KLASIFIKASI TINGKAT KEBERHASILAN PRODUKSI AYAM BROILER MENGGUNAKAN ALGORITMA SUPPORT VECTOR MACHINE (SVM)

SKRIPSI

DISUSUN OLEH

SITI AISAH PURBA

2109010017

PROGRAM STUDI SISTEM INFORMASI

FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI

UNIVERSITAS MUHAMMADIYAH SUMATERA UTARA

MEDAN

2025

KLASIFIKASI TINGKAT KEBERHASILAN PRODUKSI AYAM BROILER MENGGUNAKAN ALGORITMA SUPPORT VECTOR MACHINE (SVM)

SKRIPSI

Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer (S.Kom) dalam Program Studi Sistem Informasi pada Fakultas Ilmu Komputer dan Teknologi Informasi, Universitas Muhammadiyah Sumatera Utara

SITI AISAH PURBA
NPM. 2109010017

PROGRAM STUDI SISTEM INFORMASI
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS MUHAMMADIYAH SUMATERA UTARA
MEDAN
2025

LEMBAR PENGESAHAN

Judul Skripsi	:	Klasifikasi Tingkat Keberhasilan Produksi Ayam Broiler Menggunakan Algoritma Support Vector Machine (SVM)
Nama Mahasiswa	:	Siti Aisah Purba
NPM	:	2109010017
Program Studi	:	Sistem Informasi

Menyetujui
Komisi Pembimbing

(Dr. Al-Khowarizmi, S.kom., M.Kom)
NIDN. 0127099201

Ketua Program Studi

(Dr. Firahmi Rizky, S.Kom, M.Kom)
NIDN. 3048770671230343

Dekan

(Dr. Al-Khowarizmi, S.Kom., M.Kom.)
NIDN. 0127099201

PERNYATAAN ORISINALITAS

KLASIFIKASI TINGKAT KEBERHASILAN PRODUKSI AYAM BROILER MENGGUNAKAN ALGORITMA SUPPORT VECTOR MACHINE (SVM)

SKRIPSI

Saya menyatakan bahwa karya tulis ini adalah hasil karya sendiri, kecuali beberapa kutipan dan ringkasan yang masing-masing disebutkan sumbernya.

Medan, Juni 2025

Yang membuat pernyataan

METERAI TEMPEL

Siti Aisah Purba

NPM. 2109010017

PERNYATAAN PERSETUJUAN PUBLIKASI

KARYA ILMIAH UNTUK KEPENTINGAN AKADEMIS

Sebagai sivitas akademika Universitas Muhammadiyah Sumatera Utara, saya bertanda tangan dibawah ini:

Nama	: Siti Aisah Purba
NPM	2109010017
Program Studi	: Sistem Informasi
Karya Ilmiah	: Skripsi

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Muhammadiyah Sumatera Utara Hak Bedas Royalti Non-Eksekutif *(Non-Exclusive Royalty free Right)* atas penelitian skripsi saya yang berjudul:

KLASIFIKASI TINGKAT KEBERHASILAN PRODUKSI AYAM BROILER MENGGUNAKAN ALGORITMA SUPPORT VECTOR MACHINE (SVM)

Beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Non-Eksekutif ini, Universitas Muhammadiyah Sumatera Utara berhak menyimpan, mengalih media, memformat, mengelola dalam bentuk database, merawat dan mempublikasikan Skripsi saya ini tanpa meminta izin dari saya selama tetap mencantumkan nama saya sebagai penulis dan sebagai pemegang dan atau sebagai pemilik hak cipta.

Demikian pernyataan ini dibuat dengan sebenarnya.

Medan, Juni 2025

Yang membuat pernyataan

Siti Aisah Purba

NPM. 2109010017

RIWAYAT HIDUP

DATA PRIBADI

Nama Lengkap	: Siti Aisah Purba
Tempat dan Tanggal Lahir	: Bandar Masilam, 06 Mei 2003
Alamat Rumah	: Huta Banjar Sari
Telepon/Faks/HP	: 083876285446
E-mail	: icaoppo072@gmail.com
Instansi Tempat Kerja	: -
Alamat Kantor	: -

DATA PENDIDIKAN

SD	: SDN 095248	TAMAT: 2015
SMP	: MTSN Bandar Masilam	TAMAT: 2018
SMA	: SMK Al Washliyah 2 Perdagangan	TAMAT: 2021

KATA PENGANTAR

بسم الله الرحمن الرحيم

Alhamdulillah, segala puji dan rasa syukur penulis ucapkan kepda Allah SWT atas Rahmat dan nikmat insan, islam, kesehatan, kesejahteraan, kesabaran, kemudahan serta kesempatan kepada penulis sehingga dapat menyelesaikan penyusunan skripsi ini yang berjudul "Klasifikasi Tingkat Keberhasilan Produksi Ayam Broiler Menggunakan Algoritma Support Vector Machine (SVM)" dapat terselesaikan dengan baik. Shalawat serta salam senantiasa tercurahkan kepada Baginda Rasulullah yakni Nabi Besar Muhammad SAW sebagai Nabi dan Rasul yang diutus Allah SWT.

Penulis sangat menyadari bahwa skripsi ini masih jauh dari kata sempurna dan masih banyak kekurangan, hal ini dikarenakan keterbatasan kemampuan dan masih banyak kekurangan penulis. Untuk itu penulis mengharapkan kritik dan saran yang bersifat membangun dari semua pihak demi kesempurnaan skripsi ini.

Penyusunan skripsi ini tidak lepas dari dukungan dan bantuan berbagai pihak yang turut memotivasi dan memberikan dorongan semangat. Oleh karena itu, penulis ingin menyampaikan ucapan terimakasih yang tak terhingga kepada semua yang telah berperan, antara lain:

1. Bapak Prof. Dr. Agussani, M.AP selaku Rektor Universitas Muhammadiyah Sumatera Utara.
2. Bapak Dr. Al-Khowarizmi, S.kom., M.Kom selaku Dekan Fakultas Ilmu Komputer dan Teknologi Informasi, sekaligus dosen pembimbing penulis. Bimbingan dan masukan berharga yang diberikan oleh Bapak Al-

Khowarizmi telah memberikan kontribusi besar dalam menuntun penulis menyelesaikan skripsi ini dengan tepat waktu. Keberhasilan penulis ini tidak terlepas dari saran, arahan, dan bimbingan yang mendalam dari Bapak. Terima kasih sebesar-besarnya atas waktu, kesabaran, dan mempermudah setiap proses, selalu membuka pintu konsultasi dengan lapang. Semoga Allah SWT senantiasa melimpahkan kesehatan, keberkahan, serta balasan terbaik atas kebaikan yang Bapak berikan.

3. Bapak Halim Maulana, ST., M.Kom selaku Wakil Dekan 1 Fakultas Ilmu Komputer dan Teknologi Informasi.
4. Bapak Dr. Lutfi Basit, S.Sos., M.I.Kom selaku Wakil Dekan III Fakultas Ilmu Komputer dan Teknologi Informasi.
5. Ibu Dr. Firahmi Rizky, S.Kom, M.Kom selaku Ketua Program Studi Sistem Informasi Fakultas Ilmu Komputer dan Teknologi Informasi
6. Teristimewa untuk orang tua tercinta, cinta pertama dan panutan penulis yaitu Ayahanda Samiono Purba dan pintu surga saya Ibunda Nurjannah. Terimakasih banyak atas segala pengorbanan, dukungan, motivasi, nasihat serta do’a tulus kasih yang tidak pernah putus dipanjatkan dalam setiap sujudnya memohon ridho dari Sang pencipta agar setiap langkah anak-anaknya selalu di ridhoi dalam segala hal. Semoga kebaikan ini diganti dengan surganya Allah ya Ayah Ibu. Besar harapan penulis semoga Ayah Ibu sehat selalu, panjang umur dan bisa menyaksikan keberhasilan lainnya yang akan penulis raih di masa yang akan datang.
7. Adik laki-laki penulis, Al Afsan Nawawi Purba yang selalu membuat penulis termotivasi untuk bisa terus belajar menjadi sosok kakak yang

dapat memberikan pengaruh *positif*, baik dalam bidang akademik maupun *non*-akademik, serta berusaha menjadi panutannya dimasa yang akan datang.

8. Rizky Hidayat, seseorang yang selalu ada untuk penulis. terimakasih telah menemani setiap proses yang penulis lalui selama ini, memberikan dukungan tanpa henti, memberikan semangat dan selalu meyakinkan penulis bahwa penulis bisa mencapai impian-impiannya. semoga dilancarkan setiap niat baikmu, tetaplah menjadi sosok yang baik dan asik.
9. Terimakasih untuk sahabat penulis, Ulan Damanik, Hafizah Ayu Rizki Nasution, Puji Rahayu yang selalu menjadi pendengar yang baik untuk penulis serta memberikan semangat kepada penulis untuk segera menyelesaikan penulisan skripsi ini dan meyakinkan penulis bahwa segala masalah yang dihadapi selama proses skripsi akan berakhir.
10. Teman-Teman seperjuangan saya, Nazwa, Alifa, Jihan, Mutia, Agita, Nanda, Bintang, Rojak, Aryo, Anugrah, Rizki Perdana, Zaky, Haikal yang senantiasa mau diajak untuk berbagi cerita, mendukung, dan memberikan semangat. Semoga kebahagiaan terus bersama dengan kehidupan kalian.
11. Terakhir, penulis mengucapkan terimakasih kepada diri sendiri yang telah berusaha keras untuk meyakinkan dan menguatkan diri bahwa dapat menyelesaikan studi ini sampai selesai. Berbahagialah selalu dengan dirimu sendiri, rayakan kehadiranmu sebagai berkah dimanapun kamu menjejakkan kaki. Allah sudah merencanakan dan memberikan porsi terbaik untuk perjalanan hidupmu. Semoga langkah kebaikan selalu

menyertaimu, dan semoga Allah selalu meridhoi setiap langkah serta mengajakmu dalam lindungan-Nya, Aamiin.

KLASIFIKASI TINGKAT KEBERHASILAN PRODUKSI AYAM BROILER MENGGUNAKAN ALGORITMA SUPPORT VECTOR MACHINE (SVM)

ABSTRAK

Produksi ayam broiler merupakan salah satu sektor peternakan yang berkembang pesat di Indonesia. Penilaian tingkat keberhasilan produksi ayam broiler sangat penting bagi peternak dalam mengevaluasi performa dan mengoptimalkan hasil panen. Namun, proses evaluasi secara manual sering kali menimbulkan subjektivitas dan potensi kesalahan. Penelitian ini bertujuan untuk mengembangkan model klasifikasi tingkat keberhasilan produksi ayam broiler menggunakan algoritma *Support Vector Machine* berdasarkan tiga indikator utama, yaitu umur ayam, berat badan ayam, dan jumlah pakan. Penelitian ini juga mengimplementasikan model klasifikasi ke dalam bentuk aplikasi web interaktif menggunakan framework *Streamlit* yang dibangun dengan bahasa pemrograman Python. Data diperoleh dari peternakan Timan Farm dan melalui proses normalisasi serta pembagian data menjadi data latih dan data uji. Hasil dari pengujian model menunjukkan bahwa algoritma *Support Vector Machine* mampu mengklasifikasikan tingkat keberhasilan produksi dengan akurasi yang memadai. Aplikasi berbasis web yang dikembangkan dengan *Streamlit* memungkinkan pengguna untuk melakukan klasifikasi secara otomatis, cepat, dan akurat tanpa harus memiliki latar belakang teknis yang mendalam.

Kata Kunci: Ayam Broiler, *Support Vector Machine*, Klasifikasi, *Streamlit*, Tingkat Keberhasilan, Python.

CLASSIFICATION OF BROILER PRODUCTION SUCCESS RATE USING ALGORITHM SUPPORT VECTOR MACHINE (SVM)

ABSTRACT

Broiler production is one of the fastest growing livestock sectors in Indonesia. Assessing the success rate of broiler production is very important for farmers in evaluating performance and optimizing yields. However, the manual evaluation process often leads to subjectivity and potential errors. This study aims to develop a classification model of the success rate of broiler production using the Support Vector Machine algorithm based on three main indicators, namely chicken age, chicken weight, and feed amount. This research also implements the classification model into an interactive web application using the Streamlit framework built with the Python programming language. The data was obtained from Timan Farm and went through a normalization process and data division into training and test data. The results of model testing show that the Support Vector Machine algorithm is able to classify the success rate of production with sufficient accuracy. The web-based application developed with Streamlit allows users to perform classification automatically, quickly, and accurately without having to have an in-depth technical background.

Keywords: *Broiler Chicken, Support Vector Machine, Classification, Streamlit, Success Rate, Python.*

DAFTAR ISI

LEMBAR PENGESAHAN ..... Error! Bookmark not defined.

PERNYATAAN ORISINALITAS ..... Error! Bookmark not defined.

PERNYATAAN PERSETUJUAN PUBLIKASI ... Error! Bookmark not defined.

RIWAYAT HIDUP ..... **v**

KATA PENGANTAR ..... **vi**

ABSTRAK ..... **ix**

ABSTRACT ..... **x**

DAFTAR ISI ..... **xi**

DAFTAR TABEL ..... **xiv**

DAFTAR GAMBAR ..... **xv**

BAB I PENDAHULUAN **1**

1.1 Latar Belakang Masalah 1

1.2 Rumusan Masalah 3

1.3 Batasan Masalah 3

1.4 Tujuan Penelitian 4

1.5 Manfaat Penelitian 4

BAB II LANDASAN TEORI **5**

2.1 Klasifikasi 5

2.2 Ayam Broiler 5

2.3 *Data Mining* 6

2.3.1 Pengelompokkan *Data Mining* 7

2.3.2 Tahapan *Data Mining* 8
2.4 *Support Vector Machine* 10
2.5 *Machine Learning* 15
2.6 Python 17
2.7 *Google Collab* 19
2.8 *Streamlit* 20
2.9 *Database* 21
2.9.1 XAMPP 22
2.10 *Flowchart* 22
2.11 Penelitian Terdahulu 24
BAB III METODE PENELITIAN 28
3.1 Jenis Penelitian 28
3.2 Kerangka Berfikir 28
3.3 Analisis Kebutuhan 29
3.4 Teknik Pengumpulan Data 29
3.4.1 Studi Literatur 32
3.5 Proses Klasifikasi 32
3.5.1 *Preprocessing data* 33
3.5.2 *Data Training* dan *Data Testing* 34
3.5.3 Implementasi Model SVM 35
3.5.4 Evaluasi Model 35
BAB IV HASIL DAN PEMBAHASAN 37
4.1 Perhitungan Manual *Support Vector Machine* 37
4.1.1 Persiapan Data dan Normalisasi 37
4.1.2 Formulasi Masalah dan Asumsi *Support Vector* 39
4.1.3 Penyelesaian Sistem Persamaan Linear untuk Menemukan w dan b ... 40

4.1.4 Penentuan *Hyperplane* Optimal dan *Margin* 43
4.1.5 Verifikasi Klasifikasi Manual 44
4.2 Implementasi Model SVM dengan *Python* 46
4.2.1 Persiapan Data 46
4.2.2 Membangun Pipeline Preprocessing dan Model 51
4.2.3 Latih Model dengan Parameter Terbaik 53
4.2.4 Evaluasi Model pada Data Testing 56
4.3 Implementasi dan Hasil Berbasis Web (*Streamlit)* 57
4.3.1 Tampilan Form *Login* 57
4.3.2 Tampilan Dashboard 57
4.3.3 Tampilan *Form* Kelola Data Ayam 58
4.3.4 Tampilan *Form* Tambah Data Ayam 58
4.3.5 Tampilan *Form* Edit Data Ayam 59
4.3.6 Tampilan Pelatihan Model 60
4.3.7 Tampilan Lakukan Klasifikasi 65
4.3.8 Tampilan Riwayat Klasifikasi 66
4.3.9 Tampilan Kelola Pengguna 67
4.3.10 Tampilan Tambah Pengguna Baru 67
4.4 Pengujian Sistem Berbasis Web 68
4.5 Analisa Kelebihan dan Kelemahan Sistem 70
4.5.1 Kelebihan Sistem 70
4.5.2 Kelemahan Sistem 71
BAB V KESIMPULAN DAN SARAN 73
5.1 Kesimpulan 73
5.2 Saran 74
DAFTAR PUSAKA 76

DAFTAR TABEL

Tabel 2. 1 Simbol Flowchart 23

Tabel 2. 2 Penelitian Terdahulu 24

Tabel 3. 1 Data Ayam Broiler 30

Tabel 4. 1 Observasi Data Awal 37

Tabel 4. 2 Dataset Setelah Normalisasi 39

Tabel 4. 3 Tabel Statistik Deskriptif Dataset 48

Tabel 4. 4 Skenario Pengujian *Blackbox Testing* 68

DAFTAR GAMBAR

Gambar 2. 1 Tahapan-Tahapan Data Mining 9

Gambar 2. 2 Hyperplane di SVM yang terbagi atas dua kelas target 12

Gambar 2. 3 Tampilan Python 17

Gambar 2. 4 Contoh Tampilan Google Collab 20

Gambar 3. 1 Kerangka Berfikir 28

Gambar 3. 2 Flowchart Sistem Klasifikasi 33

Gambar 4. 1 Kode *Impor Library* Python 47

Gambar 4. 2 Kode *Load* dan Persiapan Data 47

Gambar 4. 3 Tampilan 5 Baris Pertama Dataset 48

Gambar 4. 4 Kode untuk memisahkan fitur dan label 49

Gambar 4. 5 Gambar 5 Baris Data Pertama Fitur X 50

Gambar 4. 6 Kode Pembagian Data Latih dan Data Uji 50

Gambar 4. 7 Pipeline Model SVM 52

Gambar 4. 8 Kode untuk Mencari Parameter terbaik 54

Gambar 4. 9 Tampilan *Form Login* 57

Gambar 4. 10 Tampilan Menu Dashboard 58

Gambar 4. 11 Tampilan *Form* Kelola Data Ayam 58

Gambar 4. 12 Tampilan Input Data Latih Manual 59

Gambar 4. 13 Tampilan Input Data Latih Otomatis 59

Gambar 4. 14 Tampilan *Form* Edit Data Ayam 60

Gambar 4. 15 Bagian Analisis dan Persiapan Data Latih 61

Gambar 4. 16 Heatmap Korelasi Fitur 61

Gambar 4. 17 Hasil Pelatihan dan Evaluasi Model..... 62

Gambar 4. 18 Hasil Laporan Klasifikasi 63

Gambar 4. 19 Tampilan Diagram ROC Curve..... 64

Gambar 4. 20 Tampilan Menu Klasifikasi 66

Gambar 4. 21Tampilan Riwayat Klasifikasi 67

Gambar 4. 22 Tampilan Kelola Pengguna 67

Gambar 4. 23 Tampilan Tambah Pengguna Baru 68

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Indonesia merupakan salah satu negara yang melakukan pertenakan ayam broiler. Usaha ayam broiler di Indonesia sekarang ini telah menjadi potensi yang tinggi untuk pelaku usaha dan peternak ayam. Peternakan ayam broiler adalah usaha yang memiliki potensi besar untuk dikembangkan menjadi bisnis yang besar dan menguntungkan (Hanief & Purnama, 2024). Ayam broiler ini dilahirkan melalui perkawinan silang, seleksi dan rekayasa genetik yang dilakukan oleh pembibitnya (Gustian et al., 2019). Perkembangan ayam broiler semakin hari semakin meningkat secara signifikan. Namun besarnya biaya produksi yang dibutuhkan didalam usaha peternakan ayam broiler mendorong peternak mencari cara untuk bagaimana meningkatkan hasil produksi ayam broiler (Rahmatillah et al., 2024).

Ayam broiler memiliki masa pertumbuhan yang relatif singkat dan efisiensi konversi pakan yang tinggi, sehingga menjadi pilihan utama dalam industri peternakan komersial. Namun, dalam proses produksinya, terdapat beberapa faktor yang mempengaruhi keberhasilan produksi ayam broiler meliputi bibit, pakan, manajemen atau pengelolaan, dan faktor genetik. Ketidakseimbangan dalam salah satu aspek tersebut dapat berdampak langsung terhadap tingkat keberhasilan produksi, baik dari segi kuantitas maupun kualitas hasil panen.

Timan Farm merupakan kemitraan dari PT Charoen Pokphand Indonesia, Dalam penilaian tingkat keberhasilan produksinya peternak sering kali hanya mengandalkan pengalaman dan pengamatan manual. Tingkat keberhasilan ayam

broiler ini dapat diukur dari indikator seperti berat badan ayam, jumlah pakan dan umur ayam. Namun, penilaian secara manual dapat menimbulkan kesalahan sehingga keputusan yang diambil bersifat subjektif dan kurang akurat. Maka dari itu, evaluasi keberhasilan ayam broiler menjadi aspek yang penting untuk meningkatkan efisiensi peternakan.

Untuk mengatasi permasalahan tersebut, diperlukannya pendekatan yang lebih objektif dan sistematis. Salah satu pendekatan yang dapat diterapkan adalah melalui teknik klasifikasi dalam data mining. Klasifikasi adalah suatu pekerjaan yang menilai objek data untuk memasukkannya ke dalam kelas tertentu dari sejumlah kelas yang tersedia (Putro et al., 2020). Klasifikasi pertama kali diterapkan pada bidang tanaman yang mengklasifikasi suatu spesies tertentu, seperti yang telah dilakukan oleh Carolus von Linne yang pertama kali mengklasifikasi spesies berdasarkan karakteristik fisik (Mardi, 2017).

Data mining adalah sebuah proses yang menggunakan teknik statistik, matematika, kecerdasan buatan, dan machine learning untuk mengekstraksi dan mengidentifikasi informasi yang bermanfaat dan pengetahuan yang terkait dari berbagai database besar. *Data Mining* ini tujuan utamanya adalah untuk menemukan, menggali, atau menambang pengetahuan dari data atau informasi yang dimiliki atau pola yang mungkin memberikan indikasi yang bermanfaat (Laila Sari et al., 2022).

Metode-metode yang terdapat dalam data mining untuk proses klasifikasi salah satunya yaitu *Algoritma Support Vector Machine* (SVM). SVM bekerja dengan mencari *hyperplane* optimal yang memisahkan data kedalam kelas-kelas tertentu, sehingga dapat membantu dalam mengidentifikasi pola keberhasilan

produksi ayam broiler berdasarkan indikator- indikator yang tersedia. SVM juga memiliki keunggulan dalam mengatasi *overfitting*, terutama ketika jumlah fitur lebih besar dibandingkan jumlah sampel.

Berdasarkan latar belakang yang telah diuraikan diatas maka penulis akan mencoba melakukan klasifikasi tingkat keberhasilan produksi ayam broiler dengan menggunakan algoritma SVM untuk mengembangkan model klasifikasi tingkat keberhasilan produksi ayam broiler yang diimplementasikan dalam bahasa pemrograman Python. maka penulis membuat penelitian berjudul **"Klasifikasi Tingkat Keberhasilan Produksi Ayam Broiler Menggunakan Algoritma Support Vector Machine (SVM)"**.

1.2 Rumusan Masalah

Berdasarkan penjelasan pada latar belakang di atas, berikut adalah rumusan masalah yang dapat dianalisis pada penelitian ini ialah, bagaimana cara mengklasifikasikan tingkat keberhasilan produksi ayam broiler di Timan Farm berdasarkan indikator serta untuk mengetahui bagaimana penerapan algoritma SVM dalam membangun model klasifikasi tersebut dan mengukur seberapa akurat model klasifikasi yang dihasilkan dalam klasifikasi tingkat keberhasilan produksi ayam broiler.

1.3 Batasan Masalah

Dalam penelitian ini terdapat batasan masalah yang mencakup beberapa aspek yang perlu diperhatikan untuk memastikan kejelasan dan keterbatasan dalam penelitian ini. Pertama, penelitian ini menggunakan algoritma SVM untuk klasifikasi dan memodelkan tingkat keberhasilan ayam broiler. Penelitian ini secara khusus tidak akan mencakup metode-metode klasifikasi atau model

pemodelan lainnya, dan pembatasan ini diberlakukan untuk memberikan kejelasan fokus dan konsistensi analisis terhadap Algoritma SVM sebagai pendekatan utama. Sumber data yang digunakan dalam penelitian ini hanya dilakukan pada peternakan Timan Farm.

1.4 Tujuan Penelitian

Penelitian ini memiliki tujuan utama untuk mengembangkan model klasifikasi yang dapat digunakan untuk menilai tingkat keberhasilan produksi ayam broiler di Timan Farm berdasarkan indikator yang telah ditentukan. Selain itu, penelitian ini bertujuan untuk menerapkan algoritma SVM dalam membangun sistem klasifikasi, serta mengevaluasi tingkat akurasi model yang dihasilkan dalam klasifikasi tingkat keberhasilan produksi ayam broiler. Model yang dikembangkan akan diimplementasikan dalam bahasa pemrograman Python untuk menghasilkan klasifikasi yang lebih akurat dan dapat diandalkan dalam mengoptimalkan produksi ayam broiler.

1.5 Manfaat Penelitian

Penelitian ini diharapkan dapat memberikan manfaat bagi peternak Timan Farm dalam melakukan evaluasi kinerja peternak mitra secara akurat dan objektif. Dengan adanya model klasifikasi yang dibangun menggunakan algoritma SVM, proses penilaian tingkat keberhasilan produksi ayam broiler dapat dilakukan secara otomatis dan sistematis, agar mengurangi potensi kesalahan yang mungkin terjadi dalam penilaian manual sehingga dapat meningkatkan efisiensi dan kualitas produksi.

BAB II

LANDASAN TEORI

2.1 Klasifikasi

Klasifikasi adalah sebuah proses mencari suatu himpunan model (fungsi) yang dapat mendeskripsikan dan membedakan kelas-kelas data atau konsep-konsep, dengan tujuan dapat menggunakan model tersebut untuk memprediksi kelas dari suatu objek yang mana kelasnya belum diketahui (Romli & Zy, 2020). Tujuan klasifikasi yaitu untuk memprediksi kategori dari objek tertentu atau data yang label nilainya belum terindentifikasi Dalam klasifikasi, sistem akan belajar dari data yang telah diberi label (*training data*) untuk kemudian dapat memprediksi label dari data baru yang belum diketahui (*testing data*).

2.2 Ayam Broiler

Ayam broiler merupakan salah satu jenis ayam ras yang dapat menghasilkan daging. Ayam Broiler mempunyai masa pertumbuhan yang cepat sehingga dalam waktu 4-5 minggu sudah bisa dipanen. Daging yang dihasilkan dari ayam broiler sangat empuk dan banyak disukai masyarakat (T, 2019). Produk dari ayam broiler ini mempunyai peran penting sebagai sumber protein hewani yang harganya bisa dibilang relatif murah. Ayam broiler membutuhkan pemeliharaan yang baik agar dapat mencapai produksi yang optimal.

Keberhasilan dari memproduksi ayam broiler diekspresikan dalam performans atau penampilan ayam broiler yang dapat diukur melalui mortalitas, umur ayam, konsumsi pakan, bobot badan akhir, rasio konversi pakan (FCR), dan indeks performans (IP) (Anto et al., 2025). Faktor yang dapat menentukan tingkat

keberhasilan dalam usaha peternakan ayam broiler yaitu manajemen kandang, kesehatan ayam dan pakan. (Gusdiansyah et al., 2024).

Keunggulan dari ayam broiler yaitu pertumbuhannya sangat cepat dengan bobot badan yang tinggi dalam waktu yang relatif pendek, konversi pakan kecil, siap dipotong pada usia muda serta menghasilkan kualitas daging berserat lunak (Subowo Edy, 2020). Perkembangan yang pesat dari ayam broiler juga merupakan upaya penanganan untuk mengimbangi kebutuhan masyarakat terhadap daging ayam. Namun yang menjadi penghambat didalam membudidayakan adalah mudahnya terserang penyakit pada Ayam Broiler (Patappari et al., 2024).

2.3 *Data Mining*

Penambangan data (*data mining*) adalah istilah yang digunakan untuk menggambarkan penemuan pengetahuan dalam database (Maulana & Al-Khowarizmi, 2022). Tingkatan didalam data mining bermanfaat untuk mencari pola-pola tertentu dari data yang ada pada *database* (Lukman & Herlinda, 2024). *Data Mining* banyak dipergunakan untuk melakukan pengolahan pada sekumpulan data, sehingga hasil pengelolaan tersebut dapat membantu para pengambil keputusan untuk membuat sebuah keputusan terbaik (Nas, 2021).

Data mining secara umum dikenal dengan *nama Knowledge Discovery in Database (KDD)* sehingga hasil dari data mining dapat dipergunakan untuk memperbaiki pengambilan keputusan dimasa depan. Adapun proses yang dilakukan oleh *data mining* yaitu meliputi deskripsi, prediksi, estimasi, klasifikasi, *clushtering* dan asosiasi. Dalam data mining prediksi dan klasifikasi

digunakan untuk menganalisis data yang dapat memvisualkan kelas data atau untuk memprediksi data di masa depan (Ucha Putri et al., 2021).

Menurut (Pangestu & Ridwan, 2022) Teknik yang terdapat didalam data mining, antara lain yaitu database, machine learning, statistik, dan visualisasi. Data yang dapat dipergunakan yaitu data relasi, transaksi, multimedia, web, dan text.

2.3.1 Pengelompokkan *Data Mining*

Data mining dibagi menjadi 6 kelompok yang dapat digunakan untuk menemukan, mengambil dan menambah pengetahuan dari sebuah data (Noperia et al., 2023), Adapun penjabarannya yaitu sebagai berikut:

a. Deskripsi

Deskripsi adalah menganalisis secara sederhana dalam mencoba mencari cara untuk bagaimana menggambarkan pola yang terdapat didalam data yang akan digunakan.

b. Estimasi

Estimasi hampir sama dengan klasifikasi, Namun variabel target dari estimasi lebih kearah numerik dari pada kearah kategori. Estimasi adalah perhitungan yang berbentuk angka. Model estimasi dibangun dengan menggunakan record lengkap yang menyediakan nilai dari variabel prediksi.

c. Prediksi

Prediksi hampir sama dengan estimasi dan klasifikasi yang membedakan yaitu prediksi nilai dari hasil akan ada dimasa mendatang. Terdapat beberapa metode dan teknik yang digunakan oleh estimasi dan klasifikasi bisa digunakan untuk prediksi jika keadaan yang tepat.

d. Klasifikasi

Klasifikasi hampir sama dengan estimasi dan prediksi, dalam klasifikasi terdapat target variable kategori. Sebagai contohnya dalam penggolongan pendapatan dapat dibagikan menjadi tiga kategori, yaitu pendapatan tinggi, pendapatan sedang, dan pendapatan rendah.

e. Pengklusteran

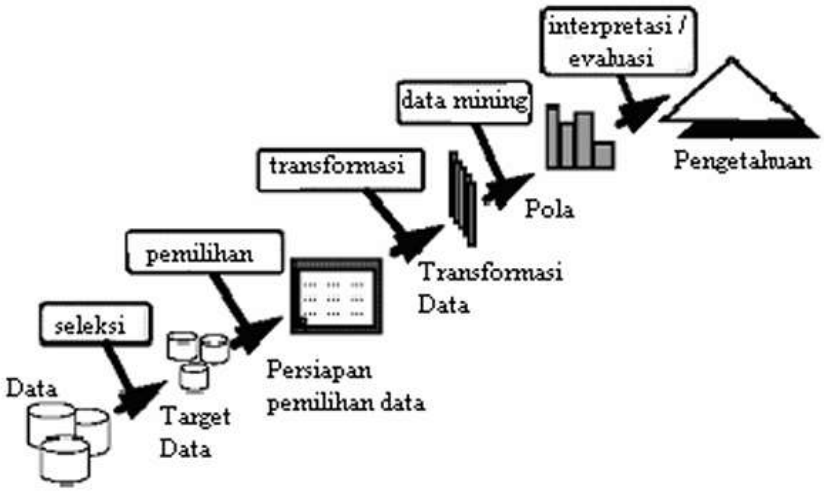
Pengklusteran dipergunakan untuk pengelompokan data, pengamatan atau menganalisis dan membentuk kelas objek-objek yang memiliki ketidakmiripan dengan record-record didalam kluster lain. Pengklusteran tidak untuk melakukan klasifikasi, mengestimasi, atau memprediksi nilai dari variabel target. Namun, algoritma pengklusteran ini mencoba melaksanakan pembagian atas keseluruhan data menjadi kelompok-kelompok yang memiliki kemiripan (homogeny), dimana kemiripan dalam satu kelompok akan bernilai maksimal, sedangkan kemiripan dengan record dalam kelompok lain akan bernilai minimal.

f. Asosiasi

Asosiasi adalah menemui atribut yang muncul dalam satu waktu untuk menandai suatu kejadian dari sebuah proses. Tugas dari asosiasi dalam *data mining* yaitu menemukan atribut yang muncul didalam satu waktu.

2.3.2 Tahapan *Data Mining*

Beberapa tahapan-tahapan yang perlu dilakukan pada *data mining*, (Wafa et al., 2022) :



Gambar 2. 1 Tahapan-Tahapan *Data Mining*

Sumber: *(Wafa et al., 2022)*

a. Seleksi data, tahapan pertama yaitu seleksi pemilihan data sebelum tahap penggalian informasi dimulai. Data yang terpilih kemudian dipisahkan dari data asli untuk digunakan dalam analisis selanjutnya. Hasil dari data yang diseleksi digunakan kedalam proses data mining.

b. Pre-processing, didalam tahapan Pre-Processing terdapat 3 step diantaranya yaitu data cleaning, menghilangkan data-data yang tidak lengkap kemudian data selection, pengambilan data yang sesuai dengan tugas analisis dan data intergation, proses penggabungan beberapa data sumber data. Proses ini bertujuan untuk menghilangkan duplikasi dan inkonsistensi, sehingga akan menghasilkan dataset yang lebih berkualitas dan bisa diandalkan untuk tahap selanjutnya.

c. Transformasi data, Tahap ini adalah langkah di mana data yang sudah terpilih akan dimodifikasi ke dalam format yang sesuai untuk melakukan pengolahan selanjutnya. Data yang sudah terpilih, akan ditransformasikan

ke dalam bentuk yang sesuai untuk prosedur penggalian lebih lanjut dengan cara melakukan proses normalisasi dan agregasi.

d. Menambang data, proses mencari pola atau informasi pada data yang sudah terpilih menggunakan metode-metode tertentu.

e. Evaluasi, pada tahapan ini menampilkan pola informasi yang telah diolah pada tahap data mining kedalam bentuk yang mudah dipahami dan dimengerti.

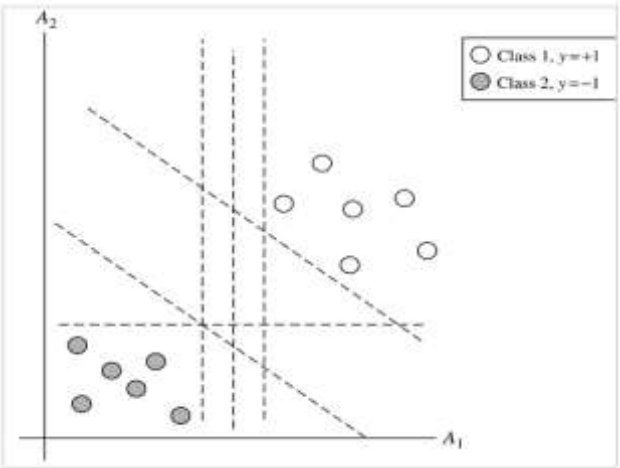
2.4 *Support Vector Machine*

Support Vector Machine (SVM) adalah sebuah algoritma dalam machine learning yang digunakan untuk klasifikasi dan regresi. Algoritma SVM bekerja dengan cara menemukan *hyperplane* (bidang pemisah) yang dapat memisahkan dua set data atau kelas yang berbeda secara optimal (Vebriani & Yustanti, 2024). SVM memiliki kelebihan yaitu mampu mengidentifikasi *hyperplane* terpisah yang memaksimalkan margin antara dua kelas yang berbeda. Namun SVM memiliki kekurangan terhadap masalah pemilihan parameter atau fitur yang sesuai. Pemilihan fitur sekaligus penyetingan parameter di SVM secara signifikan mempengaruh hasil akurasi klasifikasi (Irawan et al., 2021).

SVM merupakan algoritma berbeda dengan algoritma lainnya seperti misalnya *Neural Network* (NN), dimana SVM menggunakan data yang akan terpilih saja untuk berkontribusi pada model. Data-data yang akan berkontribusi pada model disebut dengan *support vector*. SVM memiliki kernel yang digunakan untuk mentransformasikan inputan ke bentuk yang diperlukan (Sitorus et al., 2020). Cara kerja metode SVM adalah dengan cara memisahkan beberapa kelompok data dengan garis. Garis ini disebut

hyperplane, dengan teknik SVM ini bertujuan guna mencari hyperplane yang optimal. Tingkat dari keakurasian metode SVM ini ditetapkan oleh fungsi kernel dan parameter yang dipakai. Kernel merupakan fungsi yang dipergunakan untuk memperoleh hyperplane yang optimum. Penggunaan dari kernel yang dapat dilaksanakan dalam percobaan untuk menghasilkan keakuratan yang terbaik dalam proses prediksi. SVM ini secara teoritis dan teknis, bekerja atas dasar prinsip *Structural Risk Minimization* (SRM), yaitu digunakan untuk menemukan *hyperlane* yang paling baik. Hal ini sesuai dengan prinsip dasar dari SVM, yaitu dengan memaksimalkan margin minimum antar class, fungsi kernel, dan *support vector*.

Prinsip dasar dari SVM lainnya adalah *linear classifier*, yaitu kasus klasifikasi yang dapat dipisahkan secara linear, SVM bisa juga memproses *problem non-linear* dengan cara mengoptimalkan konsep kernel pada ruang fitur yang memiliki nilai dimensi tinggi dengan memakai teknik trick_kernel. Keuntungan trick_kernel yaitu dapat mengetahui kernel mana yang tepat dalam meningkatkan performa. SVM bertujuan untuk mencari *hyperplane* pemisah yang optimal untuk memisahkan observasi yang memiliki nilai variabel target yang berbeda. *Hyperplane* ini bisa berupa garis pada dimensi dua dan datar pada dimensi lebih tinggi (Wahyudi et al., 2024). *Hyperplane* adalah batas keputusan yang memisahkan tipe dari satu kelas dengan kelas yang lain. SVM menemukan *hyperplane* menggunakan vektor pendukung dan margin (D. S. Permana & Silvanie, 2021).



Gambar 2. 2 *Hyperplane* di SVM yang terbagi atas dua kelas target

Sumber: *(D. S. Permana & Silvanie, 2021)*

Dalam algoritma SVM ada trik kernel dimana ada SVM linear dan SVM nonlinear. Dimana SVM adalah hyperlane linear yang bekerja hanya pada data yang hanya dapat dipisahkan dengan cara linear. SVM nonlinear yaitu data yang berdistribusi pada kelas yang tidak linear sering digunakan pendekatan kernel pada fitur awal data set. Dimana kernel dapat diartikan sebagai suatu fungsi yang memetakan fitur data yang memiliki dimensi awal rendah linear (Kurniawan et al., 2023).

Menurut (Nugroho et al., 2021) *Support Vector Machine* (SVM) terbagi menjadi 2 jenis yaitu:

1. SVM Linear

   SVM Linear merupakan suatu pilihan yang tepat untuk data perhitungan karena dapat dipisahkan dengan garis lurus atau *hyperplane*. SVM memisah kelas data menggunakan garis pemisah atau *hyperplane*, secara umum dinyatakan seperti persamaan 2.1

$$w.x + b = 0 \tag{2.1}$$

Hyperplane didalam SVM terdiri dari satu garis pemisah utama dan dua garis pendukung. Dengan adanya dua garis pendukung ini akan membuat SVM lebih baik dibanding dengan algoritma lainnya dalam memisahkan kelas. Cara untuk mendapatkan garis pemisah pada SVM hal pertama yang harus dilakukan yaitu mencari dua garis pendukung antara dua kelas menggunakan persamaan 2.2 dan 2.3

$$x_i.w + b \geq 1, jika\ y_i = 1 \tag{2.2}$$

$$x_i.w + b \leq -1, jika\ y_i = -1 \tag{2.3}$$

dimana b yaitu besaran skalar, w adalah normal vector, dan xi merupakan vektor data yang berhubungan dengan garis pendukung disebut dengan *support vector*. Garis pemisah optimum didapatkan dengan memaksimalkan margin antara dua garis pendukung dengan menyelesaikan masalah optimasi menggunakan persamaan 2.4 dan 2.5, yaitu dengan mencari titik minimal dari persamaan 2.2 dan 3.3

$$\min(w) = \frac{1}{2} \parallel w \parallel^2 \tag{2.4}$$

dengan batasan

$$y_i(x_i.w + b) - 1 \geq 0 \tag{2.5}$$

Masalah optimasi pada persamaan 2.4 dan 2.5 dapat diselesaikan menggunakan *Lagrange multiplier*. Sesudah masalah optimasi berhasil diselesaikan maka kelas data x_i dapat ditentukan berdasarkan persamaan 2.6

$$f(x_d) = \sum_{i=1}^{ns} a_i\ y_i x_i x_d + b \tag{2.6}$$

Dimana x_i merupakan *support vector*, *ns*=urutan dari *support vector*, x_d yaitu data yang akan diklasifikasi dan a_i adalah Lagrange multiplier.

2. Non-linear SVM

Non-Linear SVM ini memakai fungsi kernel non-linier sebagai memisahkan kelas didalam ruang fitur yang tidak linear. Non-Linear SVM efektif untuk data yang tidak bisa dipisahkan secara linier.

Pada persamaan SVM Linear data bisa dipergunakan jika terpisah secara linear. Permasalahan di dunia nyata, kelompok data pada umumnya terpisah secara non-linear. Untuk dapat menyelesaikan permasalahan non linear maka dapat menggunakan kernel trick dimana kelompok data dipetakan pada ruang dimensi yang lebih tinggi dengan menggunakan persamaan 2.7

$$\Phi: R^P \rightarrow R^q \text{ dengan } p > q \tag{2.7}$$

sehingga garis pemisah/hyperplane dapat dibentuk dalam ruang vektor yang baru. Namun untuk menentukan formula transformasi Φ cukup sulit, Maka dari itu perhitungan dot product dapat digantikan dengan fungsi kernel K $(x_i x_j)$ yang mendefinisikan fungsi transformasi Φ tersebut dengan kernel trick yang ditunjukkan dalam persamaan 2.8

$$K(x_i, x_j) = \Phi_i(x_i).\Phi_j(x_j) \tag{2.8}$$

Sebelum menggunakan kernel trick perlu mengetahui fungsi kernel yang dipakai agar dapat menentukan *support vector*. Pada umumnya terdapat 4 jenis fungsi kernel yang bisa digunakan yaitu :

1) Kernel linear, persamaan 2.9

$$K(x_{,}x_k) = x_K^T x \tag{2.9}$$

2) Kernel polynominal, persamaan 2.10

$$K\left(x_{,}x_k\right) = (x_K^T + 1)^d \tag{2.10}$$

3) Kernel sigmoid, persamaan 2.11

$$K\left(x_{,}x_k\right) = \tanh\ [k\ x_K^T x + \Phi] \tag{2.11}$$

4) Kernel radial basis function, persamaan 2.12

$$K\left(x_{,}x_k\right) = \exp\{-\| x - x_k \|_2^2/\sigma^2\} \tag{2.12}$$

Setelah menentukan kernel mana yang ingin digunakan, maka proses klasifikasi kernel dapat dilakukan dengan menggunakan persamaan 2.13

$$f\left(\Phi(x)\right) = w.\Phi(x) + b$$

$$= \sum_i^{ns} = 1_{,}x_{i\ \varepsilon}\, SV\ a_i y_i \Phi(x).\Phi(x_i) + b \tag{2.13}$$

$$= \sum_i^{ns} = 1_{,}x_{i\varepsilon} SV a_i y_i K(X, X_i) + b$$

dimana SV merupakan data pada kelompok data latih yang terpilih sebagai *support vector*.

2.5 *Machine Learning*

Machine learning adalah salah satu cabang dari kecerdasan buatan dan untuk pelaksanaan dapat menggunakan berbagai bahasa pemrograman (Sholeh et al., 2022). *Machine learning* berfokus pada pengembangan sistem yang memiliki kesanggupan untuk belajar dan membuat keputusan secara mandiri, tanpa perlu diprogram ulang oleh manusia (Pratama et al., 2023).

Cara kerja dari *machine learning* sangat bervariasi tergantung pada teknik atau metode pembelajaran yang dipegunakan dalam *Machine Learning*. Tetapi, secara umum, prinsip dasar pembelajaran mesin tetap sama tidak ada yang berbeda, yang melibatkan tahapan pengumpulan data, eksplorasi data, pemilihan model atau teknik, pelatihan model yang dipilih, dan evaluasi hasil dari *Machine learning*.

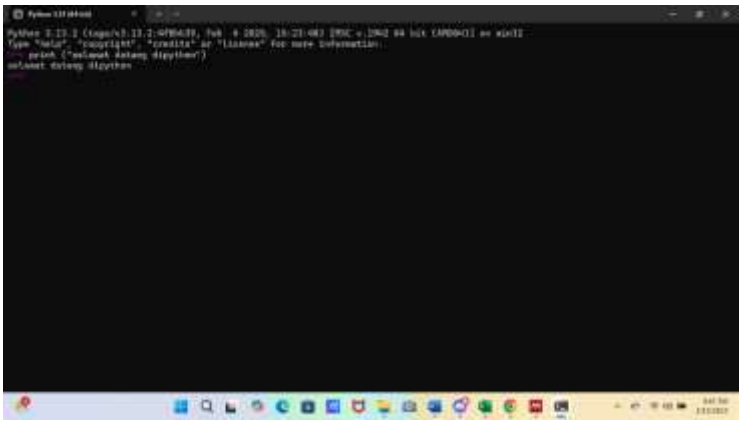
Secara garis besar *Machine Learning* di bagi menjadi 3 yaitu, (Sidik & Ansawarman, 2022):

a. *Supervised Learning* adalah sebuah algoritma *machine learning* yang proses pembelajarannya dibawah pengawasan. Yang termasuk didalam supervised learning diantaranya yaitu *classification* dan *regression*.
b. *Unsupervised Learning* adalah algoritma *machine learning* yang memproses pembelajarannya tanpa adanya pengawasan. Yang termasuk didalam unsupervised learning diantaranya yaitu *clustering* dan *dimensionality reduction*.
c. *Reinforcement Learning* adalah algoritma *machine learning* yang dapat membuat agent software mesin bekerja secara otomatis untuk menentukan perilaku ideal sehingga bisa memaksimalkan kinerja dari sebuah algoritma.

Machine learning mempunyai beberapa algoritma untuk memprediksi suatu kejadian, *yaitu Logistic Regression, K-Nearest Neighbors, Naïve-Bayes, Support Vector Machine, Neural Network, Random Forest* (Zein & Gunawan, 2022). *Machine learning* merupakan salah satu ilmu yang harus dipelajari untuk sekarang, karena *machine learning* ini adalah sistem memang kita butuhkan. Dengan memanfaatkan data, algoritma, dan model, *Machine learning* bisa membantu dalam pengambilan keputusan yang lebih cepat dan akurat. Penggunaan dari *machine learning* sudah cukup banyak dipergunakan dalam dunia industri, pertanian, dunia pendidikan dan lain sebagainya (Telaumbanua et al., 2020).

2.6 Python

Python merupakan bahasa pemrograman tingkat tinggi yang yang dibuat oleh Guido Van Rossum dan dirilis pada tahun 1991 Python juga merupakan bahasa yanng sangat populer belakangan ini. Selain itu python juga merupakan bahasa pemrograman yang multi fungsi contohnya python dapat digunakan untuk *Machine Learning* dan *Deep Larning* (Alfarizi et al., 2023).



Gambar 2. 3 Tampilan Python

Python memiliki *high-level* struktur data, *dynamic typing* dan *dynamic binding*. Python memiliki sintaks sederhana dan mudah dipelajari untuk penekanan pada kemudahan membaca dan mengurangi biaya perbaikan program. Python mendukung modul dan paket untuk mendorong kemodularan program dan *code reuse*. Interpreter Python dan standard library-nya tersedia secara gratis untuk semua *platform* dan dapat secara bebas disebarkan (Rochmawati, 2024). Python umumnya digunakan sebagi bahasa script meski pada praktiknya penggunaan bahasa ini lebih luas mencakup konteks

pemanfaatan yang umumnya dilakukan dengan bahasa script (A. F. A. Rahman et al., 2020).

Dalam analisis data python dapat melakukan kalkulasi statistik, menganalisa data maupun visualisasi data. Sedangkan pada *machine learning* python dapat digunakan untuk membuat algoritma untuk modul (B. A. C. Permana et al., 2024). Beberapa kelebihan dan kekurangan dari bahasa pemrograman python diantara lain yaitu, (Ginting et al., 2020) :

a. Pengembangan program dilakukan dengan cepat dan coding yang lebih sedikit.
b. Mendukung multi platform.
c. Memiliki sistem pengelolaan memori yang otomatis.
d. Python bersifat *Object Oriented Programming* (OOD).

Adapun beberapa kekurangan dari Bahasa pemrograman python diantara lain yaitu :

1. Python lambat dalam Eksekusi karena Python adalah interpreted language, maka dari itu lebih lambat dibandingkan bahasa yang dikompilasi seperti C++ atau Java.
2. Python mengonsumsi Memori Tinggi, Python menggunakan *dynamic typing* dan *garbage collection*, sehingga sangat memakan banyak memori.
3. Python tidak optimal untuk Aplikasi Mobile, Python tidak banyak digunakan dalam pengembangan aplikasi Android/iOS karena performanya pada python kurang baik.

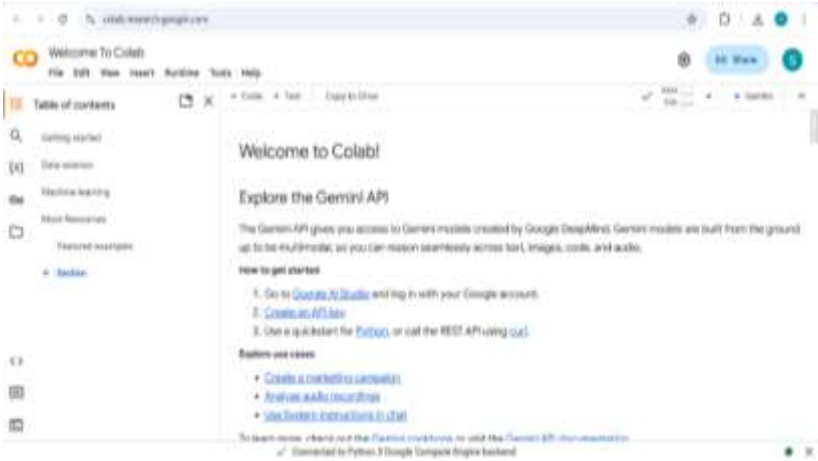
4. Python *Multi-threading* Terbatas, *Global Interpreter Lock* (GIL) membatasi eksekusi thread, membuat Python kurang efisien dalam komputasi paralel.
5. Python kurang Cocok untuk keamanan tinggi, Python sangat mudah di-*reverse engineer*, sehingga kurang bagus untuk aplikasi seperti perbankan atau militer.
6. Python tidak selalu kompatibel antara versinya, Perbedaan antara Python 2 dan Python 3 menyebabkan kode lama sering tidak bisa kompatibel.
7. Python kurang efisien dalam *High-Performance Computing*, Python tidak cocok untuk aplikasi yang memerlukan kecepatan tinggi, seperti game atau pemrosesan real-time.

Dalam konteks pemodelan prediktif, bahasa pemrograman Python telah menjadi salah satu pilihan utama bagi para praktisi dan peneliti. Python menyediakan berbagai library dan framework yang kuat, seperti scikit-learn, TensorFlow, dan Keras, yang memungkinkan implementasi yang efisien dari algoritma-algoritma supervised learning (Mestika et al., 2023).

2.7 *Google Collab*

Google Colab adalah IDE untuk pemrograman Python yang prosenya dilakukan oleh server Google dengan perangkat keras yang kuat. Dalam hal perangkat lunak, *Google Colab* menyediakan hampir semua perpustakaan yang butuhkan oleh penggunanya (Trisnawati et al., 2024) Keuntungan terbesar dari menggunakan *Google Colab* adalah bahwa *Google Colab* mempunyai kumpulan

built-in-library machine learning paling populer yang dapat dimuat dengan mudah dalam *notebook*.



Gambar 2. 4 Contoh Tampilan *Google Collab*

Google Colab menyediakan lingkungan pengembangan berbasis cloud yang memungkinkan akses mudah dan penggunaan sumber daya komputasi yang kuat tanpa memerlukan konfigurasi perangkat keras secara langsung. Dengan integrasi yang mulus dengan layanan *Google Drive* serta infrastruktur yang mendukung, *Google Colab* telah menjadi pilihan utama bagi banyak praktisi Python (Nazar, 2024).

2.8 *Streamlit*

Streamlit adalah sebuah kerangka kerja web yang ditujukan untuk menyebarkan model dan visualisasi dengan mudah menggunakan bahasa Python, yang cepat dan minimalis namun memiliki tampilan yang cukup baik serta ramah pengguna (Widi Hastomo et al., 2022). Tersedia widget bawaan untuk masukan pengguna, seperti pengunggahan gambar, penggeser, masukan teks, dan elemen *hypertext markup language* (HTML) lain yang sudah dikenal, seperti checkboxes dan radio buttons. Setiap kali pengguna berinteraksi dengan aplikasi Streamlit, skrip Python akan dijalankan kembali dari atas ke bawah. Hal tersebut merupakan

suatu konsep penting yang perlu diingat saat mempertimbangkan berbagai status aplikasi yang akan dipilih.

Streamlit merupakan aplikasi yang tidak berbayar dan pengguna tidak perlu memiliki pengetahuan pengembangan *front-end* yang mahir untuk mengoperasikannya. *Streamlit* dapat dijalankan pada editor Anaconda serta bahasa Python seri 3.7 ke atas, namun tidak mendukung pada editor *Jupyter Notebook*, sehingga harus dikonversi ke editor *Pycharm* atau *Visual Code*. Tampilan beranda yang ada pada aplikasi *Streamlit* dapat dipisahkan menjadi dua bagian, yaitu buttons, untuk pemilihan menunya, serta tampilan *visual chart*. Hal ini menyebabkan dibutuhkannya *library NumPy* serta Pandas untuk menampilkan grafik.

Salah satu kelebihan dari *streamlit* adalah developer tidak perlu mengatur tampilan website dengan CSS, HTML dan javascript karena *streamlit* telah menyediakannya melalui fungsi-fungsi yang terdapat pada framework tersebut (Putranto et al., 2023).

2.9 *Database*

Database merupakan suatu kesatuan yang dibentuk dari gabungan tabel dan file, di mana setiap tabel terdiri dari record yang disusun atas field-field yang ada di dalamnya (Wulandari & Nurmiati, 2022). *Database* menawarkan keuntungan penyimpanan data dengan format yang independen dan fleksibel. Hal ini dikarenakan *database* didefinisikan secara terpisah dari program aplikasi yang menggunakan *database* dan lingkup *database* dapat dikembangkan tanpa berdampak pada program-program yang menggunakan database tersebut (Eyni

Alfia & Waseso, 2020). Beberapa *database* yang ada saat ini adalah : Mysql, Sql Server, Ms.Access, Oracle, dan PostgreSql (Ramadhan & Mukhaiyar, 2020).

Adapun beberapa fungsi dari database yaitu:

1. Mempermudah identifikasi data dengan cara pengelompokkan data, salah satu contoh nya dengan pembuatan beberapa tabel atau field yang berbeda-beda.
2. Meminimalisir suatu data ganda.
3. Mempermudah penggunaan user dalam berbagai hal, misalnya pada saat peng-inputan data baru.
4. Penyimpanan secara digital
5. menjadi alternatif lain terkait masalah penyimpanan ruang dalam suatu aplikasi.

2.9.1 XAMPP

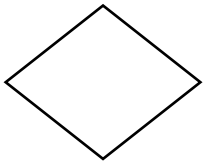
XAMPP adalah sebuah perangkat lunak bebas yang mendukung banyak sistem operasi, merupakan kompilasi dari beberapa program. XAMPP memiliki tool yang menyediakan paket perangkat lunak ke dalam satu buah paket. XAMPP merupakan salah satu paket installasi Apache, PHP, dan MySQL instant yang bisa digunakan untuk membantu proses instalasi ketiga produk tersebut (Sitanggang Rianto et al., 2022). XAMPP memiliki keunggulan yaitu mudah dioperasikan, tidak memerlukan banyak biaya serta mendukung instalasi pada Windows dan linux. (Putra & Nita, 2019).

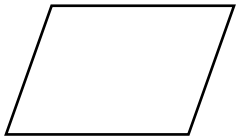
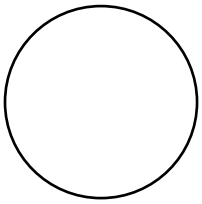
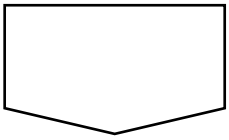
2.10 *Flowchart*

Flowchart adalah diagram yang menampilkan suatu langkah-langkah dan Keputusan didalam melakukan sebuah proses dari suatu program. Fungsi dari

flowchart yaitu penyederhanaan rangkaian prosedur untuk memudahkan pemahaman terhadap suatu informasi. Simbol *flowchart* dapat dilihat dari tabel dibawah ini:

Tabel 2. 1 Simbol Flowchart

No	Simbol Flowchart	Nama	Fungsi
1.		Terminator	Digunakan untuk menunjukkan bagian awal dan akhir dari suatu proses.
2		Process	Digunakan untuk mewakilkan suatu proses
3		Document	Menunjukkan dokumen input maupun output, baik untuk proses manual, mekanik, maupun komputer
4		decision	Digunakan untuk menyeleksi kondisi dalam program.

5		Data	Digunakan untuk mewakilkan data input/output
		On-Page Reference/Connector	Menunjukkan penghubung ke halaman yang masih sama dengan halaman yang lain
6		Off-Page Reference/Off-Page Connector	Penghubung alur dalam halaman berbeda
7		Flow	Arah alur dalam konsep (prosedur)

Sumber:(L. Rahman, 2019)

2.11 Penelitian Terdahulu

Berikut adalah table penelitian terdahulu yang mendukung kerangka teoritis penelitian ini.

Tabel 2. 2 Penelitian Terdahulu

No	Nama Peneliti	Judul Penelitian	Hasil Penelitian

1.	(Kurniawan et al., 2023)	Prediksi Hasil Panen Pertanian Salak di Daerah Tapanuli Selatan Menggunakan Algoritma SVM (Support Vector Machine)	Hasil dari penelitian yang dilakukan menunjukan bahwa algoritma SVM dapatkan nilai RMSE (*Root Mean Square Error*) sebesar 1,49071 dari perhitungan manual dengan 11 data training, 9 data testing dan dengan tingkat akurasi akurasi 44%. Pada model prediksi yang dibuat di dapatkan hasil RMSE (*Root Mean Square Error*) sebesar 1,49 dengan akurasi 44%. Dari hasil yang di dapatkan dalam perhitungan manual maupun dalam model prediksi yang dibuat keduanya sama-sama mendapatkan hasil RMSE (*Root Mean Square Error*) sebesar 1,49 dengan tingkat akurasi 44%

2.	(Basuki, 2023)	Klasifikasi Tingkat Keberhasilan Produksi Ayam Broiler Di Riau Menggunakan Algoritma K Nearest Neighbor	Hasil klasifikasi tingkat keberhasilan produksi ayam broiler menurut metode *K-Nearest Neighbor* (KNN) di Riau. Terdapat perbedaan dalam akurasi hasil tes yang dilakukan. Nilai akurasi tertinggi didapat k = 3, dengan nilai akurasi 86,49%, presisi sebesar 75,00% dan recall sebesar 70,21%. Dari percobaan yang dilakukan diketahui bahwa nilai k yang digunakan mempengaruhi hasil akhir yang diperoleh.
3.	(Cahyaningtyas et al., 2022)	Analisis Perbandingan Algoritma dan Feature Selection Untuk Klasifikasi Hasil Panen Ayam Broiler	Pada penelitian ini telah dilakukan klasifikasi data hasil panen ayam broiler dengan menggunakan teknik *data mining*. Hasil dari membandingkan 3 algoritma klasifikasi didapatkan algoritma yang terbaik untuk mengklasifikasikan data

			hasil panen ayam broiler adalah algoritma *Random Forest* dengan nilai akurasi sebanyak 89,14%. Sedangkan metode terbaik untuk meningkatkan hasil performa algoritma adalah Metode Backward dengan Peningkatan performa Metode tersebut sebesar 7,53%. Sehingga nilai Akurasi yang didapat dari Algoritma *Random Forest* Dan *Backward Elimination* Menjadi 96,67%. Selain itu Dapat diketahui atribut yang Berpengaruh dalam Peningkatan hasil panen Yaitu FCR, jumlah panen Dan berat badan. Hal ini Dapat dijadikan sebagai Perhatian lebih bagi para Peternak agar hasil panen Menjadi lebih baik.

BAB III

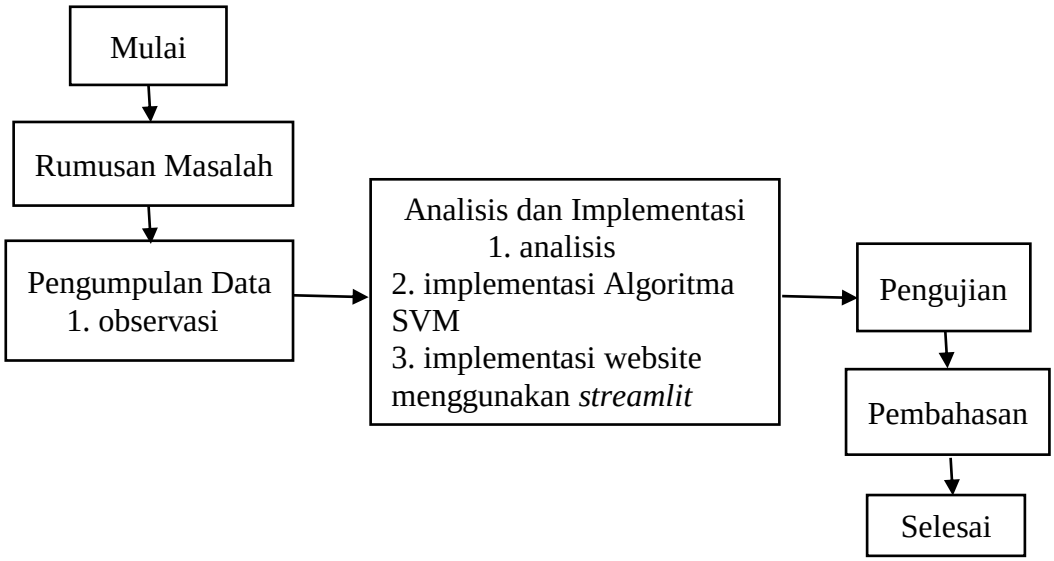
METODE PENELITIAN

3.1 Jenis Penelitian

Penelitian ini menggunakan penelitian kuantitatif yang menggunakan algoritma SVM untuk klasifikasi tingkat keberhasilan produksi ayam broiler. Sehingga hasil tersebut bisa membantu peternak dalam proses penilaian tingkat keberhasilan produksi ayam broiler yang dilakukan secara otomatis dan sistematis agar mengurangi potensi kesalahan, mengoptimalkan perawatan dan pakan, serta meningkatkan produktivitas ayam broiler. Penelitian kuantitatif merupakan kegiatan analisis penelitian di mana pada proses perhitungan menggunakan angka-angka (Veronica et al., 2022).

3.2 Kerangka Berfikir

Adapun kerangka berfikir dari penelitian ini adalah sebagai berikut:



Gambar 3. 1 Kerangka Berfikir

3.3 Analisis Kebutuhan

Dalam sistem yang sedang berjalan, terdapat beberapa aspek yang perlu dipenuhi, yaitu sistem yang mampu mengklasifikasi tingkat keberhasilan produksi ayam broiler berdasarkan indikator yang saling berhubungan, seperti umur ayam, berat badan dan jumlah pakan. Selain itu, diperlukan perangkat lunak dan perangkat keras yang mendukung kinerja sistem yang dikembangkan serta data atau sampel yang memadai untuk memaksimalkan proses kinerja sistem tersebut. Perangkat keras dan lunak yang digunakan dalam penelitian ini terdiri dari komponen hardware dan software untuk pengumpulan dan pengolahan data, sebagai berikut

1. Perangkat Lunak
   - **Python:** Python merupakan Bahasa pemrograman utama untuk implementasi algoritma, pengolahan data dan visualisasi.
   - ***Google Colab*:** *Google Colab* sebagai lingkungan pengembangan interaktif.
   - ***Streamlit*:** *Streamlit* sebagai pembuatan aplikasi web interaktif
2. Perangkat Keras

   Komputer dengan spesifikasi AMD ryzen 3, Ram 8 GB dan SSD 512 GB.

3.4 Teknik Pengumpulan Data

Dalam Penelitian ini data yang diperoleh melalui observasi yang dilakukan secara langsung di peternakan Timan Farm dengan indikator yaitu umur ayam,

berat ayam dan jumlah pakan. Data yang diambil sebanyak 100 data dengan umur ayam mulai dari 16-25 hari.

Tabel 3. 1 Data Ayam Broiler

ID	Umur Ayam	Berat Ayam	Jumlah Pakan
ayam 1	16	650	61
ayam 2	16	639	61
ayam 3	16	570	61
ayam 4	17	709	64
ayam 5	17	723	64
ayam 6	17	720	64
ayam 7	18	771	67
ayam 8	18	740	67
ayam 9	18	776	67
ayam 10	19	839	69
ayam 11	19	843	69
ayam 12	19	838	69
ayam 13	20	905	73
ayam 14	20	903	73
ayam 15	20	940	73
ayam 16	21	988	75
ayam 17	21	991	75
ayam 18	21	997	75
ayam 19	22	1066	77
ayam 20	22	1072	77
ayam 21	22	1081	77
ayam 22	23	1140	80
ayam 23	23	1201	80
ayam 24	23	1157	80
ayam 25	24	1229	83
ayam 26	24	1248	83
ayam 27	24	1250	83
ayam 28	25	1303	84
ayam 29	25	1300	84
ayam 30	25	1313	84
ayam 31	25	1292	84
ayam 32	19	824	69
ayam 33	25	1309	84
ayam 34	25	1259	84
ayam 35	20	892	73
ayam 36	23	1164	80
ayam 37	23	1196	80
ayam 38	22	1074	77
ayam 39	20	872	73

ayam 40	23	1168	80
ayam 41	20	970	73
ayam 42	16	652	61
ayam 43	19	846	69
ayam 44	22	1087	77
ayam 45	23	1184	80
ayam 46	19	860	69
ayam 47	19	879	69
ayam 48	17	684	64
ayam 49	23	1204	80
ayam 50	16	681	61
ayam 51	16	672	61
ayam 52	23	1152	80
ayam 53	20	893	73
ayam 54	18	755	67
ayam 55	19	812	69
ayam 56	24	1253	83
ayam 57	24	1215	83
ayam 58	25	1366	84
ayam 59	22	1041	77
ayam 60	20	953	73
ayam 61	21	961	75
ayam 62	23	1185	80
ayam 63	25	1323	84
ayam 64	21	986	75
ayam 65	21	1002	75
ayam 66	17	725	64
ayam 67	21	966	75
ayam 68	18	789	67
ayam 69	22	1111	77
ayam 70	21	1008	75
ayam 71	22	1043	77
ayam 72	17	702	64
ayam 73	23	1159	80
ayam 74	17	717	64
ayam 75	18	769	67
ayam 76	16	536	61
ayam 77	21	1018	75
ayam 78	19	836	69
ayam 79	23	1233	80
ayam 80	17	726	64
ayam 81	20	944	73
ayam 82	22	1109	77
ayam 83	22	1087	77

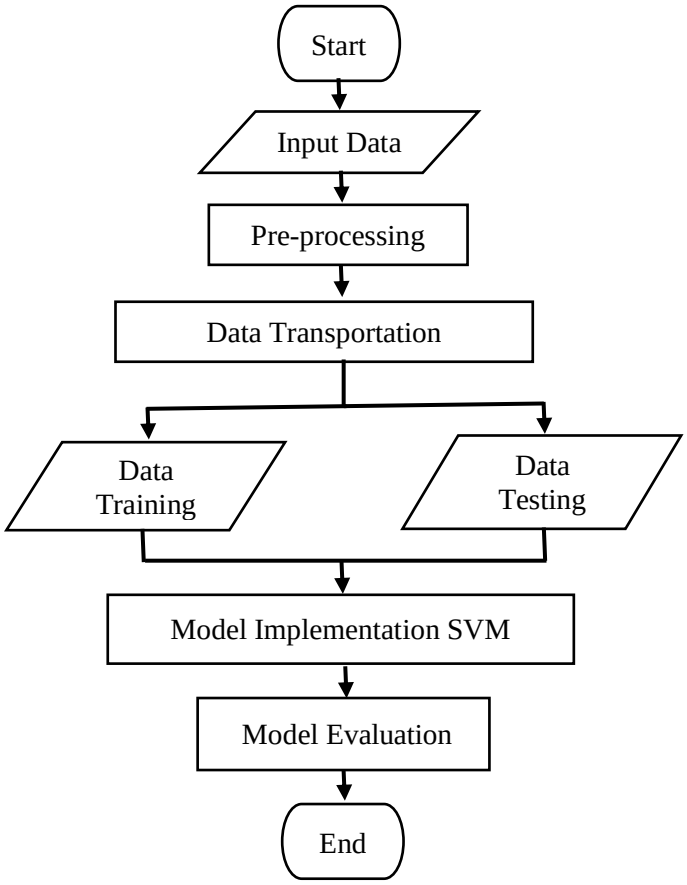
ayam 84	23	1180	80
ayam 85	19	836	69
ayam 86	22	1061	77
ayam 87	21	977	75
ayam 88	19	865	69
ayam 89	18	742	67
ayam 90	18	800	67
ayam 91	16	648	61
ayam 92	25	1336	84
ayam 93	25	1312	84
ayam 94	17	726	64
ayam 95	21	1003	75
ayam 96	24	1224	83
ayam 97	19	787	69
ayam 98	23	1163	80
ayam 99	18	724	67
ayam 100	25	1290	84

3.4.1 Studi Literatur

Studi literatur adalah sebuah metode untuk penelitian yang dilaksanakan dengan cara mengumpulkan, meninjau, dan menganalisis berbagai sumber tertulis yang berkaitan dengan topik atau permasalahan yang sedang dibahas. Sumber-sumber tersebut bisa berupa buku, jurnal ilmiah, artikel, laporan penelitian, dan dokumen lainnya .Pada tahap ini, peneliti mempelajari teori-teori yang berkaitan dengan penerapan algoritma SVM yang diambil dari jurnal sebagai referensi dan landasan teori untuk penelitian ini.

3.5 Proses Klasifikasi

Pendekatan penelitian ini diterapkan untuk mengukur tingkat keberhasilan ayam broiler untuk membantu para peternak dalam meningkatkan hasil produksi melalui penerapan klasifikasi yang dilakukan dengan memanfaatkan algoritma SVM.



***Gambar 3. 2 Flowchart* Sistem Klasifikasi**

3.5.1 *Preprocessing data*

Preprocessing data adalah proses yang dilakukan untuk persiapan dan pembersihan data sebelum dipakai dalam menganalisis atau model pembelajaran machine learning. Tujuan dari *preprocessing data* ini yaitu guna untuk meningkatkan kualitas data agar lebih akurat, terstruktur, dan siap digunakan dalam proses berikutnya. *Preprocessing data* memainkan peran penting dalam mempersiapkan data yang berkualitas untuk analisis dan pemodelan yang lebih akurat. Sebelum data digunakan dalam algoritma SVM, perlu dilakukan beberapa langkah-langkah *preprocessing* yaitu:

1. Pembersihan Data: Menghapus data yang tidak lengkap atau memiliki nilai yang tidak valid.
2. Normalisasi Data: Mengubah skala data agar fitur-fitur memiliki rentang nilai yang seragam.
3. Pemilihan Fitur: Memilih fitur yang paling relevan agar model tidak terbebani oleh informasi yang tidak signifikan

3.5.2 *Data Training* dan *Data Testing*

Sebelum mengklasifikasikan data yang telah diseleksi, Data akan dibagi menjadi dua bagian yaitu:

1. **Data pelatihan (*Training*)**

   Data Training merupakan sekumpulan data yang digunakan untuk melatih model. Model mempelajari pola dari data ini untuk membangun hubungan antara input (fitur) dan output (label). Dalam penelitian ini digunakan untuk melatih model SVM agar dapat mengenali pola-pola dan hubungan antara fitur-fitur dengan tingkat keberhasilan ayam. Data untuk membangun *hyperplane* atau garis optimal yang membedakan kelas kelas yang ada. Data pelatihan dibuat untuk melatih algoritma SVM proporsi data yang digunakan adalah 80% dari keseluruhan dataset.

2. **Data pengujian (*Testing*)**

   Dalam penelitian ini, *data testing* untuk menguji kinerja model SVM yang telah dilatih. Data ini tidak digunakan selama proses pelatihan, sehingga dapat mengukur kemampuan model dalam menggeneralisasi terhadap data baru yang belum pernah dilihat sebelumnya. Data pengujian dibuat dengan perbandingan sebesar 20% dari total dataset.

3.5.3 Implementasi Model SVM

Dalam proses pelatihan, SVM mencoba mencari *hyperplane* optimal, yaitu dengan garis atau bidang pemisah terbaik yang dapat memisahkan kelas-kelas pada data. Namun jika data dapat dipisahkan secara linier, maka linear SVM digunakan. Tetapi, jika data bersifat non-linear, maka digunakan kernel trick untuk memetakan data ke dalam dimensi yang lebih tinggi sehingga dapat dipisahkan dengan lebih baik. Setelah model dilatih, selanjutnya menguji performanya dengan menggunakan *data testing*. Model SVM akan mencoba memprediksi label berdasarkan fitur yang diberikan dalam *data testing*. Hasil prediksi ini dibandingkan dengan label asli untuk mengevaluasi akurasi model.

3.5.4 Evaluasi Model

Evaluasi model merupakan tahapan akhir krusial dalam proses klasifikasi, dimana kinerja dari model diukur menggunakan beberapa metrik evaluasi. Dalam penelitian ini, metrik yang digunakan untuk evaluasi yaitu:

a. Accuracy

Akurasi merupakan dasar yang dihitung sebagai perbandingan antara jumlah klasifikasi yang benar dibandingkan dengan total jumlah klasifikasi yang dilakukan model. Rumus akurasi sebagai berikut:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (3.1)$$

TP = Data yang bernilai positif dan diprediksi benar sebagai positif

TN = Data yang bernilai negatif dan diprediksi sebagai negatif

FP = Data yang sebenarnya negatif tetapi diprediksi sebagai positif (kesalahan tipe I)

FN = Data yang sebenarnya positif tetapi diprediksi sebagai negatif (kesalahan tipe II)

b. *Precision*

Precision untuk mengukur seberapa banyak klasifikasi yang benar dari seluruh klasifikasi positif yang dibuat oleh model.

$$Precision = \frac{TP}{TP+FP} \tag{3.2}$$

c. *Recall*

Recall untuk mengukur seberapa banyak data yang sebenarnya positif berhasil diklasifikasikan dengan benar oleh model.

$$Recall = \frac{TP}{TP+FN} \tag{3.3}$$

d. *F1-Score*

F1-Score rata-rata harmonis dari precision dan recall, digunakan Ketika adanya ketidakseimbangan kelas didalam dataset.

$$F1 = 2 \text{ X } \frac{Precision\ x\ Recall}{Precision+Recall} \tag{3.4}$$

BAB IV

HASIL DAN PEMBAHASAN

4.1 Perhitungan Manual *Support Vector Machine*

Bagian ini bertujuan untuk mendemonstrasikan prinsip-prinsip matematis yang mendasari algoritma SVM, Perhitungan manual ini dilakukan pada subset data yang representatif untuk mengilustrasikan bagaimana SVM bekerja dalam menemukan *hyperplane* optimal yang memisahkan dua kelas data. Proses ini mencakup normalisasi data, formulasi masalah, hingga penyelesaian sistem persamaan linear untuk menentukan parameter model.

4.1.1 Persiapan Data dan Normalisasi

Langkah awal dalam perhitungan adalah persiapan data. Dari keseluruhan dataset yang tersedia , dipilih lima titik data observasi yang mencakup kedua kelas, yaitu "Berhasil" (label 1) dan "Tidak Berhasil" (label 0). Pemilihan data ini merujuk pada data yang digunakan dalam draf penelitian awal.

Tabel 4. 1 Observasi Data Awal

ID Ayam	Umur (hari)	Berat (gram)	Jumlah Pakan (gram)	Label Keberhasilan
ayam 3	16	570	61	0 (Tidak Berhasil)
ayam 5	17	723	64	1 (Berhasil)
ayam 8	18	740	67	0 (Tidak Berhasil)
ayam 1	16	650	61	1 (Berhasil)
ayam 13	20	905	73	0 (Tidak Berhasil)

Fitur-fitur yang digunakan—Umur Ayam, Berat Ayam, dan Jumlah Pakan—memiliki rentang nilai yang sangat berbeda. Sebagai contoh, Berat Ayam memiliki rentang dari 536 hingga 1366 gram, sementara Umur Ayam hanya berkisar antara 16 hingga 25 hari. Perbedaan skala yang signifikan ini dapat

menyebabkan fitur dengan rentang nilai yang lebih besar mendominasi proses perhitungan dan menghasilkan model yang bias. Untuk mengatasi masalah ini, dilakukan normalisasi data menggunakan metode min-max scaler. Metode ini mengubah skala setiap fitur ke dalam rentang nilai yang seragam, biasanya antara 0 dan 1.

Rumus normalisasi *min-max* yang digunakan adalah sebagai berikut :

$$X_{scaled} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

Nilai minimum X_{min} dan nilai maksimum X_{max} untuk setiap fitur ditentukan dari keseluruhan 100 data observasi untuk memastikan proses penskalaan yang konsisten dan representatif. Rentang fitur yang digunakan adalah:

a. Umur : X_{min} = 16, X_{max} = 25
b. Berat : X_{min} = 536 X_{max} = 1366
c. Jumlah Pakan : X_{min} = 61 X_{max} = 84

Sebagai contoh, perhitung normalisasi untuk data “Ayam 8”

$X_1(Umur) = \frac{18-16}{25-16} = \frac{2}{9}$ = 0.2222

$X_2(Berat) = \frac{740-536}{1366-536} = \frac{204}{830}$ = 0.2457

$X_3(Jumlah\ Pakan) = \frac{67-61}{84-61} = \frac{6}{23}$ = 0.2608

Setelah menerapkan proses normalisasi pada kelima titik data observasi dan mengonversi label Berhasil +1 dan Tidak Berhasil menjadi -1 sesuai dengan konvensi SVM, diperoleh data yang siap untuk dianalisis sebagai berikut.

Tabel 4. 2 Dataset Setelah Normalisasi

ID Data	Umur	Berat	Pakan	Label SVM (y)
d1	0.00000	0.04096	0.00000	-1
d2	0.11111	0.22530	0.13043	+1
d3	0.22222	0.24578	0.26087	-1
d4	0.00000	0.13735	0.00000	+1
d5	0.44444	0.44458	0.52174	-1

4.1.2 Formulasi Masalah dan Asumsi *Support Vector*

Konsep fundamental dari SVM adalah menemukan sebuah *hyperplane* optimal yang dapat memisahkan data ke dalam kelas-kelas yang berbeda dengan *margin* (jarak) semaksimal mungkin. Dalam konteks tiga fitur, *hyperplane* ini merupakan sebuah bidang datar dalam ruang tiga dimensi. Persamaan umum untuk *hyperplane* linear adalah:

$$f(x) = w\,.\,x + b = w_1X_1 + w_2X_2 + w_3X_3 = 0$$

Di mana w = (w_1, w_2, w_3) adalah vektor bobot yang menentukan orientasi *hyperplane*, w = (x_1, x_2, x_3) adalah vektor fitur data, dan b adalah bias atau intersep yang menggeser *hyperplane* dari titik asal. Titik-titik data yang berada tepat di tepi *margin* disebut sebagai *support vectors*. Titik-titik inilah yang "mendukung" atau mendefinisikan posisi dan orientasi *hyperplane*. Untuk *support vectors*, berlaku kondisi :

$$y_i(w\,.\,x_i + b) = 1$$

Dalam implementasi praktis, algoritma SVM menggunakan metode optimasi kompleks seperti *Quadratic Programming* untuk secara otomatis mengidentifikasi titik mana yang merupakan *support vectors*. Namun, untuk tujuan demonstrasi manual, proses ini terlalu rumit secara komputasi. Oleh karena itu, dibuat sebuah asumsi pedagogis: kita mengasumsikan bahwa empat dari lima

titik data yang telah dinormalisasi (d1, d2, d3, dan d4) adalah *support vectors*. Asumsi ini memungkinkan kita untuk menyusun sistem persamaan linear yang dapat diselesaikan secara manual untuk menemukan parameter w dan b, sehingga dapat mengilustrasikan konsep inti SVM tanpa memerlukan perangkat lunak optimasi. *Support Vectors* yang diasumsikan adalah :

$d_1 = (0.0, 0.04096, 0.0), y_1 = -1$

$d_2 = (0.1111, 0.22530, 0.13043), y_2 = +1$

$d_3 = (0.2222, 0.24578, 0.25087), y_3 = -1$

$d_4 = (0.0, 0.13735, 0.0), y_4 = +1$

4.1.3 Penyelesaian Sistem Persamaan Linear untuk Menemukan w dan b

Berdasarkan asumsi *support vectors* diatas, empat persamaan linear dapat disusun dengan menerapkan kondisi

$$y_i(w\,.\,x_i + b) = 1$$

1. Untuk $d_1, y_1 = -1$

   $-1(w_1(0.0) + w_2(0.04096) + w_3(0.0) + b) = 1$

   $(-0{,}04096w_2 - b) = 1$

   $(0{,}04096w_2 + b) = -1$ **(**Persamaan A)

2. Untuk $d_2, y_2 = +1$

   $1(w1(0.11111) + w2(0.22530) + w3(0.13043) + b) = 1$

   $0.11111w_1 + 0.22530w_2 + 0.13043w_3 + b=1$ (Persamaan B)

3. Untuk $d_3, y_3 = -1$

   $-1(w_1(0.22222) + w_2(0.24578) + w_3(0.26087) + b) = 1$

   $(-0{,}22222w_1 - 0{,}24578w_2 - 0{,}26087w_3 - b) = 1$

   $(0{,}22222w_1 + 0{,}24578w_2 + 0{,}26087w_3 + b) = -1$ (Persamaan C)

4. Untuk $d_4, y_4 = +1$

$+1(w_1(0.0) + w_2(0.13735) + w_3(0.0) + b) = 1$

$(0{,}13735w_2 + b) = 1$ (Persamaan D)

Sistem empat persamaan ini sekarang dapat diselesaikan untuk menemukan nilai w_1, w_2 , w_3 , dan b.

Langkah 1 : Menyelesaikan untuk w_2 dan b

Langkah ini menggunakan Persamaan A dan Persamaan D untuk menemukan nilai w_2 dan b

Persamaan A : $(0{,}04096w_2 + b) = -1$

Persamaan D : $(0{,}13735w_2 + b) = 1$

Dari **Persamaan D,** dapat diekspresikan b dalam bentuk w_2 :

$$b = 1 - 0.1375w_2$$

Kemudian, substitusikan nilai b kedalam Persamaan A :

$$0.04096w_2 + (1 - 0.1375w_2) = -1$$

Sederhanakan persamaan diatas :

$0.04096w_2 - 0.1375w_2 + 1 = -1$

$-0.09654w_2 = -1 - 1$

$-0.09654w_2 = -2$

$$w_2 = \frac{-2}{-0.09654} = 20.717$$

Sekarang kita dapat menggantikan nilai w_2 ke dalam persamaan b :

$b = 1 - 0.1375(20.717)$

$b = 1 - 2.8486 = -1.8486$

$w_2 = 20.717 \quad b = -1.8486$

Langkah 2 : Menyelesaikan untuk w_1 dan w_3

Sekarang kita substitusikan nilai w_2 dan b yang telah ditemukan ke dalam Persamaan B dan Persamaan C untuk membentuk sistem dua persamaan linear baru untuk w_1 dan w_3.

Dari Persamaan B :

$0.11111\text{w1} + 0.22530\text{w2} + 0.13043\text{w3} + \text{b} = 1$

$0.11111\text{w1} + 0.22530 \,.\, 20.717 + 0.13043\text{w3} - 1.8486 = 1$

$0.11111\text{w1} + 4.6675 + 0.13043\text{w3} - 1.8486 = 1$

$0.11111\text{w1} + 2.8189 + 0.13043\text{w3} = 1$

$0.11111\text{w1} + 0.13043\text{w3} = 1 - 2.8189$

$0.11111\text{w1} + 0.13043\text{w3} = -1.8189$ (Persamaan E)

Dari Persamaan C :

$(0{,}22222w_1 + 0{,}24578w_2 + 0{,}26087w_3 + b) = -1$

$(0{,}22222w_1 + 0{,}24578(20{,}717) + 0{,}26087w_3 + (-1{,}8486) = -1$

$(0{,}22222w_1 + 5{,}0918 + 0{,}26087w_3 - 1{,}8486) = -1$

$0.22222\text{w1} + 0.26087\text{w3} + 3{,}2432 = -1$

$0.22222\text{w1} + 0.26087\text{w3} = -1 - 3{,}2432$

$0.22222\text{w1} + 0.26087\text{w3} = -4{,}2432$ (Persamaan F)

Langkah 3 : Mengeliminasi Variabel untuk Menemukan Solusi Akhir

Untuk menyelesaikan sistem Persamaan E dan F, kalikan **Persamaan E** dengan 2 untuk mengeliminasi w1 :

$2 \text{ x } (0.11111\text{w1} + 0.13043\text{w3}) = 2 \text{ x } (-1.8189)$ (Persamaan E)

$0.22222\text{w1} - 0.26086\text{w3} = -3.6378$

Sekarang, kurangkan persamaan ini dengan Persamaan F :

$(0.22222\text{w1} + \ 0.26087\text{w3}\ ) - (0.22222\ w_1\ -\ \ 0.26086\ w_3\ )\ \ = -4.2432 -$

(-3.6378)

$0.00001\ w_3 = -0{,}6054$

$$w_3 = \frac{-0.6054}{0.00001} = -60540$$

Sekarang substitusikan nilai w_3 yang ditemukan ke dalam **Persamaan E** untuk menemukan w_1:

$0.11111\text{w1}\ +\ 0.13043(-60450)\ =\ -1.8189$

$0.11111\text{w1} - 7884{,}49\ =\ -1.8189$

$0.11111\text{w1} = -1.8189 + 7884{,}49$

$0.11111\text{w1} = 7882.671$

$$w_1 = \frac{7882{,}671}{0.11111} = 70944.74$$

Dengan demikian, parameter model SVM linear yang ditemukan adalah :

$w_1 = 70944.74$

$w_2 = 20.717$

$w_3 = -60540$

$b = -1.8486$

4.1.4 Penentuan *Hyperplane* Optimal dan *Margin*

Dengan parameter yang telah dihitung, persamaan *hyperplane* optimal untuk model manual ini adalah :

$f(x) = 70944{,}74X_1 + 20{,}717X_2 - 60540X_3 - 1.8486 = 0$

Lebar margin geometris, yang merupakan jarak antara *hyperplane* pemisah dengan *support vectors*, dapat dihitung menggunakan rumus :

$$M = \frac{2}{|\,w\,|}$$

$$|\,w\,| = \sqrt{w_1^2 + w_2^2 \; + \; w_3^2}$$

$$|\,w\,| = \sqrt{(70944{,}74)^2 + \; (20.717)^2 + (-60450)^2}$$

$$|\,w\,| = \sqrt{5033156133.67 + 429{,}194 + 3654202500}$$

$$|\,w\,| = \sqrt{8687359062.86} = 93206.0033$$

Berdasarkan perhitungan diatas, maka lebar *margin* adalah :

$$M = \frac{2}{|\,w\,|}$$

$$M = \frac{2}{93206.0033} = 0.00002145$$

Margin yang sangat kecil ini mengindikasikan bahwa *support vectors* yang diasumsikan berada sangat dekat satu sama lain dalam ruang fitur yang telah dinormalisasi.

4.1.5 Verifikasi Klasifikasi Manual

Untuk memverifikasi fungsi *huperplane* yang telah ditemukan, akan mengklasifikasikan titik data d5, yang tidak diasumsikan sebagai *support vector*. Data d5 (dinormalisasi) memiliki nilai x = (0.44444,0.44458, 0.52174) dengan label asli y = -1 (tidak berhasil).

Substitusikan nilai-nilai ini ke dalam fungsi Keputusan f(x) :

$f(x) =$ 70944,74X_1 + 20,717X_2 - 60450X_3 – 1.8486

$f(d5) =$ 70944,74(0.44444) + 20,717(0.44458) - 60450(0.52174) – 1.8486

$f(d5) =$ 31530.6802 + 9,21036386 – 31539,183 – 1.8486

$f(d5) =$ -1.1410

Hasilnya adalah nilai negatif. yang sesuai dengan prediksi untuk kelas -1 (Tidak Berhasil). Ini menunjukkan bahwa *hyperplane* yang diturunkan secara manual berhasil mengklasifikasikan titik uji ini dengan benar berdasarkan asumsi yang dibuat.

Perhitungan manual dalam algoritma SVM memberikan pemahaman dasar tentang prinsip-prinsip kerja algoritma, namun memiliki keterbatasan yang signifikan ketika dihadapkan dengan data yang lebih kompleks. Dalam perhitungan manual, kita harus membuat asumsi-asumsi seperti memilih titik data tertentu sebagai support vectors, yang dapat menyebabkan solusi yang sangat bergantung pada data yang dipilih. Hal ini membatasi kemampuan model untuk menggeneralisasi dengan baik pada data yang lebih besar atau data yang lebih beragam. Selain itu, metode manual tidak dapat menangani kasus-kasus non-linear dengan efisien, sehingga menghasilkan pemisah yang kurang optimal jika dibandingkan dengan metode berbasis kernel.

Di sisi lain, penggunaan Python untuk mengimplementasikan SVM memberikan banyak keuntungan. Dengan menggunakan pustaka seperti scikit-learn, kita dapat memanfaatkan teknik optimasi canggih, seperti GridSearchCV untuk mencari kombinasi hyperparameter terbaik, serta mengaplikasikan kernel non-linear seperti RBF (*Radial Basis Function*) yang memungkinkan model untuk menangani data yang lebih kompleks. Python juga memungkinkan pemrosesan data yang lebih besar dan lebih cepat, serta penyesuaian model yang lebih mudah dengan data baru. Selain itu, Python memberikan evaluasi yang lebih lengkap dan objektif dengan menggunakan metrik seperti cross-validation, ROC-AUC, dan confusion matrix, yang tidak dapat dicapai dengan perhitungan manual. Dengan

demikian, meskipun perhitungan manual berguna untuk tujuan pembelajaran, penggunaan Python lebih efektif untuk aplikasi dunia nyata.

4.2 Implementasi Model SVM dengan *Python*

Pada bab ini, akan dijelaskan secara rinci proses implementasi algoritma *Support Vector Machine* (SVM) dalam mengklasifikasikan tingkat keberhasilan produksi ayam *broiler*. Langkah-langkah yang akan dijelaskan meliputi persiapan data, pelatihan model, evaluasi hasil, dan pengujian performa menggunakan *Python*.

4.2.1 Persiapan Data

1. *Impor Library*

Langkah pertama dalam persiapan data adalah mengimpor berbagai *library* yang diperlukan untuk analisis. *Pandas* dan *NumPy* digunakan untuk manipulasi dan perhitungan data, sementara *Matplotlib* dan *Seaborn* digunakan untuk visualisasi data agar bisa menganalisis pola dan distribusi fitur. *Sklearn* menyediakan berbagai fungsi untuk machine learning, termasuk pembagian data, normalisasi, dan evaluasi model. *Joblib* berguna untuk menyimpan model yang telah dilatih, sehingga tidak perlu melatih ulang, dan *warnings* digunakan untuk menyembunyikan peringatan yang tidak relevan selama proses analisis. Semua library ini penting untuk mempersiapkan data, melatih model SVM, dan mengevaluasi kinerjanya.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split,
GridSearchCV, StratifiedKFold
from sklearn.preprocessing import MinMaxScaler
from sklearn.svm import SVC
```

```
from sklearn.pipeline import Pipeline
from sklearn.metrics import (
    accuracy_score,
    confusion_matrix,
    classification_report,
    roc_auc_score,
    RocCurveDisplay
)
import joblib
import warnings

warnings.filterwarnings('ignore') # Untuk menyembunyikan
warning, bisa di-remark jika ingin melihat warning
```

Gambar 4. 1 Kode Impor *Library Python*

2. *Load Dataset*

Langkah berikutnya adalah memuat dataset yang digunakan untuk pelatihan dan pengujian model. Pada bagian ini, dataset dibaca menggunakan **pandas** dari file Excel yang tersedia.

```
# Load dataset
df = pd.read_excel('dataset_final_label.xlsx')

print("Dataset Awal (5 data teratas):")
display(df.head())

print(f"\nJumlah Data: {df.shape[0]}, Jumlah Fitur:
{df.shape[1]}")

print("\nInformasi Dataset:")
display(df.info())

print("\nStatistik Deskriptif:")
display(df.describe())

print("\nDistribusi Label Keseluruhan:")
display(df['Label Keberhasilan'].value_counts(normalize=True)
* 100)
display(df['Label Keberhasilan'].value_counts())
```

Gambar 4. 2 Kode *Load* dan Persiapan Data

Setelah dataset dimuat, tampilan pertama yang ditampilkan adalah 5 data teratas dari dataset untuk memberikan gambaran awal mengenai struktur data yang digunakan. Berikut adalah contoh output dataset yang ada :

```
Dataset Awal (5 data teratas):
```

	ID	Umur Ayam	Berat Ayam	Jumlah Pakan	Label Keberhasilan
0	ayam 1	16	650	61	1
1	ayam 2	16	639	61	1
2	ayam 3	16	570	61	0
3	ayam 4	17	709	64	1
4	ayam 5	17	723	64	1

```
Jumlah Data: 100, Jumlah Fitur: 5
```

Gambar 4. 3 Tampilan 5 Baris Pertama Dataset

Statistik deskriptif memberikan gambaran mengenai distribusi fitur dalam dataset:

Tabel 4. 3 Tabel Statistik Deskriptif Dataset

Fitur	count	mean	std	min	25%	50%	75%	max
Umur Ayam	100	20.65	2.78	16	18	21	23	25
Berat Ayam	100	977.20	215.68	536	788	981	1165	1366
Jumlah Pakan	100	73.66	7.24	61	67	75	80	84
Label Keberhasilan	100	0.69	0.46	0	0	1	1	1

Dari tabel statistik deskriptif diatas, dapat diberikan Kesimpulan sebagai berikut

a. Umur Ayam rata-rata adalah 20.65 hari, dengan rentang 16 hingga 25 hari.
b. Berat Ayam memiliki rata-rata 977.20 gram, dengan rentang antara 536 hingga 1366 gram.
c. Jumlah Pakan memiliki rata-rata 73.66 gram, dengan rentang antara 61 hingga 84 gram.

d. Label Keberhasilan menunjukkan bahwa sebagian besar ayam berhasil (dengan nilai rata-rata 0.69, yang berarti 69% ayam berhasil).

Distribusi label keberhasilan menunjukkan bahwa **69%** dari data menunjukkan bahwa ayam berhasil dalam produksi (Label = 1). **31%** menunjukkan bahwa ayam tidak berhasil (Label = 0).

3. Memisahkan Fitur (X) dan Label (y)

Pada bagian ini, penulis memisahkan kolom target yang berfungsi untuk klasifikasi (dalam hal ini, kolom Label Keberhasilan) dari fitur-fitur yang akan digunakan untuk pelatihan model SVM. Fitur (X) mencakup kolom-kolom yang akan digunakan sebagai input untuk model, yaitu Umur Ayam, Berat Ayam, dan Jumlah Pakan, sedangkan kolom target (y) adalah Label Keberhasilan yang berfungsi sebagai label kelas yang ingin diprediksi oleh model.

```
X = df[["Umur Ayam", "Berat Ayam", "Jumlah Pakan"]].values
y = df["Label Keberhasilan"].values

print("Fitur (5 data pertama):")
display(pd.DataFrame(X, columns=["Umur Ayam", "Berat Ayam",
"Jumlah Pakan"]).head())
print("Label (5 data pertama):", y[:5])
```

Gambar 4. 4 Kode untuk memisahkan fitur dan label

Hasil ouput dari kode ini menunjukkan bahwa :

a. Fitur (X) terdiri dari tiga kolom : Umur Ayam, Berat Ayam, dan Jumlah Pakan, dengan nilai-nilai sebagai berikut :

Fitur (X) - 5 data pertama:

	Umur Ayam	Berat Ayam	Jumlah Pakan
0	16	650	61
1	16	639	61
2	16	570	61
3	17	709	64
4	17	723	64

Gambar 4. 5 Gambar 5 Baris Data Pertama Fitur X

b. Fitur Y terdiri dari label keberhasilan dengan nilai 1 dan 0

4. Membagi dataset menjadi data latih dan uji

Pada bagian ini, data dibagi menjadi dua set: data pelatihan (training data) dan data pengujian (testing data) menggunakan fungsi train_test_split dari sklearn.model_selection. Fungsi ini membantu membagi dataset menjadi dua bagian yang digunakan untuk melatih model dan menguji akurasinya.

```
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42, stratify=y
)

print(f"Jumlah data training : {X_train.shape[0]}")
print(f"Jumlah data testing  : {X_test.shape[0]}")

print("\nDistribusi label pada data training:")
display(y_train.value_counts(normalize=True) * 100)
print("\nDistribusi label pada data testing:")
display(y_test.value_counts(normalize=True) * 100)
```

Gambar 4. 6 Kode Pembagian Data Latih dan Data Uji

Penjelasan kode :

a. **train_test_split**: Fungsi ini digunakan untuk membagi data menjadi data pelatihan dan data pengujian. Parameter test_size=0.2 menunjukkan

bahwa 20% data akan digunakan untuk pengujian, sementara 80% sisanya digunakan untuk pelatihan.

b. **random_state=42**: Ini digunakan untuk memastikan bahwa pembagian data selalu sama setiap kali kode dijalankan, sehingga hasilnya bisa direproduksi.

c. **stratify=y**: Parameter ini memastikan bahwa distribusi label pada data pelatihan dan pengujian seimbang, yaitu proporsi label "Berhasil" dan "Tidak Berhasil" pada data pelatihan dan pengujian akan sama.

Adapun hasil output dari kode terkait adalah sebagai berikut :

a. Jumlah data dibagi menjadi 80 data untuk data *training* dan 20 data untuk label testing

b. Distribusi label keberhasilan pada *data training*, data label ayam berhasil adalah 68,75% dan tidak berhasil adalah 31,25%. Dalam *data testing* distribusi label berhasil adalah 70% sedangkan yang tidak berhasil adalah 30%

Pembagian ini memastikan bahwa data pelatihan dan pengujian memiliki proporsi label yang seimbang, yang sangat penting untuk melatih model agar tidak terbias oleh salah satu kelas. Ini memastikan bahwa model dapat memprediksi kedua kelas dengan baik.Dengan pembagian dan distribusi yang adil, model SVM akan lebih akurat dan dapat menggeneralisasi hasil dengan lebih baik.

4.2.2 Membangun Pipeline Preprocessing dan Model

Pada bagian ini, penulis membangun *pipeline* untuk mempermudah alur pemrosesan data dan pelatihan model secara sistematis. *Pipeline* adalah rangkaian langkah-langkah yang diterapkan secara berurutan dalam satu objek sehingga

meminimalisir kesalahan dan memastikan semua langkah dijalankan dengan urutan yang tepat.

```
# Membuat pipeline
pipeline_svm = Pipeline([
    ('scaler', MinMaxScaler()),
    ('svc', SVC(probability=True, random_state=42)) #
probability=True untuk ROC AUC dan threshold moving
])

print("Pipeline berhasil dibuat.")
display(pipeline_svm)
```

Gambar 4. 7 Pipeline Model SVM

Pada kode ini, pipeline yang dibangun terdiri dari dua langkah utama:

1. Normalisasi Data dengan *MinMaxScaler* :

Normalisasi dilakukan menggunakan *MinMaxScaler* dari *sklearn.preprocessing*. Tujuan normalisasi ini adalah untuk mengubah skala data fitur agar berada dalam rentang [0, 1], yang sangat penting untuk algoritma SVM. Skala fitur yang seragam membantu model untuk lebih cepat dalam pelatihan dan menghindari dominasi fitur tertentu yang memiliki skala lebih besar.

2. Model SVM menggunakan SVC:

Setelah normalisasi, langkah berikutnya adalah menggunakan SVC (*Support Vector Classification*) untuk membangun model SVM. SVC adalah algoritma yang sangat cocok untuk klasifikasi dengan tujuan mencari *hyperplane* terbaik yang memisahkan dua kelas. Dalam hal ini, kita juga menambahkan parameter *probability=True* pada SVC, yang mengaktifkan kemampuan model untuk menghasilkan probabilitas prediksi. Probabilitas ini akan berguna untuk menghitung ROC AUC dan untuk melakukan *threshold moving*, yang

memungkinkan penyesuaian nilai batas keputusan (*decision threshold*) pada model untuk menyeimbangkan antara *precision* dan *recall*.

4.2.3 Latih Model dengan Parameter Terbaik

Pada bagian ini, digunakan *Grid Search* untuk melakukan *hyperparameter tuning* pada model SVM. Tujuan utama dari *grid search* adalah untuk menemukan kombinasi parameter terbaik (seperti C, kernel, gamma, dan degree) yang memberikan performa terbaik pada model. Dengan menggunakan *Stratified K-Fold Cross-Validation*, *grid search* dapat memastikan bahwa hasil tuning lebih andal dengan membagi data ke dalam beberapa bagian untuk pengujian.

```
# Tentukan parameter grid
param_grid = {
    'svc__C': [0.1, 1, 10, 100],
    'svc__kernel': ['linear', 'rbf', 'poly'],
    'svc__gamma': ['scale', 'auto', 0.01, 0.1, 1], # 'gamma'
relevan untuk 'rbf' dan 'poly'
    'svc__degree': [2, 3, 4], # 'degree' relevan untuk 'poly'
    'svc__class_weight': [None, 'balanced'] # Untuk menangani
imbalanced dataset jika perlu
}

# Stratified K-Fold Cross-Validation
cv_stratified = StratifiedKFold(n_splits=5, shuffle=True,
random_state=42)

# GridSearchCV
grid_search = GridSearchCV(
    estimator=pipeline_svm,
    param_grid=param_grid,
    cv=cv_stratified,
    scoring='roc_auc', # Bisa juga 'f1', 'accuracy',
'precision', 'recall'
    verbose=1, # Menampilkan progress
    n_jobs=-1 # Menggunakan semua core CPU
)

print("Memulai Grid Search CV...")
grid_search.fit(X_train, y_train)
```

```
print("\nParameter terbaik hasil Grid Search:")
print(grid_search.best_params_)

print(f"\nSkor {grid_search.scoring} terbaik dari Grid Search:
{grid_search.best_score_:.4f}")

# Model terbaik
best_model_pipeline = grid_search.best_estimator_
```

Gambar 4. 8 Kode untuk Mencari Parameter terbaik

Berikut dibawah ini merupakan penjelasan kode diatas :

1. Menentukan Parameter *Grid*
   a. **param_grid** : Ini adalah *dictionary* yang berisi kombinasi nilai-nilai parameter yang akan diuji. *Grid search* akan mencoba setiap kombinasi parameter untuk menemukan yang terbaik.
   b. **svc_C** : Parameter untuk pengaturan regularisasi SVM, nilai yang diuji adalah [0.1, 1, 10, 100].
   c. **svc_kernel** : Jenis kernel yang digunakan dalam SVM, yang diuji adalah 'linear', 'rbf', dan 'poly'.
   d. **svc_gamma** : Parameter untuk kernel rbf dan poly, yang menentukan skala pengaruh setiap titik data terhadap keputusan model.
   e. **svc_degree** : *Degree* untuk kernel polinomial (poly), nilai yang diuji adalah [2, 3, 4].
   f. **svc_class_weight** : Digunakan untuk menangani kelas yang tidak seimbang, diuji dengan opsi [None, 'balanced'].
2. *Stratified K-Fold Cross-Validation*

   StratifiedKFold digunakan untuk melakukan *cross-validation* dengan memastikan distribusi kelas yang seimbang di setiap *fold* (bagiannya). Parameter

n_splits=5 menunjukkan bahwa data dibagi menjadi 5 bagian, dan setiap bagian digunakan sekali untuk pengujian sementara sisanya digunakan untuk pelatihan. Dengan parameter shuffle=True, data akan diacak sebelum dibagi untuk memastikan variasi yang baik dalam pembagian data.

3. *GridSearchCV*

GridSearchCV digunakan untuk mencari parameter terbaik dengan menggunakan *grid search*. Proses ini melibatkan pencarian untuk setiap kombinasi parameter yang diberikan dalam *param_grid*. Proses pencarian menggunakan model yang sudah didefinisikan sebelumnya, yaitu *pipeline* yang berisi *MinMaxScaler* dan SVC.

4. Evaluasi dan Hasil

Setelah proses grid search selesai, hasil terbaik akan dipilih berdasarkan skor validasi yang tertinggi. Skor ini dihitung menggunakan ROC AUC, yang mengukur kemampuan model dalam membedakan antara dua kelas (berhasil dan tidak berhasil). grid_search.best_params_ menampilkan kombinasi parameter terbaik yang ditemukan oleh grid search. grid_search.best_score_ menunjukkan skor ROC AUC terbaik yang dicapai dengan kombinasi parameter tersebut.

Setelah melakukan Analisa dengan metode diatas, berikut merupakan parameter terbaik yang ditemukan :

a. svc_C = 100: Nilai C yang lebih tinggi menunjukkan regularisasi yang lebih kuat.
b. svc_kernel = 'rbf': Kernel Radial Basis Function (RBF) memberikan performa terbaik.
c. svc_gamma = ‘auto’: Skala titik data menggunakan auto.

d. svc_degree = 2: Degree kernel polinomial (untuk kernel poly).
e. svc_class_weight = None: Tidak ada penyesuaian bobot kelas.
f. Skor terbaik (ROC AUC): 0.9745, yang menunjukkan bahwa model SVM yang telah di-tune dengan parameter ini memiliki kemampuan yang sangat baik untuk membedakan antara kelas "berhasil" dan "tidak berhasil".

4.2.4 Evaluasi Model pada Data Testing

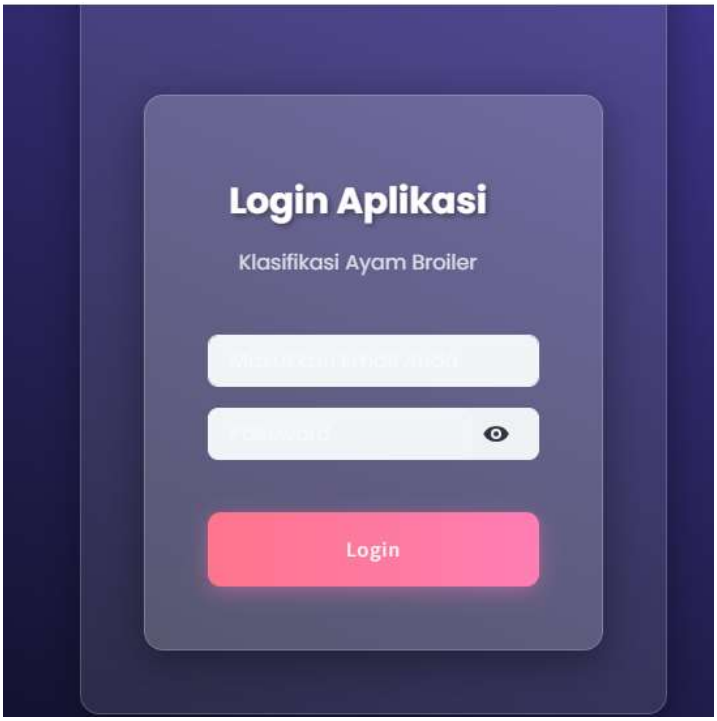
Pada bagian ini, model SVM yang sudah dioptimasi diuji dengan menggunakan data pengujian untuk melihat seberapa baik kinerjanya. Hasil evaluasi menunjukkan bahwa model memiliki akurasi 90% dan ROC AUC 1.00, yang berarti model dapat memisahkan kedua kelas ("Berhasil" dan "Tidak Berhasil") dengan sangat baik. Precision, recall, dan f1-score untuk kelas "Berhasil" sangat baik, dengan nilai recall 100%, artinya model berhasil mendeteksi semua data "Berhasil". Namun, untuk kelas "Tidak Berhasil", meskipun precision-nya 100%, recall-nya hanya 67%, yang menunjukkan ada beberapa data "Tidak Berhasil" yang tidak terdeteksi dengan sempurna.

Pada confusion matrix, model menunjukkan 14 prediksi yang benar untuk kelas "Berhasil" dan 4 untuk "Tidak Berhasil". Ada sedikit kesalahan pada data kelas "Tidak Berhasil" yang diprediksi sebagai "Berhasil". Sementara itu, ROC curve juga menunjukkan bahwa model memiliki kinerja yang sangat baik, dengan nilai AUC mencapai 1.00, yang berarti model sangat tepat dalam memisahkan kedua kelas. Secara keseluruhan, model ini bekerja sangat baik dalam mengklasifikasikan tingkat keberhasilan ayam broiler berdasarkan data yang diberikan.

4.3 Implementasi dan Hasil Berbasis Web (*Streamlit)*

4.3.1 Tampilan Form *Login*

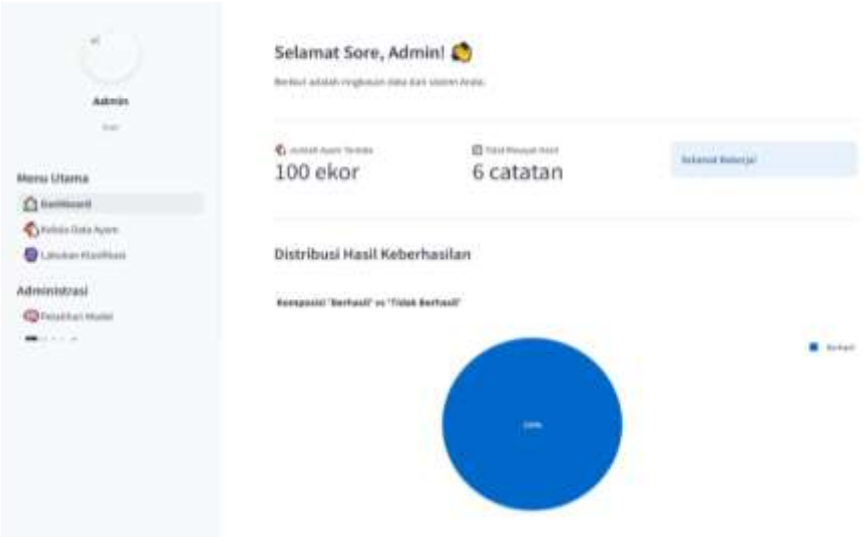
Tampilan ini menunjukkan halaman login aplikasi Klasifikasi Ayam Broiler, dengan desain modern dan minimalis. Pengguna diminta untuk memasukkan email dan password, dilengkapi dengan ikon mata untuk menampilkan password. Tombol Login yang mencolok berada di bawah form, memudahkan pengguna untuk masuk. Desain menggunakan warna gelap dengan elemen cerah, memberikan kesan elegan dan profesional, serta memudahkan navigasi. Dapat dilihat pada gambar dibawah ini



Gambar 4. 9 Tampilan *Form Login*

4.3.2 Tampilan Dashboard

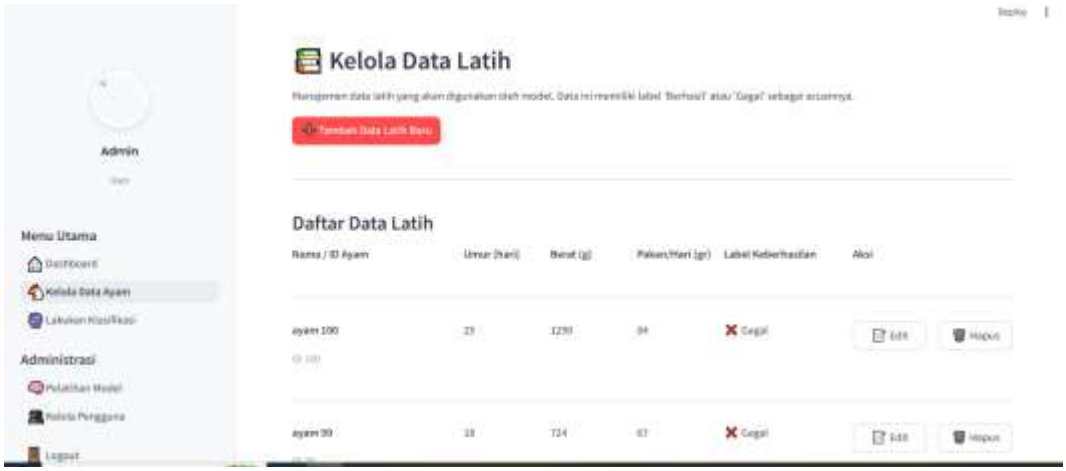
Tampilan ini menunjukkan Dashboard Admin aplikasi Klasifikasi Ayam Broiler, yang menampilkan informasi seperti jumlah ayam terdata , total riwayat hasil , dan distribusi hasil klasifikasi. Dapat dilihat pada gambar dibawah ini.



Gambar 4. 10 Tampilan Menu *Dashboard*

4.3.3 Tampilan *Form* Kelola Data Ayam

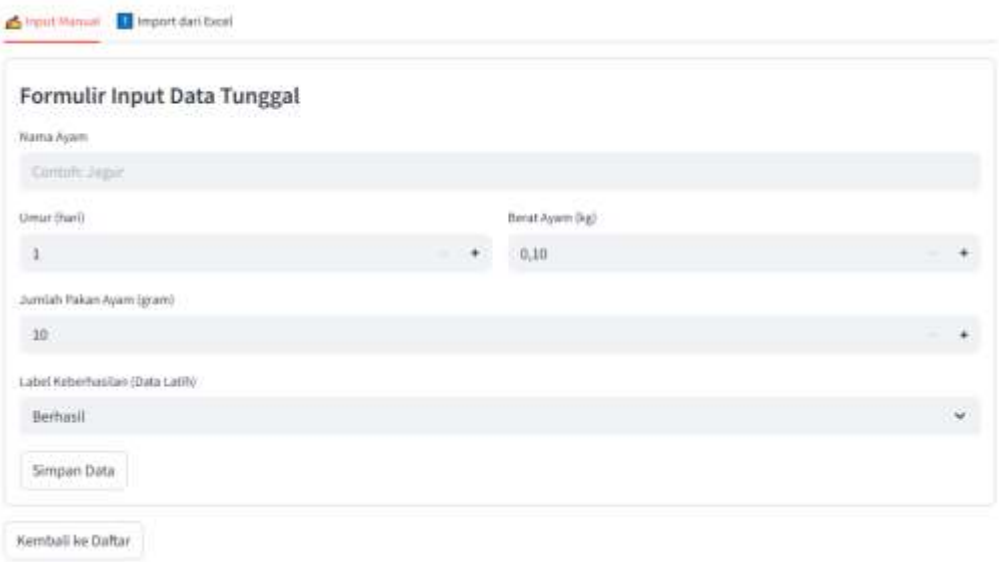
Tampilan *Form* Kelola Data Ayam menunjukkan jumlah data ayam dan data latih yang tersedia dalam aplikasi dapat dilihat pada gambar dibawah ini.



Gambar 4. 11 Tampilan Form Kelola Data Ayam

4.3.4 Tampilan *Form* Tambah Data Ayam

Tampilan formulir untuk menambahkan data ayam atau data latih baru dapat dilihat pada gambar di bawah ini. Terdapat dua cara untuk menambahkan data: pengguna dapat melakukannya secara manual, atau secara otomatis dengan mengimpor data *batch* dalam format *Excel*.



Gambar 4. 12 Tampilan Input Data Latih Manual

Gambar 4. 13 Tampilan Input Data Latih Otomatis

4.3.5 Tampilan *Form* Edit Data Ayam

Tampilan formulir untuk mengedit data ayam atau data latih dapat dilihat pada gambar di bawah ini.

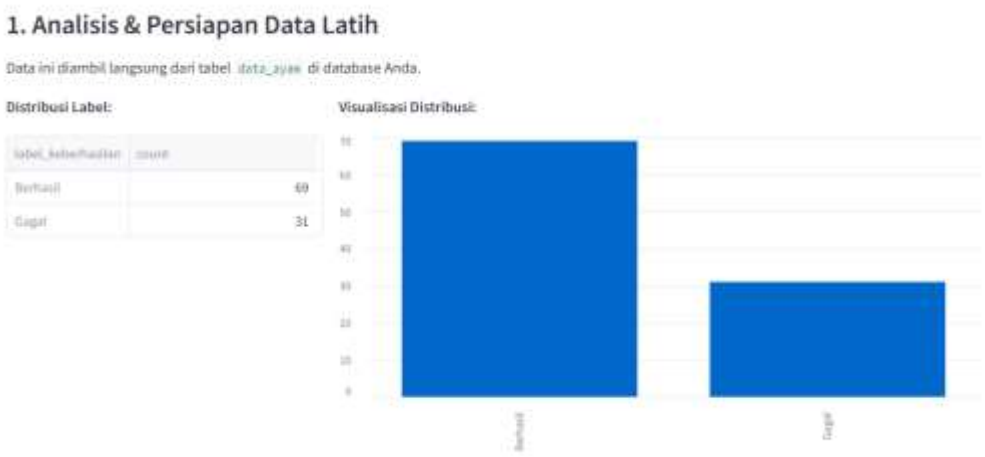
Gambar 4. 14 Tampilan Form Edit Data Ayam

4.3.6 Tampilan Pelatihan Model

Menu ini bertujuan untuk melatih model klasifikasi menggunakan data yang disediakan oleh basis data. Halaman ini terstruktur dengan beberapa bagian penting, yang dimulai dari analisis data latih hingga evaluasi hasil pelatihan model.

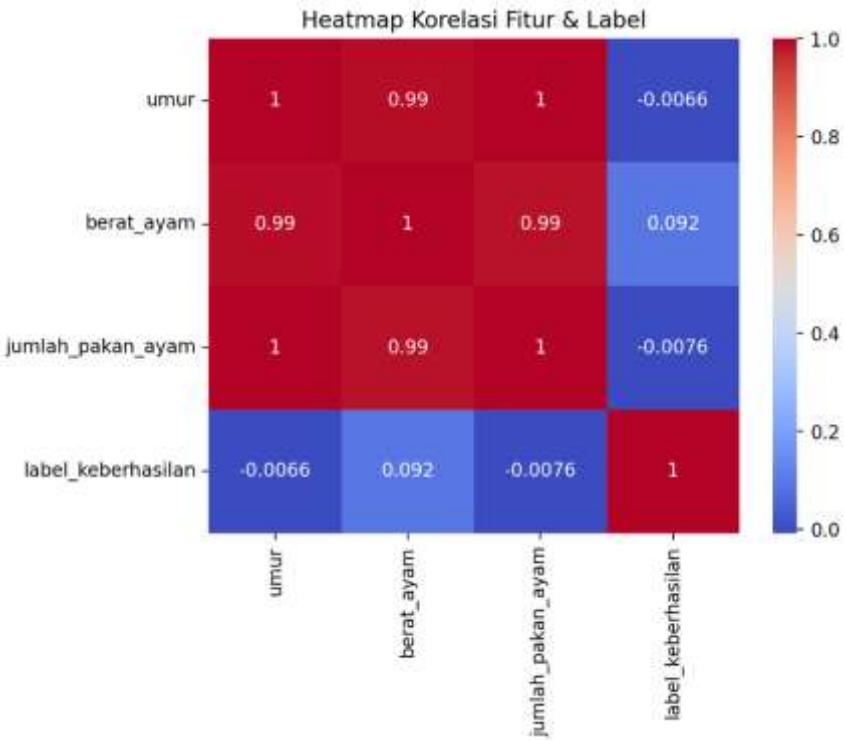
1. Analisis dan Persiapan data latih

Data latih diambil langsung dari tabel data_ayam di *database*. Jika data yang tersedia kurang dari 20 baris, aplikasi akan memberikan peringatan dan menghentikan proses lebih lanjut. Pada bagian ini, distribusi kelas (label_keberhasilan) dari data ditampilkan dalam bentuk tabel dan grafik batang. Jika distribusi label hanya mengandung satu kelas, aplikasi akan memberikan peringatan bahwa model tidak dapat dilatih.



Gambar 4. 15 Bagian Analisis dan Persiapan Data Latih

Selanjutnya, aplikasi menampilkan heatmap korelasi antar fitur untuk menunjukkan hubungan antara fitur-fitur numerik dalam dataset. Hal ini memberikan gambaran tentang sejauh mana fitur seperti umur, berat_ayam, dan jumlah_pakan_ayam berkorelasi dengan target label_keberhasilan.



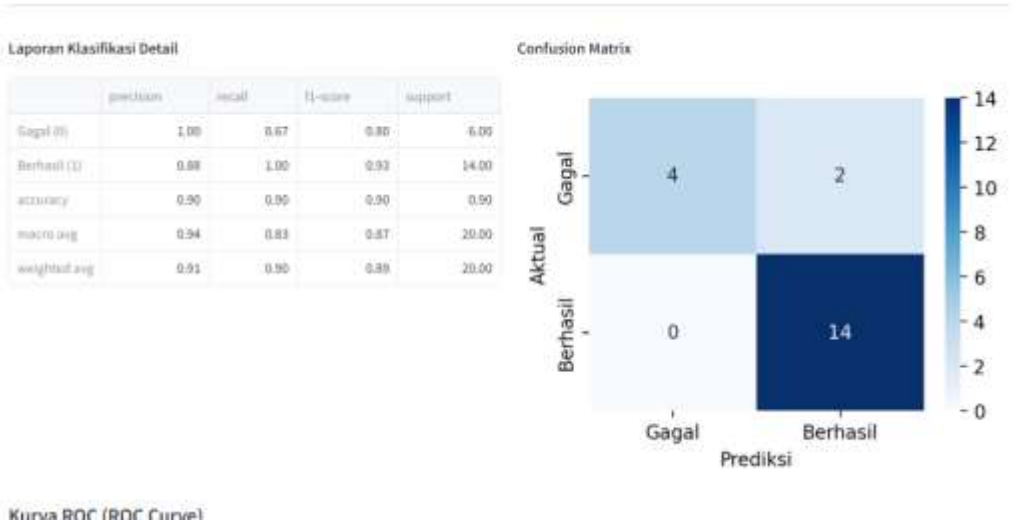
Gambar 4. 16 *Heatmap* Korelasi Fitur

Setelah memastikan bahwa data latih cukup dan memiliki distribusi kelas yang seimbang, proses pelatihan dimulai dengan membagi data menjadi dua set: data latih (80%) dan data uji (20%) menggunakan *train_test_split*. Selanjutnya, pipeline pelatihan dibuat dengan melakukan normalisasi menggunakan *MinMaxScaler*, diikuti dengan pelatihan model SVM dengan parameter yang telah ditentukan. Proses pelatihan ini dijalankan secara otomatis saat pengguna menekan tombol "Latih & Validasi Model dengan Parameter Terbaik". Aplikasi juga memberikan visualisasi distribusi kelas dan korelasi antar fitur sebelum memulai pelatihan.

Setelah pelatihan selesai, hasil evaluasi model ditampilkan dengan detail. Akurasi pada data latih dan uji dihitung untuk mengidentifikasi apakah model mengalami *overfitting*, dengan memberikan peringatan jika akurasi data latih jauh lebih tinggi dibandingkan dengan data uji. Laporan klasifikasi lengkap dengan metrik *precision*, *recall*, dan *f1-score* juga ditampilkan. Selain itu, *confusion matrix* dan kurva ROC disediakan untuk memvisualisasikan kinerja model.

Gambar 4. 17 Hasil Pelatihan dan Evaluasi Model

Gambar diatas menunjukkan hasil pelatihan dan evaluasi model SVM. Model menggunakan parameter seperti C = 100, kernel 'rbf', gamma = ‘auto’, dan degree = 2. Akurasi model pada data latih dan data uji sama-sama 90%, yang berarti model bekerja dengan baik tanpa overfitting, karena hasilnya konsisten pada kedua data tersebut. Pesan di akhir menunjukkan bahwa model cukup kuat dan efektif dalam memprediksi kelas.

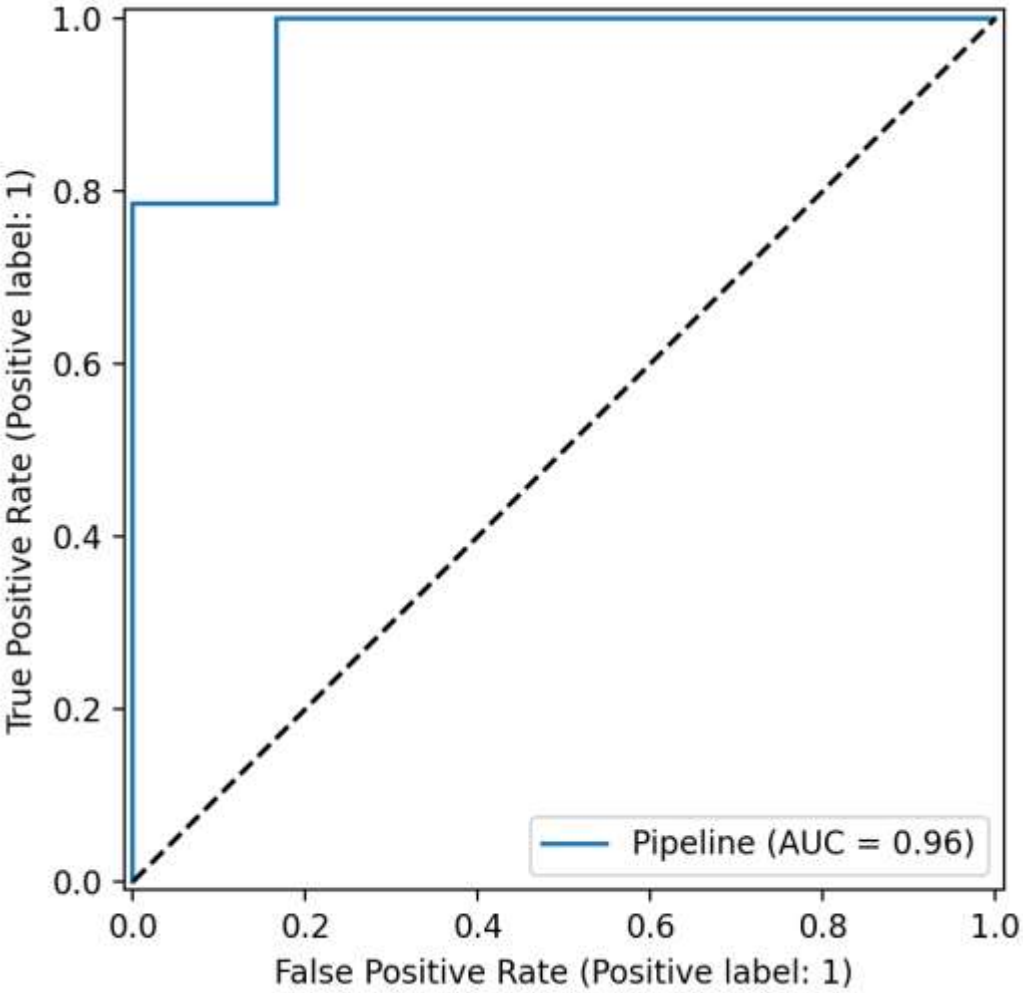


Gambar 4. 18 Hasil Laporan Klasifikasi

Gambar diatas menampilkan laporan klasifikasi dan confusion matrix dari model. Laporan klasifikasi detail menunjukkan metrik seperti *precision*, *recall*, dan *f1-score* untuk masing-masing kelas (Gagal dan Berhasil). Untuk kelas Gagal, *precision* adalah 1.00, *recall* 0.67, dan *f1-score* 0.80 dengan 6 contoh data. Sedangkan untuk kelas Berhasil, *precision* 0.88, *recall* 1.00, dan *f1-score* 0.93 dengan 14 contoh data. Akurasi keseluruhan model adalah 90%.

Confusion matrix menunjukkan hasil prediksi model: dari 6 data yang sebenarnya Gagal, model memprediksi 4 sebagai Gagal dan 2 sebagai Berhasil. Dari 14 data yang sebenarnya Berhasil, model memprediksi semuanya dengan

benar sebagai Berhasil (0 kesalahan). Ini menunjukkan bahwa model sangat akurat dalam memprediksi kelas Berhasil namun ada sedikit kesalahan dalam mengidentifikasi kelas Gagal.



Gambar 4. 19 Tampilan Diagram ROC Curve

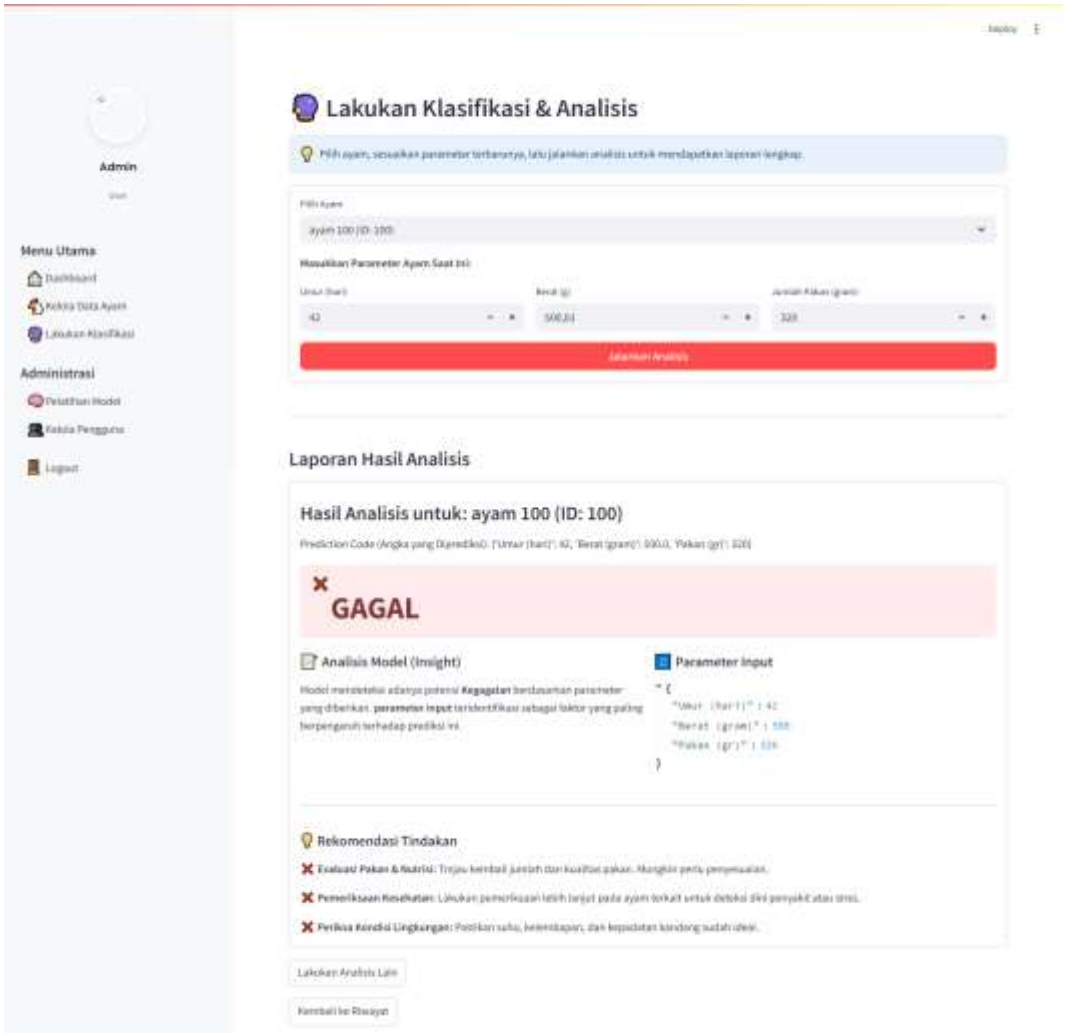
ROC curve siatas menunjukkan performa model klasifikasi dalam membedakan antara dua kelas (misalnya, Berhasil dan Gagal). Grafik ini menggambarkan hubungan antara *False Positive Rate* (FPR) di sumbu x dan *True Positive Rate (TPR)* di sumbu y. Garis putus-putus diagonal (AUC = 0.5) menunjukkan model acak, sementara garis biru menunjukkan kinerja model. Nilai AUC (*Area Under the Curve*) sebesar 0.96 menunjukkan bahwa model memiliki

kemampuan prediksi yang sangat baik, dengan AUC mendekati 1 menunjukkan pemisahan yang jelas antara kelas positif dan negatif. Model ini menunjukkan bahwa semakin rendah tingkat *False Positive*, semakin tinggi pula tingkat *True Positive*, yang menandakan prediksi model yang akurat.

Pengguna kemudian dapat memilih untuk menyimpan model yang telah dilatih menggunakan joblib atau melatih ulang model dengan parameter atau data lain jika diperlukan. Dengan cara ini, halaman pelatihan model ini menyediakan alur kerja yang jelas dan terstruktur untuk melatih, mengevaluasi, dan menyimpan model klasifikasi.

4.3.7 Tampilan Lakukan Klasifikasi

Bagian klasifikasi pada halaman ini memungkinkan pengguna untuk memilih ayam yang akan dianalisis berdasarkan ID dan memasukkan parameter seperti umur, berat, dan jumlah pakan ayam. Setelah parameter dimasukkan, pengguna dapat menekan tombol "Jalankan Analisis" untuk mendapatkan hasil klasifikasi. Hasil analisis menunjukkan prediksi model untuk ayam yang dipilih, dengan informasi apakah ayam tersebut diprediksi berhasil atau gagal berdasarkan parameter yang dimasukkan. Pada contoh ini, hasil prediksi menunjukkan status "Gagal" beserta analisis model yang menjelaskan faktor utama yang mempengaruhi prediksi tersebut. Selain itu, rekomendasi tindak lanjut disarankan, seperti evaluasi pakan, pemeriksaan kesehatan, dan pemeriksaan kondisi lingkungan untuk memastikan ayam dalam kondisi optimal.



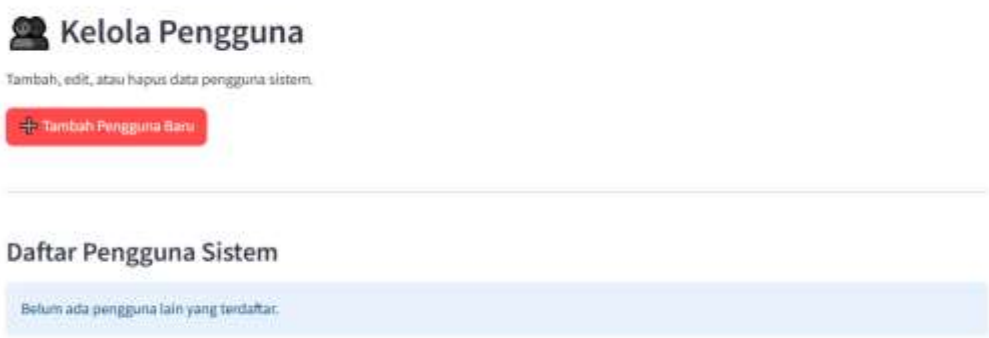
Gambar 4. 20 Tampilan Menu Klasifikasi

4.3.8 Tampilan Riwayat Klasifikasi

Halaman ini memungkinkan pengguna untuk melihat dan mengelola riwayat klasifikasi yang telah dilakukan. Setiap entri menampilkan nama ayam, hasil prediksi, dan opsi untuk menghapus riwayat. Pengguna juga dapat melakukan klasifikasi baru dengan menekan tombol "Lakukan Klasifikasi Baru".

Gambar 4. 21Tampilan Riwayat Klasifikasi

4.3.9 Tampilan Kelola Pengguna



Gambar 4. 22 Tampilan Kelola Pengguna

4.3.10 Tampilan Tambah Pengguna Baru

Halaman ini digunakan untuk menambahkan pengguna baru ke sistem. Pengguna diminta untuk memasukkan email pengguna dan password baru, kemudian dapat menyimpan data pengguna dengan menekan tombol "Simpan Pengguna". Jika pengguna ingin kembali ke daftar pengguna, mereka dapat menekan tombol "Kembali ke Daftar".

Gambar 4. 23 Tampilan Tambah Pengguna Baru

4.4 Pengujian Sistem Berbasis Web

Pengujian sistem dilakukan untuk memastikan bahwa aplikasi berbasis web yang dikembangkan dapat berjalan sesuai dengan fungsionalitas yang diharapkan dan memenuhi kebutuhan pengguna. Pengujian ini menggunakan metode *black box testing*, yang berfokus pada pengujian fungsionalitas dari perspektif pengguna tanpa memperhatikan struktur kode internal. Pengujian *black box* dilakukan dengan membuat skenario pengujian untuk setiap fitur utama pada aplikasi. Skenario ini mencakup langkah-langkah yang dilakukan pengguna, hasil yang diharapkan, dan hasil aktual dari pengujian.

Tabel 4. 4 Skenario Pengujian Blackbox Testing

No.	Skenario Pengujian	Langkah-langkah	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
1	Login Pengguna	1. Masukkan email dan password yang valid.2. Klik tombol "Login".3. Masukkan email atau password yang salah.4.	1. Pengguna berhasil masuk dan diarahkan ke halaman dashboard.2. Sistem menampilkan pesan kesalahan	1. Pengguna berhasil masuk ke dashboard.2. Pesan kesalahan ditampilkan sesuai harapan.	Berhasil

		Klik tombol "Login".	"Email atau password salah".		
2	Manajemen Data (CRUD)	1. Buka menu "Kelola Data Ayam".2. Klik "Tambah Data" dan isi formulir, lalu simpan.3. Pilih data dan klik "Edit", ubah informasi, lalu simpan.4. Pilih data dan klik "Hapus".	1. Data baru muncul di tabel.2. Informasi data berhasil diperbarui.3. Data terhapus dari tabel.	Semua fungsionalitas CRUD berjalan sesuai dengan yang diharapkan.	Berhasil
3	Pelatihan Model	1. Buka menu "Pelatihan Model".2. Klik tombol "Latih & Validasi Model".	Sistem menampilkan visualisasi data, melatih model, dan menyajikan hasil evaluasi (akurasi, confusion matrix, ROC).	Proses pelatihan dan evaluasi berjalan lancar dan menampilkan semua metrik yang relevan.	Berhasil
4	Klasifikasi Data Baru	1. Buka menu "Lakukan Klasifikasi".2. Pilih ID ayam dan masukkan data (umur, berat, pakan).3. Klik tombol "Jalankan Analisis".	Sistem menampilkan hasil prediksi ("Berhasil" atau "Gagal") beserta analisis dan rekomendasi.	Hasil prediksi, analisis faktor, dan rekomendasi ditampilkan dengan benar.	Berhasil
5	Manajemen Pengguna	1. Buka menu "Kelola Pengguna".2. Klik "Tambah Pengguna Baru", isi form, dan simpan.	Pengguna baru berhasil ditambahkan dan muncul dalam daftar pengguna.	Pengguna baru berhasil ditambahkan ke sistem.	Berhasil

Berdasarkan hasil pengujian *black box*, dapat disimpulkan bahwa seluruh fungsionalitas utama sistem klasifikasi berbasis web telah berjalan sesuai dengan rancangan dan tujuan. Sistem mampu menangani input pengguna, memproses data, melatih model, dan memberikan output yang akurat dan informatif.

4.5 Analisa Kelebihan dan Kelemahan Sistem

Berdasarkan pengembangan dan pengujian yang telah dilakukan, sistem klasifikasi ini memiliki sejumlah kelebihan yang dapat memberikan nilai tambah bagi pengguna, namun juga beberapa kelemahan yang perlu menjadi perhatian untuk pengembangan di masa depan.

4.5.1 Kelebihan Sistem

Kelebihan dari sistem yang dibuat adalah sebagai berikut :

1. Objektivitas dan Akurasi Tinggi: Sistem menyediakan alat bantu keputusan yang objektif berdasarkan data historis, mengurangi subjektivitas dalam penilaian manual. Dengan akurasi 90%, model ini memberikan tingkat kepercayaan yang tinggi dalam memprediksi keberhasilan produksi.
2. Antarmuka Pengguna yang Intuitif: Dibangun menggunakan *Streamlit*, aplikasi ini memiliki antarmuka yang sederhana dan mudah digunakan, bahkan oleh pengguna tanpa latar belakang teknis. Visualisasi data dan hasil yang jelas mempermudah pemahaman.
3. Manajemen Data Terpusat dan Mudah: Fitur CRUD (*Create, Read, Update, Delete*) memungkinkan pengguna untuk mengelola data ayam secara terpusat. Kemampuan untuk mengunggah data melalui file Excel juga meningkatkan efisiensi dalam pembaruan dataset.

4. Pendukung Keputusan Proaktif: Sistem tidak hanya memberikan label "Berhasil" atau "Gagal", tetapi juga menyajikan analisis faktor yang paling berpengaruh dan memberikan rekomendasi tindakan. Ini mengubah sistem dari alat pelaporan menjadi alat pendukung keputusan yang proaktif.
5. Fleksibilitas dan Kemampuan Adaptasi Model: Pengguna dapat melatih ulang model secara mandiri melalui aplikasi seiring dengan bertambahnya data baru. Fitur ini memastikan bahwa model dapat terus beradaptasi dan relevan dengan kondisi peternakan yang dinamis.

4.5.2 Kelemahan Sistem

Kelemahan dari sistem yang dibuat adalah sebagai berikut :

1. Ketergantungan pada Kualitas dan Kuantitas Data: Kinerja model sangat bergantung pada kualitas dan kuantitas data yang digunakan untuk pelatihan. Data yang tidak akurat, tidak lengkap, atau tidak representatif akan menghasilkan model dengan performa yang buruk.
2. Kinerja Tidak Seimbang pada Kelas Minoritas: Kelemahan paling signifikan adalah *recall* yang relatif rendah (67%) untuk kelas "Gagal". Ini berarti sistem berisiko tidak dapat mengidentifikasi sekitar sepertiga dari kasus ayam yang sebenarnya berada di jalur kegagalan, yang dapat menyebabkan kerugian finansial karena tidak adanya intervensi dini.
3. Keterbatasan Fitur yang Digunakan: Model saat ini hanya menggunakan tiga fitur (umur, berat, dan jumlah pakan). Faktor-faktor lain yang berpotensi memengaruhi keberhasilan produksi, seperti riwayat penyakit, kondisi lingkungan kandang (suhu, kelembapan), atau faktor genetik, belum

diikutsertakan. Penambahan fitur-fitur ini berpotensi meningkatkan akurasi dan keandalan model.

4. Memerlukan Pemeliharaan Teknis: Sebagai sistem berbasis perangkat lunak, aplikasi ini memerlukan pemeliharaan rutin, termasuk pembaruan *library*, pengelolaan server (jika di-*deploy* secara *online*), dan pencadangan data untuk memastikan operasional yang lancar dan aman.
5. Model Bersifat Statis Hingga Dilatih Ulang: Model tidak belajar secara real-time. Prediksi didasarkan pada versi model terakhir yang disimpan. Untuk memperbarui "pengetahuan" model, pengguna harus secara manual melakukan proses pelatihan ulang, yang mungkin tidak selalu dilakukan secara berkala.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil analisis, implementasi model, dan pengujian sistem yang telah dilakukan, berikut adalah Kesimpulan yang menjawab rumusan masalah penelitian :

1. Penelitian berhasil mengklasifikasikan Tingkat keberhasilan produksi ayam *broiler* di Timan Farm dengan menggunakan tiga indicator utama : umum ayam, berat ayam, dan jumlah pakan harian. Proses klasifikasi dilakukan melalui beberapa tahapan, yaitu persiapan data (termasuk normalisasi min-max untuk menyeragamkan skala fitur), pembagian *dataset* menjadi data latih 80% dan data uji 20%, serta penerapan algoritma *machine learning* untuk membedakan antara kategori Berhasil dan Gagal.
2. Algoritma SVM berhasil diterapkan untuk membangun model klasifikasi. Penerapan ini dilakukan dengan membangun sebuah *pipeline* pemodelan di *Python* yang mengintegrasikan normalisasi data dan klasifikasi menggunakan SVC (*Support Vector Classification*). Melalui proses optimasi hyperparameter menggunakan GridSearchCV, ditemukan bahwa model dengan kernel non-linear *Radial Basis Function* (RBF) memberikan performa terbaik, yang mengindikasikan bahwa hubungan antar fitur bersifat kompleks dan tidak dapat dipisahkan secara linear. Konfigurasi optimal yang ditemukan adalah C=100, kernel='rbf', dan gamma='auto'.
3. Model klasifikasi SVM yang dihasilkan menunjukkan tingkat akurasi yang tinggi dan andal. Pada data uji, model mencapai akurasi keseluruhan sebesar

90% dan skor ROC AUC 1.00, yang menandakan kemampuan pemisahan kelas yang sangat baik. Namun, analisis lebih dalam menunjukkan kinerja yang tidak seimbang: model mencapai *recall* 100% untuk kelas "Berhasil" (mampu mengidentifikasi semua kasus sukses) tetapi hanya 67% untuk kelas "Gagal" (melewatkan 33% kasus yang sebenarnya gagal). Meskipun presisi untuk kelas "Gagal" sempurna (100%), *recall* yang rendah menjadi titik kritis yang perlu diperhatikan.

5.2 Saran

Berdasarkan hasil penelitian dan keterbatasan yang ditemukan, beberapa saran untuk pengembangan sistem ke depan antara lain:

1. Peningkatan Kinerja Model untuk Kelas Minoritas: Untuk mengatasi rendahnya recall pada kelas "Gagal", perlu diterapkan teknik untuk menangani dataset tidak seimbang, seperti *oversampling* (misalnya SMOTE) atau *undersampling*. Selain itu, penyesuaian parameter *class_weight* pada algoritma SVM dapat membantu model lebih sensitif terhadap kelas yang jarang.
2. Penambahan Fitur Tambahan: Untuk meningkatkan akurasi, disarankan untuk menambahkan fitur-fitur lain yang mempengaruhi keberhasilan produksi, seperti riwayat penyakit, kondisi lingkungan, kualitas bibit, dan rasio konversi pakan (FCR).
3. Pengembangan Fungsionalitas Sistem: Aplikasi berbasis web dapat dikembangkan dengan fitur-fitur canggih seperti sistem peringatan dini untuk memprediksi kegagalan ayam dan dashboard analitik interaktif untuk

membantu perencanaan strategis. Integrasi dengan sensor IoT untuk pengambilan data lingkungan secara real-time juga dapat dipertimbangkan.

4. Eksplorasi Algoritma Lain: Meskipun SVM memberikan hasil yang baik, eksplorasi algoritma klasifikasi lain seperti Random Forest, Gradient Boosting, atau Neural Networks dapat dilakukan untuk membandingkan kinerja dan menemukan model yang lebih optimal sesuai dengan kebutuhan.

DAFTAR PUSAKA

Alfarizi, M. R. S., Al-farish, M. Z., Taufiqurrahman, M., Ardiansah, G., & Elgar, M. (2023). Penggunaan Python Sebagai Bahasa Pemrograman untuk Machine Learning dan Deep Learning. *Karya Ilmiah Mahasiswa Bertauhid (KARIMAH TAUHID)*, *2*(1), 1–6. https://ojs.unida.ac.id/karimahtauhid/article/view/7518/3480

Anto, A., Krismiyanto, L., & Mangisah, I. (2025). *PENAMBAHAN ENKAPSULASI EKSTRAK BUAH MENGKUDU, Zn DAN Cu PADA RANSUM TERHADAP ASUPAN PROTEIN DAN BOBOT DAGING AYAM BROILER*. *3*(1), 354–362. https://semnasfpp.uin-suska.ac.id/index.php/snipp/article/view/158/104

Basuki, B. (2023). *KLASIFIKASI TINGKAT KEBERHASILAN PRODUKSI AYAM BROILER DI RIAU MENGUNAKAN ALGORITMA K-NEAREST NEIGHTBOR*. https://repository.uin-suska.ac.id/75827/1/JURNAL REPOSITORY BENI BASUKI.pdf

Cahyaningtyas, C., Manongga, D., & Sembiring, I. (2022). Algorithm Comparison and Feature Selection for Classification of Broiler Chicken Harvest. *Jurnal Teknik Informatika (Jutif)*, *3*(6), 1717–1727. https://doi.org/10.20884/1.jutif.2022.3.6.493

Eyni Alfia, N., & Waseso, B. (2020). Perancangan Aplikasi Retensi Data Pada Database MySQL (Studi Kasus: PT. Telkomsigma). *Maret*, *2*(3), 2655–7541. https://jurnal.ikhafi.or.id/index.php/jusibi/364

Ginting, V. S., Kusrini, & Luthfi, E. T. (2020). *Keterlambatan Pembayaran Uang Sekolah Menggunakan Python*. *4*(1). https://core.ac.uk/download/pdf/328163876.pdf

Gusdiansyah, Z., Anggraeni, Baharun, A., & Rahmi, A. (2024). *Studi Kasus : Manajemen Pakan Ayam Petelur di Male Karya Prima Pasir Kerud Farm , Kabupaten Cianjur , Jawa Barat*. *3*, 6946–6952. https://ojs.unida.ac.id/karimahtauhid/article/view/13907/5386

Gustian, D., Suciati, I., & Saepudin, S. (2019). Sistem Pakar SISTEM PAKAR DENGAN ALGORITMA NAIVE BAYES UNTUK PREDIKSI HASIL PRODUKSI AYAM BROILER PLASMA (STUDI KASUS : PT.SEKAWAN SINAR SURYA). *Jurnal Riset Sistem Informasi Dan Teknologi Informasi (JURSISTEKNI)*, *1*(3), 31–42. https://doi.org/10.52005/jursistekni.v1i3.12

Hanief, R. A., & Purnama, nadia ika. (2024). Strategi Pemasaran Peternak Ayam Broiler Di Surya Endang Farm Kabupaten Deli Serdang. *Jurnal Ilmiah Manajemen, Ekonomi, & Akuntansi (MEA)*, *8*(2), 1441–1454. https://doi.org/10.31955/mea.v8i2.4166

Irawan, D., Perkasa, E. B., Yurindra, Y., Wahyuningsih, D., & Helmud, E. (2021). Perbandingan Klassifikasi SMS Berbasis Support Vector Machine, Naive Bayes Classifier, Random Forest dan Bagging Classifier. *Jurnal Sisfokom (Sistem Informasi Dan Komputer)*, *10*(3), 432–437. https://doi.org/10.32736/sisfokom.v10i3.1302

Kurniawan, R., Halim, A., & Melisa, H. (2023). *Prediksi Hasil Panen Pertanian*

Salak di Daerah Tapanuli Selatan Menggunakan Algoritma SVM (Support Vector Machine). *4*(2), 903–912. https://doi.org/10.30865/klik.v4i2.1246

Laila Sari, D., Saputra, M., & Gemasih, H. (2022). Penerapan Data Mining Dalam Proses Prediksi Perceraian Menggunakan Algoritma Naive Bayes Di Kabupaten Aceh Tengah. *Jurnal Teknik Informatika Dan Elektro*, *4*(1), 23–35. https://doi.org/10.55542/jurtie.v4i1.112

Lukman, & Herlinda. (2024). Prediksi Kelulusan Siswa dengan Metode Support Vector Machine (SVM) di SMK ADILUHUR. *STRING (Satuan Tulisan Riset Dan Inovasi Teknologi)*, *9*(1). https://journal.lppmunindra.ac.id/index.php/STRING/article/view/23355/6859

Mardi, Y. (2017). Data Mining : Klasifikasi Menggunakan Algoritma C4.5. *Edik Informatika*, *2*(2), 213–219. https://doi.org/10.22202/ei.2016.v2i2.1465

Maulana, H., & Al-Khowarizmi, A.-K. (2022). Analysis of the Effectiveness of Online Learning Using Eda Data Science and Machine Learning. *Sinkron*, *7*(1), 222–231. https://doi.org/10.33395/sinkron.v7i1.11206

Mestika, J. C., Selan, M. O., & Qadafi, M. I. (2023). Menjelajahi Teknik-Teknik Supervised Learning untuk Pemodelan Prediktif Menggunakan Python. *Buletin Ilmiah Ilmu Komputer Dan Multimedia (BIIKMA)*, *1*(1), 216–219. https://jurnalmahasiswa.com/index.php/biikma/article/view/101

Nas, C. (2021). Data Mining Prediksi Minat Calon Mahasiswa Memilih Perguruan Tinggi Menggunakan Algoritma C4.5. *Jurnal Manajemen Informatika (JAMIKA)*, *11*(2), 131–145. https://doi.org/10.34010/jamika.v11i2.5506

Nazar, R. (2024). Implementasi Pemrograman Python Menggunakan Google Colab. *Jurnal Informatika Dan Komputer (JIK)*, *15*(1), 50–56. https://journal.unmaha.ac.id/index.php/jik/article/view/253/249

Noperia, M., Ishak, I., & Sari, V. W. (2023). Implementasi Data Mining Pengelompokan Data Nilai Untuk Menentukan Minat Belajar Seni Budaya. *Megawati Noperia Ishak Vina Winda Sar*, *2*(1), 65. https://doi.org/10.53513/jursi.v2i1.5656

Nugroho, B. A., Pradana, A. K. A., & Nurfarida, E. (2021). Prediksi Waktu Kedatangan Pelanggan Servis Kendaraan Bermotor Berdasarkan Data Historis menggunakan Support Vector Machine. *Jurnal Edukasi Dan Penelitian Informatika (JEPIN)*, *7*(1), 25. https://doi.org/10.26418/jp.v7i1.42964

Pangestu, A., & Ridwan, T. (2022). Penerapan Data Mining Menggunakan Algoritma K-Means Pengelompokan Pelanggan Berdasarkan Kubikasi Air Terjual Menggunakan Weka. *JUST IT : Jurnal Sistem Informasi, Teknologi Informasi Dan Komputer*, *11*(3), 67–71. https://jurnal.umj.ac.id/index.php/just-it/article/view/11591

Patappari, A., Aksa, A. N., & Herdiansyah. (2024). *Sistem Pendukung Keputusan Identifikasi Penyakit Ayam Broiler*. *1*(2), 48–55. https://doi.org/10.25126/Rister

Permana, B. A. C., Sadali, M., & Ahmad, R. (2024). Penerapan Model Decision Tree Menggunakan Python Untuk Prediksi Faktor Dominan Penyebab Penyakit Stroke. *Infotek: Jurnal Informatika Dan Teknologi*, *7*(1), 23–31. https://doi.org/10.29408/jit.v7i1.23232

Permana, D. S., & Silvanie, A. (2021). Prediksi Penyakit Jantung Menggunakan Support Vector Machine dan Python pada Basis Data Pasien di Cleveland.

JUNIF: Jurnal Nasional Informatika, *2*(1), 29–34.

Pratama, R., Herdiansyah, M. I., Syamsuar, D., & Syazili, A. (2023). Prediksi Customer Retention Perusahaan Asuransi Menggunakan Machine Learning. *Jurnal Sisfokom (Sistem Informasi Dan Komputer)*, *12*(1), 96–104. https://doi.org/10.32736/sisfokom.v12i1.1507

Putra, A. B., & Nita, S. (2019). Perancangan dan Pembangunan Sistem Informasi E-Learning Berbasis Web (Studi Kasus Pada Madrasah Aliyah Kare Madiun). *Seminar Nasional Teknologi Informasi Dan Komunikasi 2019*, *1*(1), 81–85.

Putranto, A., Azizah, N. L., & Ratna Ika, A. I. (2023). Sistem Prediksi Penyakit Jantung Berbasis Web Menggunakan Metode SVM dan Framework Streamlit. *Jurnal Penerapan Sistem Informasi (Komputer & Manajemen)*, *4*(2), 442–452. https://archive.ics.uci.edu/ml/datasets/heart+disease

Putro, H. F., Vulandari, R. T., & Saptomo, W. L. Y. (2020). Penerapan Metode Naive Bayes Untuk Klasifikasi Pelanggan. *Jurnal Teknologi Informasi Dan Komunikasi (TIKomSiN)*, *8*(2). https://doi.org/10.30646/tikomsin.v8i2.500

Rahman, A. F. A., Sorikhi, & Wartulas, S. (2020). Prediksi Kelulusan Mahasiswa Menggunakan Algoritma C4.5 (Studi Kasus Di Universitas Peradaban). *Ijir*, *1*(2), 70–77. https://journal.peradaban.ac.id/index.php/ijir/article/view/635/521

Rahman, L. (2019). Sistem Informasi Geografis Tanah Bersertifikat Pada Desa Suluk Berbasis Website. *Prosiding Seminar Nasional Teknologi Informasi Dan Komunikasi (SENATIK)*, *2*(1), 37–44. http://prosiding.unipma.ac.id/index.php/SENATIK/article/view/1059

Rahmatillah, Z., Khairunisak, Firdus, & Allaily. (2024). Literatur Review : Potensi Bahan Lokal Alami di Indonesia Sebagai Pakan Ayam Broiler. *REKASATWA : Jurnal Ilmiah Peternakan*, *6*(2), 52–62. https://riset.unisma.ac.id/index.php/REKAPET/article/view/22770

Ramadhan, R. F., & Mukhaiyar, R. (2020). Penggunaan Database Mysql dengan Interface PhpMyAdmin sebagai Pengontrolan Smarthome Berbasis Raspberry Pi. *JTEIN: Jurnal Teknik Elektro Indonesia*, *1*(2), 129–134. https://doi.org/10.24036/jtein.v1i2.55

Rochmawati, D. R. (2024). Prediksi Cuaca Dengan Jaringan Syaraf Tiruan Menggunakan Python. *Jurnal Teknologi Komputer Dan Informatika*, *2*(2), 162–171. https://doi.org/10.59820/tekomin.v2i2.228

Romli, I., & Zy, A. T. (2020). Penentuan Jadwal Overtime Dengan Klasifikasi Data Karyawan Menggunakan Algoritma C4.5. *Jurnal Sains Komputer & Informatika (J-SAKTI*, *4*(2), 694–702.

Sholeh, M., Suraya, S., & Andayati, D. (2022). Machine Linear untuk Analisis Regresi Linier Biaya Asuransi Kesehatan dengan Menggunakan Python Jupyter Notebook. *Jurnal Edukasi Dan Penelitian Informatika (JEPIN)*, *8*(1), 20. https://doi.org/10.26418/jp.v8i1.48822

Sidik, A. D., & Ansawarman, A. (2022). Prediksi Jumlah Kendaraan Bermotor Menggunakan Machine Learning. *Formosa Journal of Multidisciplinary Research*, *1*(3), 559–568. https://doi.org/10.55927/fjmr.v1i3.745

Sitanggang Rianto, Urian Dachi Teddy, & Manurung H G Immanuel. (2022). Rancang Bangun Sistem Penjualan Tanaman Hiasberbasis Web Menggunakan Php Dan Mysql. *Tekesnos*, *4*(1), 84–90.

Sitorus, C. M., Rizal, A., & Jajuli, M. (2020). Prediksi Risiko Perjalanan Transportasi Online Dari Data Telematik Menggunakan Algoritma Support

Vector Machine. *Jurnal Teknik Informatika Dan Sistem Informasi*, *6*(2), 254–265. https://doi.org/10.28932/jutisi.v6i2.2672

Subowo Edy, S. M. (2020). SISTEM INFORMASI PETERNAKAN AYAM BROILER DI KABUPATEN PEKALONGAN BERBASIS WEB DAN ANDROID | Jurnal Surya Informatika: Membangun Informasi dan Profesionalisme. *Jurnal Surya Informatika: Membangun Informasi Dan Profesionalisme*, *6*(June), 53–65 , 53–65. https://www.researchgate.net/profile/Edy-Subowo/publication/342345365_SISTEM_INFORMASI_PETERNAKAN_AYAM_BROILER_DI_KABUPATEN_PEKALONGAN

T, N. (2019). Analisis Performans Ayam Broilerpada Kandang Tertutupdan Kandang Terbuka. *Jurnal Peternakan Nusantara*, *5*, 77–85. https://core.ac.uk/download/pdf/270148697.pdf

Telaumbanua, F. D., Hulu, P., & Dharma, T. Z. N. R. R. L. A. (2020). Penggunaan Machine Learning Di Bidang Kesehatan. *Jurnal Penelitian Teknik Informatika Universitas Prima Indonesia (UNPRI) Medan*, *2*(2), 391–399. https://doi.org/10.34012/jutikomp.v2i2.657

Trisnawati, A., Surani, D., Sri Kurniawan, B., & Fidriyanto, A. (2024). Analysis of Material Understanding Usings Google Colaboratory in Class X Informarics Subject at SMAN 5 Serang. *Cakrawala Pedagogik (Jurnal Pendidikan Dan Penelitian)*, *8*(1), 38–45. https://stkip.syekhmanshur.ac.id/jurnal/index.php/CP/article/view/627/233

Ucha Putri, S., Irawan, E., & Rizky, F. (2021). Implementasi Data Mining Untuk Prediksi Penyakit Diabetes Dengan Algoritma C4.5. *Januari*, *2*(1), 39–46. https://www.pkm.tunasbangsa.ac.id/index.php/kesatria/article/view/56/56

Vebriani, M., & Yustanti, W. (2024). Klasifikasi Deteksi Link Phising DANA Kaget Menggunakan Metode Support Vector Machine Berbasis Website. *Journal of Informatics and Computer Science*, *06*, 411–412. https://danakagetvezridd.

Veronica, A., Ernawati, Rasdiana, Abas, M., Yusriani, Hadawiah, Hidayah, N., Sabtohadi, J., Marlina, H., Mulyani, W., & Zulkarnaini. (2022). Metodologi Penelitian Kuantitatif. In *Pt. Global Eksekutif Teknologi*. https://repository.umi.ac.id/1989/2/Metodologi Penelitian Kuantitatif.pdf

Wafa, H. S., Id Hadiana, A., & Rakhmat Umbara, F. (2022). Prediksi Penyakit Diabetes Menggunakan Algoritma Support Vector Machine (SVM). *Informatics and Digital Expert (INDEX)*, *4*(1), 40–45. https://doi.org/10.36423/index.v4i1.895

Wahyudi, N., Albar, R., & Rizki, Y. (2024). *ANALISIS EFEKTIVITAS METODE SUPPORT VECTOR MACHINE (SVM) DALAM MENGURANGI BERITA HOAX BENCANA ALAM DARI DATA TWITTER ANALYSIS OF THE EFFECTIVENESS OF THE SUPPORT VECTOR MACHINE (SVM) METHOD IN REDUCING HOAX NEWS ON NATURAL*. *10*(1).

Widi Hastomo, Nur Aini, Adhitio Satyo Bayangkari Karno, & L.M. Rasdi Rere. (2022). Metode Pembelajaran Mesin untuk Memprediksi Emisi Manure Management. *Jurnal Nasional Teknik Elektro Dan Teknologi Informasi*, *11*(2), 131–139. https://doi.org/10.22146/jnteti.v11i2.2586

Wulandari, T., & Nurmiati, S. (2022). Rancang Bangun Sistem Pemesanan Wedding Organizer Menggunakan Metode Rad di Shofia Ahmad Wedding.

Jurnal Rekasaya Informasi, *11*(69), 79–85.
Zein, S., & Gunawan, G. (2022). Prediksi Hasil FIFA World Cup Qatar 2022 Menggunakan Machine Learning dengan Python. *Jurnal Riset Matematika*, 153–162. https://doi.org/10.29313/jrm.v2i2.1382

LAMPIRAN

Lampiran 1. Pengambilan Dataset Ayam Broiler

Lampiran 2. Surat Penetapan Dosen Pembimbing

MAJELIS PENDIDIKAN TINGGI PENELITIAN & PENGEMBANGAN PIMPINAN PUSAT MUHAMMADIYAH

UNIVERSITAS MUHAMMADIYAH SUMATERA UTARA

FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI

UMSU Terakreditasi A Berdasarkan Keputusan Badan Akreditasi Nasional Perguruan Tinggi No. 89/SK/BAN-PT/Akred/PT/III/2019

Pusat Administrasi: Jalan Mukhtar Basri No. 3 Medan 20238 Telp. (061) 6622400 - 66224567 Fax. (061) 6625474 - 6631003

umsumedan umsumedan umsumedan umsumedan

PENETAPAN DOSEN PEMBIMBING
PROPOSAL/SKRIPSI MAHASISWA
NOMOR : 964/II.3-AU/UMSU-09/F/2024

Assalamu'alaikum Warahmatullahi Wabarakatuh

Dekan Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Muhammadiyah Sumatera Utara, berdasarkan Persetujuan permohonan judul penelitian Proposal / Skripsi dari Ketua / Sekretaris.

Program Studi : **Sistem Informasi**
Pada tanggal : **16 Desember 2024**

Dengan ini menetapkan Dosen Pembimbing Proposal / Skripsi Mahasiswa.

Nama : **Siti Aisah Purba**
NPM : **2109010017**
Semester : **VII (Tujuh)**
Program studi : **Sistem Informasi**
Judul Proposal / Skripsi : **Prediksi Tingkat Keberhasilan Produksi Ayam Boiler Menggunakan Algoritma Support Vector Machine (SVM) Di PT.Japfa Bandar Masilam**

Dosen Pembimbing : **Dr. Al-Khowarizmi,S.Kom.,M.Kom.**

Dengan demikian di izinkan menulis Proposal / Skripsi dengan ketentuan

1. Penulisan berpedoman pada buku panduan penulisan Proposal / Skripsi Fakultas Ilmu Komputer dan Teknologi Informasi UMSU
2. Pelaksanaan Sidang Skripsi harus berjarak 3 bulan setelah dikeluarkannya Surat Penetapan Dosen Pembimbing Skripsi.
3. **Proyek Proposal / Skripsi** dinyatakan " **BATAL** " bila tidak selesai sebelum Masa Kadaluarsa tanggal : **16 Desember 2025**
4. Revisi judul……

Wassalamu'alaikum Warahmatullahi Wabarakatuh.

Ditetapkan di : Medan
Pada Tanggal : 15 Jumadil Akhir 1446 H
16 Desember 2024 M

Dekan

Dr. Al-Khowarizmi,S.Kom.,M.Kom.
NIDN : 0127099201

Cc. File

Lampiran 3. Surat Balasan Izin Penelitian

Timan Farm

Jln. Sangnawaluh Perdagangan II, Kec. Bandar, Kab. Simalungun
Telp: +62 85358232107

Nomor : 091/TFM/IV/2025

Perihal : **Balasan Permohonan Izin Riset**

Perdagangan, 30 April 2025

Kepada Yth : Dekan
Universitas Muhammadiyah
Sumatera Utara
Fakultas Ilmu Komputer dan
Teknologi Informasi

Tempat

Dengan Hormat,
Yang bertanda tangan di bawah ini:

Nama	: Timan
Jabatan	: Pemilik Peternakan

Menerangkan bahwa:

Nama	: Siti Aisah Purba
NPM	: 2109010017
Program Studi	: Sistem Informasi
Semester	: VIII (Delapan)

Telah kami setujui untuk mengadakan kegiatan riset pada Timan Farm Kecamatan Bandar Kabupaten Simalungun sebagai syarat penyusunan skripsi dengan judul:

"KLASIFIKASI TINGKAT KEBERHASILAN PRODUKSI AYAM BROILER MENGGUNAKAN ALGORITMA SUPPORT VECTOR MACHINE (SVM)".

Demikian surat ini kami sampaikan atas perhatian serta kerja samanya kami ucapkan terima kasih.

Hormat Kami,

Pemilik Peternakan

TIMAN

Lampiran 4. Lembar Bimbingan

MAJELIS PENDIDIKAN TINGGI PENELITIAN & PENGEMBANGAN PIMPINAN PUSAT MUHAMMADIYAH

UNIVERSITAS MUHAMMADIYAH SUMATERA UTARA

FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI

UMSU Terakreditasi A Berdasarkan Keputusan Badan Akreditasi Nasional Perguruan Tinggi No. 89/SK/BAN-PT/Akred/PT/III/2019

Pusat Administrasi: Jalan Muhtar Basri No. 3 Medan 20238 Telp. (061) 6622400 - 66224567 Fax. (061) 6625474 - 6631003

umsumedan umsumedan umsumedan umsumedan

Berita Acara Pembimbingan Skripsi

Nama Mahasiswa : Siti Aisah Purba
NPM : 2109010017
Nama Dosen Pembimbing :

Program Studi : Sistem Informasi
Konsentrasi :
Judul Penelitian : Klasifikasi tingkat kebernasilan produksi ayam broiler menggunakan algoritma support vector machine

Item	Hasil Evaluasi	Tanggal	Paraf Dosen
	Revisi Judul	10/02/2025	
	Revisi bab 1	10/02/2025	
	Revisi bab 2	12/02/2025	
	Revisi bab 3 (dataset)	14/02/2025	
	ACC Seminar Proposal	18/02/2025	
	Revisi Program	14/05/2025	
	Revisi bab 4,5	19/06/2025	
	ACC Sidang	3/07/2025	

Medan, 11 Juli 2025

Diketahui oleh :
Ketua Program Studi
Sistem Informasi

(Martiano S.Pd, S.Kom.,M.Kom)

Disetujui oleh :
Dosen Pembimbing

(Dr.Al-Khowarizmi,S.Kom.,M.Kom)