu dokumen html. Tag ini merupakan perintah wajib bagi pemogram *Web* untuk menuliskan tag pertama dalam dokumen html.
2. Elemen *Head* `</head>` merupakan tag berikutnya setelah html (`<html>`), yang berfungsi untuk menuliskan keterangan tentang dokumen *Web* yang akan ditampilkan, elemen ini nantinya akan diakhiri dengan tanda penutup `</head>`.
3. Elemen Title `<title>` merupakan suatu elemen yang harus dituliskan didala elemen *head* yang digunakan untuk memberikan judul/ informasi pada caption browser *Web* tentang topic/ tema atau judul dari suatu dokumen *Web* yang ditampilkan pada browser
4. Elemen Body `<body>` merupakan bagian utama dalam dokumen *Web*. Jika kita ingin menampilkan suatu teks atau informasi atau yang dikenal dengan sebutan konten, maka kita harus meletakkan teks tersebut pada elemen *body*.

2.2.6. XAMPP

XAMPP adalah sebagai server yang berdiri sendiri (*localhost*) yang terdiri beberapa program seperti *Apache HTTP Server*, *MySQL database*, dan penerjemah bahasa yang dituliskan dengan bahasa pemrograman PHP dan Perl. Nama XAMPP sendiri merupakan singkatan dari X (empat sistem operasi apapun), *Apache*, *MySQL*, *PHP* dan *Perl*. Program ini tersedia dalam GNU *General Publik license* dan bebas, merupakan *Web* server yang mudah untuk digunakan yang dapat menampilkan halaman *Web* yang dinamis secara gratis dan mendukung instalasi di *Linux* dan *Windows*. (Saputra, 2012)



Gambar 2.12 Logo XAMPP
Sumber : Aplikasi

2.2.7. Notepad++

Adalah program aplikasi yang berguna untuk mengedit teks dan script kode pemrograman. Software Notepad++ dibuat dan dikembangkan oleh tim Notepad++. Perangkat lunak komputer ini memiliki kelebihan pada peningkatan kemampuan sebuah program teks editor, lebih dari sekedar program Notepad bawaan windows. Notepad++ bisa mengenal tag dan kode dalam berbagai bahasa pemrograman dan bersifat *opensource* (gratis). (Saputra, 2012)



Gambar 2.13 Logo Notepad++
Sumber : Aplikasi

2.2.8. MySQL

Sejarah diciptakan pada tahun 1979, oleh Michael "Monty" widenius, seorang programmer komputer asal Swedia. Monty mengembangkan sebuah sistem *database* sederhana dinamakan UNIREG yang menggunakan koneksi Low-level ISAM database engine dengan *indexing*. MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (*database management*

system) atau DBMS yang Multithread, multi-user, dengan sekitar 6 juta instalasi diseluruh dunia. MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis di bawah lisensi GNU *General Public License*(GPL). (Saputra, 2012)



Gambar 2.14 Logo MySQL
Sumber : Data Penelitian (2018)

1. Keistimewaan MySQL

MySQL memiliki beberapa keistimewaan, antara lain:

- a. Probabilitas. MySQL dapat berjalan stabil pada berbagai sistem operasi seperti Windows, Linux, FreeBSD, Mac OS X Server, Solaris, Amiga.
- b. Perangkat lunak sumber terbuka. MySQL didistribusikan sebagai perangkat lunak sumber terbuka, dibawah lisensi GPL sehingga dapat digunakan secara gratis.
- c. Multi-user. MySQL dapat digunakan beberapa pengguna dalam waktu yang bersamaan tanpa mengalami masalah atau konflik.
- d. *Performance tuning*. MySQL memiliki kecepatan yang menakjubkan menangani query sederhana, dengan kata lain dapat memproses lebih banyak SQL per satuan waktu.

- e. Ragam tipe data. MySQL memiliki ragam tipe data yang sangat kaya, seperti *signed/unsigned integer, float, double, char, text, date, timestamp*, dan lain-lain.
- f. Perintah dan fungsi. MySQL memiliki operator dan fungsi secara penuh yang mendukung perintah *Select* dan *Where* dalam perintah (*query*).
- g. Keamanan. MySQL memiliki beberapa lapisan keamanan seperti level subnetmask, nama host, dan izin akses *user* dengan perizinan yang mendetail secara sandi terenskripsi.
- h. Skalabilitas dan pembatasan. MySQL mampu menangani basis data dalam skala besar, dengan jumlah rekaman (*records*) lebih dari 50 juta dan 60 ribu tabel serta 5 miliar baris. Selain itu batas indek dapat ditampung mencapai 32 indeks pada tiap tabelnya.
- i. Konektivitas. MySQL dapat melakukan koneksi dengan klien menggunakan protokol TCP/IP, *Unix socket* (UNIX), atau named pipes (NT).
- j. Lokalisasi. MySQL dapat mendeteksi pesan kesalahan pada klien dengan menggunakan lebih dari dua puluh bahasa. Meskipun demikian, bahasa indonesia belum termasuk didalamnya.
- k. Antar muka. MySQL memiliki antar muka (*interface*) terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API (*Application Programming Interface*).
- l. Klien dan peralatan. MySQL dilengkapi dengan berbagai peralatan (*tool*) yang dapat digunakan untuk administrasi basis data, dan pada setiap peralatan yang ada disertakan petunjuk *online*.

m. Struktur tabel. MySQL memiliki struktur tabel yang lebih fleksibel dalam menagani *ALTER TABLE*, dibandingkan basis data lainnya semacam *PostgreSQL* ataupun *Oracle*.

2. Kelebihan MySQL

- a. Berlisensi GPL dan multi platform.
- b. dapat diintegrasikan dengan bebrapa bahasa pemrograman seperti Net, java, python, perl yang merupakan bahasa pemrograman yang paling domain dikalangan programmer.
- c. Mendukung ODBC untuk sistem operasi windows sehingga bisa digunakan aplikasi yang berjalan di windows.
- d. Bisa dijalankan pada spesifikasi hardware yang rendah karena lebih hemat *resource memory* (dibandingkan database lain) sehingga mudah digunakan untuk bahan pembelajaran.
- e. MySQL dapat mendeteksi pesan kesalahan pada klien dengan menggunakan lebih dari 20 bahasa meskipun bahasa indonesia belum termasuk di dalamnya.

3. Kekurangan MySQL

1. Banyak mengklaim kurang support terhadap pemrograman Visual/Desktop, sehingga sedikit yang menggunakan untuk aplikasi visual.
2. Karena berlisensi GPL sehingga sulit mendapatkan *update* untuk problem yang urgent, sehingga perusahaan skala menengah keatas lebih memilih RDBMS berlisensi dan support seperti *Oracle* dan MS SQL server.

3. Sangat diragukan dalam menangani data skala besar, karena ada beberapa opini yang pro dan kontrak terhadap kemampuan MySQL terhadap pengolahan data yang besar.

2.4 Penelitian Terdahulu

Penelitian yang dilakukan oleh peneliti merupakan penelitian terapan yang dilakukan dengan mengambil beberapa penelitian terdahulu yang relevan dengan penelitian ini, yang dijadikan peneliti sebagai referensi. Dari penelitian terdahulu dapat disimpulkan.

Restia Dwi Oktavianing Tyas, Arief Andy Soebroto, M. Tamzil Furqon (2015). **Pengembangan sistem pakar diagnosa penyakit sapi potong dengan metode fuzzy k-nearest neighbour**. Hasil dan pembahasan merupakan tahap hasil perancangan dan implementasi sistem pakar. . Pengujian sistem pakar diagnosa penyakit sapi potong terdiri dari tiga bagian yaitu pengujian black box, pengujian variasi nilai variabel k dan variasi data latih dan pengujian nilai variabel m.(Technology, Dwi, Tyas, Soebroto, & Furqon, 2015).

Reski Mai Candra, Weni Rahim (2014). **Sistem pakar diagnosa bibit unggul sapi dan kambing dengan metode certainty factor**. Berdasarkan dari hasil analisa, perancangan dan implementasi pada sistem ini, dapat diambil beberapa kesimpulan diantaranya adalah. Metode Certainty Factor dapat di terapkan untuk menentukan jenis induk yang dapat menghasilkan bibit unggul. Berdasarkan hasil pengujian user acceptance test menyatakan bahwa sistem dapat memudahkan para peternak menentukan calon induk sapi dan kambing yang dapat

menghasilkan bibit unggul. Berdasarkan pengujian oleh pakar, menyatakan bahwa sistem Dianosa Bibit Unggul Sapi Dan Kambing Dengan Metode Certainty Factor sesuai dengan yang diterapkan oleh pakar. (Ilmiah et al., 2014).

I Kadek Dwi Gandika Supartha, Ida Nirmala Sari(2014). **Sistem pakar diagnosa awal penyakit kulit pada sapi bali dengan metode forward chaining dan certainty factor. Setelah dilakukan analisis dan pengujian terhadap Sistem.** Pakar Diagnosa Awal Penyakit Kulit pada Sapi Bali dengan Menggunakan Metode forward chaining dan Certainty Factor, dapat disimpulkan bahwa. Sistem ini dapat digunakan untuk mengetahui jenis penyakit yang dialami oleh sapi Bali di BPTU Sapi Bali berdasarkan inputan gejala-gejala yang dimasukan oleh user. Sistem ini mampu menyimpan representasi pengetahuan pakar dengan metode Foward Chaining dan Certainty Factor berdasarkan nilai Measure of Believe (MB) dan Measure of Disbelieve (MD). Selain menghasilkan penyakit sistem pakar ini dapat menjelaskan definisi dan cara pengobatan penyakit kulit pada sapi Bali. (Dwi, Supartha, Sari, & Informatika, 2014)

Indra Fauz i Rohman, Prihastuti Harsani, Arie Qur'ania. (2016) **Aplikasi Diagnosis Penyakit Sapi Menggunakan Metode Certainty Factors Berbasis Android.** Berdasarkan penelitian yang dilakukan pada aplikasi diagnosispenyakit sapi, maka dapat disimpulkan bahwa metode *Certainty Factors* dapat diimplementasikan dalam aplikasi Diagnosis Penyakit sapi dan penerapan metode *Certainty Factors* sangat membantu dalam pembuatan aplikasi ini, dengan mengandalkan bobot dari masing- masing gejala. Setelah melalui pengujian didapat bahwa Implementasi aplikasi dengan perhitungan manual hasilnya sama

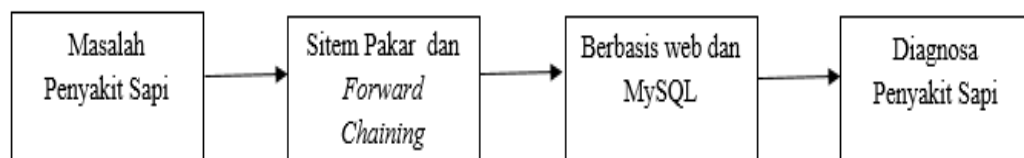
atau valid. Dengan tingkat Akurasi dari aplikasi dengan metode certainty factors ini mencapai 86%. Berdasarkan hasil pengujian yang telah dilakukan pada perangkat android versi 4.0 atau Ice Cream Sandwich sampai android versi 5.0 atau Lollipop, semua fitur yang ada pada aplikasi dapat berjalan dengan sangat optimal. Aplikasi ini akan sangat membantu kedua belah pihak, antara peternak dan dokter hewan. Peternak sangat terbantu karena aplikasi ini sudah dapat menentukan penyakit yang diderita sapi sehingga lebih cepat dalam penanganannya dan dokter hewan pun terbantu dengan aplikasi ini karena dokter dapat dengan cepat memfokuskan penanganan sesuai saran aplikasi. (Harsani, Komputer, & Bogor, 2016)

Alexius Endy Budianto.(2015) **Aplikasi Sistem Pakar Menggunakan Metode Backward Chaining Untuk Analisis Penyakit Hewan Ternak.** Berdasarkan permasalahan yang telah dibahas dan diselesaikan melalui laporan ini maka terdapat beberapa kesimpulan yaitu untuk membangun sistem pakar ini dibutuhkan inputan gejala-gejala yang menggunakan metode backward chaining atau runut balik. Berdasarkan hasil pengujian aplikasi sistem pakar ini berguna untuk membantu para peternak sapi untuk mengenali gejala-gejala yang ditimbulkan dan memberikan tindakan yang tepat untuk mengatasi serangan penyakit pada hewan ternak sapi. (Budianto & Kom, 2015)

2.5 Kerangka Berpikir

Sistem pakar untuk mendiagnosa penyakit sapi merupakan suatu sistem pakar yang dirancang sebagai alternatif untuk membantu masyarakat khusus nya

peternak sapi dalam memberikan kemudahan dan solusi terhadap penyakit sapi. Dengan menggunakan metode Forward Chaining berbasis *Web* sangat membantu para peternak sapi dimanapun dan kapanpun dalam mengetahui dan menangani gejala apa yang ada pada sapi tersebut sehingga bisa meminimalisir ternak sapi yang terjangkit penyakit sampai berakibat kematian. Dengan berbasiskan sistem *Web* diharapkan tingkat kemudahan pengaksesan sistem pakar ini lebih mudah dan efisien.



Gambar 2.15 Kerangka Berfikir
Sumber : Data Penelitian (2018)

Uraian Kerangka Berfikir:

1. Masalah Penyakit sapi: penelitian ini mengangkat tiga variabel penyakit metabolik pada ternak sapi yang pertama penyakit *Milk Fever (Parturient-Hypocalcemia)*, Kedua penyakit *Grass Tetany (Hypomagnesemia)*, ketiga Penyakit *Asetonemia (Ketosis)*.
2. Sistem pakar dan metode *Forward Chaining*: penelitian ini membangun sistem pakar untuk mendiagnosa penyakit pada hewan ternak dengan menggunakan metode *Forward Chaining*.
3. Berbasis *Web* dan MySQL: penelitian ini menggunakan tools berbasis *Web* dan MySQL
4. Diagnosa Penyakit Sapi: berdasarkan hasil analisis penelitian maka didapatkan diagnosa jenis penyakit sapi serta cara penanganannya.

BAB III METODE PENELITIAN

2.3. Desain Penelitian

Supaya penelitian ini dapat selesai dan berjalan sesuai dengan tujuan yang diinginkan . oleh sebab itu, membuat desain penelitian keadaan yang akan terjadi pada saat dilapangan. Desain dibawah ini di rancang oleh peneliti untuk menyelesaikan masalah yang ada.



Gambar 3.1 Desain Penelitian
Sumber : Data Penelitian (2018)

Berdasarkan gambar diatas, peneliti dapat menyampaikan penjelasan uraian sebagai berikut:

1. Identifikasi Masalah

Penelitian dimulai dengan melakukan studi pendahuluan agar dapat memecahkan permasalahan yang di hadapi dan bisa melanjutkan proses selanjutnya.

2. Merumuskan Masalah

Setelah mendapatkan informasi yang spesifik maka penelitian ini merumuskan masalah agar bisa di jawab dengan melalui penelitian.

3. Menentukan Tujuan Penelitian

Pada penelitian ini bertujuan untuk mengetahui bagaimana sistem pakar untuk mendiagnosa penyakit sapi menggunakan metode *Forward Chaining*.

4. Mempelajari Literatur

Untuk mendukung jalannya penelitian. Maka peneliti mencari dan mempelajari sumber-sumber pengetahuan berupa buku-buku teori, jurnal-jurnal penelitian dan sumber pustaka yang berkaitan dengan penelitian. Diantaranya kecerdasan buatan, sistem pakar, PHP, UML, dan *MySQL*.

5. Menganalisa data-data yang telah didapatkan

Setelah data-data yang berkaitan dengan penyakit sapi didapat baik melalui studi literatur dengan Dokter hewan sebagai pakar. Setelah data-data sudah didapat dan di analisa kemudian data tersebut disederhanakan

dan dikelompokkan agar lebih mudah dilakukan proses pengolahan data yang dibutuhkan dalam sistem pakar.

6. Mengelola Data Sistem Pakar Dengan Metode *Forward Chaining*

Sistem pakar dapat menghasilkan suatu kesimpulan berdasarkan aturan atau kaidah yang ada. Oleh karena itu, data-data yang sudah dianalisa kemudian diolah menggunakan metode *forward chaining* untuk membuat kaidah (*rule*) yang digunakan saat sistem pakar melakukan penelusuran sebelum menyimpulkan hasil.

7. Mengimplementasikan Kedalam Program Berbasis *Web*

Penelitian ini melakukan kegiatan perancangan mulai dari basis pengetahuan, desain UML, desain *database*. Dan desain antarmuka. Setelah dilakukan pengkodean untuk mentranslasikan desain yang telah dibuat kedalam program perangkat lunak sehingga menghasilkan sebuah program komputer. Pengkodean dilakukan menggunakan bahasa pemrograman PHP dikombinasikan dengan bahasa pemrograman HTML, dan *database MySQL* melalui *editor* teks *notepad++*.

8. Menguji Hasil Penelitian

Proses ini bertujuan untuk memberikan informasi kepada masyarakat umum atau petenak sapi perah memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan. Sistem juga diuji dengan membandingkan hasil diagnosa pakar dengan hasil diagnosa sistem untuk melihat apakah sistem berjalan dengan baik.

9. Menarik Kesimpulan.

Tahapan terakhir penelitian ini yaitu menyimpulkan hasil penelitian yang berisi jawaban singkat terhadap rumusan masalah berdasarkan data-data yang ada. Dan penelitian ini juga memberi solusi untuk membantu *users* dalam memecahkan permasalahan yang ada.

3.2. Pengumpulan data

Teknik pengumpulan data disini bertujuan untuk mendapatkan hasil yang benar-benar valid dan dapat di percaya. Dengan adanya data yang sudah di peroleh data dapat di terapkan pada sistem pakar. Teknik pengumpulan data yang digunakan dalam penelitian ini adalah:

3.2.2. Wawancara

Pengambilan data ini melakukan wawancara secara terstruktur atau tatap muka dengan pakar hewan Drh. Jusak Wira Hardja, M.Si. yang berlokasi di Happy Pet Shop di Ruko Glass Centre no.14-15, Lubuk Baja, Kota Batam, Kepri.

3.2.3. Studi Literatur

Studi literatur adalah segala usaha yang dilakukan oleh peneliti untuk mendapat informasi yang relevan dengan topik atau masalah yang sedang diteliti. Dengan cara mengumpulkan, membaca, dan memahami referensi teoritis yang berasal dari buku-buku teori, jurnal-jurnal penelitian yang berkaitan dengan judul yang diangkat.

3.3. Operasional Variabel

Variabel yang digunakan dalam penelitian ini adalah proses diagnosa penyakit sapi menggunakan metode *forward chaining* berbasis *Web*.

Berikut ini adalah tabel variabel dan indikator dalam penelitian ini yaitu (Tabel 3.1):

Tabel 3.1 Variabel Dan Indikator

Variabel	Indikator
PENYAKIT SAPI	Penyakit <i>Milk Fever</i> (<i>Parturient hypocalcemia</i>)
	Penyakit <i>Grass Tetany</i> (<i>Hypomagnesemia</i>)
	Penyakit <i>Asetonemia</i> (<i>Ketosis</i>)

Sumber: Data Penelitian (2018)

Dalam **Tabel 3.1** diatas menjelaskan hubungan antara variabel dan indikator. Variabelnya adalah Penyakit Sapi, sedangkan indikatornya adalah 3 jenis penyakit pada sapi: Penyakit *Milk Fever* (*parturient hypocalcemia*), Penyakit *Grass Tetany* (*Hypomagnesemia*) dan Penyakit *Asetonemia* (*Ketosis*).

3.4. Metode Perancangan Sistem

Dalam perancangan sistem dimana diperlukan suatu keahlian untuk perancangan elemen-elemen komputer yang menggunakan sistem pemilihan peralatan dan program untuk sistem yang baru. Dibawah ini akan membahas metode perancangan sistem pakar penyakit sapi.

3.4.1. Desain Basis Pengetahuan

Desan basis pengetahuan bisa dilakukan setelah peneliti melakukan proses pengumpulandata dan fakta melalui wawancara dengan Pakar dan studi literatur tentang materi yang berkaitan dengan data-data. Sumber fakta yang didapat berupa data-data yang berhubungan dengan penyakit sapi. Pengetahuan dan fakta tersebut ditampilkan dalam tabel berikut:

Tabel 3.2 Tabel Jenis Penyakit, penyebab dan Solusi

Kode	Jenis Penyakit	Penyebab	Solusi
P01	Penyakit <i>Milk Fever (Parturient hypocalcemia)</i>	<ol style="list-style-type: none"> 1. tingkat produksi susu 2. umur 3. asupan 4. saluran pencernaan 5. keseimbangan diet. 	<ol style="list-style-type: none"> 1. Mengembalikan kadar kalsium darah pada kondisi normal. 2. Jangan memberikan rumput mengandung kalsium dan magnesium yang rendah.
P02	Penyakit <i>Grass Tetany (Hypomagnesemia)</i>	<ol style="list-style-type: none"> 1. Menurunnya kadar magnesium. 	<ol style="list-style-type: none"> 1. Memberikan terapi magnesium 2. Segera berikan larutan magnesium sulfat 25% sebanyak 400 ml secara oral. 3. Hindari pemberian secara intravenus karena dapat mempercepat timbulnya serangan jantung yang

			<p>bersifat fatal.</p> <ol style="list-style-type: none"> 4. Bila terlihat gejala <i>spasmus</i>, maka berikan sedativa untuk menenangkan hewan dan mengurangi resiko serangan jantung. 5. Berikan kombinasi preparat magnesium dan garam kalsium secara perlahan melalui intravena. 6. Memberikan suplemen magnesium pada masa - masa kritis seperti pada musim hujan dan pengembalaan pada rumput basah.
P03	Penyakit Asetonemia (<i>Ketosis</i>)	<ol style="list-style-type: none"> 1. Gangguan metabolisme karbohidrat 2. energi tidak seimbang. 	<ol style="list-style-type: none"> 1. Penyuntikan obat untuk meningkatkan kadar glukosa dan mempercepat metabolisme hati. 2. Obat yang diberikan secara oral untuk meningkatkan kadar gula darah dan memperbaiki metabolisme seperti sodium <i>propionat</i> dan <i>gliserol</i>. 3. Perawatan individual secara terpisah pada hewan yang sakit dengan memberikan pakan khusus seperti <i>molases</i>.

Sumber: Data Penelitian (2018)

Dalam **Tabel 3.2** menjelaskan solusi yang harus dilakukan sesuai jenis penyakit sapi yang diderita. Kemudian pemberian kode untuk setiap jenis penyakit sapi tersebut. Sistem pakar yang menggunakan metode *forward chaining*

pada penelitian ini digunakan untuk melakukan diagnosa penyakit pada hewan sapi sehingga data solusi tidak diberikan kode.

Tabel 3.3 Tabel Gejala Penyakit

Kode	Ciri – Ciri Tipe Kepribadian
G01	Stadium pertama ditandai dengan berbaring pada sternal (<i>sternal recumbency</i>). Dan kepala yang mengarah kebelakang.
G02	Stadium kedua melibatkan kolaps dan koma.
G03	Penyakit ini berlangsung secara akut (sangat cepat) sapi terlihat menderita sangat parah atau mati tanpa gejala klinis.
G04	Sapi terlihat berlari – lari secara liar dan jatuh dengan gejala <i>konvulsi</i> .
G05	<i>Hyperthermia</i> kontraksi otot yang parah berkaitan dengan tetany serta detak jantung yang tidak teratur dan berbunyi.
G06	Sapi akan mengalami hipersensitif.
G07	Sapi akan terlihat galak serta sulit untuk dipegang gejala ini berlangsung beberapa hari sampai memasuki stadium konvulsi
G08	penurunan produksi susu dan bau aseton dari pernapasan, susu dan urin sapi penderita.
G09	Sapi tidak mau mengkonsumsi konsentrat meskipun tetap mengkonsumsi rumput kering atau <i>silase</i> .
G10	Bulu terlihat kusam dan sapi mengalami <i>lethargy</i> (lemah).
G11	Berhentinya ruminasi dan penurunan produksi susu secara drastis.
G12	Aseton dapat mempengaruhi otak dimana sapi menjadi liar.
G13	Keluar air liur dari mulut.
G14	Menjilati benda sekitar dan berdiri dengan menjulurkan kepala ke atas.

Sumber: Data Penelitian (2018)

Pada **Tabel 3.3** menjelaskan gejala dari setiap jenis penyakit sapi dan kemudian diberi kode.

Data aturan merupakan data yang berisi relasi antara data jenis penyakit dan data gejala penyakit yang telah diberi kode sebelumnya. Relasi antar data tersebut

disusun berdasarkan sumber pengetahuan dan fakta yang telah didapatkan. Data aturan ini disusun untuk memudahkan peneliti dalam menyusun kaidah yang akan digunakan sebagai basis pengetahuan dalam sistem pakar pada penelitian ini. Susunan data aturan yang digunakan dapat dilihat pada tabel dibawah ini:

Tabel 3.4 Tabel Data Aturan

Kode Jenis Penyakit	Kode Gejala Penyakit
P01	G01, G02
P02	G03, G04, G05, G06, G07
P03	G08, G09, G10, G11 G12, G13, G14

Sumber: Data Penelitian (2018)

Berdasarkan **Tabel 3.4** diatas, maka kaidah (*rule*) yang akan digunakan dalam sistem pakar adalah sebagai berikut:

1. Kaidah 1: *IF G01 AND G02 THEN P01*
2. Kaidah 2: *IF G03 AND G04 AND G05 AND G06 AND G07 THEN P02*
3. Kaidah 3: *IF G08 AND G09 AND G10 AND G11 AND G12 AND G13 AND G14 THEN P03*

Berdasarkan data aturan yang telah disusun pada **Tabel 3.4**, dapat dijelaskan sebagai berikut:

1. Jika gejalanya adalah stadium pertama ditandai dengan berbaring pada sternal (*sternal recumbency*) (G01), stadium kedua melibatkan kolaps dan koma

- (G02) maka jenis penyakitnya adalah Penyakit *Milk Fever (Parturient hypocalcemia)* (P01).
2. Jika gejalanya adalah penyakit ini berlangsung secara akut (sangat cepat) sapi terlihat menderita sangat parah atau mati tanpa gejala klinis (G03), sapi terlihat berlari – lari secara liar dan jatuh dengan gejala konvulsi (kejang-kejang) (G04), *hyperthermia* kontraksi otot yang parah berkaitan dengan tetany serta detak jantung yang tidak teratur dan berbunyi (G05), sapi akan mengalami hipersensitif (G06), sapi akan terlihat galak serta sulit untuk dipegang gejala ini berlangsung beberapa hari sampai memasuki stadium konvulsi (G07), hemat (CK12) maka jenis penyakitnya adalah Penyakit *Grass Tetany (Hypomagnesemia)* (P02).
 3. Jika gejalanya adalah penurunan produksi susu dan bau aseton dari pernapasan, susu dan urin sapi penderita (G08), Sapi tidak mau mengkonsumsi konsentrat meskipun tetap mengkonsumsi rumput kering atau silase (G09), Bulu terlihat kusam dan sapi mengalami lethargy (lemah) (G10), Berhentinya ruminasi dan penurunan produksi susu secara drastis (G11), aseton dapat mempengaruhi otak dimana sapi menjadi liar (G12), keluar air liur dari mulut (G13), Menjilati benda sekitar dan berdiri dengan menjulurkan kepala ke atas (G14) maka jenis penyakitnya adalah Penyakit Asetonemia (*Ketosis*) (P03).

Berdasarkan kaidah yang telah dibuat tersebut maka tabel keputusannya adalah berikut:

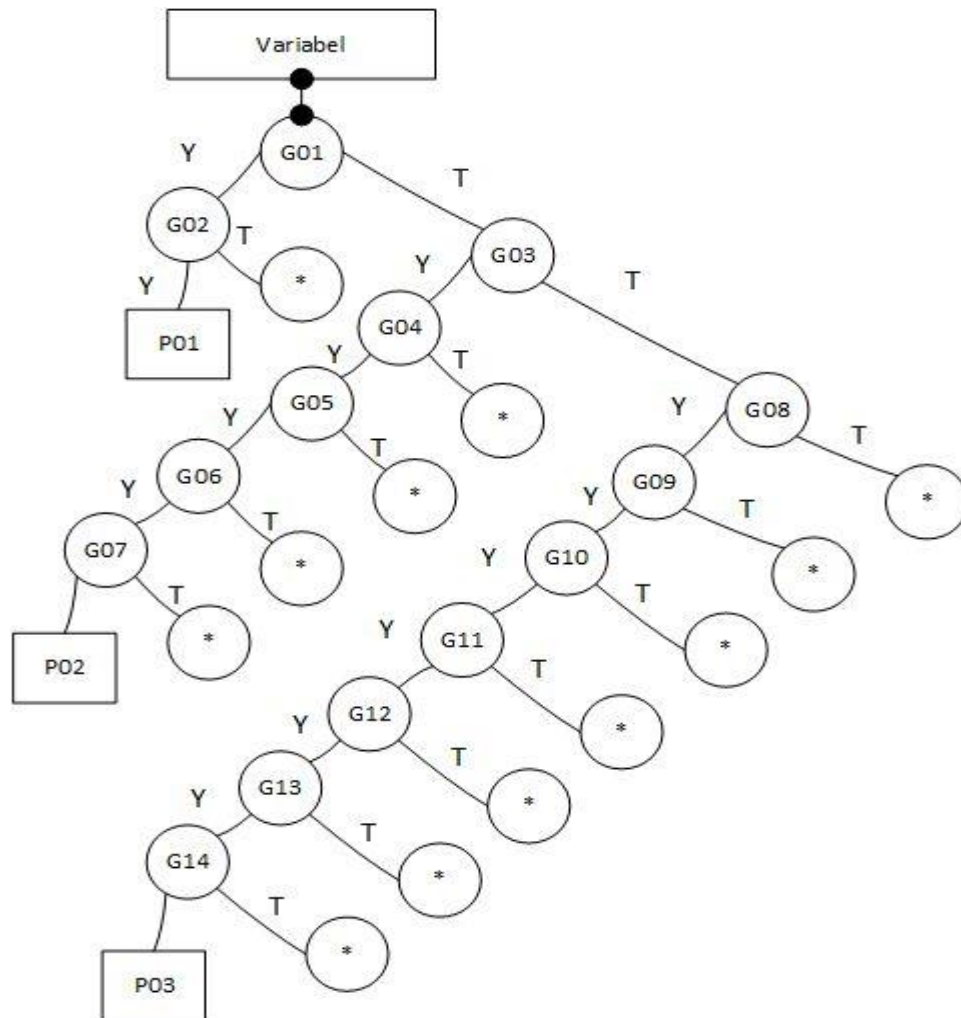
Tabel 3.5 keputusan

Penyakit Gejala	P01	P03	P03
G01	√		
G02	√		
G03		√	
G04		√	
G05		√	
G06		√	
G07		√	
G08			√
G09			√
G10			√
G11			√
G12			√
G13			√
G14			√

Sumber: Data Penelitian (2018)

Pada **Tabel 3.5** diatas, baris gejala diberi tanda centang untuk kolom penyakit yang memenuhi aturan dari masing-masing gejala. Hal ini dibuat untuk memudahkan dalam menyusun aturan kaidah produksi sistem pakar yang akan dibuat.

Berdasarkan tabel keputusan (**Tabel 3.5**) diatas maka dapat dibuat pohon keputusan (**Gambar 3.2**) sebagai berikut:



Gambar 3.2 Pohon Keputusan
Sumber: Data Penelitian (2018)

Keterangan:

Y: Ya

T: Tidak

*: Tidak ada data /terdiagnosis

P01 – P03: Keterangan lengkap pada **Tabel 3.2**

G01 – G14: Keterangan lengkap pada **Tabel 3.3**

Pada **Gambar 3.2** menunjukkan pohon keputusan yang memperlihatkan hubungan antara gejala penyakit dan jenis penyakit sapi.

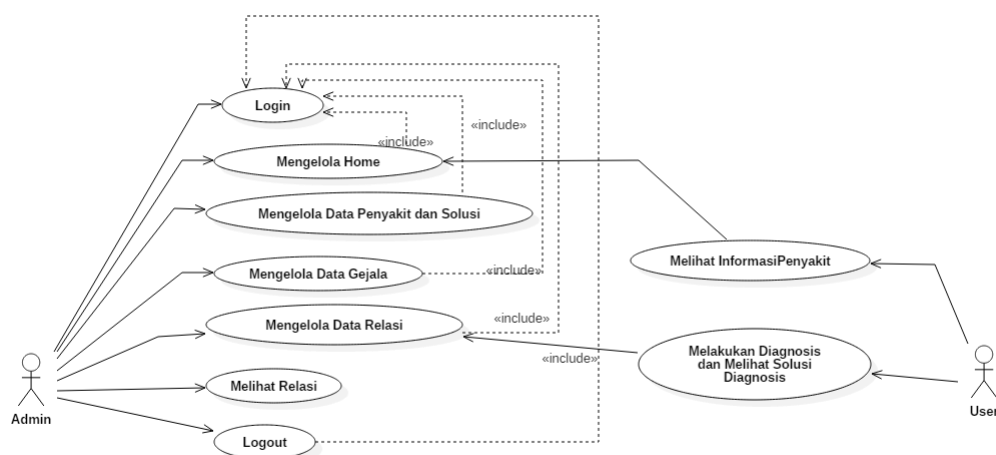
3.4.2. Desain UML (*Unified Modeling Language*)

UML (Unified Modeling Language) adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek (A.S. & Shalahuddin, 2011: 113)

Desain *UML* dibuat untuk memudahkan dalam pembuatan program. Pemodelan *UML* menggunakan alat bantu *software StarUML* versi 2.8.0. Berikut ini adalah diagram UML yang digunakan dalam perancangan program:

3.4.2.1. Use case Diagram

Use case diagram yang akan digunakan pada sistem diagnosis penyakit sapi seperti pada **Gambar 3.3**:



Gambar 3.3 Use case Diagram
Sumber: Data Penelitian (2018)

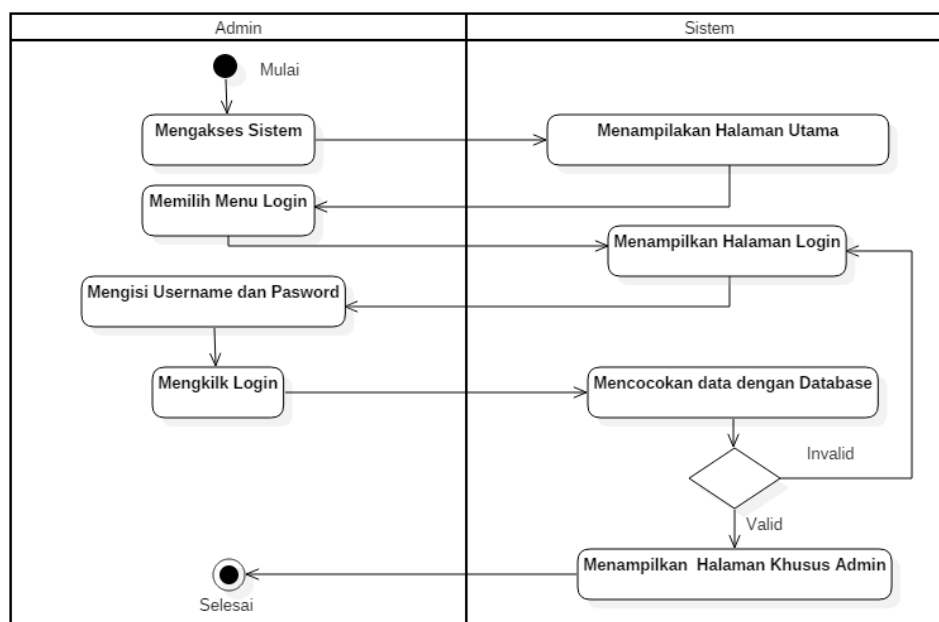
Terdapat 2 aktor yaitu *admin* dan *user*. *Admin* melakukan interaksi dengan sistem berupa mengelola *home*, mengelola data penyakit dan solusi, mengelola data gejala, mengolah data relasi, melihat relasi dan *logout*. Semua interaksi dapat dilakukan setelah *admin* melakukan *Login* pada menu *Login*. Sedangkan *user* berinteraksi dengan sistem yaitu dapat melihat menu home atau informasi penyakit sapi dan melakukan diagnosis mengenai penyakit sapi dan dapat melihat hasil dari diagnosis yang dilakukan yaitu berupa solusi.

3.4.2.2. Activity Diagram

Berikut ini adalah *activity diagram* yang dirancang dalam penelitian ini:

a. Activity Diagram Login Admin

Activity diagram Login admin merupakan *UML* yang menggambarkan kegiatan pengguna pada halaman khusus *admin*.

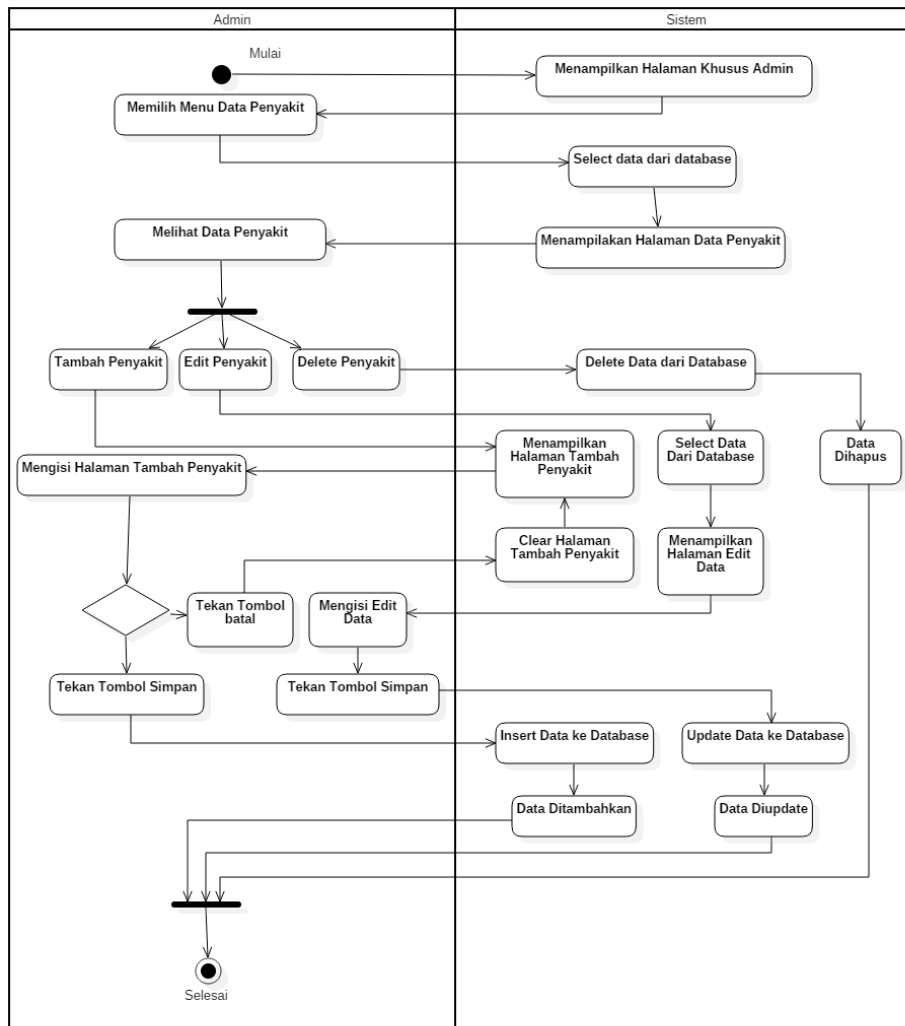


Gambar 3.4 Activity Diagram Login
Sumber: Data Penelitian (2018)

Pada **Gambar 3.4** diatas, proses *Login admin* adalah *admin* mengases sistem, kemudian sistem akan menampilkan halaman utama. Kemudian *admin* akan memilih menu admin dan sistem akan menampilkan halaman khusus admin. Admin akan mengisi *username* dan *password* pada menu *Login*, kemudian klik tombol *Login*. Maka sistem akan mengecek *username* dan *password* kemudian dicocokkan dengan data yang ada di *database*. Jika *username* dan *password* tidak sesuai dengan yang ada di *database* maka sistem akan menampilkan pesan kesalahan dan sistem kembali menampilkan halaman menu *Login*, apabila benar maka sistem akan menampilkan halaman khusus *admin*.

b. *Activity diagram* Mengelola Menu Penyakit

Activity diagram mengelola menu penyakit merupakan kegiatan *admin* dalam mengelola data jenis penyakit dan solusi yang digunakan. Berikut ini gambar *activity diagram* mengelola menu penyakit (**Gambar 3.5**):



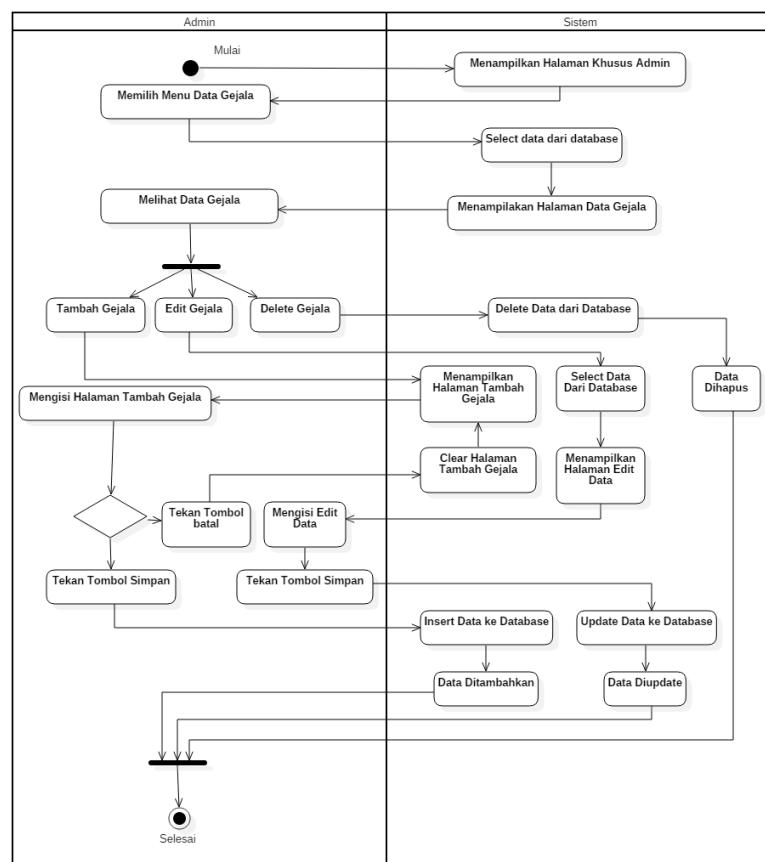
Gambar 3.5 Activity Diagram Mengelola Menu Penyakit
Sumber: Data Penelitian (2018)

Pada **Gambar 3.5** diatas, halaman khusus *admin* yang terbuka setelah *admin* melakukan *Login*. Sistem akan menampilkan menu-menu pada halaman khusus *admin*, kemudian *admin* memilih menu penyakit. Sistem memanggil data dari database dan menampilkan halaman data penyakit. *Admin* melihat 3 pilihan yaitu tambah, *edit*, dan *delete*. Jika *admin* mengklik tombol tambah maka sistem akan menampilkan halaman tambah penyakit, kemudian admin mengisi data penyakit dan solusi. Kemudian memilih tombol simpan maka data akan

dimasukkan ke *database* kemudian data ditambahkan di *database*, jika *admin* menekan tombol batal maka sistem akan membersihkan halaman tambah penyakit. Jika *admin* memilih tombol *edit*, maka sistem akan mengambil data dari *database*. *Admin* mengedit data kemudian klik tombol simpan. Sistem akan melakukan *update database*. Jika *admin* menekan tombol *delete* maka data yang ada di *database* akan terhapus maka proses selesai.

c. Activity Diagram Mengelola Gejala

Activity diagram mengelola menu gejala merupakan kegiatan *admin* dalam mengelola gejala penyakit yang digunakan. Berikut ini gambar *activity diagram* mengelola menu gejala (**Gambar 3.6**):

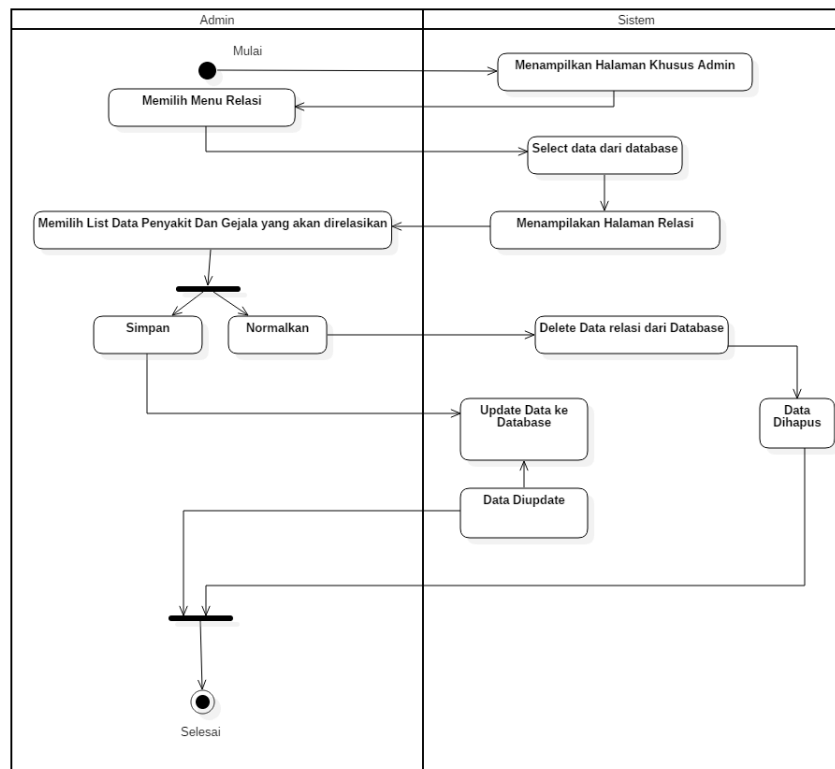


Gambar 3.6 Activity Diagram Mengelola Gejala
Sumber: Data Penelitian (2018)

Pada **Gambar 3.6**, Halaman khusus *admin* yang terbuka setelah *admin* melakukan *Login*. Sistem akan menampilkan menu-menu pada halaman khusus *admin*, kemudian *admin* memilih menu gejala. Sistem memanggil data dari database dan menampilkan halaman data gejala. *Admin* melihat 3 pilihan yaitu tambah, *edit*, dan *delete*. Jika *admin* mengklik tombol tambah maka sistem akan menampilkan halaman tambah gejala, kemudian *admin* mengisi data gejala. Kemudian memilih tombol simpan maka data akan dimasukkan ke database kemudian data ditambahkan di database, jika *admin* menekan tombol batal maka sistem akan membersihkan halaman tambah gejala. Jika *admin* memilih tombol *edit*, maka sistem akan mengambil data dari database. *Admin* mengedit data kemudian klik tombol simpan. Sistem akan melakukan *update database*. Jika *admin* menekan tombol *delete* maka data yang ada di database akan terhapus maka proses selesai.

d. *Activity Diagram* Mengelola Menu Relasi

Activity diagram mengelola menu relasi merupakan diagram *UML* yang menggambarkan kegiatan *admin* dalam mengelola data relasi sesuai dengan data aturan penelitian ini. Berikut ini gambar *activity diagram* mengelola data relasi (**Gambar 3.7**):

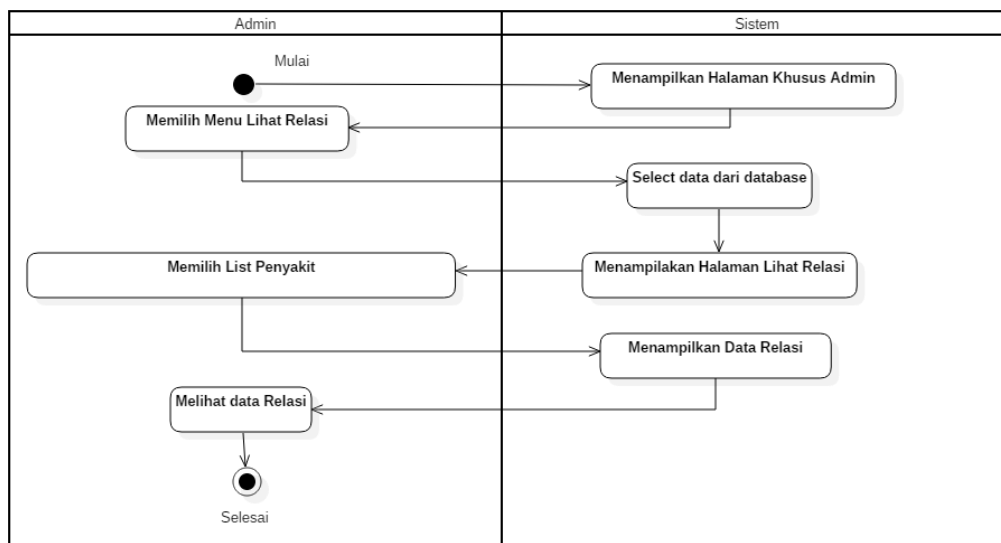


Gambar 3.7 Activity Diagram Mengelola Menu Relasi
Sumber: Data Penelitian (2018)

Pada **Gambar 3.7** diatas, *admin* mulai mengakses sistem dan sistem menampilkan halaman khusus admin. *admin* memilih menu drelasi kemudian sistem menampilkan halama menu relasi. Admin dapat memilih jenis penyakit dan gejala yang akan direlasikan. *Admin* dapat melakukan 2 pilihan yaitu simpan dan normalkan. Pertama, jika *admin* menekan tombol simpan maka sistem akan mengupdate data relasi ke *database*. Kedua, jika Admin menekan tombol normalkan , maka sistem akan menghapus data relasi antar data penyakit dan gejala dan data relasi akan dihapus dari *database* maka proses selesai.

e. *Activity Diagram* Mengelola Menu Lihat Relasi

Activity diagram mengelola menu lihat relasi merupakan diagram *UML* yang menggambarkan kegiatan *admin* dalam data relasi atau aturan dari *rule forward chaining*. Berikut ini gambar *activity diagram* mengelola menu lihat relasi (**Gambar 3.8**):

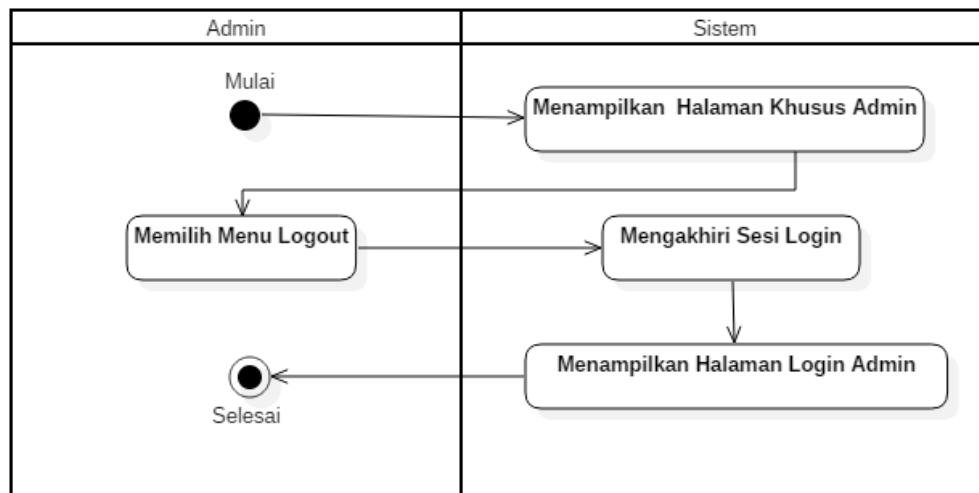


Gambar 3.8 *Activity Diagram* Mengelola Menu Lihat Relasi
Sumber: Data Penelitian (2018)

Pada **Gambar 3.8** diatas, *admin* mulai dengan mengakses sistem dan sistem menampilkan halaman khusus admin. *Admin* memilih menu lihat relasi kemudian sistem akan mengambil data dari *database* dan menampilkan halaman lihat relasi. *Admin* dapat melihat list penyakit yang akan dilihat relasinya, maka sistem akan menampilkan halaman data relasi yang telah dibuat. *Admin* akan melihat data relasi apakah telah sesuai dengan aturan penelitian ini, maka proses selesai.

f. *Activity diagram* menu *logout*

Activity diagram menu *logout* merupakan diagram *UML* yang menggambarkan kegiatan admin dalam menggunakan menu *logout*. Berikut ini gambar *activity diagram* menu *logout* (**Gambar 3.9**):

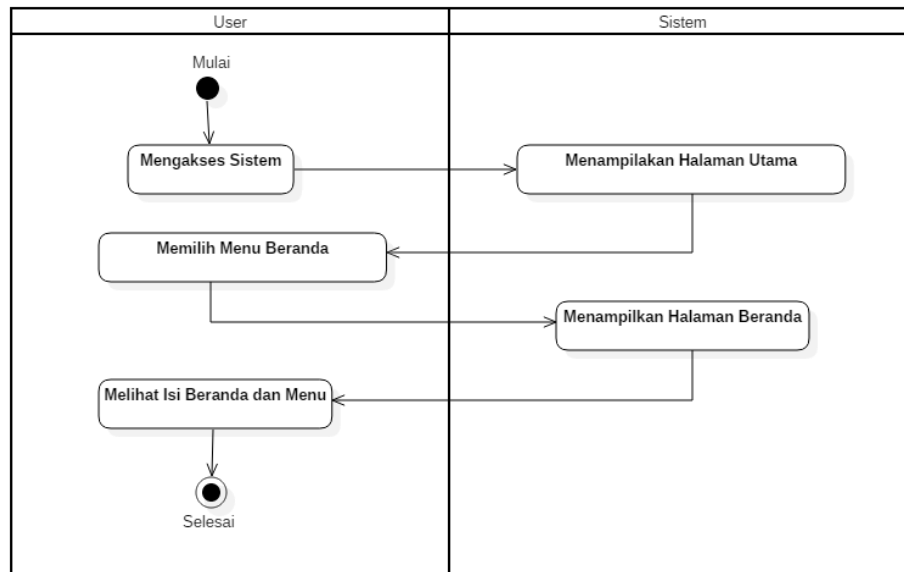


Gambar 3.9 *Activity Diagram Logout*
Sumber: Data Penelitian (2018)

Pada **Gambar 3.9** diatas, *admin* mulai dengan mengakses sistem dan sistem menampilkan halaman khusus admin. *Admin* memilih menu *logout* kemudian sistem akan mengakhiri sesi *Login admin* dan sistem akan menampilkan halaman menu *Login*.

g. *Activity Diagram* Melihat Menu Beranda

Activity diagram melihat menu beranda merupakan diagram *UML* yang menggambarkan kegiatan *user* melihat menu beranda. Berikut ini gambar *activity diagram* melihat menu beranda (**Gambar 3.10**):

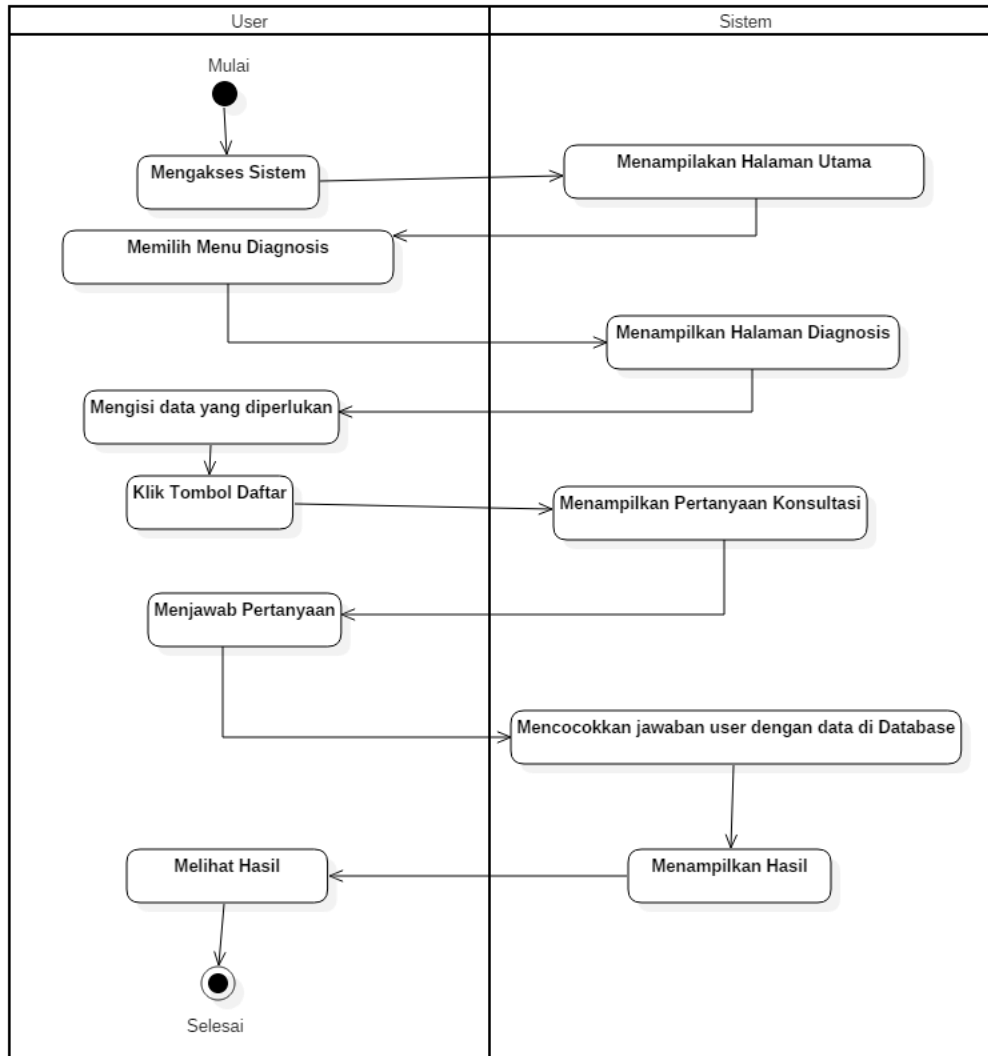


Gambar 3.10 Activity Diagram Melihat Menu Beranda
Sumber: Data Penelitian (2018)

Pada **Gambar 3.10** diatas, *user* mulai dengan mengakses sistem dan sistem menampilkan halaman utama. *User* memilih menu beranda kemudian sistem menampilkan halaman beranda. *User* dapat melihat info mengenai penyakit sapi yang bermanfaat di sistem maka proses melihat menu beranda selesai.

h. Activity diagram Menu Diagnosis

Activity diagram menu diagnosis merupakan diagram *UML* yang menggambarkan kegiatan *user* melakukan diagnosis penyakit sapi. Berikut ini gambar *activity diagram* menu diagnosis (**Gambar 3.11**):



Gambar 3.11 Activity Diagram Menu Diagnosis
 Sumber: Data Penelitian (2018)

Pada **Gambar 3.11** diatas, *user* mulai dengan mengakses sistem dan sistem menampilkan halaman utama. *User* memilih menu diagnosis kemudian sistem menampilkan halaman menu diagnosis. *User* akan mengisi data terlebih dahulu dan menekan tombol daftar, Sistem akan menampilkan halaman pertanyaan diagnosis. *User* akan menjawab pertanyaan sesuai dengan apa yang terjadi pada hewan sapinya. Kemudian sistem akan mencocokkan jawaban dengan data yang ada didatabases lalu sistem akan menampilkan hasil atau solusinya. *User* dapat

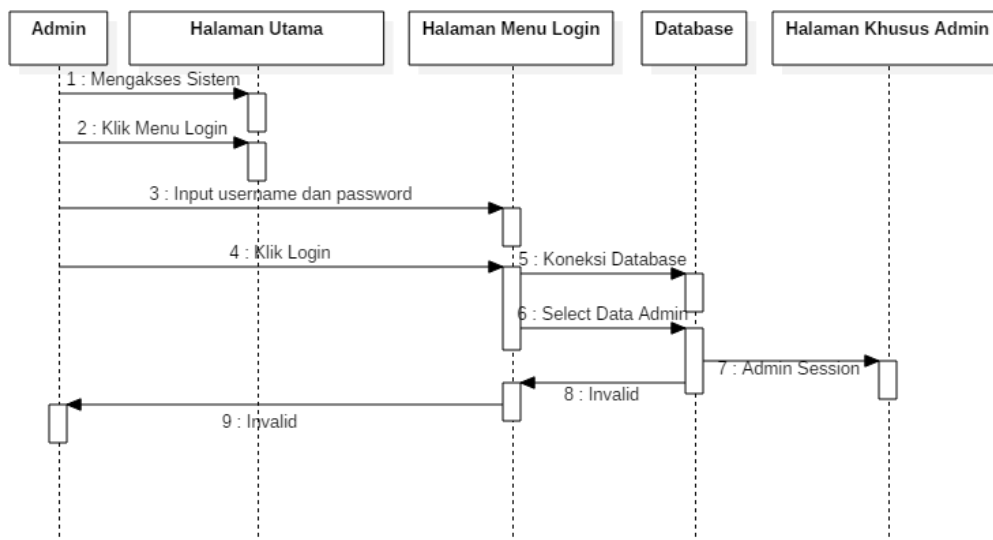
melihat solusi dari gejala yang terjadi pada hewan sapinya. maka proses melakukan diagnosis selesai.

3.4.2.3. Sequence Diagram

Sequence Diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterim antar objek. (A.S & Shalahuddin, 2011, p.137)

a. Sequence diagram Login Admin

Sequence diagram Login admin merupakan urutan waktu kegiatan *admin* saat melakukan *Login*. Berikut ini gambar *sequence diagram Login admin* (**Gambar 3.12**):



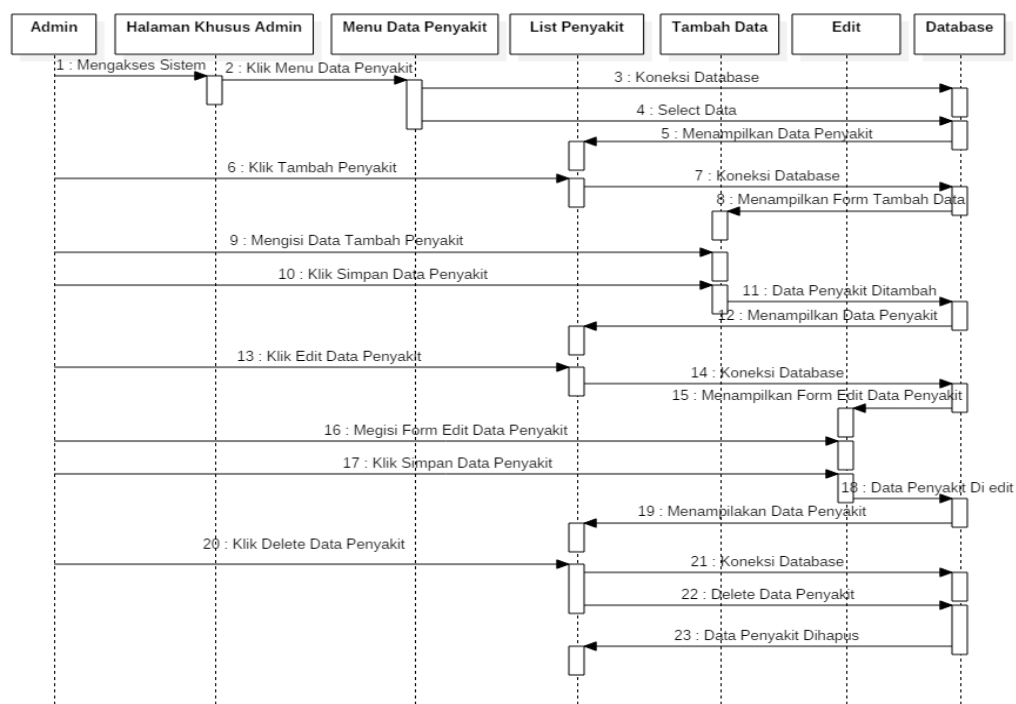
Gambar 3.12 Sequence Diagram Login Admin
Sumber: Data Penelitian (2018)

Admin mengakses sistem lalu sistem menampilkan halaman utama. *Admin* mengklik menu admin, maka sistem akan menampilkan halaman *Login* admin.

Admin akan memasukkan *username* dan *password* ke menu *Login* kemudian *admin* melakukan klik tombol *Login*. Dari Halaman *Login* Admin, sistem akan mengecek *username* dan *password* yang sudah dimasukkan kemudian sistem akan terhubung dengan *database*, setelah itu *username* dan *password* akan dicocokkan oleh sistem. Jika *username* dan *password* valid maka halaman khusus admin akan ditampilkan. Jika *username* dan *password* invalid (tidak sesuai dengan *database*) maka sistem akan menampilkan pesan gagal di halaman menu *Login*.

b. Sequence Diagram Mengelola Menu Penyakit

Sequence diagram mengelola menu penyakit merupakan urutan waktu kegiatan *admin* saat mengelola data penyakit dan solusi. Berikut ini gambar *sequence diagram* mengelola menu penyakit (**Gambar 3.13**)

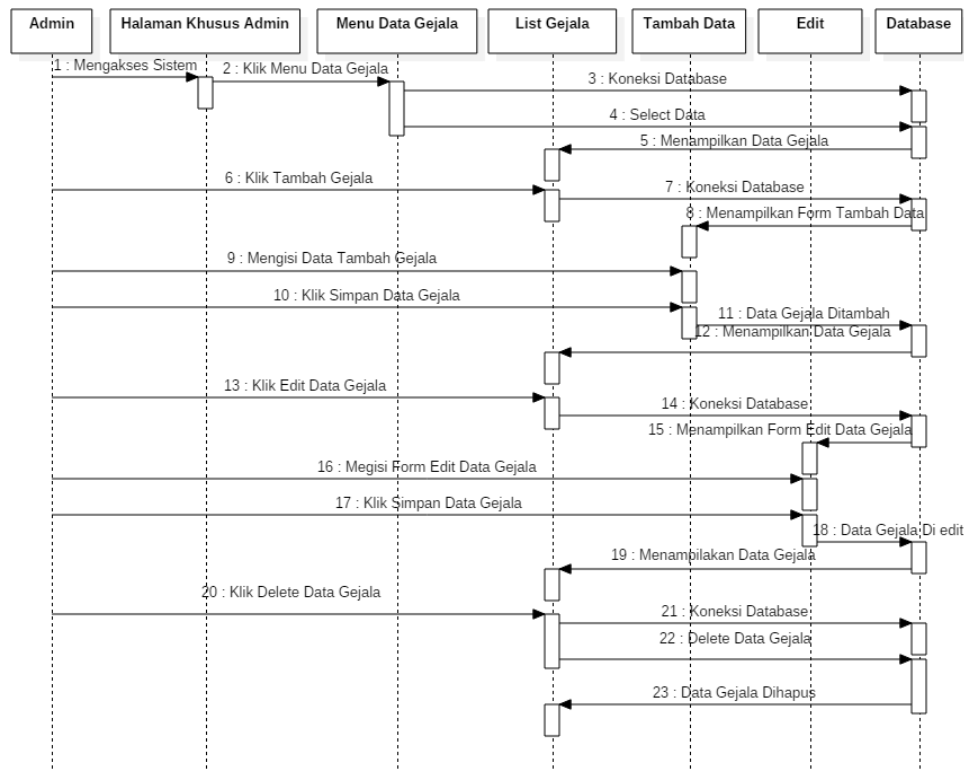


Gambar 3.13 Sequence Diagram Mengelola Menu Penyakit
Sumber: Data Penelitian (2018)

Admin mengakses sistem, kemudian menampilkan halaman khusus admin. Lalu *admin* mengklik menu penyakit, *admin* menekan tombol tambah penyakit. Sistem akan menampilkan form tambah data. Kemudian admin akan mengisi form tambah penyakit lalu admin mengklik tombol simpan. Data ditambahkan ke *database* dan sistem akan menampilkan kembali menu penyakit. Kemudian admin klik *edit* data penyakit dan sistem akan menampilkan form edit data, kemudian admin akan mengisi form edit data penyakit lalu mengklik tombol simpan. Data user diupdate pada database, kemudian sistem akan menampilkan form menu data penyakit kembali. Admin mengklik tombol *delete*, sistem pun akan mendelete data penyakit pada *database*.

c. *Sequence Diagram* Mengelola Gejala

Sequence diagram mengelola gejala merupakan urutan waktu kegiatan *admin* saat melakukan pengelolaan gejala penyakit. Berikut ini gambar *sequence diagram* mengelola data gejala (**Gambar 3.14**):

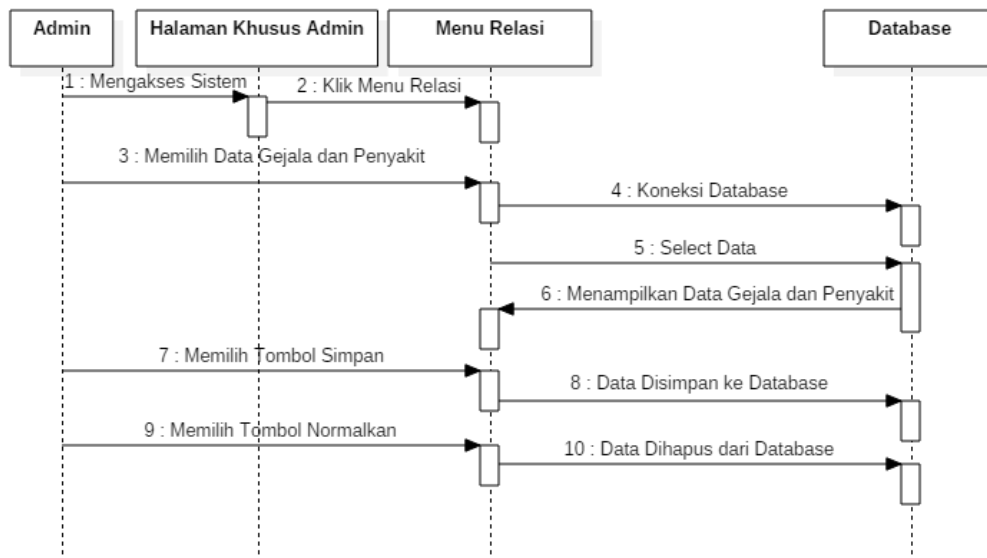


Gambar 3.14 Sequence Diagram Mengelola Gejala
Sumber: Data Penelitian (2018)

Admin mengakses sistem, kemudian menampilkan halaman khusus admin. Lalu *admin* mengklik menu gejala, *admin* menekan tombol tambah gejala. Sistem akan menampilkan form tambah data. Kemudian admin akan mengisi form tambah gejala lalu admin mengklik tombol simpan. Data ditambahkan ke *database* dan sistem akan menampilkan kembali menu gejala. Kemudian admin klik *edit* data gejala dan sistem akan menampilkan form edit data, kemudian admin akan mengisi form edit data gejala lalu mengklik tombol simpan. Data user diupdate pada *database*, kemudian sistem akan menampilkan form menu data gejala kembali. Admin mengklik tombol *delete*, sistem pun akan mendelete data gejala pada *database*.

d. *Sequence Diagram* Mengelola Menu Relasi

Sequence diagram mengelola menu relasi merupakan urutan waktu kegiatan *admin* saat mengelola data relasi sesuai aturan kaidah. Berikut ini gambar *sequence diagram* mengelola menu relasi (**Gambar 3.15**):

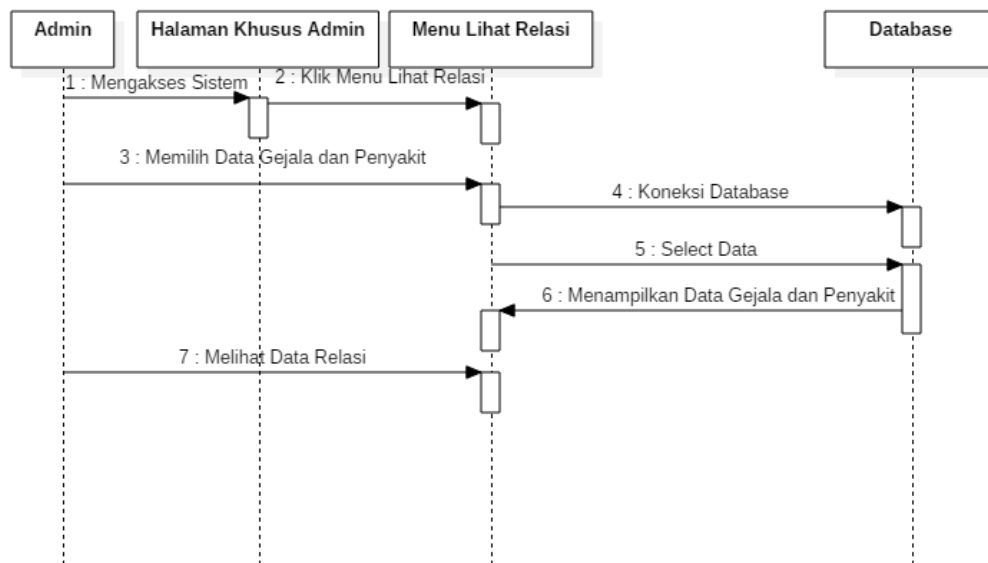


Gambar 3.15 *Sequence Diagram* Mengelola Menu Relasi
Sumber: Data Penelitian (2018)

Admin mengakses sistem dan menampilkan halaman khusus admin. kemudian admin mengklik menu relasi, Sistem akan menampilkan halaman menu relasi dan admin akan memilih data penyakit yang akan direlasikan ke data gejala kemudian sistem terkoneksi dengan *database* dan mengambil data, kemudian menampilkan data gejala dan penyakit. Jika *admin* menekan tombol simpan maka sistem akan menambahkan data ke *database*. Bila *admin* memilih tombol normalkan maka data gejala dan penyakit yang direlasikan akan dihapus dari *database*.

e. *Sequence Diagram* Mengelola Menu Lihat Relasi

Sequence diagram mengelola menu lihat relasi merupakan urutan waktu kegiatan admin saat melihat relasi yang telah sesuai dengan aturan rule pada sistem. Berikut ini gambar *sequence diagram* mengelola menu lihat relasi (Gambar 3.16):

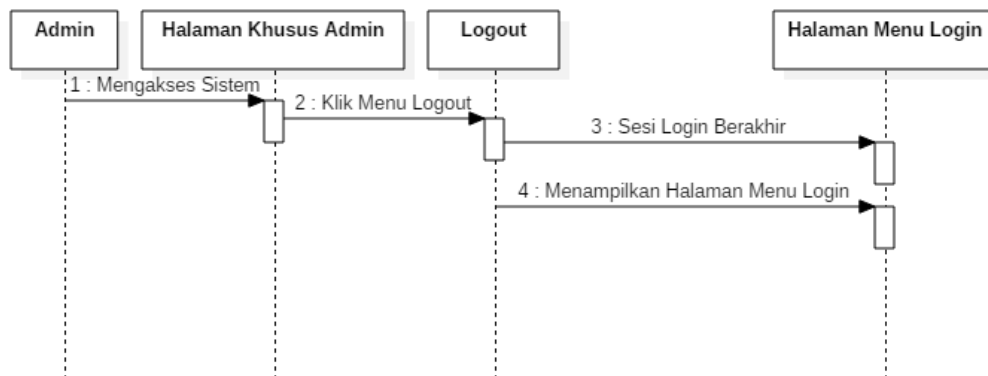


Gambar 3.16 *Sequence diagram* Mengelola Menu Lihat Relasi
Sumber: Data Penelitian (2018)

Admin mengakses sistem dan menampilkan halaman khusus admin. kemudian admin mengklik menu lihat relasi, kemudian sistem menampilkan data relasi yang telah dibuat pada menu relasi. *Admin* memilih data gejala dan penyakit yang akan dilihat relasinya. Sistem akan terkoneksi ke *database* dan sistem akan menampilkan data gejala dan penyakit yang telah direlasikan.

f. *Sequence diagram* logout admin

Sequence diagram logout admin merupakan urutan waktu kegiatan admin saat keluar dari sistem. Berikut ini gambar *sequence diagram logout* admin (**Gambar 3.17**):

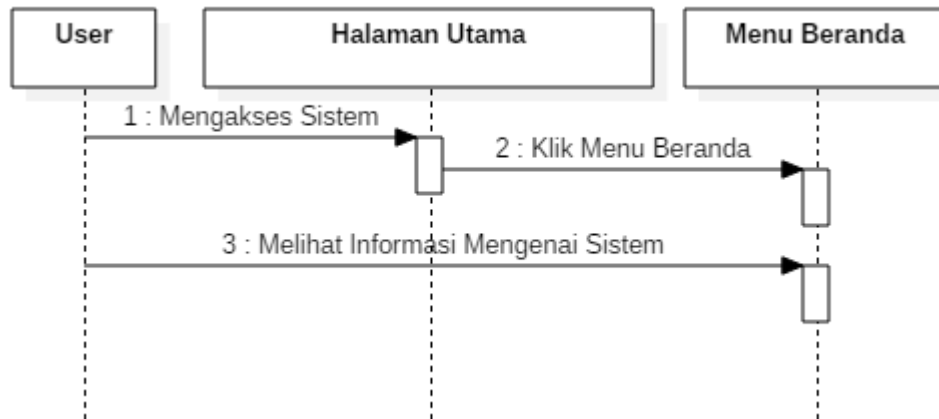


Gambar 3.17 *Sequence diagram logout* admin
Sumber: Data Penelitian (2018)

Admin mengakses halaman khusus admin lalu mengklik menu *logout* dan sesi *Login* admin pada sistem pun berakhir kemudian akan menampilkan halaman menu *Login*.

g. *Sequence diagram* Melihat Menu Beranda

Sequence diagram melihat menu beranda merupakan urutan waktu kegiatan pengguna (*user*) saat mengakses sistem dan melihat informasi mengenai sistem pakar. Berikut ini gambar *sequence diagram* melihat menu beranda (**Gambar 3.18**):



Gambar 3.18 *Sequence diagram* Melihat Menu Beranda

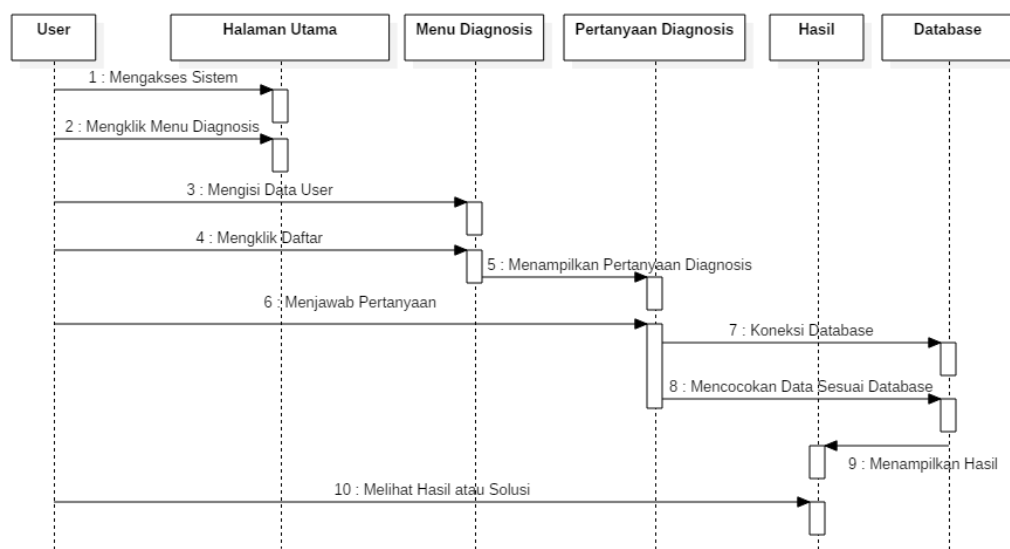
Sumber: Data Penelitian (2018)

User mengakses sistem, sistem akan menampilkan halaman utama sistem.

User memilih menu beranda dan dapat melihat informasi mengenai penyakit sapi.

h. *Sequence Diagram* Menu Diagnosis

Sequence diagram menu diagnosis merupakan urutan waktu kegiatan pengguna (*user*) saat melakukan diagnosis penyakit sapi. Berikut ini gambar *sequence diagram* menu diagnosis (**Gambar 3.19**):



Gambar 3.19 *Sequence Diagram* Menu Diagnosis

Sumber: Data Penelitian (2018)

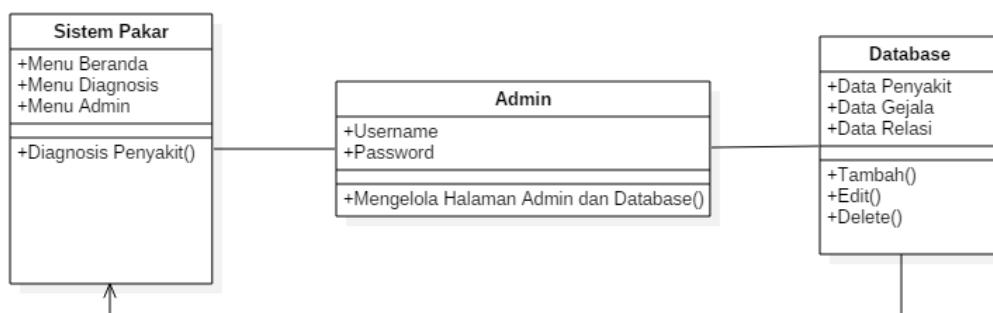
User mengakses sistem dan mengklik menu diagnosis pada halaman utama sistem. Sistem akan menampilkan halaman menu diagnosis. *User* akan mengisi data yang diperlukan untuk buku tamu yang tersimpan pada database panel dengan menekan tombol daftar. Sistem akan menampilkan pertanyaan-pertanyaan diagnosis mengenai penyakit sapi. Sistem akan melakukan koneksi ke database untuk mencocokkan jawaban dari setiap pertanyaan yang dijawab oleh *User*, kemudian sistem akan menampilkan hasil diagnosis yaitu berupa solusi yang bisa dilihat oleh *User*.

3.4.2.4. Class Diagram

Dalam penelitian ini hanya dibuat 2 macam class diagram yaitu class diagram user dan class diagram admin. Berikut ini adalah gambar-gambar class diagram yang digunakan dalam sistem pakar pada penelitian ini:

a. Class Diagram Admin

Class diagram admin merupakan urutan kegiatan admin saat mengakses sistem pakar diagnosis penyakit sapi. Berikut ini gambar *class diagram admin*.

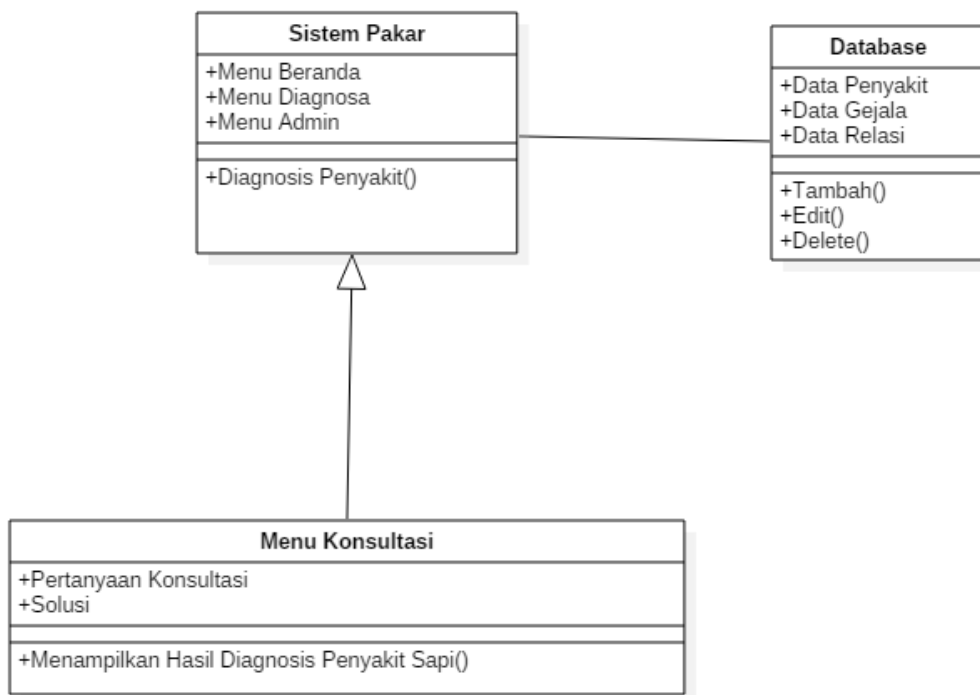


Gambar 3.20 Class Diagram Admin
Sumber: Data Penelitian (2018)

Struktur *class diagram admin* pada **Gambar 3.20** Diatas menggambarkan bahwa pada sistem pakar memiliki atribut berupa menu beranda, menu diagnosis, dan menu *admin* yang memiliki fungsi yaitu mendiagnosis penyakit pada hewan sapi. Sistem pakar terhubung dengan *database* yang memiliki atribut data penyakit, data gejala, data relasi serta memiliki fungsi tambah, *edit* dan *delete*, dan melakukan *Login* terlebih dahulu pada sistem.

b. *Class Diagram User*

Class diagram user merupakan urutan kegiatan user saat mengakses sistem pakar diagnosis penyakit sapi. Berikut ini gambar *class diagram user*.



Gambar 3.21 *Class Diagram User*
Sumber: Data Penelitian (2018)

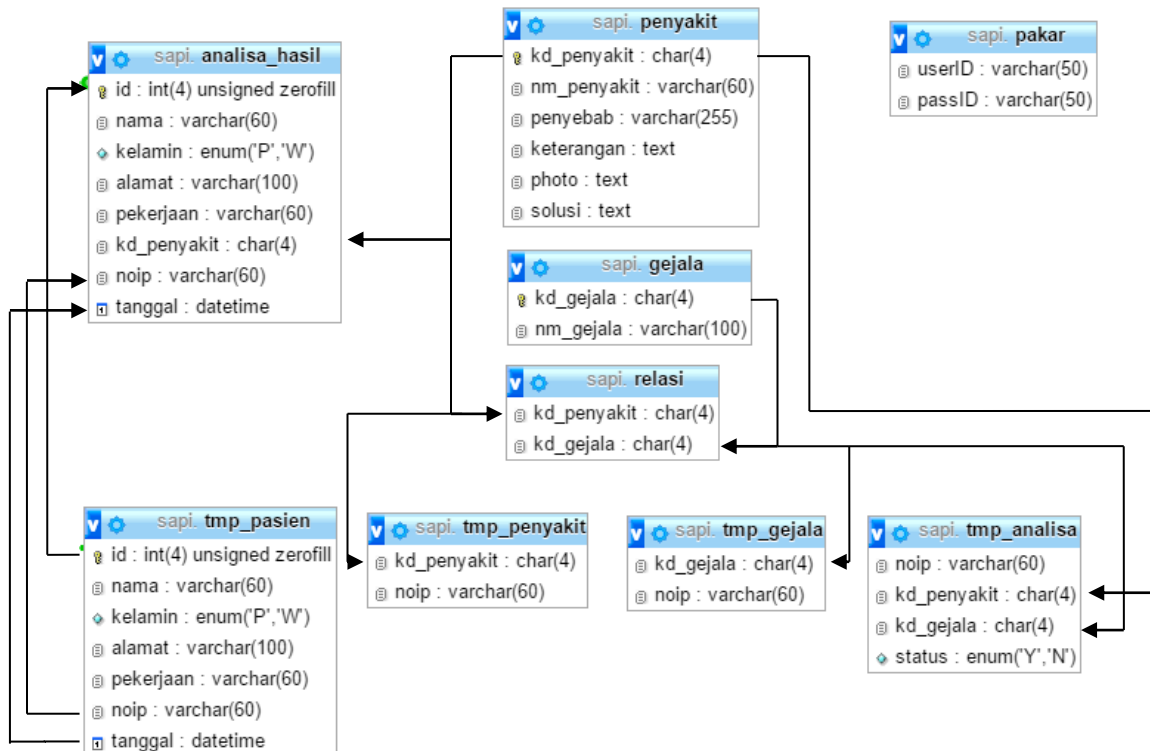
Struktur *class diagram user* pada **Gambar 3.21** diatas menggambarkan bahwa pada sistem pakar memiliki atribut berupa menu beranda, menu diagnosis,

dan menu *admin* yang memiliki fungsi yaitu mendiagnosis penyakit pada hewan sapi. Sistem pakar terhubung dengan *database* yang memiliki atribut data penyakit, data gejala, data relasi serta memiliki fungsi tambah, *edit* dan *delete*.

Pada menu diagnosis yang dapat diakses user memiliki pertanyaan-pertanyaan dan solusi untuk diagnosa pada hewan sapi.

3.4.3. Desain Database

PDM (Physical Data Model) adalah model yang menggunakan sejumlah table untuk menggambarkan data serta hubungan antar data-data tersebut. *PDM* merupakan konsep yang menerangkan detail bagaimana data disimpan di dalam *database*. (A.S. & Shalahuddin, 2011) Berikut ini adalah konsep database yang dibuat dalam penelitian.



Gambar 3.22 Desain Physical Data Model
Sumber: Data Penelitian (2018)

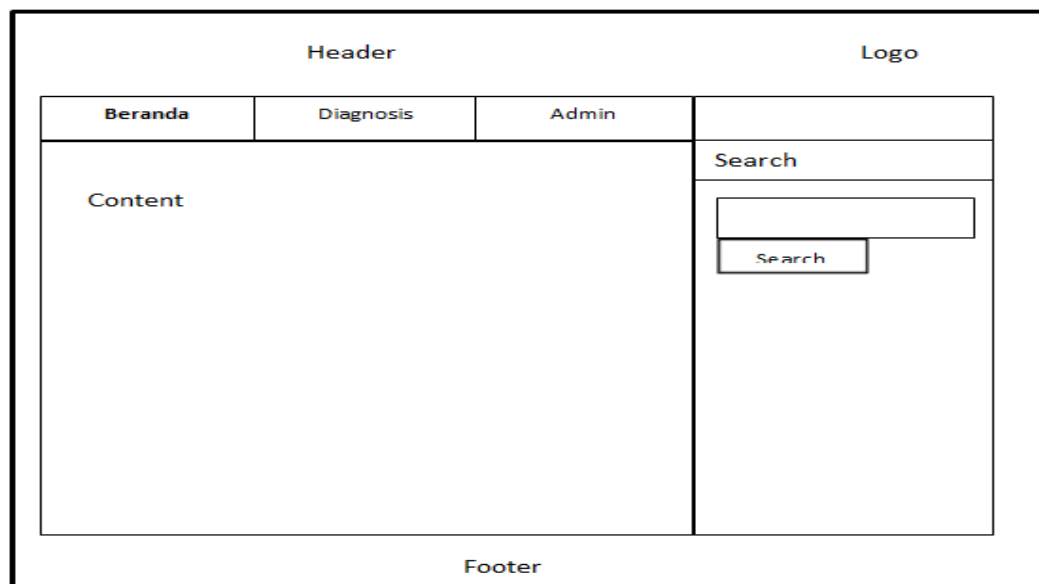
Pada **Gambar 3.22** Diatas, terdapat 5 tabel *database* yang terdiri dari: tabel analisa hasil menyimpan identitas *user*, tabel penyakit menyimpan jenis penyakit dan kode penyakit, tabel gejala menyimpan jenis gejala dan kode gejala, tabel relasi menyimpan kode penyakit dan kode gajala, tabel pakar menyimpan nama pakar dan id pakar, serta memiliki 4 tabel temporary yaitu tabel temp pasien, tabel temp penyakit, tabel temp gejala dan tabel temp analisa.

3.4.4. Desain Antarmuka

Berikut ini adalah desain antarmuka yang akan dibuat pada aplikasi sistem pakar mendeteksi penyakit sapi:

1. Halaman Beranda

Halaman beranda menampilkan informasi tentang aplikasi sistem pakar mendeteksi penyakit sapi.



Gambar 3.23 Halaman Beranda
Sumber: Data Penelitian (2018)

2. Halaman Diagnosis Daftar

Halaman Diagnosis Daftar difungsikan untuk user memasukkan data pribadi awal penggunaan sistem dan kemudian akan diarahkan ke pertanyaan diagnosis penyakit sapi.

Header			Logo
Beranda	Diagnosis	Admin	
MASUKAN DATA Nama <input type="text"/> Kelamin <input type="radio"/> Pria <input type="radio"/> Wanita Alamat <input type="text"/> Jenis Sapi <input type="text"/> <input type="button" value="Daftar"/>			Search <input type="text"/> <input type="button" value="Search"/>
Footer			

Gambar 3.24 Halaman Diagnosis
Sumber: Data Penelitian (2018)

3. Halaman Diagnosis Pertanyaan

Halaman Diagnosis Pertanyaan berfungsi untuk user melakukan diagnosis terhadap penyakit yang tertampak pada sapi. User akan menjawab pertanyaan-pertanyaan yang ditampilkan oleh sistem

Header			Logo
Beranda	Diagnosis	Admin	
<p>JAWABLAH PERTANYAAN BERIKUT :</p> <p>Pertanyaan ?</p> <p><input type="radio"/> Benar (YA) <input type="radio"/> Salah (TIDAK)</p> <p><input type="button" value="Jawab"/></p>			<p>Search</p> <p><input type="text"/></p> <p><input type="button" value="Search"/></p>
Footer			

Gambar 3.25 Halaman Diagnosis Pertanyaan
Sumber: Data Penelitian (2018)

4. Halaman Diagnosis Solusi

. Halaman Diagnosis Solusi ini menampilkan solusi dari setiap pertanyaan yang telah user jawab. Solusi tersebut berasal dari database sistem.

Header			Logo
Beranda	Diagnosis	Admin	
<p>Solusi</p> <p>Solusinya adalah :</p> <p><input type="button" value="OK"/> <input type="button" value="CANCEL"/></p>			
Footer			

Gambar 3.26 Halaman Diagnosis Solusi
Sumber: Data Penelitian (2018)

5. Halaman Admin

Halaman Admin berfungsi untuk menambahkan data gejala penyakit dan solusi penyakit pada sapi Admin akan melakukan pengeditan data sesuai data yang didapat dari Pakar.

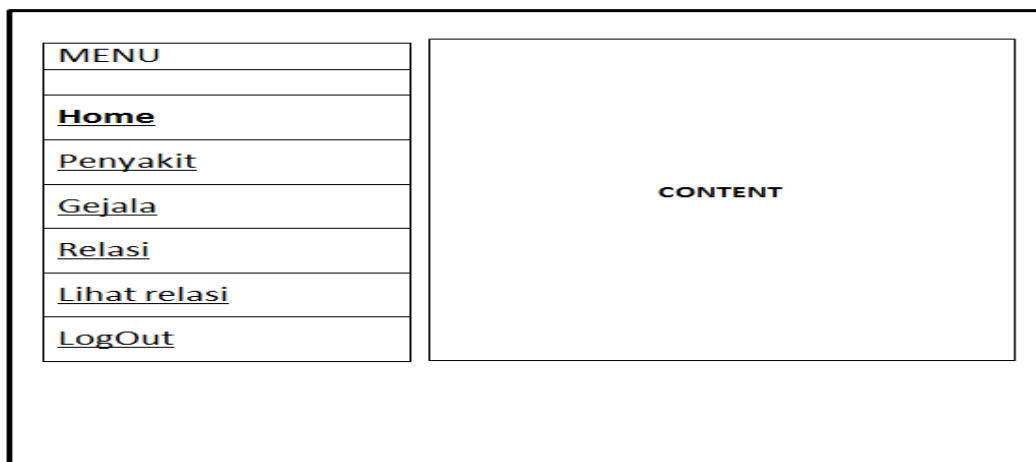


The image shows a sign-in form for an admin user. It consists of a central rectangular box. In the top right corner of this box, the text "SIGN IN" is displayed. Below this text are two stacked input fields: the top one is labeled "Username" and the bottom one is labeled "Password". At the bottom center of the box is a blue button with the text "SIGN IN" in white.

Gambar 3.27 Halaman Admin
Sumber: Data Penelitian (2018)

6. Halaman Home Admin

Halaman Home Admin adalah halaman yang menampilkan keterangan mengenai halaman admin tersebut.



The image shows the home page of an admin interface. It is divided into two main sections. On the left is a vertical menu table with the following items: "MENU", "Home", "Penyakit", "Gejala", "Relasi", "Lihat relasi", and "LogOut". On the right is a large, empty rectangular area labeled "CONTENT".

Gambar 3.28 Halaman *Home* admin
Sumber: Data Penelitian (2018)

7. Halaman Menu Penyakit

Halaman Menu Penyakit berfungsi untuk admin menambahkan, mengedit dan menghapus data penyakit pada sapi.

MENU	DAFTAR SEMUA PENYAKIT			
Home	ID	Nama Penyakit	Pilihan	
Penyakit	P001	Penyakit	Edit	Delete
Gejala			Tambah	
Relasi				
Lihat relasi				
LogOut				

Gambar 3.29 Halaman Menu Penyakit
Sumber: Data Penelitian (2018)

8. Halaman Tambah Penyakit

Halaman tambah penyakit ini berfungsi untuk admin menambahkan mengedit dan menghapus data penyakit pada sapi.

MENU	ENTRY DATA	
Home	ID :	<input type="text" value="P001"/>
Penyakit	Penyakit :	<input type="text"/>
Gejala	Penyebab :	<input type="text"/>
Relasi	Keterangan :	<input type="text"/>
Lihat relasi	Solusi :	<input type="text"/>
LogOut		
		<input type="button" value="Simpan"/> <input type="button" value="Batal"/>

Gambar 3.30 Halaman Tambah Penyakit
Sumber: Data Penelitian (2018)

9. Halaman Menu Gejala

Halaman Menu Gejala ini berfungsi untuk admin menambahkan mengedit dan menghapus data gejala penyakit pada sapi.

MENU	DAFTAR SEMUA GEJALA			
Home	ID	Nama Gejala	Pilihan	
Penyakit	G001	Gejala	Edit	Delete
Gejala			Tambah	
Relasi				
Lihat relasi				
LogOut				

Gambar 3.31 Halaman Menu Gejala
Sumber: Data Penelitian (2018)

10. Halaman Tambah Gejala

Halaman tambah gejala ini berfungsi untuk admin menambahkan mengedit dan menghapus data gejala penyakit pada sapi.

MENU	MASUKAN DATA GEJALA	
Home	Kode :	<input type="text" value="P001"/>
Penyakit	Gejala :	<input type="text"/>
Gejala		
Relasi		
Lihat relasi		
LogOut		
		<input type="button" value="Simpan"/> <input type="button" value="Batal"/>

Gambar 3.32 Halaman Tambah Gejala
Sumber: Data Penelitian (2018)

11. Halaman Menu Relasi

Halaman relasi adalah halaman untuk admin membuat relasi antar data penyakit dengan data gejala sesuai dengan tabel keputusan yang telah dibuat.

<table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td style="text-align: center;">MENU</td></tr> <tr><td> </td></tr> <tr><td>Home</td></tr> <tr><td>Penyakit</td></tr> <tr><td>Gejala</td></tr> <tr><td>Relasi</td></tr> <tr><td>Lihat relasi</td></tr> <tr><td>LogOut</td></tr> </table>	MENU		Home	Penyakit	Gejala	Relasi	Lihat relasi	LogOut	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td style="text-align: center;">INPUT RELASI</td></tr> <tr><td style="text-align: center;">Data Penyakit</td></tr> <tr><td style="text-align: center;">Data Gejala</td></tr> <tr> <td style="text-align: center;">Simpan</td> <td style="text-align: center;">Normalkan</td> </tr> </table>	INPUT RELASI	Data Penyakit	Data Gejala	Simpan	Normalkan
MENU														
Home														
Penyakit														
Gejala														
Relasi														
Lihat relasi														
LogOut														
INPUT RELASI														
Data Penyakit														
Data Gejala														
Simpan	Normalkan													

Gambar 3.33 Halaman Menu Relasi

Sumber: Data Penelitian (2018)

12. Halaman Menu Lihat Relasi

Halaman menu lihat relasi ini adalah halaman yang menampilkan relasi yang telah dibuat antara data penyakit dan data gejala penyakit.

<table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td style="text-align: center;">MENU</td></tr> <tr><td> </td></tr> <tr><td>Home</td></tr> <tr><td>Penyakit</td></tr> <tr><td>Gejala</td></tr> <tr><td>Relasi</td></tr> <tr><td>Lihat relasi</td></tr> <tr><td>LogOut</td></tr> </table>	MENU		Home	Penyakit	Gejala	Relasi	Lihat relasi	LogOut	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td colspan="3" style="text-align: center;">TAMPILKAN GEJALA PER PENYAKIT</td></tr> <tr> <td style="width: 20%;">Penyakit</td> <td colspan="2">List Daftar Penyakit</td> </tr> <tr> <td> </td> <td colspan="2" style="text-align: center;"><input type="button" value="Tampil"/></td> </tr> <tr><td colspan="3"> </td></tr> <tr><td colspan="3">NAMA PENYAKIT :</td></tr> <tr><td colspan="3"> </td></tr> <tr><td colspan="3">DAFTAR GEJALA</td></tr> <tr> <td>Penyakit</td> <td>Gejala</td> <td>Nama Gejala</td> </tr> <tr> <td>P001</td> <td>G001</td> <td> </td> </tr> <tr> <td> </td> <td> </td> <td> </td> </tr> <tr> <td> </td> <td> </td> <td> </td> </tr> </table>	TAMPILKAN GEJALA PER PENYAKIT			Penyakit	List Daftar Penyakit			<input type="button" value="Tampil"/>					NAMA PENYAKIT :						DAFTAR GEJALA			Penyakit	Gejala	Nama Gejala	P001	G001							
MENU																																										
Home																																										
Penyakit																																										
Gejala																																										
Relasi																																										
Lihat relasi																																										
LogOut																																										
TAMPILKAN GEJALA PER PENYAKIT																																										
Penyakit	List Daftar Penyakit																																									
	<input type="button" value="Tampil"/>																																									
NAMA PENYAKIT :																																										
DAFTAR GEJALA																																										
Penyakit	Gejala	Nama Gejala																																								
P001	G001																																									

Gambar 3.34 Rancangan Halaman Menu Konsultasi

Sumber: Data Penelitian (2018)

3.5. Lokasi dan Jadwal Penelitian

3.5.1. Lokasi Penelitian

Penelitian ini dilakukan di Peternakan Pak Ali, Kampung Tembesi Bengkel RT.004 RW.001 Kelurahan Kibing, Batu Aji, Kota Batam-Kepulauan Riau

Alasan peneliti memilih ini sebagai lokasi penelitian adalah:

1. Ketersediaan data untuk melakukan penelitian.
2. Mudah mendapatkan data yang dibutuhkan.
3. Lokasi yang mudah ditempuh.
4. Efisiensi biaya dan waktu.

3.5.2. Jadwal Penelitian

Jadwal penelitian perlu dibuat untuk menggambarkan kapan dan berapa lama waktu yang diperlukan untuk melakukan setiap langkah dalam penelitian. Selain itu, jadwal penelitian juga merupakan target (*deadline*) bagi peneliti yang bersangkutan untuk dapat melaksanakan dan menyelesaikan penelitian. Berikut ini adalah tabel jadwal kegiatan yang dilakukan selama penelitian berlangsung:

Tabel 3.6 Tabel Jadwal Penelitian

No	Kegiatan	Jadwal																					
		September 2017				Oktober 2017				November 2017				Desember 2017				Januari 2018				Februari 2018	
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2
1	Pemilihan Judul																						
2	Pengajuan Judul																						
3	Pengumpulan Data																						
4	Penyusunan Bab I																						
5	Penyusunan Bab II																						
6	Penyusunan Bab III																						
7	Penyusunan Bab IV																						
8	Penyusunan Bab V, Daftar Pustaka, Lampiran																						

Sumber: Data Penelitian (2018)