

SKRIPSI

**SISTEM PAKAR DIAGNOSA PENYAKIT PADA SAPI
MENGUNAKAN METODE *RATIONAL UNIFIED
PROCESS***

*Sebagai salah satu syarat untuk menyelesaikan Program Studi Sarjana Terapan
Rekayasa Perangkat Lunak Jurusan Teknik Informatika*



Oleh:

DIMAS MALIK SURYANDA

63042011228

**PROGRAM STUDI REKAYASA PERANGKAT LUNAK
JURUSAN TEKNIK INFORMATIKA
POLITEKNIK NEGERI BENGKALIS**

2024

HALAMAN PENGESAHAN

SKRIPSI

SISTEM PAKAR DIAGNOSA PENYAKIT PADA SAPI MENGGUNAKAN METODE *RATIONAL UNIFIED PROCESS*

DIMAS MALIK SURYANDA


6304201228

Telah diujikan dan dinyatakan lulus ujian skripsi pada tanggal 18 Januari 2024 oleh tim penguji
Program Studi Sarjana Terapan Rekayasa Perangkat Lunak


Pembimbing Utama

Bengkalis, 18 Januari 2024
Anggota Tim Penguji

Jaroji, M.Kom.
NIP. 198611072015041002


Fajar Rathawati, M.Cs
NIP.198312122019032011


Eva Yumami, S.Kom, M.T
NIP. 198904182022032008


Lidya Wati, M. Kom
NIP. 198908222014042001


Mengetahui,
Ketua Jurusan Teknik Informatika






Kasmawi, M.Kom
NIP. 197706072014041001

LEMBAR PENGESAHAN

Kami dengan sebenarnya menyatakan bahwa, kami telah membaca keseluruhan dari Laporan Skripsi ini, dan kami berpendapat bahwa Laporan Skripsi ini layak dan memenuhi syarat untuk memperoleh gelar sarjana.

Tanda Tangan : 
Nama Penguji : Fajar Ratnawati, M.Cs
Tanggal Pengujian : 18 januari 2024

Tanda Tangan : 
Nama Penguji : Eva Yumami, S.Kom,M.T
Tanggal Pengujian : 18 januari 2024

Tanda Tangan : 
Nama Penguji : Lidya Wati, M.Kom
Tanggal Pengujian : 18 januari 2024

PERNYATAAN KEASLIAN SKRIPSI

Saya menyatakan dengan sesungguhnya bahwa Laporan Skripsi ini adalah asli hasil karya saya dan tidak terdapat karya yang pernah dilakukan untuk memperoleh gelar kesarjanaan di perguruan tinggi, dan sepanjang pengetahuan saya juga tidak terdapat karya atau pendapat yang pernah ditulis atau dipublikasikan oleh orang lain, kecuali yang secara tertulis disebutkan sumbernya dalam naskah dan dalam daftar pustaka.

Bengkalis, 18 Januari 2024



Dimas Malik Suryanda
6304201228

LEMBAR PERSEMBAHAN



Alhamdulillahirobbi'alamin, dengan rahmat dan ridha Allah, amanah ini telah selesai, tetapi ini bukan akhir dari perjalanan, melainkan awal perjuangan.

Skripsi ini penulis persembahkan untuk :

Ayah dan Ibu tercinta

Adek Tercinta

Ayah Surbakti dan Ibu Ana Supriatin

Adek Ahmad Hibrizi Fitriasyah

Azmi Kholis

Dan sahabat saya Hairul Fadli

Yendri Pranata

Fadlin Azhary

Berbagai ungkapan syukur atas kasih sayang dan bimbingan yang telah diberikan oleh bapak dan ibu, serta nasehat dan dorongan untuk menjadi orang yang berguna bagi orang lain. Pencapaian ini adalah wujud dari kesadaran penulis untuk berterima kasih dan berbakti kepada kedua orang tua.

Untuk adek adek tercinta terima kasih atas cintanya dan bantuannya. Semoga perbuatan baikmu mendapat balasan kebaikan yang banyak dan berlimpah, serta selalu dalam perlindungan Allah SWT.

Sahabat-sahabat seperjuangan, terima kasih atas semangatnya, bantuannya, dan dukungannya. Penulis berharap pertemanan kita menjadi persaudaraan abadi.

Dan

Almamater Tercinta Politeknik Negeri Bengkalis

SISTEM PAKAR DIAGNOSA PENYAKIT PADA SAPI MENGUNAKAN METODE *RATIONAL UNIFIED PROCESS*

Nama Mahasiswa : Dimas Malik Suryanda
NIM : 6304201228
Dosen Pembimbing : Jaroji, M. Kom

Abstrak

Sapi merupakan hewan ternak yang memiliki potensi ekonomi. Di samping potensi ekonomi yang besar, terdapat risiko yang cukup besar yaitu penyakit pada sapi. Penyakit pada hewan ternak sapi dapat menular dengan cepat, dan dapat berakibat fatal, yaitu kematian. Penyakit tersebut dapat timbul disebabkan oleh bakteri, virus, jamur, dan parasit. Untuk mencegah agar penyakit sapi tidak menular, maka peternak sapi harus mengetahui terlebih dahulu penyakit pada hewan ternak sapi, sehingga dapat dilakukan pencegahan dan pengobatan terhadap hewan ternak sapi sedini mungkin. Dengan menggunakan Metode *Rational Unified Process* (RUP) membangun sistem pakar diagnosa penyakit pada hewan sapi untuk mendukung proses tersebut sistem pakar dibangun menggunakan *Algoritma Random Forest*, di mana penerapan *Algoritma Random Forest* ini berbasis *website* sehingga bisa memberikan hasil diagnosa yang cepat dan praktis dan memberikan informasi yang sesuai dengan jenis penyakit.

Kata Kunci: Sistem Pakar, Penyakit, *Website*, Metode *Rational Unified Process*

EXPERT SYSTEM FOR DIAGNOSING DISEASES IN CATTLE USING THE RATIONAL UNIFIED PROCESS METHOD

Student's Name : Dimas Malik Suryanda
NIM : 6304201228
Supervisor : Jaroji, M. Kom

Abstract

Cattle are livestock that have economic potential. Alongside their significant economic potential, there is a considerable risk, namely diseases in cattle. Diseases in cattle can spread rapidly and can be fatal, leading to death. These diseases can be caused by bacteria, viruses, fungi, and parasites. To prevent the spread of diseases in cattle, cattle farmers must first be aware of the diseases in their livestock so that prevention and treatment can be carried out as early as possible. Using the Rational Unified Process (RUP), a expert system for diagnosing diseases in cattle is built to support this process. The expert system is constructed using the Random Forest Algorithm, with its implementation based on a website to provide quick and practical diagnostic results, offering information relevant to the type of disease.

Keywords : *Expert System, Disease, Website, Rational Unified Process Method*

KATA PENGANTAR

Segala Puji dan syukur penulis panjatkan ke hadirat Allah SWT yang telah melimpahkan segala rahmatnya sehingga penulis dapat menyelesaikan laporan skripsi dengan judul “Sistem Pakar Diagnosa Penyakit Pada Sapi Menggunakan Metode *Rational Unified Process*” guna memenuhi sebagian persyaratan untuk melanjutkan ke tahap pembuatan skripsi.

Penulis menyadari kekurangan serta keterbatasan yang ada. Sehingga dalam menyelesaikan laporan skripsi ini, penulis memperoleh bantuan dari berbagai pihak, dalam kesempatan ini penulis menyampaikan ucapan terima kasih kepada:

1. Bapak Johnny Custer, ST, MT selaku Direktur Politeknik Negeri Bengkalis,
2. Bapak Kasmawi, M. Kom selaku Ketua Jurusan Teknik Informatika Politeknik Negeri Bengkalis,
3. Bapak Fajri profesio putra, M. Cs selaku Ketua Program Studi Rekayasa Perangkat Lunak Politeknik Negeri Bengkalis,
4. Bapak Jaroji, M. Kom selaku dosen pembimbing yang memberikan bimbingan dengan baik hingga skripsi dapat diselesaikan.
5. Ibu Ryci Rahmatil Fiska, M. Kom dan Ibu Eva Yumami, S. Kom., M.T. selaku Koordinator Skripsi Program Studi Sarjana Terapan Rekayasa Perangkat Lunak Politeknik Negeri Bengkalis.
6. Seluruh Dosen Program Studi Sarjana Terapan Rekayasa Perangkat Lunak Politeknik Negeri Bengkalis yang telah membimbing penulis dalam menyelesaikan skripsi ini.
7. Kepada ayah tercinta Surbakti dan ibu tercinta Ana Supriatin serta adik-adikku tercinta Ahmad Hibrizi Fitriasyah dan Azmi Kholis yang telah senantiasa memberikan doa dan dorongan serta bantuan materi maupun non materi.

8. Orang-orang yang saya sayangi dan teman sekelas selalu memberi support, semangat, memberi dukungan, masukan, dan ilmu kepada saya dalam penulisan skripsi.

Penyusun menyadari adanya keterbatasan dalam penyusunan laporan skripsi ini. Besar harapan penyusun akan saran dan kritik yang bersifat membangun. Akhirnya penyusun berharap agar laporan ini dapat bermanfaat bagi penyusun dan bagi pembaca.

Bengkalis, 18 Januari 2024

Penulis

Dimas Malik Suryanda

6304201228

DAFTAR ISI

	Halaman
SKRIPSI.....	i
HALAMAN PENGESAHAN	ii
PERNYATAAN KEASLIAN SKRIPSI.....	iv
LEMBAR PERSEMBAHAN	v
Abstrak.....	vi
Abstract.....	vii
KATA PENGANTAR.....	viii
DAFTAR ISI.....	x
DAFTAR TABEL	xii
DAFTAR GAMBAR.....	xiii
DAFTAR LAMPIRAN	xiv
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan.....	2
1.4 Manfaat	2
BAB II TINJAUAN PUSTAKA.....	4
2.1. Kajian Terdahulu.....	4
2.2. Landasan Teori	5
2.2.1. Sistem Pakar	5
2.2.2. Aplikasi	6
2.2.3. <i>Unified Modeling Language (UML)</i>.....	6
2.2.4. <i>Entity Relationship Diagram (ERD)</i>	8
2.2.5. <i>Website</i>	8
2.2.6. <i>Diagnosis</i>	8
2.2.7. <i>Algoritma</i>	9
2.2.8. <i>Random Forest</i>	10

2.2.9. <i>MySql</i>	10
2.2.10. <i>Rational Unified Process</i>	11
BAB III METODOLOGI PENELITIAN	13
3.1. Data dan Alat Penelitian	13
3.2.1 Alat Penelitian	13
3.2.2 Data Penelitian	13
3.2. Prosedur Penelitian	13
3.2.1 <i>Inception (Permulaan)</i>	14
3.2.2 <i>Elaboration (Perencanaan)</i>	18
3.2.3 <i>Construction (Konstruksi)</i>	23
3.2.4 <i>Transition (Transisi)</i>	25
3.2.5 <i>Algoritma Random Forest</i>	26
BAB IV HASIL DAN PEMBAHASAN	32
4.1. Hasil dan Pembahasan	32
4.1.1 Deskripsi	32
4.1.2 Prosedur Eksperimen	32
BAB V PENUTUP	46
5.1 Kesimpulan	46
5.2 Saran	46
DAFTAR PUSTAKA	47
LAMPIRAN	49

DAFTAR TABEL

	Halaman
Tabel 2.1 Simbol-Simbol <i>Use Case Diagram</i>	7
Tabel 3.1 Wawancara.....	15
Tabel 3.2 <i>Value Stories</i>	17
Tabel 3.3 Sampel Data Set.....	26
Tabel 3.4 <i>Entropy</i> Total	27
Tabel 4.1 Pengujian <i>User</i> 1	41
Tabel 4.2 Pengujian <i>User</i> 2.....	42
Tabel 4.3 Pengujian <i>User</i> 3.....	43
Tabel 4.4 Pengujian <i>User</i> 4.....	43
Tabel 4.5 Pengujian <i>User</i> 5.....	44
Tabel 4.6 Validasi Pakar	45

DAFTAR GAMBAR

	Halaman
Gambar 3.1 Tahapan Pengembangan Sistem.....	14
Gambar 3.2 Sistem yang Sedang Berjalan.....	16
Gambar 3.3 Gambaran Sistem Yang di usulkan	16
Gambar 3.4 <i>Use Case Diagram</i>	17
Gambar 3.5 <i>Activity Diagram</i> Diagnosa Penyakit.....	19
Gambar 3.6 <i>Activity Diagram</i> Konsultasi.....	20
Gambar 3.7 Halaman Utama.....	21
Gambar 3.8 Halaman Form Diagnosa Penyakit.....	21
Gambar 3.9 .Halaman Hasil Diagnosa.....	22
Gambar 3.10 Halaman <i>Dashboard</i>	23
Gambar 3.11 Halaman Diagnosa	23
Gambar 3.12 Halaman Form Diagnosa.....	24
Gambar 3.13 Halaman Hasil Diagnosa.....	25
Gambar 4.1 Dataset.....	33
Gambar 4.2 Dataset setelah proses pengisian data null	34
Gambar 4.3 Pengecekan info dataset	34
Gambar 4.4 Pengecekan data null dari setiap kolom	35
Gambar 4.5 Proses <i>Stemming</i>	36
Gambar 4.6 Proses pembagian variabel dataset.....	36
Gambar 4.7 Proses tokenisasi	37
Gambar 4.8 Proses pembagian dataset.....	37
Gambar 4.9 Proses TF-IDF vectorizer	38
Gambar 4.10 Halaman dashboard.....	39
Gambar 4.11 Halaman diagnosa	40
Gambar 4.12 Halaman form diagnosa	40
Gambar 4.13 Halaman Hasil Diagnosa.....	41

DAFTAR LAMPIRAN

	Halaman
Lampiran 1. Penyakit Hewan Ternak Sapi	49
Lampiran 2. Kode Program <i>Dashboard</i>	53
Lampiran 3. Kode Program Diagnosa.....	56
Lampiran 4. Kode Program Halaman <i>Form</i> Diagnosa	59
Lampiran 5. Kode Program Hasil Diagnosa	61

BAB I

PENDAHULUAN

1.1 Latar Belakang

Menurut Dinas Tanaman Pangan, Hortikultura dan Peternakan Kabupaten Bengkalis Tahun (2020). Pemerintah Kabupaten Bengkalis mempunyai potensi peternakan yang cukup besar dengan produk unggulannya yaitu hewan ternak sapi. produk unggulan peternakan tersebut berkembang dan terfokus dalam kawasan pengembangan pusat produksi. Dari berbagai macam jenis hewan ternak yang banyak dipelihara oleh peternak di pedesaan adalah sapi. Sapi menduduki peringkat pertama sebagai komoditas unggulan, di tahun 2020 jumlah populasi hewan ternak sapi mencapai 3.037.00 ekor, salah satu program pemerintah untuk menyejahterakan rakyat adalah melalui program bantuan ternak sapi.

Sapi merupakan hewan ternak yang banyak memiliki potensi ekonomi. Di samping potensi ekonomi yang besar, terdapat risiko yang cukup besar yaitu penyakit pada sapi. Penyakit pada hewan ternak sapi dapat menular dengan cepat, dan dapat berakibat fatal, yaitu kematian. Penyakit tersebut dapat timbul disebabkan oleh bakteri, 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.

Berdasarkan permasalahan di atas maka diusulkan penelitian untuk membangun Sistem Pakar Diagnosa Penyakit pada Sapi Menggunakan *Rational Unified Process*. Dengan adanya sistem pakar diharapkan dapat membantu penanganan sebagai tindakan awal untuk mengatasi penyakit pada hewan ternak sapi. Untuk mendukung proses tersebut sistem pakar dibangun menggunakan algoritma Random forest, di mana penerapan algoritma ini berbasis *website* sehingga bisa memberikan hasil diagnosa yang cepat dan praktis dan memberikan informasi yang sesuai dengan jenis penyakit.

1.2 Rumusan Masalah

Atas dasar masalah-masalah yang ada, maka rumusan masalah yang dapat diangkat yaitu Bagaimana membuat model diagnosa penyakit pada sapi yang optimal dan menerapkannya pada aplikasi berbasis *website* menggunakan metode *Rational Unified Process*.

1.3 Tujuan

Pelaksanaan penelitian ini adalah merancang dan membangun sistem pakar diagnosa penyakit pada hewan sapi dengan menggunakan Algoritma Random forest dengan pendekatan *machine learning* yang berbasis *website*. Tujuannya sebagai berikut:

1. Membuat model diagnosa penyakit sapi yang optimal,
2. Membuat aplikasi diagnosa penyakit sapi berbasis web.

1.4 Manfaat

Berdasarkan tujuan penelitian di atas, manfaat penelitian yang dapat dihasilkan adalah sebagai berikut:

1. Mempermudah kan peternak untuk mendapatkan informasi mengenai penyakit pada hewan sapi.
2. Mempermudah peternak dalam mengambil tindakan yang tepat.

BAB II

TINJAUAN PUSTAKA

2.1. Kajian Terdahulu

Untuk mendukung penelitian yang dilakukan, penulis mengambil beberapa penelitian yang telah dilakukan sebelumnya sebagai referensi diantaranya sebagai berikut:

Berdasarkan penelitian yang dilakukan oleh Wardani, dkk (2022) dalam penelitiannya membuat aplikasi Sistem Pakar Untuk Diagnosa Penyakit Sapi Menggunakan Metode *Bayes*. Digunakan untuk memberikan tindakan pencegahan secara umum untuk membantu peternak sapi menghindari kerugian ekonomi yang lebih besar dengan mengetahui lebih awal penyakit sapi yang menyerang [1].

Berdasarkan penelitian yang dilakukan oleh Irawan, dkk (2021) dalam penelitiannya membuat aplikasi penerapan *Backward Chaining*, aplikasi tersebut digunakan untuk pencegahan penyakit pada sapi akan tetapi, aplikasi yang dibangun belum membuat sistem diagnosa penyakit sapi berbasis *website* menggunakan *Machine Learning* [2].

Berdasarkan penelitian yang dilakukan oleh Rozaq, dkk (2022) dalam penelitiannya membuat aplikasi Sistem Informasi Diagnosa Penyakit Hewan Ternak Pada Dinas Peternakan dan Kesehatan Hewan. Memiliki kekurangan pada Sistem informasi diagnosa penyakit hewan ternak sering kali mengandalkan pelaporan manual oleh peternak atau Dinas Peternakan. Hal ini dapat memunculkan risiko kesalahan manusia dalam memasukkan data atau kurangnya kesadaran akan pentingnya pelaporan penyakit hewan. Ketergantungan pada pelaporan manual juga dapat menyebabkan keterlambatan dalam mendapatkan data dan mengurangi efisiensi sistem [3].

Berdasarkan penelitian yang dilakukan oleh Anggraini & Afidh (2023) dalam penelitiannya membuat Sistem Pakar Diagnosa Penyakit Sapi Menggunakan Metode CBR dan Algoritma *Similarity Sorgenfrei*. Sistem pakar CBR membutuhkan basis data kasus yang lengkap. Jika basis data kasus tidak cukup luas

atau tidak mencakup berbagai variasi penyakit dan gejala yang mungkin terjadi pada sapi, maka kemampuan sistem untuk memberikan diagnosis yang akurat dan relevan dapat terbatas [4].

Berdasarkan penelitian yang dilakukan oleh Nuzzakir (2020) Penelitian ini membahas tentang Penerapan Metode *Case Based Reasoning* (CBR) untuk Sistem Pakar Diagnosa Penyakit Ternak Sapi. Diawali dengan pengumpulan data dengan melakukan konsultasi kepada pakar yang berada pada Dinas Pertanian Bidang Kesehatan Hewan Kabupaten Pohuwato. Data yang diperoleh berupa data nama penyakit dan data gejala. Data tersebut diolah berdasarkan langkah-langkah perhitungan metode CBR sehingga diperoleh hasil diagnosa dan solusi yang diberikan untuk penanganan penyakit. Peneliti telah melakukan analisa dan membuat *listing* program untuk membangun sebuah sistem yang nantinya akan digunakan oleh para peternak [5].

2.2. Landasan Teori

2.2.1. Sistem Pakar

Sistem pakar adalah suatu bentuk kecerdasan buatan yang dikembangkan untuk memodelkan pengetahuan dan keterampilan seorang ahli dalam suatu bidang tertentu. Teori dasar tentang sistem pakar didasarkan pada prinsip bahwa seorang ahli manusia memiliki pengetahuan yang berharga dan pengalaman yang luas dalam suatu domain tertentu. Dalam sistem pakar, pengetahuan dan keterampilan tersebut ditransfer ke dalam sebuah sistem komputer sehingga sistem tersebut dapat memberikan solusi atau rekomendasi yang serupa dengan seorang ahli manusia.

Menurut Nilsson (1998), sistem pakar beroperasi berdasarkan dua komponen utama, yaitu basis pengetahuan dan mesin inferensi. Ahli lainnya Davis (1983), mengemukakan bahwa sistem pakar harus memenuhi empat kriteria utama, yaitu memiliki basis pengetahuan yang kaya dan akurat, mampu menjawab pertanyaan dan memberikan penjelasan, dapat memecahkan masalah secara efektif, dan mampu belajar dari pengalaman. Dengan demikian, sistem pakar diharapkan dapat memberikan solusi yang handal dan memiliki kemampuan untuk beradaptasi dengan perubahan yang terjadi dalam domain yang ditangani.

Pengembangan sistem pakar juga melibatkan langkah-langkah seperti identifikasi dan akuisisi pengetahuan dari ahli manusia, representasi pengetahuan yang sesuai, serta pengujian dan evaluasi sistem yang dikembangkan. Ahli lainnya, Buchanan dan Shortliffe (1984), mengemukakan bahwa sistem pakar dapat digunakan dalam berbagai bidang, termasuk bidang kedokteran, keuangan, manufaktur, dan sebagainya. Keunggulan dari sistem pakar terletak pada kemampuannya untuk memberikan solusi yang cepat dan konsisten, serta dapat digunakan sebagai alat bantu dalam pengambilan keputusan.

2.2.2. Aplikasi




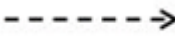
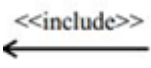
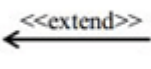
Menurut Kinaswara, dkk (2019) aplikasi adalah perangkat lunak yang diciptakan dengan berbagai komponen atribut yang sesuai dengan pengguna agar dapat membantu pengguna dalam mengolah setiap data agar menghasilkan *input* dan *output*. Aplikasi mempunyai arti yaitu pemecahan masalah yang menggunakan salah satu teknik pemrosesan data aplikasi yang biasanya berpacu pada sebuah komputasi yang diinginkan atau diharapkan maupun pemrosesan data yang diharapkan [6].

2.2.3. Unified Modeling Language (UML)

Menurut Nistrina & Sahidah (2022) UML memiliki fungsi yang membantu untuk pendeskripsian dan desain sistem perangkat lunak, jenis-jenis UML adalah *use case diagram* yang berfungsi untuk mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem yang akan dibangun, Menurut James Rumbaugh dalam bukunya yang berjudul "*The Unified Modeling Language Reference Manual*", *Use case* adalah unit koheren dari fungsionalitas yang terlihat secara eksternal yang disediakan oleh unit sistem dan diekspresikan oleh urutan pesan yang dipertukarkan oleh unit sistem dan satu atau lebih aktor dari unit sistem. Tujuan dari *use case* adalah untuk mendefinisikan bagian dari perilaku yang koheren tanpa mengungkapkan struktur internal sistem. Dalam model, eksekusi setiap *use case* tidak tergantung pada yang lain, meskipun implementasi kasus penggunaan dapat membuat ketergantungan implisit di antara mereka karena objek

bersama. Setiap *use case* mewakili bagian ortogonal fungsionalitas yang eksekusinya dapat dicampur dengan eksekusi *use case* lainnya [7].

Tabel 2. 1 Simbol-simbol Use Case Diagram

Simbol	Nama	Keterangan
	Aktor	Mewakili peran orang, sistem yang lain, atau alat ketika berkomunikasi dengan <i>use case</i>
	<i>Use case</i>	Abstraksi dan interaksi antara sistem dan aktor
	<i>Assosiation</i>	Abstraksi dari penghubung antara aktor dengan <i>use case</i>
	<i>Generalisasi</i>	Menunjukkan spesialisasi aktor untuk dapat berpartisipasi dengan <i>use case</i>
	<i>Include</i>	Menunjukkan bahwa suatu <i>use case</i> seluruhnya merupakan fungsionalitas dari <i>use case</i> lainnya.
	<i>Extend</i>	Menunjukkan bahwa suatu <i>use case</i> merupakan tambahan fungsional dari <i>use case</i> lainnya jika suatu kondisi terpenuhi.

Class diagram merupakan pembentukan utama dari sistem berorientasi objek yang mempresentasikan suatu class beserta atributnya, *Activity Diagram* menggambarkan *workflow* (aliran kerja), menggambarkan aktivitas sistem bukan apa yang dilakukan oleh aktor. Beberapa jenis UML yaitu:

- a. *Sequence diagram*
- b. *State Machine diagram*
- c. *Communication diagram*
- d. *Deployment diagram*
- e. *Component diagram*
- f. *Object diagram*
- g. *Composite structure diagram*
- h. *Interaction overview diagram*
- i. *Package diagram*
- j. *Diagram training*

2.2.4. Entity Relationship Diagram (ERD)

Menurut Afiifah, dkk (2022) ERD merupakan pemodelan data atau sistem dalam *database*, ERD digunakan untuk memodelkan struktur dan hubungan antar data yang relatif komplit [8]. ERD memiliki tiga elemen dasar yaitu entitas, atribut, dan relasi.

- a. Entitas yaitu berupa objek, contoh manusia, tempat, benda, atau kondisi mengenai data yang dibutuhkan,
- b. Atribut merupakan informasi yang terdapat pada entitas,
- c. Relasi merupakan hubungan dua atau lebih entitas. Macam-macam relasi (*one to one, one two many, many to many*).

2.2.5. Website

Menurut Manullang, dkk (2021) *Website* merupakan sebuah kumpulan halaman-halaman web beserta *file- file* pendukungnya, seperti *file* gambar, video, dan *file* digital lainnya yang disimpan pada sebuah *web server* yang umumnya dapat diakses melalui internet.

Website tersebut dapat dibedakan menjadi 2 yaitu web bersifat statis dan dinamis. Bersifat statis apabila isi informasinya tetap dan isi informasinya hanya dari pemilik *website* sedangkan web yang bersifat dinamis apabila isi informasinya selalu berubah-ubah dan dapat diubah oleh pemilik maupun pengguna *website* [9].

2.2.6. Diagnosis

Menurut Hanif (2022) diagnosis berasal dari bahasa Yunani "*gronis*", 45 yang berarti pengetahuan. Melansir Britannica, diagnosis adalah proses penentuan sifat dari suatu penyakit atau gangguan dan membedakan dengan kondisi lain. Secara tradisional, diagnosis telah didefinisikan sebagai seni mengidentifikasi satu sebagai penyebab paling mungkin dari penyakit seseorang. Diagnosis tidak bisa dilakukan oleh sembarang orang. Diagnosis harus dilakukan oleh dokter dan tenaga kesehatan ahli. Sebuah diagnosis medis merupakan langkah kompleks yang melibatkan riwayat pasien, pemeriksaan dan pengujian [10].

2.2.7. Algoritma

Algoritma menurut Kani (2020) adalah suatu upaya dengan urutan operasi yang disusun secara logis dan sistematis untuk menyelesaikan suatu masalah untuk menghasilkan suatu output tertentu. Perkembangan ilmu pengetahuan dan teknologi telah memberikan manusia kemampuan untuk menghasilkan karya yang semakin canggih dan kompleks. Meskipun komputer umumnya dapat melakukan perhitungan dengan cepat dibandingkan manusia, komputer tidak dapat memecahkan masalah dengan cara yang sama tanpa manusia mengajari mereka langkah-langkah yang telah ditentukan sebelumnya (algoritma). Algoritma dapat digunakan tidak hanya untuk menyelesaikan masalah komputer, tetapi juga untuk menyelesaikan masalah sehari-hari yang membutuhkan serangkaian proses atau langkah proses [11].

Algoritma merupakan langkah-langkah sistematis yang digunakan untuk memecahkan masalah atau menyelesaikan tugas tertentu. Landasan teori dalam algoritma meliputi konsep dasar seperti notasi, variabel, pengulangan, dan pengambilan keputusan. Notasi dalam algoritma digunakan untuk menggambarkan langkah-langkah secara jelas dan terstruktur, sering kali menggunakan *pseudo code* atau bahasa pemrograman yang terstandarisasi. Variabel digunakan untuk menyimpan dan memanipulasi data dalam algoritma, sedangkan pengulangan dan pengambilan keputusan memungkinkan algoritma untuk menjalankan serangkaian instruksi berdasarkan kondisi yang ditentukan.

Algoritma juga mencakup struktur data, yaitu cara penyimpanan dan pengorganisasian data dalam komputer. Struktur data merupakan komponen penting dalam algoritma karena mempengaruhi kecepatan dan efisiensi dalam pemrosesan data. Beberapa struktur data umum meliputi *array*, *linked list*, *stack*, *queue*, *tree*, dan *graph*. Setiap struktur data memiliki kelebihan dan kelemahan masing-masing tergantung pada jenis masalah yang ingin diselesaikan. Dalam pemilihan algoritma, pemahaman yang baik tentang struktur data akan membantu dalam memilih dan mengimplementasikan algoritma yang sesuai.

2.2.8. Random Forest

Menurut Breiman (2001) dalam bukunya yang berjudul “*Random Forests*”, *Random Forest* adalah sebuah algoritma yang menggabungkan beberapa pohon keputusan yang dibuat secara acak untuk menghasilkan prediksi yang lebih akurat. *Random Forest* bekerja dalam dua fase. Fase pertama yaitu menggabungkan sejumlah N *decision tree* untuk membuat *Random Forest*. Kemudian fase kedua adalah membuat prediksi untuk setiap *tree* yang dibuat pada fase pertama [12]. Cara kerja algoritma *Random Forest* dapat dijabarkan dalam langkah-langkah berikut:

- a. Algoritma memilih sampel acak dari *database* yang disediakan.
- b. Membuat *decision tree* untuk setiap sampel yang dipilih. Kemudian akan didapatkan hasil prediksi dari setiap *decision tree* yang telah dibuat.
- c. Dilakukan proses *voting* untuk setiap hasil prediksi. Untuk masalah klasifikasi menggunakan modus (nilai yang paling sering muncul), sedangkan untuk masalah regresi akan menggunakan mean (nilai rata-rata).
- d. Algoritma akan memilih hasil prediksi yang paling banyak dipilih (*vote* terbanyak) sebagai prediksi akhir.

2.2.9. MySQL

MySQL merupakan sebuah perangkat lunak atau software sistem manajemen basis data SQL atau DBMS *Multithread* dan multi *user*. *MySQL* sebenarnya merupakan turunan dari salah satu konsep utama dalam database untuk pemilihan atau seleksi dan pemasukan data yang memungkinkan pengoperasian data dikerjakan secara mudah dan otomatis.

MySQL merupakan sistem manajemen *database* yang bersifat relational. Artinya data yang dikelola dalam *database* yang akan diletakkan pada beberapa tabel yang terpisah sehingga manipulasi data akan jauh lebih cepat. *SQL* juga dapat diartikan sebagai antarmuka standar untuk sistem manajemen relasional, termasuk sistem yang beroperasi pada komputer pribadi. *SQL* memungkinkan seorang pengguna untuk mengetahui di mana lokasinya, atau bagaimana informasi tersebut disusun. *SQL* lebih mudah digunakan dibandingkan dengan bahasa pemrograman, tetapi rumit dibandingkan software lembar kerja dan pengolah data. Sebuah pernyataan *SQL* yang sederhana dapat menghasilkan set permintaan untuk

informasi yang tersimpan pada komputer yang berbeda di berbagai lokasi yang tersebar, sehingga membutuhkan waktu dan sumber daya komputasi yang banyak. *SQLite* dapat digunakan untuk investigasi interaktif, atau pembuatan laporan ad hoc atau disisipkan dalam program aplikasi.

SQL juga merupakan bahasa pemrograman yang dirancang khusus untuk mengirimkan suatu perintah query (pengaksesan data berdasarkan pengalamatan tertentu) terhadap sebuah database. Kebanyakan *software database* mengimplementasikan *SQL* secara sedikit berbeda, tapi seluruh *database SQL* mendukung subset standar yang ada [13].

2.2.10. Rational Unified Process

RUP adalah pendekatan pengembangan perangkat lunak yang dilakukan berulang-ulang (*iterative*), fokus pada arsitektur (*architecture-centric*), lebih diarahkan berdasarkan penggunaan kasus (*use case driven*). RUP juga merupakan proses rekayasa perangkat lunak dengan pendefinisian yang baik (*well defined*) dan penstrukturan yang baik (*well structured*). RUP menyediakan pendefinisian struktur yang baik untuk alur hidup proyek perangkat lunak. RUP adalah sebuah produk proses perangkat lunak yang dikembangkan oleh *Rational Software* yang diakuisisi oleh IBM dibulan Februari 2003 [14]. Metode RUP memiliki 4 fase yaitu:

2.2.10.1 Fase *Inception*.

Tahap pertama memodelkan proses bisnis yang dibutuhkan (*business modelling*) dan mendefinisikan kebutuhan akan sistem yang akan dibuat (*requirements*).

2.2.10.2 Fase *Elaboration*

Tahap kedua lebih ditujukan pada perencanaan arsitektur sistem. Tahap ini lebih pada analisis dan desain sistem.

2.2.10.3 Fase *Construction*

Tahap ketiga, dimana kita mengembangkan komponen dan fitur-fitur sistem. Implementasi dan pengujian sistem yang tertuju pada implementasi perangkat lunak pada kode program.

2.2.10.4 Fase *Transition*

Tahap dimana kita *deployment* atau instalasi sistem agar dapat digunakan dan dipahami oleh pengguna. Aktifitas pada tahap ini termasuk pada pelatihan pengguna dan pemeliharaan.

BAB III

METODOLOGI PENELITIAN

3.1. Data dan Alat Penelitian

Pada penelitian ini penulis menggunakan data dan alat penelitian digunakan dalam penelitian ini. Adapun data dan alat penelitian adalah sebagai berikut:

3.2.1 Alat Penelitian

Adapun alat yang digunakan untuk melakukan penelitian ini:

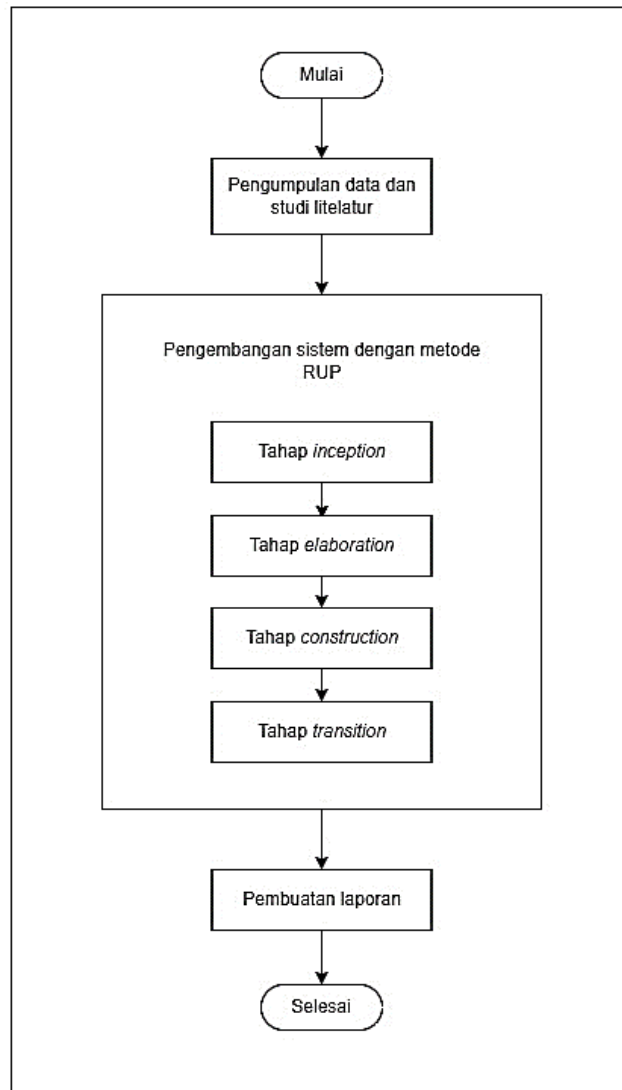
1. Perangkat Keras (*Hardware*)
 - a. Laptop : *Processor Core™ i5-3320M*
 - b. Printer : *Ip2770*
 - c. Memori : *8,00 GB*
 - d. *System Type* : *64-bit Operating System*
2. Perangkat Lunak (*Software*)
 - a. Sistem Operasi : *Windows 10*
 - b. Bahasa pemrograman : *PHP, Python*
 - c. *Text Editor* : *Ms. Word dan Visual Studio Code*
 - d. Pengolahan gambar : *Ms. Visio, Mockup*
 - e. *Server* : *Apache (XAMPP)*

3.2.2 Data Penelitian

Dalam perancangan Sistem Pakar Diagnosa Penyakit Hewan Ternak Sapi ini data yang dibutuhkan adalah data gejala, data penyakit, data *treatment*. Adapun data sampel yang digunakan terlampir pada Tabel 1.

3.2. Prosedur Penelitian

Prosedur penelitian ini ada beberapa tahapan yang akan dilakukan oleh peneliti dengan menggunakan model *Rational Unified Process* (RUP) adalah sebagai berikut:



Gambar 3.1 Tahapan Pengembangan Sistem

3.2.1 *Inception* (Permulaan)

3.2.1.1 Pengumpulan data

Pada tahapan ini peneliti melakukan wawancara untuk mengumpulkan dan menganalisis data dan informasi tentang penyakit pada sapi yang diberikan oleh dokter hewan.

Tabel 3. 1 Wawancara

No.	Pertanyaan	Jawaban
1.	Bagaimana sistem mendeteksi penyakit hewan ternak sapi pada saat ini pak ?	Sistem diagnosa pada saat ini kepada peternak masih dilakukan secara manual, peternak datang ke puskesmas ,pusat kesehatan hewan atau biasanya menghubungi saya , dan masyarakat menunggu hingga saya datang ke lokasi.
2.	Apakah sebelumnya dokter sudah pernah menggunakan aplikasi diagnose hewan?	Sebelumnya saya sudah menggunakan aplikasi ini akan tetapi hanya seputaran dokter dan admin saja yang bisa menggunakan aplikasi ini.
3.	Apakah dengan perancangan sistem ini dapat membantu peternak pak?	Dengan adanya sistem ini semoga membantu permasalahan yang terjadi di lokasi.
4	Apakah ada masukan untuk aplikasi website yang akan saya buat ini pak?	Saya berharap aplikasi sistem pakar dapat menjadi alat bantu yang handal, memberikan panduan yang cepat dan akurat dalam mendeteksi serta mengelola penyakit pada hewan ternak sapi. Kemampuan aplikasi untuk menyediakan informasi terkini akan menjadi nilai tambah yang signifikan

3.2.1.2 Analisis kebutuhan

Pada tahap ini akan dijelaskan tentang rencana hasil penelitian yang dilakukan dan dikembangkan aplikasi yang dibangun akan diberi nama Sistem Pakar Diagnosa Peternakan Pada Sapi Menggunakan Algoritma Random Forest dan metode RUP. Sistem yang sedang berjalan sebelumnya masih kurang lengkap untuk dapat melakukan diagnosis penyakit yang dialami oleh hewan sapi. Sistem yang diusulkan adalah aplikasi *website* yang mempermudah kan bagi peternak untuk lebih awal mengetahui gejala penyakit yang dialami oleh hewan sapi dan juga bisa membantu proses informasi tentang penyakit yang terjadi lebih cepat.

a. Analisis kebutuhan fungsional

Analisa kebutuhan fungsional merupakan merupakan analisis fungsi-fungsi yang dibutuhkan dalam sistem. Fungsi yang dibutuhkan dalam *website* diagnosa penyakit hewan sapi diantaranya:

- 1) *User* dapat masuk ke aplikasi.
- 2) *User* dapat melihat hasil diagnosa.

3) *User* dapat melakukan konsultasi.

b. Analisis kebutuhan non fungsional

1) *Usability* Hal ini yang akan berkaitan dengan kemudahan penggunaan sistem atau perangkat lunak ini oleh *user*,

2) *Portability* Mencakup kemudahan dalam pengaksesan sistem aplikasi ini, khususnya terkait dengan faktor waktu dan lokasi saat pengaksesan aplikasi, yang digunakan untuk mengakses, baik dalam perangkat lunak, perangkat keras, maupun jaringan,

3) *Reliability sejauh mana sistem dapat diandalkan.*

3.2.1.3 Analisis sistem perancangan

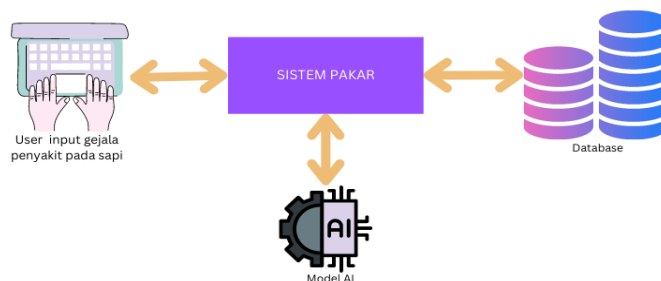
a. Berikut gambaran dari rancangan sistem yang sedang berjalan saat ini:



Gambar 3. 2 Sistem yang Sedang Berjalan

Berdasarkan dari pengumpulan data maka sistem yang berjalan saat ini berupa *user* datang langsung kepada dokter hewan untuk mendapatkan data penyakit.

b. Rancangan sistem yang diusulkan



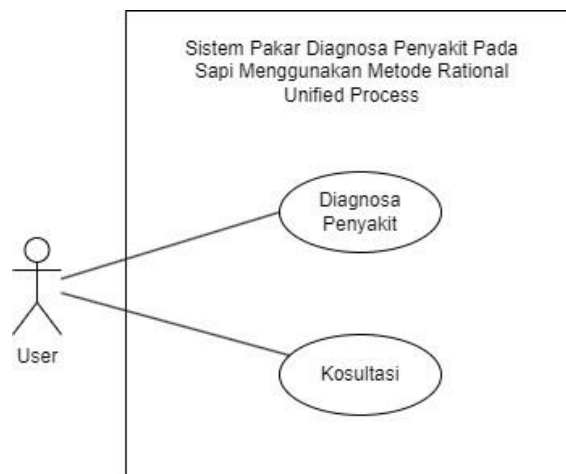
Gambar 3. 3 Gambaran Sistem Yang di usulkan

Rancangan sistem yang diusulkan untuk aplikasi sistem pakar diagnosa penyakit pada sapi menggunakan RUP yaitu, *user*

menginput gejala pada sapi kemudian diolah dengan menggunakan Model AI dengan mengolah data yang ada di *database*

3.2.1.4 Use Case Diagram

Use Case Diagram adalah salah satu dari berbagai jenis diagram UML (*Unified Modelling Language*) yang mendeskripsikan apa yang sistem lakukan tanpa mendeskripsikan bagaimana sistem menyelesaikannya (Finandhita, 2018). Serta menggambarkan hubungan interaksi sistem dan aktor.



Gambar 3. 4 *Use Case Diagram*

3.2.1.5 Value Stories

Berikut adalah *value* yang ditentukan:

Tabel 3. 2 Value Stories

User Stories Admin		Value
US-01	User dapat masuk ke aplikasi	5
US-02	User dapat melihat hasil diagnosa.	5
US-03	User dapat melakukan konsultasi	5

Keterangan penilaian:

- 5 : sangat diutamakan
- 4 : diutamakan kedua
- 3 : diutamakan ketiga
- 2 : diutamakan keempat

3.2.2 *Elaboration* (Perencanaan)

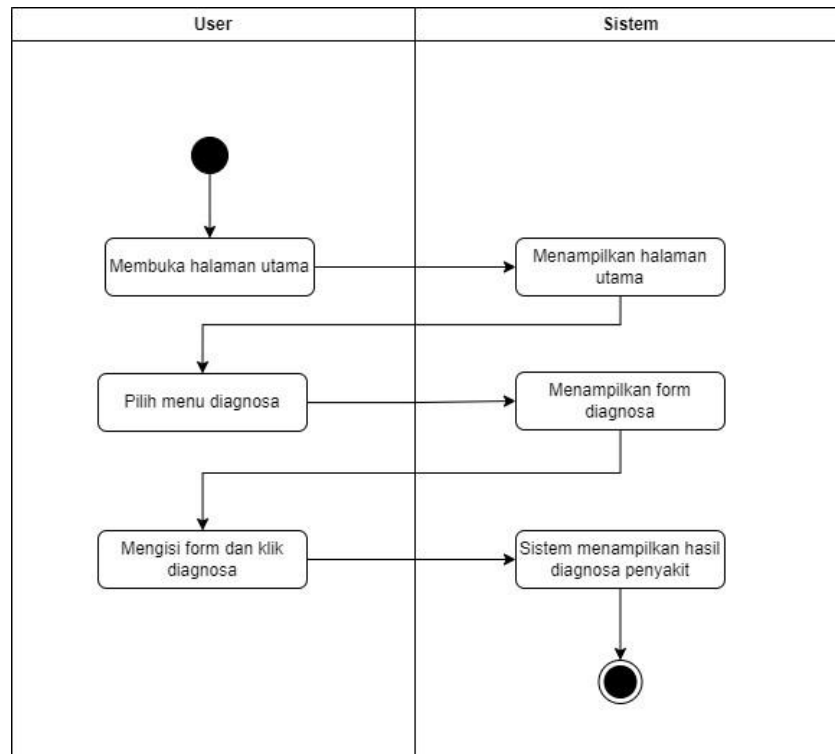
Tahap *Elaboration* merupakan lanjutan dari tahap *inception* di mana pada tahap *inception*, analisis and desain akan dilengkapi dengan menambahkan *Activity Diagram*, dan *Entity Relationship Diagram*. Dilanjutkan dengan tahap *implementation* yaitu prototipe yang berupa digital prototipe dari desain interface sistem.

3.2.2.1. *Activity Diagram*

Activity Diagram adalah *diagram* yang menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem yang diusulkan. *Activity Diagram* dalam bahasa Indonesia berarti diagram aktivitas, yaitu diagram yang dapat memodelkan proses-proses yang terjadi pada sebuah sistem. Runtutan proses dari suatu sistem digambarkan secara vertikal. *Activity Diagram* merupakan pengembangan dari *Use Case* yang memiliki alur aktivitas.

a. *Activity Diagram* Diagnosa Penyakit

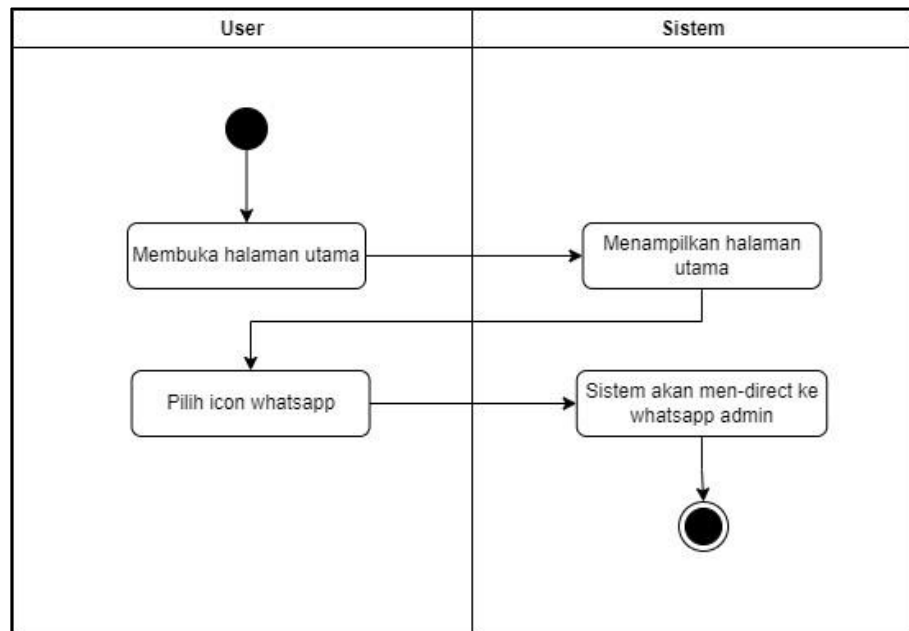
Gambar 3.5 Setelah *user* berhasil *Login* dan sistem menampilkan halaman dashboard, *user* memilih menu diagnosa agar dapat memperoleh informasi mengenai diagnosa penyakit pada sapi. Setelah memilih menu diagnosa, sistem akan menampilkan form diagnosa penyakit yang harus di isi oleh *user*. Setelah *user* mengisi form tersebut tekan tombol diagnosa dan sistem secara otomatis akan menampilkan mengenai penyakit yang di derita oleh sapi berdasarkan informasi yang sudah di input kan oleh *user* sebelumnya.



Gambar 3. 5 *Activity Diagram* Diagnosa Penyakit

b. *Activity Diagram* Konsultasi

Gambar 3.6 Apabila *user* telah berhasil memperoleh informasi mengenai diagnosa penyakit yang di derita oleh sapi dan ingin berkonsultasi dengan dokter hewan, *user* dapat memilih *ikon whatsapp* yang nantinya akan mengarahkan ke whatsapp dokter hewan tersebut.

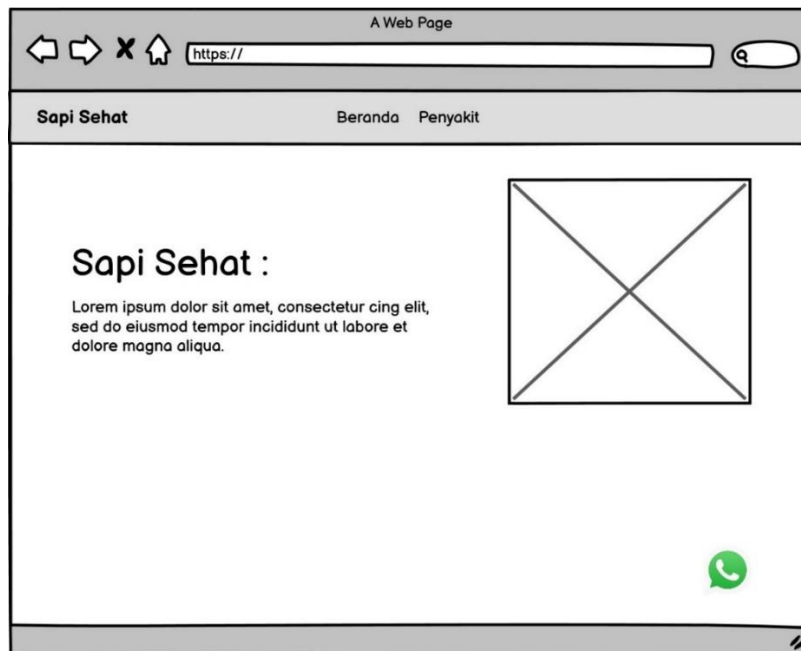


Gambar 3.6 Activity Diagram Konsultasi

3.2.2.2. Design Wireframe

1. Halaman Utama

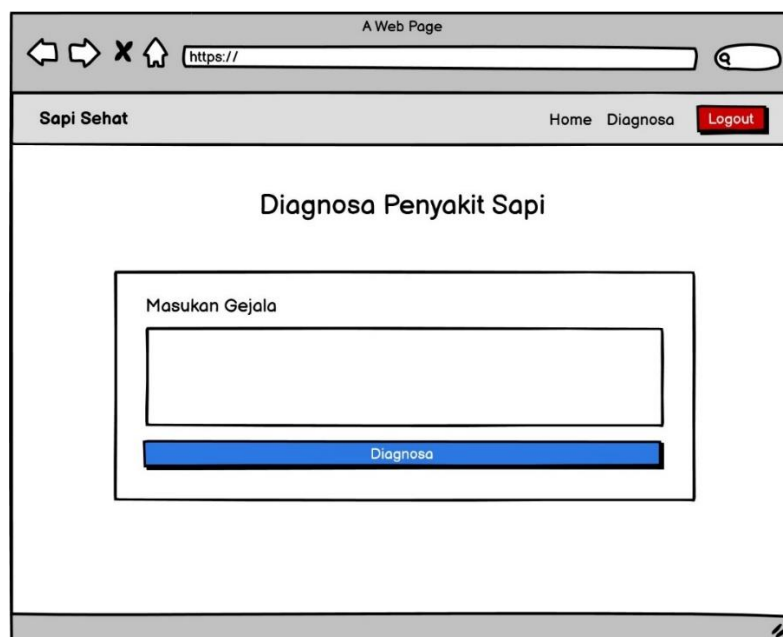
Halaman utama atau beranda (*homepage*) adalah halaman awal yang dilihat oleh pengunjung saat mereka mengakses suatu situs web atau aplikasi dari Sistem Pakar Diagnosa Penyakit Pada Sapi sebelum melakukan proses *Login*. Pada halaman utama memiliki beberapa fungsi atau fitur yaitu menu Beranda, Penyakit, *Login* dan menu Daftar. Apabila *user* hanya akan melihat informasi yang ada pada halaman beranda *user* tidak perlu mendaftar atau *Login* ke dalam sistem. Sedangkan bila *user* ingin menggunakan fitur yang ada pada sistem untuk memperoleh informasi mengenai diagnosa penyakit pada sapi, *user* diharuskan mendaftar atau *Login* terlebih dahulu dengan memilih menu masuk atau daftar apabila belum memiliki akun.



Gambar 3. 7 Halaman Utama

2. Halaman Form Diagnosa Penyakit

Setelah *user* memilih menu penyakit pada proses sebelumnya, sistem akan menampilkan form input diagnosa penyakit yang dapat di isi oleh *user* seperti yang terdapat pada Gambar 3.8



Gambar 3. 8 Halaman Form Diagnosa Penyakit

3. Halaman Hasil Diagnosa

Setelah *user* berhasil selesai mengisi form diagnosa penyakit pada proses sebelumnya, sistem akan menampilkan informasi diagnosa penyakit berdasarkan data penyakit atau gejala yang di input kan oleh *user*.

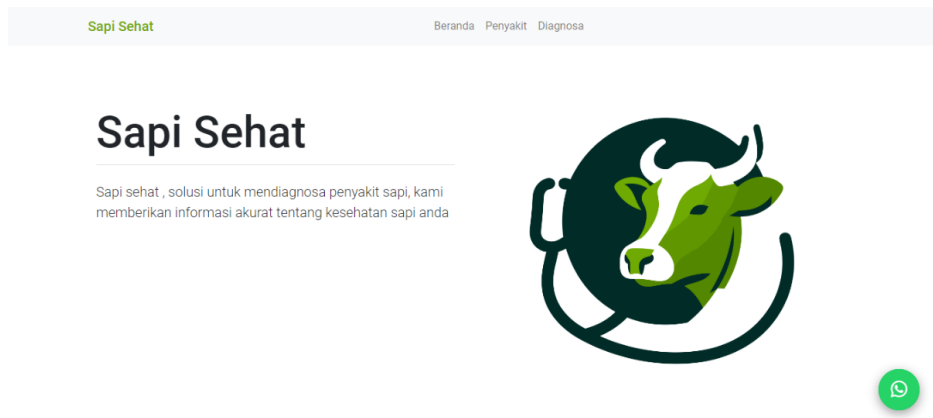
The screenshot shows a web browser window with the title 'A Web Page'. The address bar contains 'https://'. The page header includes 'Sapi Sehat', 'Home', 'Diagnosa', and a 'Logout' button. The main content area is titled 'Diagnosa Penyakit Sapi' and contains a form with the following elements:

- Masukan Gejala:** A text input field for entering symptoms.
- Diagnosa:** A blue button to submit the form.
- Hasil Diagnosa :** A section displaying the results of the diagnosis, including:
 - Penyakit :** A blue box containing the text 'lorem ipsum dolor..'
 - Risiko :** A red box containing the text 'Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt'.
 - Penanganan :** A teal box containing the text 'Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt'.
 - Tingkat Akurasi :** A green box containing the text '98.10%'.

Gambar 3. 9 .Halaman Hasil Diagnosa

3.2.3 Construction (Konstruksi)

Pada tahap ini penulis melakukan pembuatan *website*. Setelah proses pembuatan *website* selesai, maka penulis akan langsung menguji apakah sistem berjalan dengan baik atau sebaliknya.



Gambar 3. 10 Halaman *Dashboard*

Gambar 3.11 merupakan tampilan awal dari aplikasi *Sapi Sehat*. Pada halaman *dashboard* ini terdapat halaman tampilan *dashboard website sapi sehat*.



Gambar 3. 11 Halaman *Diagnosa*

Gambar 3.11 merupakan tampilan profil dari *website sapi sehat*. Masyarakat dapat melihat tampilan dari berbagai dianogsa penyakit yang terdapat pada sapi.

Diagnosa Penyakit Sapi

Masukkan gejala :

Diagnosa

Diagnosa Penyakit Sapi

Masukkan gejala :

Diagnosa

Gambar 3. 12 Halaman Form Diagnosa

Gambar 4.12 merupakan tampilan dari form masukan gejala pada aplikasi Sapi Sehat. Masyarakat/*user* diharuskan mengisi form tersebut apabila ingin mengetahui atau memperoleh informasi mengenai penyakit yang di derita oleh sapi.

Diagnosa Penyakit Sapi

Masukkan gejala :

Hasil Diagnosa:

Penyakit:
Abses (Kumpulan nanah yang terbentuk di dalam jaringan tubuh)

Risiko:
Gangguan reproduksi.

Penanganan:
1.000 unit dengan GUSANEX dan 8.000 ml dengan Dexatozoon Injeksi dan 8.000 ml dengan B-SANPLEX dan 8.000 ml dengan VIT-OXY LA

Tingkat Akurasi:
100,0%

Gambar 3. 13 Halaman Hasil Diagnosa

Gambar 3.13 merupakan tampilan dari hasil diagnosa penyakit yang di derita oleh sapi berdasarkan informasi gejala yang telah di inputkan oleh masyarakat atau *user*. Dari hasil diagnosa terdapat informasi yang di sajikan seperti informasi mengenai penyakit, risiko, cara menangani dan tingkat akurasi dari generate informasi tersebut.

3.2.4 *Transition (Transisi)*

Pada tahap akhir peneliti memfokuskan pada penyampaian aplikasi yang sudah jadi dan melakukan pengujian kepada pengguna. Pengujian ini bertujuan aplikasi yang dibangun sudah menjawab permasalahan yang terjadi serta kualitas yang dihasilkan sudah memadai sesuai dengan apa yang dibutuhkan.

3.2.5 Algoritma Random Forest

Definisi algoritma *random forest*, adalah setiap tree pada random forest akan mengeluarkan prediksi kelas. Prediksi kelas dengan vote terbanyak menjadi kandidat prediksi pada model. Semakin banyak jumlah tree maka akan menghasilkan akurasi yang lebih tinggi. Algoritma ini didasarkan pada konsep *assemble learning*, yakni proses menggabungkan beberapa pengklasifikasian untuk mencapai masalah yang kompleks dan untuk meningkatkan kinerja model. Contoh gejala yaitu batuk dan kecurusan.

1. Sampel Data Set

Tabel 3.3 Sampel Data Set

No.	Gejala	Penyakit
1	Batuk, Kecurusan, Gangguan sistem pencernaan	Parasit internal
2	Batuk, Demam	Pneumonia
3	Batuk, Demam, Nafsu makan menurun, Lelah	Pneumonia
4	Batuk, Lelah, Nafsu makan menurun	Tuberculosis

2. Frekuensi

No	Batuk	Kecurusan	Gangguan sistem pencernaan	Demam	Nafsu makan menurun	Lelah	Prediksi
1	1	1	1	0	0	0	Parasit internal
2	1	0	0	1	0	0	Pneumonia
3	1	0	0	1	1	1	Pneumonia
4	1	0	0	0	1	0	Tuberculosis

Note : 1 = iya

0 = tidak

Parasit Internal = 1

Pneumonia = 2

Tuberculosis = 1

Entropy = (s) = -

$$= (s) - \left(\frac{1}{4} \log_2 \frac{1}{4} + \frac{2}{4} \log_2 \frac{2}{4} + \frac{1}{4} \log_2 \frac{1}{4} \right)$$

$$\begin{aligned}
&= -(0.25 \times \log_2 0.25 + 0.5 \log_2 0.5 + 0.25 \log_2 0.25) \\
&= 0,5 + 0,5 + 0,5 = 1.5
\end{aligned}$$

3 Gain

1. Gain batuk

1 = 4 sample

Parasit interna (1)

Pneumonia (2)

Tuberculosis (1)

$$\begin{aligned}
&= \left(\frac{1}{4} \log_2 \frac{1}{4} + \frac{2}{4} \log_2 \frac{2}{4} + \frac{1}{4} \log_2 \frac{1}{4} \right) \\
&= -(0.25 \times (-2)) + 0,5 \times (-1) + 0,25 \times (-2) \\
&= 0.5 + 0.5 + 0.5 \\
&= 1.5
\end{aligned}$$

0 = 0 sample

$$\begin{aligned}
\text{Gain batuk} = H(S) &= \left(\frac{4}{4} \times H(\text{batuk}) \right) = 1.5 - \left(\frac{4}{4} \times 1.5 \right) \\
&= 1.5 - 1.5 \\
&= 0
\end{aligned}$$

2. Gain kecurusan

1 = 1 sample

Parasit interna (1)

$$\begin{aligned}
&= \left(\frac{1}{4} \log_2 1 \right) \\
&= 0
\end{aligned}$$

0 = 2 sample

Pneumonia (2)

Tuberculosis (1)

$$\begin{aligned} &= \left(\frac{2}{2} \log_2 \frac{2}{2} + \frac{1}{2} \log_2 \frac{1}{2}\right) \\ &= -(1 \times 0) + 0,5 \times (-1) \\ &= 0 + 0,5 \\ &= 0,5 \end{aligned}$$

Gain kekurangan

$$\begin{aligned} &= 1,5 \left(\frac{1}{4} \times 0 + \frac{3}{4} \times 0,5\right) \\ &= 1,5 - (0,25 + 0,375) \\ &= 1,5 - 0,625 \\ &= 0,875 \end{aligned}$$

3. Gangguan sistem pencernaan

1 = 1 sample

Parasit interna (1)

$$\begin{aligned} &= \left(\frac{1}{4} \log_2 1\right) \\ &= 0 \end{aligned}$$

0 = 2 sample

Pneumonia (2)

Tuberculosis (1)

$$\begin{aligned} &= \left(\frac{2}{2} \log_2 \frac{2}{2} + \frac{1}{2} \log_2 \frac{1}{2}\right) \\ &= -(1 \times 0) + 0,5 \times (-1) \\ &= 0 + 0,5 \\ &= 0,5 \end{aligned}$$

Gain gangguan sistem pencernaan

$$= 1,5 \left(\frac{1}{4} \times 0 + \frac{3}{4} \times 0,5\right)$$

$$= 1.5 - (0.25 + 0.375)$$

$$= 1.5 - 0.125$$

4. Demam

1 = 1 sample

Parasit interna (2)

$$= \left(\frac{2}{1} \log_2 2\right)$$

$$= 2$$

0 = 2 sample

Parasit internal (1)

Tuberpolosis (1)

$$= \left(\frac{1}{2} \log_2 1 + \frac{1}{2} \log_2 \frac{1}{2}\right)$$

$$= 1$$

Gain Demam

$$= 1.5 \left(\frac{2}{4} \times 1 + \frac{2}{4} \times 1\right)$$

$$= 1.5 - (0.5 + 0.5)$$

$$= 1.5 - 1$$

$$= 0.5$$

5. Nafsu makan menurun

1 = 2 sample

Pneumonia (1)

Tuberpolosis (1)

$$= \left(\frac{1}{2} \log_2 1 + \frac{1}{2} \log_2 1\right)$$

$$= -(0.5 \times 1 + 0.5 \times 1)$$

$$= 0.5 + 0.5$$

$$= 1$$

0 = 2 sample

Parasit internal (1)

Penomonia (1)

$$\begin{aligned} &= \left(\frac{1}{2} \log_2 1 + \frac{1}{2} \log_2 1\right) \\ &= -(0.5 \times 1 + 0.5 \times 1) \\ &= 0.5 + 0.5 \\ &= 1 \end{aligned}$$

Gain Nafsu makan menurun

$$\begin{aligned} &= 1.5 \left(\frac{2}{4} \times 1 + \frac{2}{4} \times 1\right) \\ &= 1.5 - (0.5 + 0.5) \\ &= 1.5 - 1 \\ &= 0.5 \end{aligned}$$

6. Lelah

1 = 1 sample

Pneumonia (1)

$$\begin{aligned} &= \left(\frac{1}{1} \log_2 1\right) \\ &= 0 \end{aligned}$$

0 = 4 sample

Parasit interna (1)

Pneumonia (1)

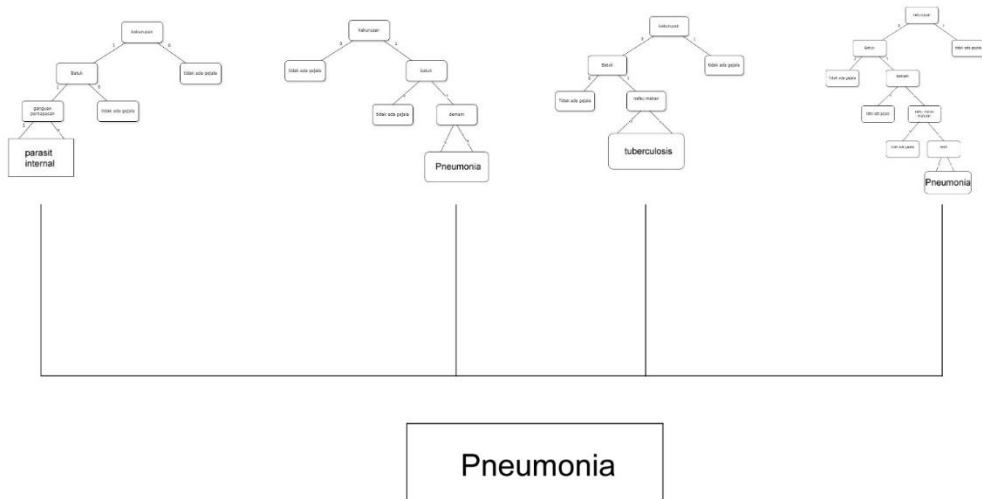
Tuberculosis (1)

$$\begin{aligned} &= \left(\frac{1}{3} \log_2 1 + \frac{2}{3} \log_2 1 + \frac{1}{3} \log_2 1\right) \\ &= -(0.33 \times 0 + 0.33 \times 0 + 0.33 \times 0) \\ &= 0 \end{aligned}$$

Gain lelah

$$\begin{aligned} &= 1.5 \left(\frac{1}{4} \times 0 + \frac{3}{4} \times 0 \right) \\ &= 1.5 - (0 + 0) \\ &= 1.5 - 0 \\ &= 1.5 \end{aligned}$$

4. Hasil Akhir



Gambar 3.14 Hasil Akhir Perhitungan

BAB IV

HASIL DAN PEMBAHASAN

4.1. Hasil dan Pembahasan

Penelitian bertujuan untuk merancang dan membangun sistem pakar yang dapat digunakan untuk diagnosa penyakit pada hewan sapi yang optimal dengan menggunakan Algoritma dan pendekatan *machine learning* yang berbasis *website*.

4.1.1 Deskripsi

Pada penelitian ini menggunakan data set yang telah dipilih yang dimana terdiri dari 4 kolom yaitu kolom penyakit, gejala, penanganan dan risiko yang Dimana kolom gejala digunakan untuk melakukan pembelajaran model, sedangkan untuk penyakit, penanganan dan risiko merupakan target dari prediksi model. Data set yang digunakan pada penelitian ini merupakan data set teks yang terdiri dari 457 total data. Dengan diberikan gejala yang lengkap, maka diharapkan model machine learning yang dibangun dapat melakukan proses prediksi penyakit, penanganan dan risiko dengan baik.

4.1.2 Prosedur Eksperimen

Prosedur yang diberikan yaitu berupa tahapan dan Langkah- langkah yang dilakukan untuk mengembangkan system pakar diagnosa penyakit pada sapi.

4.1.2.1. Data set

Data set yang digunakan pada penelitian ini didapatkan langsung setelah dilakukan wawancara dengan dokter hewan mengenai penyakit, jenis penanganan dan risiko yang dihasilkan dari gejala tertentu. Dalam data set ini, terdiri dari 4 kolom yaitu kolom penyakit, gejala, penanganan dan risiko, yang Dimana nantinya model akan melakukan proses pengenalan pola dengan menggunakan data pada kolom gejala untuk melakukan proses prediksi penyakit, penanganan dan risiko berdasarkan gejala yang terkait. Data set yang digunakan berjumlah total 457 data. Dengan total data sebanyak 457 data tersebut yang Dimana langsung didapatkan

dari dokter hewan, maka memungkinkan untuk dapat melatih model sehingga dapat belajar berbagai macam gejala, penyakit, penanganan dan risiko yang ada pada sapi. Sehingga, dengan hal tersebut, diharapkan bahwa dengan digunakannya data tersebut sebagai data pelatihan dan data pengujian model, dapat membangun model yang memiliki akurasi yang baik untuk melakukan proses diagnosa penyakit pada sapi dan dapat membantu para peternak untuk mengetahui penyakit, penanganan dan risiko berdasarkan gejala pada sapi mereka.

4.1.2.2.Data Preprocessing

Dari total 457 data yang akan digunakan untuk proses Pembangunan model machine learning diagnosa penyakit sapi, terdapat beberapa data yang kosong. Untuk sebelumnya, pada gambar 4.1 diberikan visualisasi dari data set yang digunakan untuk melakukan Pembangunan model.

```
dataset_path = 'sapi1.xlsx'
df = pd.read_excel(dataset_path)
df.head()
```

	penyakit	gejala	penanganan	risiko
0	Abses	Abses	1.000 unit dengan GUSANEX dan 8.000 ml dengan ...	Penyebaran Infeksi, Gangguan kesehatan, Kematian...
1	Avitaminosis	Anemia, Anorexia, Bulu Kusam, Demam, Diare, Ga...	1.000 botol dengan PYRONIL-2500 BOLLUS dan 8.00...	Kekurangan vitamin (tidak terlalu beresiko)
2	Babesiosis	Bulu kusam,Gatal,Kelainan kulit,Perkultitan	8000 gram dengan TRYPONIL, @3000 ml dengan Int...	Penurunan kesehatan, Kematian, Penurunan produ...
3	Bovine Ephemeral Fever	Anorexia, Demam, Demam, Demam, Diare, Duraasi g...	10.000 ml dengan B-SANPLEX dan 1.000 botol den...	Bisa menurunkan produksi daging
4	Corpus Luteum Persisten	Anestrus	5000 ml dengan CAPRIGLADIN INJ, @10000 ml deng...	Gangguan reproduksi

Gambar 4.1 Data Set

Gambar 4.1 menunjukkan 5 data teratas yang digunakan untuk Pembangunan model *machine learning*. Apabila dilakukan proses analisis, maka dapat dilihat bahwa data set yang digunakan sudah memberikan data yang cukup kompleks dan baik sehingga dapat memungkinkan Pembangunan model machine learning yang baik. Namun, karena masih terdapat beberapa data yang kosong pada kolom risiko, maka dilakukan proses mengisi semua data pada kolom risiko yang kosong dengan menggunakan string “Bisa menurunkan produksi daging”. Untuk visualisasi proses dan hasil data set yang didapatkan setelah proses mengisi data diberikan pada gambar 4.2.

```
df["risiko"].fillna("Bisa menurunkan produksi daging", inplace=True)
df.head()
```

	penyakit	gejala	penanganan	risiko
0	Abses	Abses	1.000 unit dengan GUSANEX dan 8.000 ml dengan ...	Penyebaran infeksi, Gangguan kesehatan, Kematian...
1	Avitaminosis	Anemia, Anorexia, Bulu Kusam, Demam, Diare, Ga...	1.000 bolus dengan PYRONIL-2500 BOLUS dan 8.00...	Kekurangan vitamin (tidak terialu beresiko)
2	Babesiosis	Bulu kusam,Gatal,Kelainan kulit,Perkultian	8000 gram dengan TRYPONIL; @3000 ml dengan Int...	Penurunan kesehatan, Kematian, Penurunan produ...
3	Bovine Ephemeral Fever	Anorexia, Demam, Demam, Demam, Diare, Durasi g...	10.000 ml dengan B-SANPLEX dan 1.000 bolus den...	Bisa menurunkan produksi daging
4	Corpus Luteum Persisten	Anestrus	5000 ml dengan CAPRIGLADIN INJ; @10000 ml deng...	Gangguan reproduksi

Gambar 4.2 Data Set Setelah Proses Pengisian Data Null

Gambar 4.2 menunjukkan hasil proses mengisi data kosong pada kolom risiko dengan string “Bisa menurunkan produksi daging”. Tujuan dilakukannya proses mengisi data kosong tersebut yaitu agar performa model yang dibangun dapat menghasilkan akurasi yang baik, maka dilakukan proses mengisi semua data dan memastikan tidak ada data yang kosong, karena dengan adanya data kosong tersebut, maka bisa berpengaruh pada kualitas model dalam melakukan proses diagnosa penyakit pada sapi. Sehingga setelah dilakukan proses pengisian data kosong tersebut, maka ketika di lakukan proses pengecekan seluruh data set, menunjukkan tidak adanya data Null. Untuk hasil pengecekan data set diberikan pada gambar 4.3 dan 4.4.

```
df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 456 entries, 0 to 455
Data columns (total 4 columns):
#   Column      Non-Null Count  Dtype
---  ---
0   penyakit    456 non-null    object
1   gejala      456 non-null    object
2   penanganan  456 non-null    object
3   risiko     456 non-null    object
dtypes: object(4)
memory usage: 14.4+ KB
```

Gambar 4.3 Pengecekan Info Data Set

```
# Menghitung jumlah missing value di setiap kolom
missing_values_count = df.isna().sum()
print("Jumlah missing value di setiap kolom:")
print(missing_values_count)

# Menghitung jumlah baris yang memiliki setidaknya satu missing value
missing_rows_count = df.isna().any(axis=1).sum()
print(f"Jumlah baris dengan setidaknya satu missing value: {missing_rows_count}")

Jumlah missing value di setiap kolom:
penyakit      0
gejala        0
penanganan    0
risiko         0
dtype: int64
Jumlah baris dengan setidaknya satu missing value: 0
```

Gambar 4.4 Pengecekan Data Null Dari Setiap Kolom

Gambar 4.3 dan 4.4 menunjukkan hasil proses pengecekan data set yang sudah dilakukan. Dapat dilihat pada gambar 4.3, ketika dilakukan pengecekan info data set menghasilkan data pada setiap kolom yaitu sebanyak “456 *non-null*”, yang Dimana, dengan hasil tersebut, menunjukkan pada setiap kolom tidak terdapat data yang kosong dan mulai dari data pada Index ke 0 hingga 456 (sehingga total data yaitu sebanyak 457 data) datanya terisi semua. Dapat dilihat pula pada gambar 4.4, ketika dilakukan proses pencarian nilai yang missing value atau nilai yang kosong, pada setiap kolom mendapatkan hasil 0, yang Dimana setelah dilakukan proses ini, maka dapat dipastikan bahwa data set yang digunakan sudah cukup baik dan tidak terdapat data yang kosong.

Setelah dilakukan *preprocessing* data berupa pengisian nilai *null*, maka dapat dilakukan lagi *preprocessing* data yaitu berupa *stemming*. Tujuan dilakukannya proses *stemming* ini yaitu untuk menjadikan suatu kata yang memiliki imbuhan atau akhiran- akhiran morfologis menjadi data dasar atau stem. Sehingga dengan setiap kata diubah menjadi kata dasar, maka akan mempermudah dan membantu mempercepat waktu proses running pelatihan dan pengujian model. Dengan dilakukannya proses *stemming* juga mempermudah analisis teks dan pengelompokan kata-kata dalam proses analisis sentimen. Pada penelitian ini, proses *stemming* data akan menggunakan library Sastrawi yang Dimana merupakan library untuk proses *stemming* data set dengan Bahasa Indonesia. Untuk proses *stemming* yang dilakukan, diberikan pada gambar 4.5.


```

# Inialisasi stemmer Bahasa Indonesia
factory = StemmerFactory()
stemmer = factory.create_stemmer()

# Fungsi untuk melakukan stemming
def stem_text(text):
    return stemmer.stem(text)

# Melakukan stemming pada semua kolom teks
for col in ['gejala', 'penanganan', 'risiko']:
    df[col] = df[col].apply(stem_text)

```

Gambar 4.5 Proses *Stemming*

Gambar 4.5 menunjukkan proses *stemming* yang dilakukan. Dapat dilihat pada gambar 4.5, menunjukkan bahwa proses *stemming* diberikan pada kolom gejala, penanganan dan risiko, karena pada kolom tersebut berisi data- data yang berupa Bahasa Indonesia, sedangkan, proses *stemming* tidak dilakukan pada kolom penyakit karena pada kolom penyakit terdiri dari istilah istilah medis yang Dimana apabila dilakukan proses *stemming* dapat merusak makna asli dari istilah yang ada, oleh karena itu, pada kolom penyakit tidak dilakukan proses *stemming*. Setelah dilakukan proses *stemming*, maka dapat dilakukan proses pembagian data set menjadi variabel *predictor* dan variabel target, yang Dimana, variabel predictor merupakan kolom gejala, sedangkan variabel target merupakan kolom penyakit, penanganan dan diagnosa. Untuk visualisasi pembagian variabel diberikan pada gambar 4.6.

```

x = df['gejala']
y = df[['penyakit', 'penanganan', 'risiko']]

```

Gambar 4.6 Proses Pembagian Variabel Data Set

Setelah dilakukan proses pembagian *variabel* pada data set, maka akan dilakukan proses tokenizing atau proses pemecahan teks menjadi beberapa unit terpisah, sehingga lebih mempermudah model machine learning dalam melakukan proses analisis dan pengolahan data. Untuk visualisasi proses tokenizing diberikan pada gambar 4.7.

```

# Librari NLTK untuk proses tokenizing
import nltk
nltk.download('punkt')
from nltk.tokenize import word_tokenize

# Fungsi untuk melakukan tokenizing
def tokenize_text(text):
    return ' '.join(word_tokenize(text))

[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data] Package punkt is already up-to-date!

df['gejala'] = df['gejala'].apply(tokenize_text)

```

Gambar 4.7 Proses Tokenisasi

Gambar 4.7 menunjukkan proses tokenizing pada data set. Proses tokenizing dilakukan pada data set dan kolom gejala. Tokenisasi pada kolom gejala dalam data kesehatan bertujuan memisahkan teks gejala menjadi unit-unit terpisah (token), mempermudah analisis individu, pencarian informasi, dan pemrosesan data. Tahapan ini mendukung klasifikasi penyakit, diagnosis, serta pra-pemrosesan untuk model bahasa alami, memungkinkan identifikasi pola, pencarian kata kunci, dan analisis klustering gejala dalam konteks medis. Sehingga dengan hal tersebut, memungkinkan model dapat dengan mudah melakukan proses pengenalan pola dan analisis data set.

4.1.2.3. Pembagian Data dan Transformasi Data

Langkah krusial dalam machine learning adalah membagi data menjadi data latih (*train data*) dan data uji (*test data*). Proses pembagian ini dilakukan dengan menerapkan teknik acak (*random-split*) dan validasi data. Data latih berfungsi sebagai set pelatihan untuk mengajarkan model, sementara data uji digunakan untuk menguji kinerja model. Dengan adanya pemisahan ini, diharapkan dapat memperoleh estimasi yang lebih akurat, memungkinkan model untuk melakukan prediksi dengan baik pada data yang belum pernah dilihat sebelumnya. Untuk visualisasi pembagian data set diberikan pada gambar 4.8.

```

# Memisahkan data menjadi training set (70%) dan testing set (30%)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=24)

```

Gambar 4.8 Proses Pembagian Data Set

Gambar 4.8 menunjukkan proses pembagian data set yang Dimana dapat dilihat bahwa terdapat 70% data yang digunakan sebagai data pelatihan dan 30% sebagai data pengujian. Yang Dimana berarti sebanyak 70% data akan digunakan dalam model melakukan proses pengenalan pola dari data dan sebanyak 30% akan digunakan dalam model melakukan proses evaluasi berdasarkan hasil Latihan model yang sudah dilakukan. Setelah dilakukan proses pembagian data menjadi data pelatihan dan pengujian, maka akan dilakukan proses konversi dari data yang awalnya merupakan data string menjadi data berbentuk numerik atau *vector*. Tujuan dilakukannya proses konversi ini yaitu karena model machine learning hanya dapat melakukan pemrosesan dengan menggunakan data bit atau numerik, sehingga dengan dilakukannya proses konversi data menjadi data vector, dapat mempermudah dan menjadikan model dapat belajar pola dari data dengan baik. Dalam penelitian ini, proses konversi data menjadi vector menggunakan TF-IDF *Vectorizer*. Tujuan digunakannya TF-IDF *Vectorizer* yaitu pada TF-IDF *Vectorizer* dapat mengukur seberapa penting sebuah kata dalam suatu dokumen terkait dengan keseluruhan korpus atau koleksi dokumen. TF-IDF, yang merupakan singkatan dari *Term Frequency-Inverse Document Frequency*, memberikan skor tinggi untuk kata-kata yang sering muncul dalam suatu dokumen tetapi jarang muncul di seluruh koleksi dokumen. Dengan menggunakan TF-IDF *vectorizer*, kita dapat mengubah teks menjadi representasi numerik yang dapat digunakan sebagai fitur dalam model machine learning, memungkinkan analisis dan pemodelan teks yang lebih efektif dan relevan dengan konteks data yang diberikan. Untuk proses konversi data menjadi vector diberikan pada gambar 4.9.

```
# Inisialisasi TF-IDF vectorizer
vectorizer = TfidfVectorizer()

# Melakukan TF-IDF transformation
X_train_tfidf = vectorizer.fit_transform(X_train)
X_test_tfidf = vectorizer.transform(X_test)
```

Gambar 4.9 Proses TF-IDF *Vectorizer*

Gambar 4.9 menunjukkan proses konversi data string menjadi data *vector*. Dapat dilihat pada gambar 4.1.9, bahwa proses konversi *vector* diberikan pada

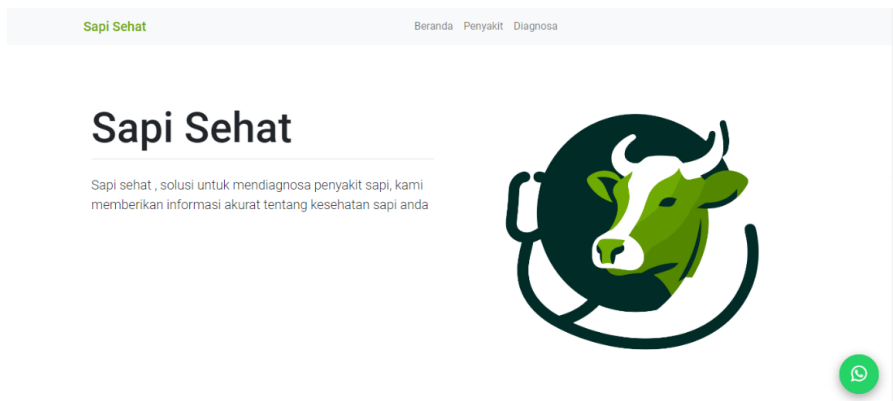
X_train dan X_test, yang Dimana merupakan data yang digunakan untuk melakukan proses pengenalan pola dari data. Sehingga dengan dilakukan konversi data menjadi *vector*, diharapkan untuk model dapat melakukan pengenalan pola dengan baik.

4.1.2.4. *Modelling* dan Evaluasi Model

Setelah dilakukan proses preprocessing data set dan konversi data sehingga dapat siap digunakan untuk membangun model, maka selanjutnya dapat dilakukan proses modelling dengan menggunakan data set yang sudah diproses dan dibagi menjadi data pelatihan serta data pengujian. Dalam penelitian ini, algoritma Random Forest yang paling optimal. Tujuan dipilihnya algoritma Random Forest algoritma proses klasifikasi berdasarkan target, sehingga dengan menggunakan *random forest* diharapkan dapat lebih baik untuk melakukan proses diagnosa penyakit pada sapi.

4.1.2.5. Implementasi

Tahapan ini berisi *screenshot* hasil tampilan dan potongan kode. Proses konstruksi aplikasi menggunakan bahasa pemrograman *PYTHON*.



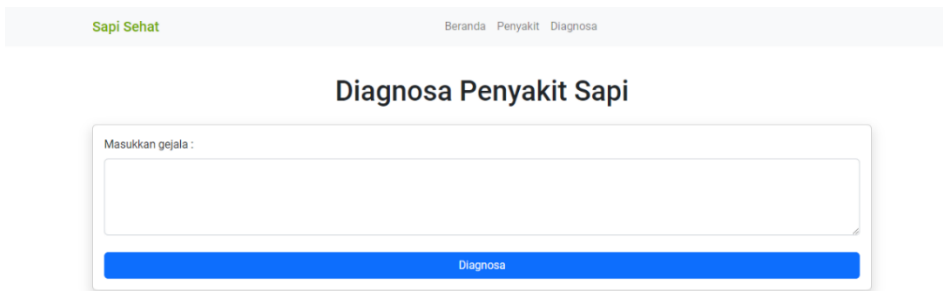
Gambar 4.10 Halaman dashboard

Gambar 4.10 merupakan tampilan awal dari aplikasi jika masyarakat/*user* sudah berhasil *Login* ke dalam *website Sapi Sehat*. Pada halaman *dashboard* ini terdapat halaman tampilan *dashboard website sapi sehat*.



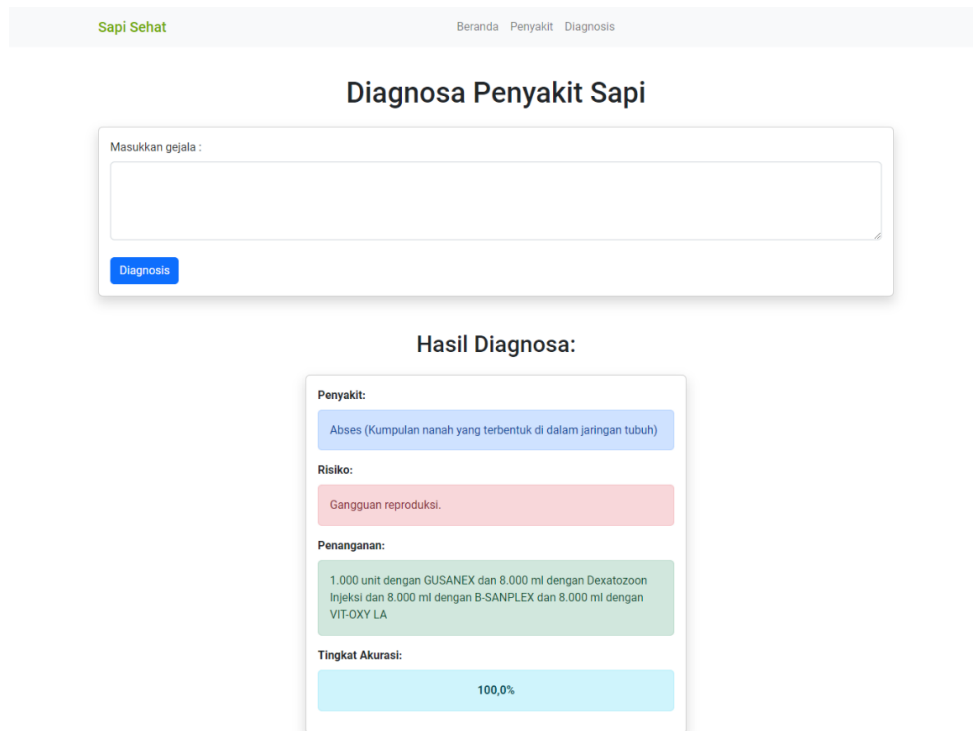
Gambar 4.11 Halaman Diagnosa

Gambar 4.13. merupakan tampilan profil dari *website* jika masyarakat/*user* sudah *Login* ke *website* sapi sehat. Masyarakat dapat melihat tampilan dari berbagai dianogsa penyakit yang terdapat pada sapi.



Gambar 4.12 Halaman Form Diagnosa

Gambar 4.14. merupakan tampilan dari form masukan gejala pada aplikasi Sapi Sehat. Masyarakat/*user* diharuskan mengisi form tersebut apabila ingin mengetahui atau memperoleh informasi mengenai penyakit yang di derita oleh sapi.



Gambar 4.13 Halaman Hasil Diagnosa

Gambar 4.15. merupakan tampilan dari hasil diagnosa penyakit yang di derita oleh sapi berdasarkan informasi gejala yang telah di inputkan oleh masyarakat atau *user*. Dari hasil diagnosa terdapat informasi yang di sajikan seperti informasi mengenai penyakit, risiko, cara menangani dan tingkat akurasi dari generate informasi tersebut

4.1.2.6. Pengujian (*Testing*)

Kegiatan yang dilakukan pada tahap ini adalah untuk meminimalisir kesalahan dan memastikan output yang dihasilkan sesuai dengan yang diharapkan.

a. Deskripsi Pengujian *Blackbox*

Pengujian ini dilakukan dengan menggunakan *web browser*. Teknik yang saya pakai dalam pengujian ini adalah pengujian black box. Hal ini dikarenakan pengujian black box berfokus pada hasil yang dikeluarkan. *Blackbox testing* memiliki keuntungan dan kekurangan dalam implementasinya, salah satu kelebihanannya yaitu membantu dalam hal penemuan aspek yang tidak terpenuhi dari spesifikasi kebutuhan yang

diberikan dalam pengembangan perangkat lunak, dan kekurangan dari blackbox testing adalah pengujian tidak bisa dilakukan sepenuhnya dikarenakan pengetahuan, penguji terbatas tentang perangkat lunak yang diuji [15].

b. Prosedur Pengujian

Dalam pengujian ini peneliti menggunakan teknik pengujian *blackbox*. Adapun prosedur pengujian ini adalah sebagai berikut:

1. Menyiapkan beberapa *user* untuk menggunakan aplikasi sipemurah melalui *web browser* yang berbeda-beda.
2. Melakukan proses *login* aplikasi.
3. Melakukan pengujian dari tiap-tiap proses dengan berbagai *test cast*.
4. Melakukan pengamatan terhadap tiap-tiap proses.

4.1.2.7. Proses Pengujian

Tabel 4.1 Pengujian User 1

Nama Pengguna : Ahmad Dahlan Web Browser : Google Chrome Alamat			
Aksi	Yang Diharapkan	Hasil Yang di dapatkan	Kesimpulan
Membuka halaman website	Menampilkan halaman website	Setelah melakukan pengujian <i>user</i> tidak menemukan adanya <i>error</i> pada website	Berhasil
Menekan menu diagnosa	Mampu melihat, menginput, mengubah, dan menghapus data penyakit pada hewan.	<i>User</i> melakukan pengujian dengan memasukkan diagnosa dan tidak menemukan adanya <i>error</i> , semua fungsi berjalan dengan baik.	Berhasil
Fitur konsultasi	Mampu menghubungkan <i>user</i> dengan dokter hewan dengan API Whatsap	<i>User</i> tidak menemukan adanya <i>error</i> .	Berhasil

Tabel 4.2 Pengujian User 2

Nama Pengguna : Suherman Web Browser : Google Chrome			
Aksi	Yang Diharapkan	Hasil Yang di dapatkan	Kesimpulan

Membuka halaman website	Menampilkan halaman website	Setelah melakukan pengujian <i>user</i> tidak menemukan adanya <i>error</i> pada website	Berhasil
Menekan menu diagnosa	Mampu melihat, menginput, mengubah, dan menghapus data penyakit pada hewan.	<i>User</i> melakukan pengujian dengan memasukkan diagnosa dan tidak menemukan adanya <i>error</i> , semua fungsi berjalan dengan baik.	Berhasil
Fitur konsultasi	Mampu menghubungkan <i>user</i> dengan dokter hewan dengan API Whatsap	<i>User</i> tidak menemukan adanya <i>error</i> .	Berhasil

Tabel 4.3 Pengujian User 3

Nama Pengguna : Zunaidi Web Browser : Google Chrome			
Aksi	Yang Diharapkan	Hasil Yang di dapatkan	Kesimpulan
Membuka halaman website	Menampilkan halaman website	Setelah melakukan pengujian <i>user</i> tidak menemukan adanya <i>error</i> pada website	Berhasil
Menekan menu diagnosa	Mampu melihat, menginput, mengubah, dan menghapus data penyakit pada hewan.	<i>User</i> melakukan pengujian dengan memasukkan diagnosa dan tidak menemukan adanya <i>error</i> , semua fungsi berjalan dengan baik.	Berhasil
Fitur konsultasi	Mampu menghubungkan <i>user</i> dengan dokter hewan dengan API Whatsap	<i>User</i> tidak menemukan adanya <i>error</i> .	Berhasil

Tabel 4.4 Pengujian User 4

Nama Pengguna : Sakimin Web Browser : Google Chrome			
Aksi	Yang Diharapkan	Hasil yang di dapatkan	Kesimpulan
Membuka halaman website	Menampilkan halaman website	Setelah melakukan pengujian <i>user</i> tidak menemukan adanya <i>error</i> pada website	Berhasil
Menekan menu diagnosa	Mampu melihat, menginput, mengubah, dan menghapus data penyakit pada hewan.	<i>User</i> melakukan pengujian dengan memasukkan diagnosa dan tidak menemukan adanya <i>error</i> , semua	Berhasil

		fungsi berjalan dengan baik.	
Fitur konsultasi	Mampu menghubungkan <i>user</i> dengan dokter hewan dengan API Whatsap	<i>User</i> tidak menemukan adanya <i>error</i> .	Berhasil

Tabel 4.5 Pengujian User 5

Nama Pengguna : Darmin			
Web Browser : Google Chrome			
Aksi	Yang Diharapkan	Hasil Yang di dapatkan	Kesimpulan
Membuka halaman website	Menampilkan halaman website	Setelah melakukan pengujian <i>user</i> tidak menemukan adanya <i>error</i> pada website	Berhasil
Menekan menu diagnosa	Mampu melihat, menginput, mengubah, dan menghapus data penyakit pada hewan.	<i>User</i> melakukan pengujian dengan memasukkan diagnosa dan tidak menemukan adanya <i>error</i> , semua fungsi berjalan dengan baik.	Berhasil
Fitur konsultasi	Mampu menghubungkan <i>user</i> dengan dokter hewan dengan API Whatsap	<i>User</i> tidak menemukan adanya <i>error</i> .	Berhasil

Dari hasil pengujian yang telah dilakukan oleh masing-masing *user* dengan berbagai web browser aplikasi dapat berjalan dengan baik, tidak terdapat *error* dan mudah dala penggunaannya.

4.1.2.8. Validasi Sistem Pakar

Tabel 4.6 Validasi Pakar

Daftar Penyakit	Daftar Gejala	Hasil Pakar	Hasil sistem	Kesimpulan	Pengujian
Bovine Ephemeral Feve	Kaku pada otot-otot tubuh, Kebingungan, Kehilangan nafsu makan	p	p	sesuai	TF
Kecelakaan	Buta, Luka berdarah, Pincang,	P	P	Sesuai	TF
keguguran	Keguguran, Keguguran Muda, Lahir mati	P	P	sesuai	TF
Eutokia	Lahir mati, Lahir normal	P	P	Sesuai	TE

Distokia	Kesulitan dalam proses persalinan, Keterlambatan dalam proses persalinan	P	N	Tidak Sesuai	NF
Urolothiasis	Kesulitan kencing	P	P	Sesuai	TF
Stomatitis	Anorexia, Demam, Liur berdarah	P	P	Sesuai	TF
Radang mata	Borok Perkulitan, Bulu Rontok, Daun Telinga Keropeng	p	N	Tidak sesuai	NF
Masitis	Penurunan berat badan, Kehilangan nafsu makan	P	N	Tidak Sesuai	NF
Fraktur	Patah tulang kaki	P	P	Sesuai	TF
Endometritis	Rahim bernanah	P	P	Sesuai	TF
Konjunctivitas	Buta, Kornea Mata Keruh	P	P	Sesuai	TF

Keterangan :

- P : Positif
- N : *Negative*
- TF : *True false*
- NF : *None False*

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan hasil dari penelitian yang telah dilakukan ini dapat disimpulkan bahwa Sistem Pakar Diagnosa Penyakit Pada Sapi ini merupakan solusi inovatif untuk membantu masyarakat dalam mendeteksi penyakit pada sapi berdasarkan gejala-gejala yang diinputkan. Penerapan Metode *Rational Unified Process* (RUP) dalam pengembangan sistem ini memberikan beberapa keuntungan dan karakteristik yang dapat meningkatkan keandalan dan efisiensi sistem. Selain itu, Sistem Pakar Diagnosa Penyakit pada Sapi ini memiliki potensi untuk memberikan dampak positif yang signifikan bagi kesehatan sapi dan kemudahan akses informasi bagi masyarakat peternak.

5.2. Saran

Saran yang dapat diusulkan untuk pengembangan Sistem Pakar Diagnosa Penyakit pada Sapi menggunakan Metode *Rational Unified Process* (RUP) mencakup melibatkan aktif pemangku kepentingan atau *stackholder* terkait, melakukan analisis domain medis yang lebih mendalam bersama dokter hewan, dan mengadopsi pendekatan iteratif untuk penyempurnaan berbasis umpan balik pengguna. Penting juga untuk memfokuskan perhatian pada keamanan data dan privasi, memberikan pelatihan pengguna yang memadai, serta merencanakan pemeliharaan dan pembaruan sistem secara berkala. Sistem yang terbuka, dokumentasi komprehensif, evaluasi dan peningkatan terus-menerus, serta responsivitas terhadap perkembangan teknologi juga menjadi poin kunci untuk meningkatkan kualitas dan keberlanjutan sistem ini dalam memberikan layanan di bidang kesehatan hewan.

DAFTAR PUSTAKA

- [1] P. E. Wardani, Y. Siagian, and M. Ihsan, “Sistem Pakar Untuk Diagnosa Penyakit Sapi Menggunakan Metode Bayes,” *Build. Informatics, Technol. Sci.*, vol. 4, no. 2, pp. 413–421, 2022, doi: 10.47065/bits.v4i2.2197.
- [2] M. D. Irawan, A. Widarma, and Y. H. Siregar, “Universitas Islam Negeri Sumatera Utara, Indonesia 1 Jl. Williem Iskandar Pasar V Medan Estate 20371, Indonesia 1 Teknik Informatika, Fakultas Teknik,” *Jl. Jend. Ahmad Yani*, vol. 11, 2121, doi: 10.34010/jati.v11i1.
- [3] I. B. Rozaq, R. Gupitha, I. Komputer, U. Subang, and S. Informasi, “SISTEM INFORMASI DIAGNOSA PENYAKIT HEWAN TERNAK Abstrak Pendahuluan Kajian Teori,” vol. 9, no. 1, pp. 22–35, 2022.
- [4] N. Anggraini, R. F. Fahlevie Afidh, M. dan Dosen, P. Teknik Informatika, S. Dumai, and J. Utama Karya Bukit Batrem Kota Dumai, “Sistem Pakar Diagnosa Penyakit Sapi Menggunakan Metode CBR Dan Algoritma Similarity Sorgenfrei,” *J. Eng. Technol. Innov. (JETI) Februari*, vol. 2, no. 1, pp. 1–10, 2023.
- [5] S. Muharni and S. Andriyanto, “Sistem Diagnosa Penyakit Jantung Berbasis Case Based Reasoning (CBR),” *Pros. Semin. Nas. Darmajaya*, vol. 1, pp. 1–11, 2021, [Online]. Available: <https://jurnal.darmajaya.ac.id/index.php/PSND/article/view/2910>
- [6] T. A. Kinaswara, N. R. Hidayati, and F. Nugrahanti, “Rancang Bangun Aplikasi Inventaris Berbasis Website Pada Kelurahan Bantengan | Kinaswara | Prosiding Seminar Nasional Teknologi Informasi dan Komunikasi (SENATIK),” *Pros. Semin. Nas. Teknol. Inf. dan Komun.*, vol. 2, no. 1, pp. 71–75, 2019, [Online]. Available: <http://prosiding.unipma.ac.id/index.php/SENATIK/article/view/1073>
- [7] K. Nistrina and L. Sahidah, “Unified Modelling Language (Uml) Untuk Perancangan Sistem Informasi Penerimaan Siswa Baru Di Smk Marga Insan

- Kamil,” *J. Sist. Informasi, J-SIKA*, vol. 4, no. 1, pp. 17–23, 2022.
- [8] K. 'Afiifah, Z. F. Azzahra, and A. D. Anggoro, “Analisis Teknik Entity-Relationship Diagram dalam Perancangan Database Sebuah Literature Review,” *Intech*, vol. 3, no. 2, pp. 18–22, 2022, doi: 10.54895/intech.v3i2.1682.
- [9] A. H. Manullang, M. Aritonang, and M. J. Purba, “Sistem Informasi Bimbingan Belajar Number One Medan Berbasis Web,” *TAMIKA J. Tugas Akhir Manaj. Inform. Komputerisasi Akunt.*, vol. 1, no. 1, pp. 44–49, 2021, doi: 10.46880/tamika.vol1no1.pp44-49.
- [10] I. R. Perpustakaan and U. Airlangga, *Self-Acceptance*. 2014. doi: 10.1007/978-94-007-0753-5_103694.
- [11] Kani, “Modul Pengantar Algoritma dan Pemrograman,” *Algoritm. dan Bhs. Pemrograman*, vol. 1, pp. 1–36, 2020, [Online]. Available: <https://pustaka.ut.ac.id/lib/wp-content/uploads/pdfmk/MSIM4203-M1.pdf>
- [12] Putri Choirunisa, “Implementasi Artificial Intelligence Untuk Memprediksi Harga Penjualan Rumah Menggunakan Metode Random Forest Dan Flask,” *UII.ac.id*, pp. 1–101, 2020.
- [13] D. S. D. R. RD Indrawan, “Rancang Bangun Sistem Penerimaan Santri Baru Berbasis Web Menggunakan Codeigniter 4 pada Pondok Pesantren Bilal Bin Rabah,” *Repository.UsahidSolo.Ac.Id*, pp. 8–23, 2021.
- [14] L. Yahya and M. Mulyati, “Rancang Bangun Sistem Informasi Menggunakan Metode Rational Unified Process (Rup) Pada Salon Mobil Scuto,” *Klik - J. Ilmu Komput.*, vol. 3, no. 2, pp. 101–106, 2022, doi: 10.56869/klik.v3i2.383.
- [15] A. C. Praniffa, A. Syahri, F. Sandes, U. Fariha, Q. A. Giansyah, and M. L. Hamzah, “Pengujian Black Box Dan White Box Sistem Informasi Parkir Berbasis Web Black Box and White Box Testing of Web-Based Parking Information System,” *J. Test. dan Implementasi Sist. Inf.*, vol. 1, no. 1, pp. 1–16, 2023.

LAMPIRAN

Lampiran 1. Penyakit Hewan Ternak Sapi

Penyakit	Gejala	Penanganan	Risiko
<i>Abses</i>	<i>Abses</i>	1.000 unit dengan <i>Gusanex</i> dan 8.000 ml dengan <i>Dexatozoon</i> Injeksi dan 8.000 ml dengan <i>B-Sanplex</i> dan 8.000 ml dengan <i>Vit-Oxy LA</i>	Penyebaran infeksi, Gangguan kesehatan, Kematian, Penurunan produktivitas, Kesulitan reproduksi, Biaya perawatan, Penularan ke hewan lain
<i>Avitaminosis</i>	<i>Anemia</i>	1.000 <i>Bolus</i> dengan <i>Pyronil-2500 Bolus</i> dan 8.000 ml dengan <i>Vitol-140</i> , 1000 dengan <i>Pyronil-@2500</i> bolus dengan <i>B-Sanplex</i> , 1001 dengan <i>Pyronil-@2500</i> bolus dengan <i>B-Sanplex</i> , 8.000 ml dengan <i>Vitol</i> , 1.000 bolus dengan <i>Pyronil-2500 Bolus</i> dan 10.000 ml dengan <i>Vitol-140</i> dan 8.000 ml dengan <i>Biodin1003</i> dengan <i>Pyronil-@2500</i> bolus dengan <i>B-Sanplex</i> 2.000 ml dengan <i>B-Sanplex-140</i> , 8.000 ml dengan <i>B-Sanplex</i> dan 1.000 bolus dengan <i>Pyronil-2500 Bolus</i> , 1.000 <i>Bolus</i> dengan <i>Pyronil-2500 Bolus</i> dan 10.000 ml dengan <i>Vitol-140</i> dan 8.000 ml dengan <i>Biodin</i>	Kekurangan vitamin (tidak terlalu beresiko)
	<i>Anorexia</i>		
	Bulu kusam		
	Demam		
	Diare		
	Gangguan otot		
	Gangguan pencernaan		
	Gangguan penglihatan		
	Gangguan pertumbuhan pada anak hewan		
	Gangguan pertumbuhan pada anak-anak		
	Gangguan reproduksi		
	Gangguan sistem kekebalan tubuh		
	Gangguan sistem saraf		
	Gangguan tulang		
	Keguguran pada betina		
	Kehilangan nafsu makan		
	Kejang		
	Kekurusan		
	Lelah		
	Lemah		
	Kerontokan rambut		
	Kesulitan dalam perkawinan		
	Masalah gigi dan gusi seperti berdarah atau gusi yang bengkak		
Masalah kulit seperti kering, bersisik, dan keriput			
Masalah pernapasan			
Penglihatan kabur			
Perubahan perilaku			
Pincang			
Sakit otot			

	Sakit sendi		
	Sembelit		
Babesiosis	Bulu kusam	8000 gram dengan <i>Tryponil</i> ; @3000 ml dengan <i>Intrafer-200 B12</i> , 1000 gram dengan <i>Super-P</i> ; @8000 ml dengan <i>Vitol-140</i> ; dan @1000 ml dengan <i>Medoxy LA</i>	Penurunan kesehatan, Kematian, Penurunan produktivitas, Penularan, Biaya perawatan, Pembatasan perdagangan, Gangguan reproduksi
	Gatal		
	Kelainan kulit		
	Perkulitan		
Bovine Ephemeral Fever	<i>Anorexia</i>	10.000 ml dengan <i>B-Sanplex</i> dan 1.000 bolus dengan <i>Pyronil-2500 Bolus</i> , 8.000 ml dengan <i>Vet-Oxy LA</i> dan 8.000 ml dengan <i>Sulpidon Injeksi</i> dan 8.000 ml dengan <i>B-Sanplex</i> , 10.000 ml dengan <i>Sulpidon</i> , 8001 ml dengan <i>B-Sanplex</i> ; @8000 ml dengan <i>Sulpido Injeksi</i> dan @8000 ml dengan <i>Vet-Oxy LA</i> 8.000 ml dengan <i>B-Sanplex</i> dan 10.000 ml dengan <i>Sulpidon Injeksi</i> dan 8.000 ml dengan <i>Vet-Oxy LA Injeksi</i> dan 8.000 ml dengan <i>Vet-Oxy LA</i> dan 8.000 ml dengan <i>B-Sanplex</i> , 8001 ml dengan <i>B-Sanplex</i> ; @8000 ml dengan <i>Sulpido Injeksi</i> dan @8000 ml dengan <i>Vet-Oxy LA</i>	Bisa menurunkan produksi daging
	Demam		
	Demam		
	Demam		
	Diare, Durasi gejala biasanya 3-4 hari		
	Gemetar		
	Kaku pada otot-otot tubuh		
	Kebingungan		
	Kehilangan nafsu makan		
	Lemah		
	Kelesuan		
	Kemungkinan terjadinya kejang		
	Kesadaran menurun		
	Kesulitan berdiri		
	Kesulitan bergerak		
	Muntah		
	Pembengkakan pada sendi-sendi kaki		
	Penurunan produksi susu pada sapi perah		
	Penurunan suhu tubuh setelah demam		
	Pincang		
	Produksi air liur meningkat		
	Sakit otot		
	Sakit sendi		
Sempoyongan			
Terjadi kejang-kejang			
Tremor			
Corpus Luteum Persisten	<i>Anestrus</i>	5000 ml dengan <i>Caprigladin Inj</i> ; @10000 ml dengan <i>Vitol-140</i>	Gangguan reproduksi
Distokia	Gelisah	1.000 unit dengan <i>Traksi</i> dan 10.000 ml dengan <i>Penstrep-400</i> dan 8.000 ml dengan <i>Biodin</i> , 3.000 ml dengan <i>Biodin</i> dan 1.000 unit dengan 2.000 ml dengan <i>Vet-Oxy LA</i> dan 1.000 ml dengan <i>Oxytocin</i>	Komplikasi persalinan sapi
	Janin terlilit tali pusar		
	Keluar cairan atau darah dalam jumlah yang tidak normal dari vulva		
	Kepala atau anggota badan janin terlihat tidak normal atau terdistorsi		

	Kesulitan dalam proses persalinan Keterlambatan dalam proses persalinan Lama waktu persalinan yang tidak normal Pendarahan atau keluarnya cairan dari vulva Posisi atau presentasi janin yang tidak benar (seperti kepala tersendat) Posisi atau presentasi janin yang tidak benar (misalnya, kaki belakang) Sapi kesulitan untuk melahirkan Sapi menunjukkan tanda-tanda kesakitan yang tidak biasa Sapi merintih Sapi tampak gelisah Sapi tampak tidak nyaman		
Endometritis	Rahim bernanah	10000 ml dengan <i>Penstrep-400</i> ; @10000 ml dengan <i>Vitol-140</i>	Gangguan kesehatan sapi
Enteritis	Berat badan turun secara signifikan Darah dalam tinja (terkadang) Dehidrasi Demam Diare Diare Gatal atau terasa seperti ada benda asing di mata Gejala flu seperti pilek atau sakit tenggorokan (pada beberapa kasus) Penurunan berat badan yang signifikan Kehilangan nafsu makan Lelah Keluarnya sekresi atau mata berair Kram perut atau nyeri perut Kurangnya urin Lemas Mata cekung Mata merah Mata iritasi Mencret Mengedipkan mata lebih sering dari biasanya Mual Mulut kering Muntah	2.000 ml dengan <i>Sulfa Strong</i> dan 2.000 ml dengan <i>Vitol-140</i> , 10.000 ml dengan <i>Sulfa Strong</i> dan 10.000 ml dengan <i>Biodin</i> , 10.000 ml dengan <i>Sulfa Strong</i> dan 8.000 ml dengan <i>Vitol-140</i> , 2.000 ml dengan <i>Sulfa Strong</i> dan 2.000 ml dengan <i>B-Sanplex</i>	Gangguan pencernaan sapi

	Pembengkakan pada kelopak mata		
	Penglihatan kabur		
	Penurunan produksi susu pada hewan ternak (jika berlaku)		
	Perubahan perilaku		
	Perut kembung		
	Produksi lendir atau nanah pada mata (terutama jika infeksi bakteri)		
	Rasa nyeri atau ketidaknyamanan di mata		
	Sensasi panas di mata		
	Sensitif terhadap cahaya (<i>fotofobia</i>)		
	Suhu tubuh yang meningkat		
<i>Eutokia</i>	Lahir mati	10.000 ml dengan <i>Sulfa Strong</i> dan 10.000 ml dengan <i>Biodin</i> , 1000 bolus dengan <i>Lagantor BSA Bolus</i> ; @ 1000 ml dengan <i>Vitol-140</i> , 8000 ml dengan <i>Vitol-140</i> ; @ 1000 ml dengan <i>Pyronil-2500 Bolus</i>	Persalinan sapi
	Lahir normal		
	Tanda lain		
<i>Fasciolosis</i>	Bulu kusam	8000 ml dengan <i>Vitol-140</i> ; @ 10000 ml dengan <i>Abenol-100 Oral</i>	Infeksi parasit pada sapi
	Kekurusan		
	Rahang bawah bengkak		
<i>Fraktur</i>	Patah tulang kaki	0.500 ml dengan <i>Dexatozoom Injeksi</i> ; @0.500 ml dengan <i>Penstrep-400</i>	Cedera tulang sapi

Untuk dataset yang digunakan lebih lengkap diberikan pada github:

<https://github.com/Dims56/Dataset>

Lampiran 2. Kode Program *Dashboard*

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Sapi Sehat</title>
    <link rel="icon" type="image/x-icon" href="{{ url_for('static',
filename='img/Logo.png') }}" />
    <!-- CSS CDN AOS -->
    <link href="https://unpkg.com/aos@2.3.1/dist/aos.css" rel="stylesheet" />
    <!-- CSS CDN Bootstrap -->
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
rbsA2VBKQhggwzxH7pPCaAqO46MgnOM80zW1RWuH61DGLwZJEdK2K
adq2F9CUG65" crossorigin="anonymous" />
    <!-- Icon CDN Bootstrap -->
    <link href="https://cdn.jsdelivr.net/npm/bootstrap-
icons@1.11.1/font/bootstrap-icons.css" rel="stylesheet" />
    <!-- Animate.css -->
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/animate.css/4.1.1/animate.min.css"
/>
    <!-- My CSS -->
    <link rel="stylesheet" href="{{ url_for('static', filename='css/style.css') }}"
/>
  </head>
  <body class="animate__animated animate__fadeIn">
    <!-- Navbar Start -->
    <nav class="navbar navbar-expand-lg bg-light fixed-top">
      <div class="container">
        <a class="navbar-brand-landing" href="/">Sapi Sehat</a>
        <button class="navbar-toggler" type="button" data-bs-toggle="collapse"
data-bs-target="#navbarSupportedContent" aria-
controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle
navigation">
          <span class="navbar-toggler-icon"></span>
        </button>
        <div class="collapse navbar-collapse" id="navbarSupportedContent">
          <ul class="navbar-nav mx-auto mb-2 mb-lg-0">
            <li class="nav-item">
              <a class="nav-link" href="#Beranda">Beranda</a>
            </li>
            <li class="nav-item">
              <a class="nav-link" href="#Penyakit">Penyakit</a>
            </li>
          </ul>
        </div>
      </div>
    </nav>
  </body>
</html>
```

```

        </li>
    </ul>
    <form class="form-inline my-3 my-lg-0">
        <a href="{{ url_for('login') }}" class="btn btn-primer me-2">Masuk</a>
        <a href="{{ url_for('register') }}" class="btn btn-sekunder">Daftar</a>
    </form>
</div>
</div>
</nav>
<!-- Navbar End -->

<!-- Hero Section Start -->
<div id="Beranda" class="container col-xxl-10 py-5">
    <div class="row">
        <div class="col-lg-6">
            <div class="container text-start pt-5 mt-5">
                <h1 class="judul">
                    Sapi Sehat : Algoritma
                    <h1 class="judul-algoritma pb-2">Random Forest</h1>
                </h1>
                <p class="lead hero-center border-top pt-4">Sapi Sehat, solusi cerdas
                untuk mendiagnosa penyakit sapi. Kami menggunakan algoritma Random
                Forest untuk memberikan informasi akurat tentang kesehatan sapi Anda.</p>
            </div>
        </div>
        <div class="col-lg-6 mt-3 justify-content-center text-center">
            
        </div>
    </div>
</div>
<!-- Hero Section End -->

<!-- Penyakit Section Start -->
<div class="container pt-5 pb-5" id="Penyakit">
    <div class="content text-center border-top mb-5">
        <div class="row">
            <div class="col-md-12 pt-5">
                <div class="titlepage">
                    <h1 class="header-penyakit pb-2">Penyakit</h1>
                    <p class="lead hero-center">Terdapat 500 penyakit yang dapat di
                    diagnosa oleh sistem kami dengan teknologi kecerdasan buatan yaitu Algoritma
                    Random Forest</p>
                </div>
            </div>
        </div>
    </div>
</div>

```

```
<div class="container justify-content-center penyakit-list">
  <p>Abses</p>
  <p>Avitaminosis</p>
  <p>Babesiosis</p>
  <p>Bovine Ephemeral Fever</p>
  <p>Corpus Luteum Persisten</p>
  <p>Distokia</p>
  <p>Endometritis</p>
  <p>Enteritis</p>
  <p>Eutokia</p>
  <p>Fasciolosis</p>
  <p>Fraktur</p>
  <p>Gingivitis</p>
  <p>Hernia</p>
  <p>Hipofungsi Ovari</p>
  <p>Indigesti</p>
  <p>Kawin Berulang</p>
  <p>Kecelakaan</p>
  <p>Keguguran</p>
  <p>Kekurangan Calsium</p>
  <p>Keracunan</p>
  <p>Konjunktivitas</p>
  <p>Malnutrisi</p>
  <p>Mastisis</p>
  <p>Miasis</p>
  <p>Orf</p>
  <p>Otitis Interna</p>
  <p>Papilloma</p>
  <p>Pneumonia</p>
  <p>Prolapsus Dubur</p>
  <p>Q fever</p>
  <p>Radang Mata</p>
  <p>Retensio Secundarium</p>
  <p>Scabies</p>
  <p>Stomatitis</p>
  <p>Surra</p>
  <p>Tympany</p>
  <p>Urolothiasis</p>
</div>
</div>
</div>
</div>
<!-- Penyakit Section End -->

<!-- WhatsApp Floating Button Start -->
```

```

<div class="whatsapp-float">
  <a href="https://api.whatsapp.com/send?phone=+6281269849594"
target="_blank">
    <i class="bi bi-whatsapp"></i>
  </a>
</div>
<!-- WhatsApp Floating Button End -->

<!-- Footer Start -->
<footer class="border-top bg-white p-3">
  <div class="container">
    <p class="text-center pt-3">© 2023 Skripsi - Dimas Malik Suryanda</p>
  </div>
</footer>
<!-- Footer End -->

<!-- My JavaScript -->
<script src="{{ url_for('static', filename='js/script.js') }}"></script>
<!-- JavaScript CDN Bootstrap -->
<script
src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.11.6/dist/umd/popper.min.
js" integrity="sha384-
oBqDVMmMz9ATKxIep9tiCxS/Z9fNfEXiDAYTujMAeBAsjFuCZSmKbSSUn
Qlmh/jp3" crossorigin="anonymous"></script>
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/js/bootstrap.min.js"
integrity="sha384-
cuYeSxntonz0PPNlHhBs68uyIAVpIIOZZ5JqeqvYYIcEL727kskC66kF92t6Xl
2V" crossorigin="anonymous"></script>
</body>
</html>

```

Lampiran 3. Kode Program Diagnosa

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>Sapi Sehat - Diagnosa</title>
  <link rel="icon" type="image/x-icon" href="{{ url_for('static',
filename='img/Logo.png') }}" />
  <!-- CSS CDN AOS -->
  <link href="https://unpkg.com/aos@2.3.1/dist/aos.css" rel="stylesheet" />
  <!-- CSS CDN Bootstrap -->

```

```

<link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
rbsA2VBKQhggwzxH7pPCaAqO46MgnOM80zW1RWuH61DGLwZJEdK2K
adq2F9CUG65" crossorigin="anonymous" />
<!-- Icon CDN Bootstrap -->
<link href="https://cdn.jsdelivr.net/npm/bootstrap-
icons@1.11.1/font/bootstrap-icons.css" rel="stylesheet" />
<!-- My CSS -->
<link rel="stylesheet" href="{{ url_for('static', filename='css/style.css') }}"
/>
</head>
<body>
<!-- Navbar Start -->
<nav class="navbar navbar-d navbar-expand-lg navbar-dark">
<div class="container">
<b class="navbar-brand" href="/dashboard">Sapi Sehat</b>
<button class="navbar-toggler" type="button" data-bs-toggle="collapse"
data-bs-target="#navbarSupportedContent" aria-
controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle
navigation">
<span class="navbar-toggler-icon"></span>
</button>
<div class="collapse navbar-collapse" id="navbarSupportedContent">
<ul class="navbar-nav ms-auto mb-2 mb-lg-0">
<li class="nav-item">
<a class="nav-link" href="/dashboard">Home</a>
</li>
<li class="nav-item">
<a class="nav-link active" href="/diagnosa">Diagnosa</a>
</li>
</ul>
<form action="{{ url_for('logout') }}" method="post" class="d-flex">
<button class="btn btn-danger" type="submit">Logout</button>
</form>
</div>
</div>
</nav>
<!-- Navbar End -->

<div class="container mt-5 mb-3">
<h1 class="mb-4 text-center">Diagnosa Penyakit Sapi</h1>
<form method="POST" action="/predict">
<div class="card p-3 shadow">
<label class="mb-2" for="symptoms">Masukkan 3 gejala atau
lebih:</label>

```


Lampiran 4. Kode Program Halaman *Form* Diagnosa

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Sapi Sehat - Diagnosa</title>
    <link rel="icon" type="image/x-icon" href="{{ url_for('static',
filename='img/Logo.png') }}" />
    <!-- CSS CDN AOS -->
    <link href="https://unpkg.com/aos@2.3.1/dist/aos.css" rel="stylesheet" />
    <!-- CSS CDN Bootstrap -->
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
rbsA2VBKQhggwzxH7pPCaAqO46MgnOM80zW1RWuH61DGLwZJEdK2K
adq2F9CUG65" crossorigin="anonymous" />
    <!-- Icon CDN Bootstrap -->
    <link href="https://cdn.jsdelivr.net/npm/bootstrap-
icons@1.11.1/font/bootstrap-icons.css" rel="stylesheet" />
    <!-- My CSS -->
    <link rel="stylesheet" href="{{ url_for('static', filename='css/style.css') }}"
/>
  </head>
  <body>
    <!-- Navbar Start -->
    <nav class="navbar navbar-d navbar-expand-lg navbar-dark">
      <div class="container">
        <b class="navbar-brand" href="/dashboard">Sapi Sehat</b>
        <button class="navbar-toggler" type="button" data-bs-toggle="collapse"
data-bs-target="#navbarSupportedContent" aria-
controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle
navigation">
          <span class="navbar-toggler-icon"></span>
        </button>
        <div class="collapse navbar-collapse" id="navbarSupportedContent">
          <ul class="navbar-nav ms-auto mb-2 mb-lg-0">
            <li class="nav-item">
              <a class="nav-link" href="/dashboard">Home</a>
            </li>
            <li class="nav-item">
              <a class="nav-link active" href="/diagnosa">Diagnosa</a>
            </li>
          </ul>
        </div>
      </div>
    </nav>
  </body>
</html>
```



```

</div>

<!-- JavaScript CDN Bootstrap -->
<script
src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.11.6/dist/umd/popper.min.
js" integrity="sha384-
oBqDVmMz9ATKxIep9tiCxs/Z9fNfEXiDAYTujMAeBAsjFuCZSmKbSSUn
Qlmh/jp3" crossorigin="anonymous"></script>
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/js/bootstrap.min.js"
integrity="sha384-
cuYeSxntonz0PPNlHhBs68uyIAVpIIOZZ5JqeqvYYIcEL727kskC66kF92t6Xl
2V" crossorigin="anonymous"></script>
</body>
</html>

```

Lampiran 5. Kode Program Hasil Diagnosa

```

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8" />
<meta name="viewport" content="width=device-width, initial-scale=1.0" />
<title>Sapi Sehat - Diagnosa</title>
<link rel="icon" type="image/x-icon" href="{{ url_for('static',
filename='img/Logo.png') }}" />
<!-- CSS CDN AOS -->
<link href="https://unpkg.com/aos@2.3.1/dist/aos.css" rel="stylesheet" />
<!-- CSS CDN Bootstrap -->
<link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
rbsA2VBKQhggwzxH7pPCaAqO46MgnOM80zW1RWuH61DGLwZJEdK2K
adq2F9CUG65" crossorigin="anonymous" />
<!-- Icon CDN Bootstrap -->
<link href="https://cdn.jsdelivr.net/npm/bootstrap-
icons@1.11.1/font/bootstrap-icons.css" rel="stylesheet" />
<!-- My CSS -->
<link rel="stylesheet" href="{{ url_for('static', filename='css/style.css') }}"
/>
</head>
<body>
<!-- Navbar Start -->
<nav class="navbar navbar-d navbar-expand-lg navbar-dark">
<div class="container">

```

```

    <b class="navbar-brand" href="/dashboard">Sapi Sehat</b>
    <button class="navbar-toggler" type="button" data-bs-toggle="collapse"
data-bs-target="#navbarSupportedContent" aria-
controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle
navigation">
    <span class="navbar-toggler-icon"></span>
</button>
<div class="collapse navbar-collapse" id="navbarSupportedContent">
    <ul class="navbar-nav ms-auto mb-2 mb-lg-0">
    <li class="nav-item">
    <a class="nav-link" href="/dashboard">Home</a>
    </li>
    <li class="nav-item">
    <a class="nav-link active" href="/diagnosa">Diagnosa</a>
    </li>
    </ul>
    <form action="{{ url_for('logout') }}" method="post" class="d-flex">
    <button class="btn btn-danger" type="submit">Logout</button>
    </form>
</div>
</div>
</nav>
<!-- Navbar End -->

<div class="container mt-5 mb-3">
    <h1 class="mb-4 text-center">Diagnosa Penyakit Sapi</h1>
    <form method="POST" action="/predict">
    <div class="card p-3 shadow">
    <label class="mb-2" for="symptoms">Masukkan 3 gejala atau
lebih:</label>
    <textarea class="form-control" id="symptoms" name="symptoms"
rows="4" cols="50" required></textarea><br />
    <input class="btn btn-primary" type="submit" value="Diagnosa" />
    </div>
</form>

{% if result %}
<div class="container">
    <h2 class="text-center pt-5 pb-3">Hasil Diagnosa:</h2>
    <div class="row justify-content-center">
    <div class="col-md-6">
    <div class="card shadow mb-5">
    <div class="card-body">
    <b>Penyakit:</b>
    <div class="alert alert-primary mt-2" role="alert">{{
result['Penyakit'] }}</div>

```

