BAB III
METODE PENELITIAN

3.1 Objek Penelitian

Objek dalam penelitian ini adalah proses klasifikasi penyakit kambing berdasarkan kombinasi data gejala dan data klinis menggunakan algoritma *Random Forest*. Penelitian ini berfokus pada bagaimana teknik *machine learning* dapat digunakan untuk mengelompokkan berbagai jenis penyakit kambing secara otomatis berdasarkan pola yang muncul pada data, seperti suhu tubuh, umur, dan gejala-gejala yang tercatat. Dataset yang digunakan diperoleh dari *platform kaggle*, yang menyediakan kumpulan data penyakit ternak dengan karakteristik gejala yang cukup beragam sehingga sesuai untuk tujuan pemodelan. Pemilihan objek ini didasarkan pada kebutuhan untuk meningkatkan ketepatan identifikasi penyakit, karena di lapangan proses diagnosis masih banyak bergantung pada pengamatan subjektif dan seringkali sulit dilakukan ketika beberapa penyakit memiliki gejala yang mirip. Dengan demikian, penelitian ini bertujuan menghasilkan model klasifikasi yang mampu melakukan prediksi jenis penyakit secara lebih konsisten berdasarkan data.

Seluruh proses penelitian dilakukan secara daring dengan memanfaatkan dataset yang tersedia pada *platform kaggle* tersebut. Proses analisis dilakukan secara terstruktur agar informasi terkait gejala dan kondisi klinis kambing dapat diproses dengan tepat sesuai kebutuhan penelitian. Pendekatan ini mencakup tahapan pengumpulan data, pra-pemrosesan, transformasi fitur, pelatihan model, hingga evaluasi model hasil klasifikasi.

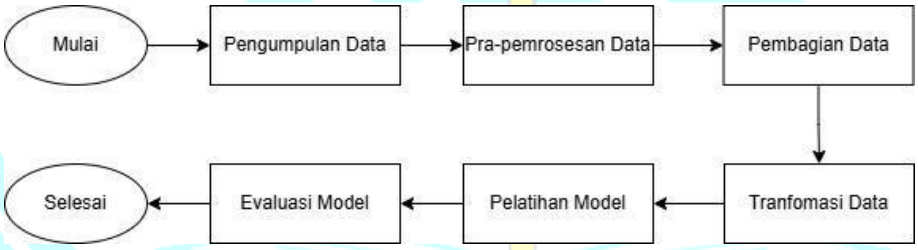
Kegiatan penelitian dilakukan secara bertahap dan terencana, mulai dari persiapan hingga penyusunan hasil akhir. Perencanaan waktu dilakukan agar setiap tahapan penelitian dapat berjalan secara sistematis, cermat, dan sesuai dengan tujuan yang telah ditetapkan, sehingga hasil yang diperoleh dapat dipertanggungjawabkan secara ilmiah. Adapun waktu pelaksanaan penelitian dapat dilihat pada Tabel 3.1 berikut:

Tabel 3. 1 Jadwal Pelaksanaan Penelitian

No	Kegiatan Penelitian	1	2	3	4	5
1.	Studi Literatur dan Pengumpulan Referensi					
2.	Pengumpulan Dataset					
3.	Pra-Pemrosesan Data					
4.	Pembagian Data					
5.	Transformasi Data					
6.	Pelatihan Model					
7.	Evaluasi Model					
8.	Penyusunan Laporan					

3.2 Prosedur Penelitian

Prosedur penelitian ini disusun secara sistematis dan bertahap untuk memastikan langkah-langkah yang dilakukan untuk proses klasifikasi penyakit kambing menggunakan algoritma *Random Forest*. Diagram alur prosedur penelitian ditunjukkan pada Gambar 3.1:



Gambar 3. 1 Diagram Alur Penelitian

(Sumber: Penulis, 2025)

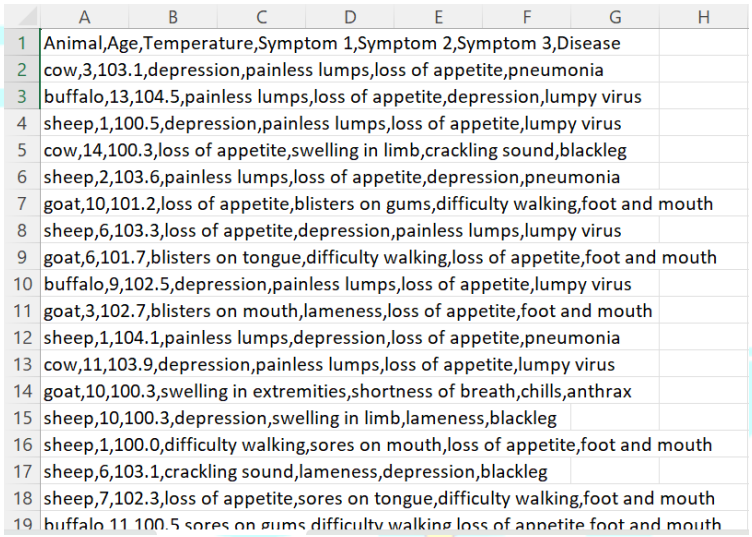
3.2.1 Pengumpulan Data

Data Data yang digunakan dalam penelitian ini bersumber dari platform Kaggle melalui dataset berjudul "*Livestock Symptoms and Diseases*" https://www.kaggle.com/datasets/researcher1548/livestock-*Symptoms*-and-diseases yang dapat diakses secara terbuka oleh publik. Dataset tersebut berisi data gejala klinis, kondisi fisik, dan diagnosis penyakit pada berbagai jenis hewan ternak yang dikumpulkan dan dipublikasikan sebagai data sekunder untuk keperluan penelitian dan pembelajaran.

Sebagai dataset publik, data ini telah digunakan oleh berbagai pengguna untuk beragam tujuan analisis, seperti eksplorasi data, pengembangan model klasifikasi penyakit ternak, serta studi penerapan algoritma machine learning. Namun demikian, penggunaan data dalam setiap penelitian dapat menghasilkan

temuan yang berbeda karena dipengaruhi oleh fokus objek penelitian, tahapan pengolahan data, pemilihan metode, serta strategi analisis yang diterapkan.

Pada penelitian ini, dataset dimanfaatkan sebagai sumber data mentah yang selanjutnya diproses melalui tahapan seleksi, pra-pemrosesan, dan pemodelan sesuai dengan tujuan penelitian, yaitu klasifikasi penyakit kambing. Dengan demikian, meskipun dataset bersifat umum dan telah digunakan sebelumnya, proses analisis dan hasil yang diperoleh dalam penelitian ini bersifat berbeda dan dapat dipertanggungjawabkan secara ilmiah.



	A	B	C	D	E	F	G	H
1	Animal,Age,Temperature,Symptom 1,Symptom 2,Symptom 3,Disease							
2	cow,3,103.1,depression,painless lumps,loss of appetite,pneumonia							
3	buffalo,13,104.5,painless lumps,loss of appetite,depression,lumpy virus							
4	sheep,1,100.5,depression,painless lumps,loss of appetite,lumpy virus							
5	cow,14,100.3,loss of appetite,swelling in limb,crackling sound,blackleg							
6	sheep,2,103.6,painless lumps,loss of appetite,depression,pneumonia							
7	goat,10,101.2,loss of appetite,blisters on gums,difficulty walking,foot and mouth							
8	sheep,6,103.3,loss of appetite,depression,painless lumps,lumpy virus							
9	goat,6,101.7,blisters on tongue,difficulty walking,loss of appetite,foot and mouth							
10	buffalo,9,102.5,depression,painless lumps,loss of appetite,lumpy virus							
11	goat,3,102.7,blisters on mouth,lameness,loss of appetite,foot and mouth							
12	sheep,1,104.1,painless lumps,depression,loss of appetite,pneumonia							
13	cow,11,103.9,depression,painless lumps,loss of appetite,lumpy virus							
14	goat,10,100.3,swelling in extremities,shortness of breath,chills,anthrax							
15	sheep,10,100.3,depression,swelling in limb,lameness,blackleg							
16	sheep,1,100.0,difficulty walking,sores on mouth,loss of appetite,foot and mouth							
17	sheep,6,103.1,crackling sound,lameness,depression,blackleg							
18	sheep,7,102.3,loss of appetite,sores on tongue,difficulty walking,foot and mouth							
19	buffalo,11,100.5,sores on gums,difficulty walking,loss of appetite,foot and mouth							

Gambar 3. 2 Pengumpulan Dataset

(Sumber: Penulis, 2025)

3.2.2 Pra-Pemrosesan Data

Pra-pemrosesan data dilakukan untuk memastikan bahwa data berada dalam kondisi bersih, konsisten, dan siap digunakan pada tahap pemodelan. Tahapan ini penting karena kualitas data sangat berpengaruh terhadap performa algoritma *machine learning* yang digunakan. Pada penelitian ini, pra-pemrosesan diterapkan pada dataset yang telah dikumpulkan, dengan tujuan menyesuaikan data agar sesuai dengan fokus penelitian serta mempersiapkan untuk proses klasifikasi penyakit kambing. Adapun langkah-langkah pra-pemrosesan yang dilakukan adalah sebagai berikut:

1. Seleksi Data

Seleksi data dilakukan untuk memastikan bahwa data yang digunakan sesuai dengan fokus penelitian, yaitu klasifikasi penyakit pada kambing. Pada tahap ini,

dataset awal yang mencakup berbagai jenis hewan ternak disaring berdasarkan kategori jenis hewan, sehingga hanya data kambing yang digunakan untuk dianalisis lebih lanjut.

Proses ini bertujuan untuk menghindari perbedaan karakteristik penyakit antarjenis ternak yang berpotensi memengaruhi pola data serta menurunkan kinerja model klasifikasi. Dengan membatasi data hanya pada kambing, model dapat mempelajari pola gejala dan data klinis yang lebih spesifik, sehingga hasil klasifikasi menjadi lebih relevan dan tidak bias akibat pencampuran data dari hewan lain. Setelah proses seleksi dilakukan, jumlah data berkurang dari 43.778 baris menjadi 10.628 baris, dengan jumlah atribut tetap sebanyak 7 kolom. Data hasil seleksi ini selanjutnya digunakan pada tahap pra-pemrosesan lanjutan dan pemodelan klasifikasi penyakit kambing.

2. Penanganan Nilai Hilang (*Missing Value*)

Pada tahap penanganan nilai hilang, dataset diperiksa terlebih dahulu untuk mengetahui apakah terdapat data kosong pada fitur numerik maupun kategorikal. Nilai hilang tidak dihapus, melainkan diatasi melalui proses imputasi agar tidak mengurangi jumlah data yang tersedia. Untuk fitur numerik seperti *Age* dan *Temperature*, nilai kosong diisi menggunakan nilai median karena metode ini lebih stabil terhadap keberadaan outlier dan dapat mempertahankan distribusi data. Sementara itu, nilai hilang pada fitur gejala diisi dengan label "*unknown*" sebagai kategori penanda agar informasi tetap tercatat tanpa menghilangkan baris data. Proses ini memastikan bahwa dataset berada dalam kondisi lengkap dan dapat diproses oleh algoritma pada tahap pemodelan.

Tabel 3. 2 Contoh Data Sebelum Penanganan Nilai Hilang

Usia	Suhu Tubuh	Gejala 1	Gejala 2	Gejala 3	Penyakit
3	*NaN*	*fever*	*coughing*	*NaN*	*pneumonia*
NaN	39.1	*Loss of appetite*	*NaN*	*Swelling leg*	*Blackleg*
4	40.0	*NaN*	*chillls*	*Sores on tongue*	*Foot and mouth*

Pada tabel 3.2 menunjukkan contoh data sebelum dilakukan penanganan nilai hilang. Terlihat bahwa beberapa kolom, seperti Suhu Tubuh, Usia, dan beberapa gejala, masih memiliki nilai kosong. Kondisi ini dapat mengganggu proses pemodelan karena algoritma tidak dapat memproses nilai kosong secara langsung.

Tabel 3. 3 Contoh data Setelah Penanganan Nilai Hilang

Usia	Suhu Tubuh	Gejala 1	Gejala 2	Gejala 3	Penyakit
3	39.5	*fever*	*coughing*	*unknown*	*pneumonia*
3	39.1	*Loss of appetite*	*unknown*	*Swelling leg*	*Blackleg*
4	40.0	*unknown*	*chillls*	*Sores on tongue*	*Foot and mouth*

Pada tabel 3.3 menampilkan data setelah nilai kosong ditangani. Nilai hilang pada kolom numerik diisi menggunakan *median*, yaitu nilai tengah yang dihitung dari seluruh data pada kolom tersebut setelah diurutkan, sehingga nilai pengganti tidak dipilih secara sembarangan. Sementara itu, nilai kosong pada kolom gejala diganti dengan label "*unknown*" karena fitur gejala bersifat kategorikal dan tidak dapat dihitung *median* atau *mean*-nya. Penggantian ini dilakukan untuk menandai bahwa informasi gejala tidak tercatat tanpa menghapus baris data. Dengan demikian, proses imputasi dilakukan secara sistematis dan bukan berdasarkan perkiraan, sehingga data menjadi lengkap dan siap digunakan untuk tahap pemodelan.

3. Pembersihan Duplikasi Data

Setelah proses penanganan nilai hilang selesai, dataset diperiksa untuk memastikan bahwa tidak terdapat baris yang muncul lebih dari satu kali. Baris yang terduplikasi perlu dihapus karena dapat membuat model mempelajari pola yang sama secara berulang, sehingga memengaruhi keseimbangan data dan mengurangi kemampuan model dalam mengenali variasi kasus yang berbeda. Dengan menghilangkan data yang duplikat, proses pembelajaran menjadi lebih objektif dan hasil model tidak terpengaruh oleh pengulangan data.

Tabel 3. 4 Contoh Data Sebelum Penghapusan Duplikasi

Usia	Suhu Tubuh	Gejala 1	Gejala 2	Gejala 3	Penyakit
3	39.5	*fever*	*coughing*	*Loss of appetite*	*pneumonia*
3	39.5	*fever*	*coughing*	*Loss of appetite*	*pneumonia*
4	40.0	*weakness*	*chillls*	*Sores on tongue*	*Foot and mouth*

Pada tabel 3.4 memperlihatkan adanya baris data yang sama, seperti pada baris pertama dan kedua yang memiliki nilai yang sama pada seluruh kolom. Baris

seperti ini disebut duplikasi dan dapat menimbulkan bias karena model akan menganggap pola tersebut lebih dominan daripada yang sebenarnya.

Tabel 3. 5 Contoh Data Setelah Penghapusan Duplikasi

Usia	Suhu Tubuh	Gejala 1	Gejala 2	Gejala 3	Penyakit
3	39.5	*fever*	*coughing*	*Loss of appetite*	*pneumonia*
4	40.0	*weakness*	*chillls*	*Sores on tongue*	*Foot and mouth*

Pada tabel 3.5 menujukkan data setelah proses pembersihan, hanya satu baris yang mewakili data yang sebelumnya duplikat. Tabel 3.5 menunjukkan bahwa dataset kini lebih bersih dan setiap baris data bersifat unik. Dengan demikian, model dapat mempelajari variasi data secara lebih seimbang tanpa dipengaruhi oleh pengulangan baris yang tidak perlu.

4. Penentuan Fitur dan Label

Setelah data melalui tahap pra-pemrosesan, langkah selanjutnya adalah penentuan fitur (X) dan label (y) yang digunakan dalam proses pemodelan. Penentuan fitur dilakukan melalui pendekatan *feature selection* berbasis pengetahuan domain, yaitu dengan mempertimbangkan relevansi klinis setiap variabel terhadap diagnosis penyakit kambing. Pendekatan ini dipilih karena jumlah fitur yang digunakan relatif sedikit dan seluruhnya memiliki keterkaitan langsung dengan kondisi kesehatan ternak.

Fitur yang digunakan dalam penelitian ini terdiri dari *Age* dan *Temperature* sebagai fitur numerik, serta *Symptom* 1, *Symptom* 2, dan *Symptom* 3 sebagai fitur kategorikal. Variabel-variabel tersebut dipilih karena secara umum digunakan dalam praktik diagnosis penyakit kambing dan mampu merepresentasikan kondisi fisik serta gejala klinis yang muncul pada ternak. Sementara itu, kolom Animal tidak disertakan sebagai fitur karena data telah difilter sehingga hanya mencakup kambing, sehingga kolom tersebut tidak lagi memberikan informasi diskriminatif bagi proses klasifikasi.

Label atau variabel target (y) ditetapkan pada kolom *Disease*, yang merepresentasikan jenis penyakit kambing yang akan diprediksi oleh model. Pemilihan fitur dan label ini disesuaikan dengan tujuan penelitian, yaitu membangun model klasifikasi penyakit kambing yang relevan secara klinis dan konsisten dengan data yang digunakan.

3.2.3 Pembagian Data

Pembagian dataset merupakan tahap penting dalam proses machine learning untuk memastikan model mampu melakukan generalisasi terhadap data baru. Berbagai penelitian menunjukkan bahwa rasio pembagian data memiliki pengaruh terhadap performa model. Menurut penelitian oleh (Bichri et al., 2024), variasi rasio *train–test split* seperti 60:40, 70:30, 80:20, 85:15 hingga 90:10 dapat menghasilkan performa yang berbeda, di mana penggunaan data latih lebih dari 80% cenderung memberikan hasil yang lebih baik.

Hal ini sejalan dengan penelitian (Fahrudin et al., 2024), yang menyatakan bahwa pembagian data umumnya berada pada rentang 70% hingga 80% untuk data latih dan 20% hingga 30% untuk data uji, karena proporsi tersebut mampu menjaga keseimbangan antara proses pembelajaran model dan evaluasi performa, sehingga semakin besar proporsi data latih, maka model cenderung memiliki performa yang lebih baik karena mampu mempelajari pola data secara lebih optimal.

Penelitian oleh (Suci Amaliah et al., 2022) menerapkan skema yang sama dan menunjukkan bahwa pembagian ini efektif untuk menghasilkan model yang terlatih dengan baik sekaligus memberikan evaluasi yang akurat. Temuan yang sejalan pada penelitian (Indaryono et al., 2024) yang menunjukkan hasil evaluasi yang konsisten menggunakan pembagian 80% data latih dan 20% data uji.

Berdasarkan pertimbangan tersebut, pada penelitian ini dataset dibagi menggunakan metode *train–test split* dengan dua variasi pembagian data, yaitu 80% data latih dan 20% data uji, serta 85% data latih dan 15% data uji. Kedua variasi tersebut akan digunakan dalam proses pelatihan dan pengujian model untuk melihat pengaruh perbedaan proporsi data latih terhadap kinerja model.

Pemilihan rasio 80:20 didasarkan pada pertimbangan keseimbangan antara data pelatihan dan data pengujian, sedangkan rasio 85:15 digunakan untuk memberikan tambahan data latih sehingga model memiliki kesempatan belajar yang lebih banyak. Dengan penggunaan kedua variasi tersebut, diharapkan model yang dihasilkan dapat memberikan performa yang optimal.

Proses pembagian data dilakukan menggunakan teknik stratified sampling agar proporsi setiap kelas penyakit tetap terjaga pada data latih maupun data uji. Pendekatan ini penting untuk memastikan seluruh kategori penyakit terwakili

secara proporsional sehingga evaluasi performa model dapat dilakukan secara lebih tepat. Selain itu, pembagian data dilakukan sebelum tahap transformasi untuk mencegah data leakage, sehingga hasil evaluasi benar-benar mencerminkan kemampuan model dalam melakukan generalisasi.

3.2.4 Transformasi Data

Transformasi data dilakukan untuk menyiapkan fitur agar dapat diproses secara optimal oleh algoritma machine learning. Pada dataset ini, fitur yang digunakan terdiri dari fitur numerik (*Age dan Temperature*) serta fitur kategorikal berupa gejala (*Symptom 1-3*). Karena kedua jenis fitur tersebut memiliki karakteristik yang berbeda, maka diperlukan proses transformasi agar model dapat membaca dan mempelajarinya secara konsisten. Transformasi dilakukan setelah pembagian data train-test untuk menghindari terjadinya data *leakage*.

1. Standarisasi Fitur Numerik

Fitur numerik seperti *Age* dan *Temperature* akan distandarisasi menggunakan metode *StandardScaler*. Proses ini bertujuan untuk menyesuaikan skala data sehingga perbedaan nilai antar fitur tidak terlalu besar. Meskipun algoritma *Random Forest* tidak sensitif terhadap perbedaan skala data, proses standarisasi tetap akan dilakukan untuk menjaga konsistensi data, memastikan *pipeline preprocessing* berjalan secara seragam, serta menghindari perbedaan skala yang terlalu jauh antar fitur. Proses *fitting* pada *StandardScaler* akan dilakukan menggunakan data latih, sedangkan data uji hanya akan ditransformasikan menggunakan parameter yang diperoleh dari data latih guna mencegah terjadinya kebocoran data.

2. *Encoding* Fitur Kategorikal

Fitur gejala pada kolom *Symptom* 1–3 masih berbentuk teks, sehingga tidak dapat diproses langsung oleh model *Random Forest*. Meskipun algoritma *Random Forest* mampu menangani data kategorikal, model ini tetap memerlukan data dalam bentuk numerik, bukan teks. Oleh karena itu, nilai kosong pada fitur gejala diisi menggunakan SimpleImputer dengan strategi modus.

Selanjutnya, seluruh nilai gejala diubah ke dalam bentuk numerik menggunakan teknik label encoding. Proses ini dilakukan dengan menggabungkan

seluruh gejala dari ketiga kolom untuk memperoleh daftar gejala yang unik, kemudian setiap gejala diberikan kode numerik secara konsisten.

Pendekatan ini memungkinkan setiap gejala direpresentasikan dalam bentuk angka tanpa mengubah makna kategorinya, sehingga model dapat memproses data dengan baik. Selain itu, penggunaan kode yang konsisten memastikan bahwa gejala yang sama memiliki representasi yang sama meskipun muncul pada kolom yang berbeda.

3.2.5 Pelatihan Model

Setelah tahapan pra-pemrosesan dan pembagian data menggunakan metode train–test split dilakukan, proses dilanjutkan dengan pelatihan model menggunakan algoritma *Random Forest* Classifier. Pelatihan dilakukan menggunakan data latih yang telah melalui proses transformasi dalam pipeline, sehingga alur pemodelan berlangsung konsisten dan terhindar dari data leak*Age*. *Random Forest* dipilih karena mampu mengolah kombinasi fitur numerik dan kategorikal, memiliki performa yang stabil pada klasifikasi multikelas, serta relatif tahan terhadap overfitting melalui pendekatan ensemble. Pada tahap ini, model mempelajari hubungan antara pola gejala klinis dan jenis penyakit berdasarkan variasi data latih yang tersedia.

1. Menyiapkan Data Latih (80%)

Data latih sebesar 80% diperoleh dari pembagian dataset menggunakan metode train–test split. Data latih ini menjadi dasar pembelajaran model untuk mengenali keterkaitan antara gejala klinis dan jenis penyakit kambing sehingga model dapat memahami karakteristik setiap penyakit secara lebih akurat.

2. Pelatihan Model *Random Forest*

Pada tahap pelatihan, *Random Forest* membangun sejumlah pohon keputusan *(decision tree)*, di mana setiap pohon dilatih menggunakan subset data dan subset fitur yang dipilih secara acak. Setiap pohon menghasilkan prediksinya masing-masing, kemudian hasil akhir ditentukan melalui mekanisme *majority voting*, yaitu memilih kelas penyakit yang memperoleh suara terbanyak dari seluruh pohon. Parameter seperti jumlah pohon *(n_estimators)* dan kedalaman pohon digunakan untuk mengatur kompleksitas model serta meningkatkan performanya.

3. Validasi dan Evaluasi

Setelah pelatihan selesai, model dievaluasi menggunakan data uji sebesar 20% yang tidak pernah dilihat selama proses pelatihan. Evaluasi dilakukan dengan membandingkan hasil prediksi model terhadap label sebenarnya, kemudian dihitung beberapa metrik seperti akurasi, *precision*, *recall*, dan *F1-score* untuk menilai kualitas prediksi secara kuantitatif. Selain itu, *confusion matrix* digunakan untuk melihat distribusi prediksi pada tiap kelas penyakit dan mengidentifikasi kelas mana yang masih sering tertukar.

3.2.6 Evaluasi Model

Setelah proses pelatihan diselesaikan, tahap berikutnya adalah melakukan evaluasi model untuk mengetahui sejauh mana kemampuan *Random Forest* dalam mengklasifikasikan penyakit kambing secara akurat. Evaluasi dilakukan menggunakan data uji, yaitu data yang tidak dilibatkan dalam proses pelatihan, sehingga performa yang diperoleh dapat menggambarkan kemampuan model secara objektif dan menghindari bias akibat *overfitting*. Pada tahap ini, beberapa metrik digunakan untuk menilai kualitas prediksi model dari berbagai sudut pandang:

1. *Confusion Matrix*

Confusion matrix digunakan untuk menggambarkan performa model secara lebih rinci dengan menampilkan jumlah prediksi benar dan salah pada setiap kelas penyakit. Metode ini tidak hanya memberikan gambaran hasil klasifikasi secara umum, tetapi juga mampu menunjukkan pola kesalahan model, seperti penyakit yang sering tertukar atau gejala yang sulit dibedakan oleh model.

Pada penelitian ini, proses klasifikasi bersifat multikelas karena terdapat beberapa jenis penyakit kambing yang harus dibedakan. Oleh karena itu, *confusion matrix* disajikan dalam bentuk tabel K × K, di mana K merupakan jumlah kelas penyakit. Untuk memperjelas cara membacanya, bagian-bagian pada *confusion matrix* dapat dijelaskan sebagai berikut:

a. Baris menunjukkan label penyakit yang sebenarnya (aktual).
b. Kolom menunjukkan label penyakit yang diprediksi oleh model.
c. Nilai diagonal merupakan jumlah prediksi yang benar untuk setiap kelas penyakit.

d. Nilai di luar diagonal merupakan jumlah prediksi yang salah, yaitu ketika suatu penyakit diklasifikasikan sebagai penyakit lain.

Melalui *confusion matrix*, peneliti dapat melihat kelas penyakit mana yang memiliki tingkat prediksi akurat serta kelas mana yang masih sering tertukar, terutama pada penyakit yang memiliki pola gejala mirip.

2. *Classification Report*

Classification report digunakan untuk memberikan evaluasi model secara lebih rinci terhadap setiap kelas penyakit. Laporan ini menampilkan metrik yang digunakan untuk menilai ketepatan prediksi, kemampuan model dalam menemukan kasus, serta keseimbangan antara keduanya. Adapun metrik yang digunakan dalam *classification report* meliputi *precision*, *recall*, dan *f1-score*, yang dijelaskan sebagai berikut:

a. *Precision*

Precision digunakan untuk mengukur ketepatan prediksi positif, yaitu seberapa banyak prediksi positif yang benar-benar sesuai dengan kondisi sebenarnya. Metrik ini penting ketika tujuan evaluasi adalah memastikan model tidak terlalu sering salah dalam memberikan prediksi positif.

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

Nilai *Precision* dihitung dengan membandingkan jumlah *True Positive* (TP) terhadap seluruh prediksi positif yang dibuat model (TP + FP (*False Positive*)). Semakin tinggi *Precision*, semakin sedikit kesalahan model dalam memprediksi suatu penyakit ketika sebenarnya penyakit tersebut tidak ada. Dengan demikian, *Precision* membantu menilai seberapa "tepat" model saat menyatakan bahwa suatu kondisi adalah positif.

b. *Recall*

Recall digunakan untuk mengukur kemampuan model dalam menemukan seluruh kasus positif yang ada pada data sebenarnya. Metrik ini menunjukkan seberapa banyak penyakit yang berhasil dideteksi oleh model dan penting digunakan ketika melewatkan kasus positif harus dihindari.

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

Recall dihitung dengan membandingkan jumlah *True Positive* (TP) terhadap seluruh kasus positif yang sebenarnya ada (TP + FN (*False Negatif*)). Nilai *recall* yang tinggi menunjukkan bahwa model mampu mengenali sebagian besar penyakit yang seharusnya terdeteksi. Jika *recall* rendah, berarti model sering melewatkan kasus penyakit sehingga berpotensi menyebabkan kesalahan diagnosis.

c. *F1-score*

F1-score adalah metrik yang menggabungkan *Precision* dan *Recall* ke dalam satu nilai yang seimbang. Metrik ini digunakan ketika kedua aspek ketepatan prediksi positif dan kemampuan mendeteksi kasus positif sama-sama penting untuk dievaluasi.

$$F1 = 2\ x\ \frac{Precision\ x\ Recall}{Precision + Recall} \quad (4)$$

F1-score dihitung menggunakan rata-rata harmonis antara *Precision* dan *Recall*. Nilainya akan tinggi hanya jika kedua metrik tersebut memiliki nilai yang baik. Oleh karena itu, *F1-score* sangat berguna pada dataset dengan distribusi kelas tidak seimbang karena memberikan penilaian yang lebih stabil dibandingkan hanya melihat *Precision* atau *Recall* saja.

d. Akurasi

Selain metrik pada classification report, akurasi digunakan untuk mengukur kinerja model secara keseluruhan dengan menghitung persentase prediksi yang benar terhadap seluruh data uji. Nilai ini memberikan gambaran umum mengenai kemampuan model dalam melakukan klasifikasi penyakit kambing. Perhitungan akurasi dapat dirumuskan sebagai berikut:

$$Accuracy = \frac{\text{Jumlah prediksi benar}}{\text{Total data}}$$

Nilai akurasi menunjukkan proporsi jumlah prediksi yang benar dibandingkan dengan keseluruhan data. Semakin tinggi nilai akurasi, maka semakin baik kemampuan model dalam melakukan klasifikasi secara umum.

Namun, akurasi hanya menunjukkan performa global dan tidak dapat merepresentasikan kemampuan model dalam mengenali masing-masing kelas penyakit secara rinci. Oleh karena itu, akurasi digunakan sebagai indikator awal dan perlu didukung oleh metrik lain seperti *Precision*, *Recall*, dan f1-score untuk mendapatkan evaluasi yang lebih menyeluruh.