BAB II

LANDASAN TEORI

2.1 Penelitian Terdahulu

Dalam penelitian sebelumnya mengimplementasikan *internet of things* dalam penelitiannya, Alat nya digunakan untuk mengetahui suhu tubuh pada sapi. Pendteksian suhu tubuh tersebut digunakan sebagai acuan peternak sapi untuk mengetahui lebih awal terkait penyakit yang ada pada sapi tersebut.

No	Penulis	Judul Jurnal	Nama Jurnal	Tahun	Hasil
1	Saharuddin R. Sokku, Sabran F Harun	Deteksi Sapi Sehat Berdasarkan Suhu Tubuh Berbasis Sensor MLX90614 dan Mikrokontroller	PROSIDING SEMINAR NASIONAL LP2M UNM - 2019 "Peran Penelitian dalam Menunjang Percepatan Pembangunan Berkelanjutan di Indonesia"	2019	alat pendeteksi sapi sehat bersarkan suhu tubuh menggunakan sensor MLX90614 berbasis mikrokontroller dengan mengunakan rangkaian LCD, Sensor MLX90614, Rangkaian Arduino nano sebagai mikrokontroller, dan buzzer dapat bekerja secara baik dan dapat

					dioperasikan dengan baik
2	Ibman Andika, Dewi Maharani, Mardalius	Penerapan Teorema Bayes pada Sistem Pakar Pendeteksi Penyakit Domba	Edumatic: Jurnal Pendidikan Informatika	2022	Dalam penelitian ini, hasil analisis dari teorima bayes menunjukkan bahwa penyakit yang dialami pada domba adalah cacingan dengan Tingkat probabilitas 60,71%
3	Sarah Lasniari, Jasril, Suwanto Sanjaya, Febi Yanto, Muhammad Affandes	Klasifikasi Citra Daging Babi dan Daging Sapi Menggunakan Deep Learning Arsitektur ResNet-50 dengan Augmentasi Citra	Jurnal Sistem Komputer dan Informatika (JSON)	2022	Dataset citra daging asli berjumlah 457 citra, yang setelah melalui proses augmentasi meningkat menjadi 2742 citra dan terbagi dalam tiga kelas. Pengujian dilakukan dengan membandingkan dua skema, yaitu data asli dan data augmentasi, dengan distribusi data pelatihan dan pengujian sebesar

					90%:10%. Confusion Matrix menunjukkan bahwa model mencapai kinerja klasifikasi tertinggi dengan rata-rata akurasi 87,64%, recall 87,59%, dan precision 90,90%. Dari visualisasi proses pelatihan dan pengujian, tidak ditemukan tanda-tanda overfitting.
4	Sza Sza Amulya Larasati, Elok Nuraida Kusuma Dewi, Brahma Hanif Farhansyah, Fitra Abdurrachman Bachtiar, Fajar Pradana	Penerapan Decision Tree dan Random Forest Dalam Deteksi Tingkat Stres Manusia Berdasarkan Kondisi Tidur	Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK)	2023	Hyperparameter tuning dilakukan dengan menggunakan teknik k-fold cross validation, dan model dirancang menggunakan algoritma klasifikasi Decision Tree serta Random Forest. Hasilnya menunjukkan bahwa lima fitur,

					yaitu tingkat mendengkur, laju respirasi, pergerakan anggota tubuh termasuk bola mata, serta detak jantung saat tidur, berkorelasi positif dengan tingkat stres. Semakin tinggi nilai dari kelima fitur ini, semakin tinggi tingkat stres yang diindikasikan. Sebaliknya, tiga fitur lainnya, yaitu suhu tubuh, kadar oksigen, dan waktu tidur, berkorelasi negatif dengan tingkat stres. Dengan kata lain, semakin tinggi nilai dari ketiga fitur tersebut, semakin rendah tingkat stres yang dialami. Model Decision Tree

					mencapai akurasi 0,99, sedangkan Random Forest mencapai akurasi 1,0.
5	Naisah Marito Putry, Betha Nurina Sari, M.Kom	KOMPARASI ALGORITMA KNN DAN NAÏVEBAYES UNTUKKLASIFIKASI DIAGNOSIS PENYAKIT DIABETES MELITUS	Evolusi: Jurnal Sains dan Manajemen	2022	Penelitian ini membandingkan dua algoritma, KNN dan Naïve Bayes, dalam mengklasifikasikan diagnosis penyakit diabetes melitus. Hasil analisis menunjukkan bahwa akurasi Naïve Bayes lebih tinggi dibandingkan KNN, dengan nilai akurasi tertinggi sebesar 80% untuk Naïve Bayes dan 75% untuk KNN. Selain itu, recall tertinggi dihasilkan oleh KNN dengan nilai 0.92, sementara presisi tertinggi dicapai oleh Naïve Bayes

					dengan nilai 0.86. Penelitian ini memberikan wawasan berharga mengenai efektivitas algoritma Naïve Bayes dan KNN dalam diagnosis diabetes melitus dan dapat menjadi referensi serta dasar pengembangan ilmu pengetahuan untuk penelitian selanjutnya.
6	Annisa Nurba Iffah'da, Anita Desiani	IMPLEMENTASI ALGORITMA K-NEAREST NEIGHBOR (K-NN) DAN SINGLE LAYER PERCEPTRON (SLP) DALAM PREDIKSI PENYAKIT SIROSIS BILIARI PRIMER	Jurnal Ilmiah Informatika	2022	Penelitian ini membandingkan akurasi algoritma K-Nearest Neighbor (K-NN) dan Single Layer Perceptron (SLP) dalam mendeteksi dini penyakit sirosis biliari primer. Hasilnya menunjukkan bahwa K-NN memiliki akurasi

					76.2% dan SLP memiliki akurasi 62%, mengindikasikan bahwa kedua algoritma efektif dalam deteksi dini penyakit ini. Namun, K-NN terbukti lebih unggul dengan akurasi, presisi, dan recall yang lebih tinggi. K-NN mencapai presisi sebesar 77% dan recall sebesar 75%, menunjukkan kemampuannya dalam mengurangi angka kematian global. Sebaliknya, meskipun K-NN unggul dalam deteksi awal, SLP lebih baik dalam menemukan kembali informasi pada pasien dengan sirosis biliari primer, dengan

					nilai recall sebesar 65%.
7	Annida Purnamawati, Wawan Nugroho, Destiana Putri, Wahyutama Fitri Hidayat	Deteksi Penyakit Daun pada Tanaman Padi Menggunakan Algoritma Decision Tree, Random Forest, Naïve Bayes, SVM dan KNN	InfoTekJar: Jurnal Nasional Informatika dan Teknologi Jaringan	2020	Penelitian mengenai klasifikasi penyakit tanaman menggunakan machine learning menghasilkan tiga kategori model: Overfit (Random Forest, Decision Tree, dan Naive Bayes), Underfit (SVM), dan Good (KNN). Dari kelima algoritma tersebut, KNN terbukti sebagai metode terbaik dengan akurasi 87%, menunjukkan konsistensi yang baik pada data pelatihan dan pengujian. KNN tidak mengalami overfitting, sehingga memiliki kemampuan generalisasi yang

					lebih baik dibandingkan model lainnya. Ini penting karena data gambar yang digunakan dalam produksi mungkin berbeda dari data pelatihan. Model KNN berfokus pada fitur umum untuk memprediksi penyakit daun daripada hanya mengandalkan fitur spesifik dari data pelatihan.
8	Deo Haganta Depari, Yuni Widiastiwi, Mayanda Mega Santoni	Perbandingan Model Decision Tree, Naive Bayes dan Random Forest untuk Prediksi Klasifikasi Penyakit Jantung	Jurnal Informatik	2022	Berdasarkan penelitian ini, beberapa kesimpulan dapat diambil: a. Model Naive Bayes dan Decision Tree memberikan nilai precision tertinggi untuk pasien tanpa penyakit jantung (0) dan dengan

					penyakit jantung (1). b. Model Decision Tree, Random Forest, dan Naive Bayes memberikan nilai recall tertinggi untuk kedua kategori pasien tersebut. c. Nilai f1-score tertinggi untuk pasien tanpa dan dengan penyakit jantung diperoleh dari model Naive Bayes dan Random Forest. d. Nilai akurasi adalah 71% untuk Naive Bayes, 72% untuk Decision Tree, dan 75% untuk Random Forest. e. Random Forest adalah metode terbaik dengan akurasi 75%. Namun, untuk kasus yang

					memerlukan kecepatan, Random Forest kurang efisien karena membutuhkan waktu signifikan (69 detik) untuk mencapai akurasi tinggi dengan 1000 pohon. Sebaliknya, Decision Tree memiliki akurasi 72% dengan waktu eksekusi hanya 0.118 detik, menjadikannya lebih cocok untuk skenario yang memerlukan kecepatan.
9	Achmad Afifuddin, Lukman Hakim	Deteksi Penyakit Diabetes Mellitus Menggunakan Algoritma Decision Tree Model Arsitektur C4.5	Jurnal Krisnadana	2023	Pengembangan aplikasi untuk deteksi awal penyakit diabetes menggunakan algoritma klasifikasi C4.5 menunjukkan beberapa temuan penting:

					Algoritma C4.5 memudahkan proses pengambilan keputusan klinis awal dengan mengkonversi data menjadi struktur pohon keputusan, yang ditentukan berdasarkan nilai entropy dan gain dari setiap atribut data. Akurasi prediksi aplikasi ini meningkat dengan penambahan volume data. Ini menegaskan bahwa semakin banyak data yang tersedia, semakin tinggi kemungkinan untuk mendapatkan hasil prediksi yang tepat.

					Aplikasi ini berhasil memprediksi diabetes mellitus menggunakan algoritma C4.5 dengan tingkat akurasi yang sangat tinggi, yaitu 96%.
10	Muhammad Abid Wiratama, Windha Mega Pradnya	OPTIMASI ALGORITMA DATA MINING MENGGUNAKAN BACKWARD ELIMINATION UNTUK KLASIFIKASI PENYAKIT DIABETES	Jurnal Nasional Pendidikan Teknik Informatika: JANAPATI	2022	Pada penelitian ini, ditemukan bahwa sebelum pengoptimalan, algoritma KNN mencapai akurasi 92,8% dengan AUC 0,942, Naïve Bayes memiliki akurasi 88,0% dan AUC 0,912, serta C4.5 mencatatkan akurasi 96,7% dan AUC 0,956. Setelah proses pengoptimalan, kinerja algoritma meningkat signifikan dimana KNN mencapai akurasi tertinggi sebesar 97,6% dan

					AUC 0,973, Naïve Bayes meningkat menjadi akurasi 89,4% dengan AUC 0,958, dan C4.5 mencapai akurasi 97,5% dengan AUC tertinggi di antara semua yaitu 0,988. Berdasarkan hasil ini, algoritma KNN yang telah dioptimasi terbukti paling akurat dengan skor 97,6%, sementara C4.5 yang dioptimasi menunjukkan kinerja terbaik dalam hal AUC dengan skor 0,988.

2.2 Teori yang digunakan

2.2.1 *Machine Learning*

Machine Learning merupakan cabang dari sebuah kecerdasan buatan yang memungkinkan computer untuk belajar dari data tanpa perlu deprogram secara eksplisit. Proses pembelajaran ini mencakup pengidentifikasian pola dalam data untuk membuat prediksi atau keputusan. Tujuan utama dari algoritma *machine learning* adalah

untuk mengekstrak pengetahuan dengan cara yang dapat berguna untuk penyelidikan logis[3]. Terdapat beberapa pendekatan dalam *machine learning*, termasuk *supervised learning*, *unsupervised learning*, dan *reinforcement learning*.

Supervised learning melibatkan penggunaan data yang telah diberi label untuk melatih model agar bisa membuat prediksi atau mengambil Keputusan terhadap data baru yang belum pernah dilihat sebelumnya. *Unsupervised learning*, di sisi lain, melibatkan penggunaan data yang tidak memiliki label untuk menemukan struktur atau pola yang tidak terlihat sebelumnya dalam data. *Reinforcement learning* melibatkan interaksi agen (biasanya disimulasikan sebagai komputer) dengan lingkungan untuk belajar membuat keputusan yang optimal.

Machine learning telah diterapkan dalam berbagai bidang, termasuk pengenalan wajah, deteksi penipuan, pemrosesan bahasa alami, dan permainan komputer. Dengan kemampuannya untuk belajar dari data, *machine learning* telah menjadi salah satu alat yang paling berharga dalam mengatasi masalah yang rumit dan mendorong invovasi dalam teknologi.

2.2.2 *Data Mining*

Data mining adalah proses menemukan pola yang bermanfaat atau informasi yang berguna dari Kumpulan data yang besar dan kompleks[6]. Tujuan utama dari *data mining* adalah untuk mengungkapkan wawasan yang tidak terlihat sebelumnya, yang dapat digunakan untuk pengambilan Keputusan yang lebih baik dalam berbagai bidang seperti bisnis, ilmu pengetahuan, kesehatan, dan lain-lain. Proses ini melibatkan penggunaan teknik statistic, matematika, dan kecerdasan buatan untuk menganlisis data dan mengidentifikasi pola yang signifikan.

Salah satu teknik yang sering digunakan dalam *data mining* adalah *clustering*, yang digunakan untuk mengelompokkan data ke dalam kelompok-kelompok yang serupa berdasarkan karakgteristik tertentu. Teknik lainnya adalah *classification*, yang digunakan untuk memprediksi kategori atau kelas dari data berdasarkan pola yang ditemukan dalam data pelatihan. Selain itu, terdapat juga teknik *association*, yang digunakan untuk menemukan hubungan atau asosiasi antara item-item dalam kumpulan data.

Data mining memiliki peran yang penting dalam mengoptimalkan proses pengambilan Keputusan dengan mengidentifikasi pola atau tren yang dapat membantu organisasi atau individu dalam mengambil langkah-langkah yang lebih baik di masa depan. Dengan kemampuannya untuk menganalisis data yang besar dan kompleks, data mining telah menjadi salah satu alat yang sangat berharga dalam menghadapi tantangan dalam mengelola dan memahami informasi dalam dunia yang terus berkembang.

2.2.3 Pengertian "Penyakit"

Penyakit adalah kegagalan dari mekanisme adaptasi suatu organisme untuk bereaksi secara tepat terhadap rangsangan atau tekanan sehingga timbul gangguan pada fungsi struktur, bagian, organ atau sistem dari tubuh. Penyakit adalah suatu kondisi abnormal dalam tubuh atau pikiran seseorang yang menyebabkan gangguan fungsi normal dan kesejahteraan individu. Penyakit dapat disebabkan oleh berbagai faktor seperti infeksi mikroorganisme (virus, bakteri, jamur, parasit), keturunan genetik, faktor lingkungan, gaya hidup, atau kombinasi dari beberapa faktor tersebut. Penyakit bisa bersifat akut (jangka pendek) atau kronis (jangka panjang), dan dapat mempengaruhi berbagai sistem dalam tubuh, termasuk sistem pernapasan, pencernaan, saraf, dan lainnya. Pengelolaan dan penanganan penyakit memerlukan diagnosis yang tepat dan intervensi medis yang sesuai untuk memulihkan atau mempertahankan kesehatan individu. Penyakit juga

bisa terjadi pada semua makhluk hidup, manusia dan termasuk juga dengan hewan.

2.2.4 Pengertian "Hewan Ternak Kambing"

Hewan ternak kambing adalah jenis hewan domestik yang dibudidayakan oleh manusia untuk berbagai tujuan seperti produksi daging, susu, serat (wol), serta kulit. Kambing (Capra aegagrus hircus) termasuk dalam keluarga Bovidae dan genus Capra. Kambing ternak memiliki peran penting dalam sektor peternakan dan agribisnis karena kemampuannya beradaptasi dengan berbagai lingkungan serta kemudahan dalam pemeliharaannya. Selain itu, kambing juga memiliki nilai ekonomi yang signifikan bagi peternak karena produk-produk yang dihasilkan memiliki permintaan tinggi di pasar. Pemeliharaan kambing memerlukan pengetahuan tentang manajemen pakan, kesehatan, reproduksi, dan perawatan umum untuk memastikan produktivitas dan kesejahteraan hewan yang optimal.

2.2.5 Pengertian "Suhu tubuh"

Suhu tubuh adalah ukuran dari panas yang dihasilkan oleh tubuh makhluk hidup sebagai hasil dari metabolism dan berbagai biologis lainnya. Pada hewan, termasuk hewan ternak seperti kambing, suhu tubuh merupakann indikator penting dari kesehatan dan kondisi fisiologis. Suhu tubuh diatur oleh pusat pengatur suhu di otak, yaitu hipotalamus, yang memastikan bahwa suhu tubuh dalam rentang yang normal dan optimal untuk fungsi tubuh.

Berikut ini adalah pentingnya suhu tubuh dalam kesehatan hewan ternak:

1. Indikator kesehatan: Perubuhana suhu tubuh sering kali menjadi tanda awal adanya penyakit atau gangguan kesehatan. Suhu tubuh yang meningkat (hipertermia) bisa menunjukkan adanya infeksi, peradangan, atau kondisi lainnya, sementara suhu tubuh yang

menurun (hipotermia) bisa menjadi indikasi penyakit tertentu atau kondisi lingkungan yang ekstrem.

2. Homeostasis: Suhu tubuh yang stabil penting untuk menjaga homeostasis, yaitu kondisi keseimbangan internal tubuh. Suhu yang terlalu tinggi atau terlalu rendah dapat mengganggu fungsi enzim dan metabolisme, yang berdampak pada kesehatan dan produktivitas hewan.
3. Faktor pengaruh: Suhu tubuh hewan ternak dipengaruhi oleh berbagai faktor, termasuk suhu lingkungan, aktivitas fisik, kondisi kesehatan, dan faktor stres. Pemantauan suhu tubuh secara teratur dapat membantu dalam pengelolaan kesehatan hewan ternak dan pencegahan penyakit.

Dalam konteks penelitian deteksi penyakit pada hewan ternak kambing, suhu tubuh menjadi parameter kunci. Pengukuran dan pemantauan suhu tubuh dapat digunakan untuk mendeteksi dini adanya gangguan kesehatan atau penyakit. Dengan memanfaatkan teknologi dan metodologi yang tepat, perubahan suhu tubuh dapat dianalisis untuk menentukan pola yang berkaitan dengan kondisi kesehatan hewan, sehingga memungkinkan tindakan pencegahan dan penanganan yang lebih efektif.

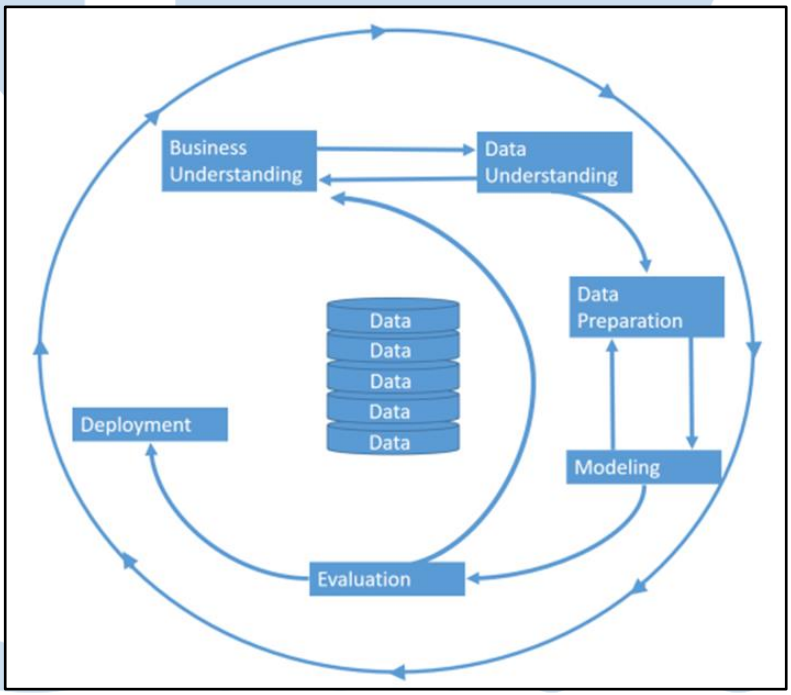
Suhu tubuh bukan hanya indikator pasif, tetapi juga alat diagnostik aktif dalam manajemen kesehatan hewan ternak, termasuk kambing. Pemahaman yang mendalam tentang suhu tubuh dan pengaruhnya dapat memberikan dasar yang kuat untuk penelitian dan praktik dalam bidang peternakan.

2.3 *Framework* Big Data

Dalam pengembangan model dalam penelitian ini, terdapat beberapa metodologi yang dapat diterapkan, seperti CRISP-DM, KDD, dan SEMMA. Berikut ini adalah penjelasan mengenai langkah-langkah yang harus diikuti dalam setiap metodologi tersebut.

2.3.1 CRISP-DM (*Cross-Industry Standard Process for Data Mining*)

CRISP-DM adalah metodologi yang digunakan untuk memandu langkah-langkah dalam proses data mining. Metodologi ini terdiri dari enam tahap utama yaitu *business understanding*, *data understanding*, *data preparation*, *modeling*, *evaluation*, dan *deployment*[7]. CRISP-DM memiliki 6 tahapan yang harus dilalui saat mengembangkan model *machine learning*. Hasil dari setiap tahap akan menentukan apakah kita harus melanjutkan ke tahap berikutnya atau kembali mengulangi tahap tersebut. Oleh karena itu, tahapan dalam CRISP-DM bersifat fleksibel karena memungkinkan kita untuk mengulangi suatu tahapan yang sama guna mencapai hasil yang paling optimal. Gambar … menunjukkan kerangak dari metode CRISP-DM.



Gambar 2. 1 Tahapan CRISP-DM [8]

Berikut ini adalah tahapan pada CRISP-DM:

1. *Business understanding*: Memahami tujuan dan kebutuhan proyek dari sudut pandang bisnis, lalu mengubahnya menjadi masalah yang akan menjadi dasar untuk merancang rencana proyek yang bertujuan mencapai tujuan bisnis.

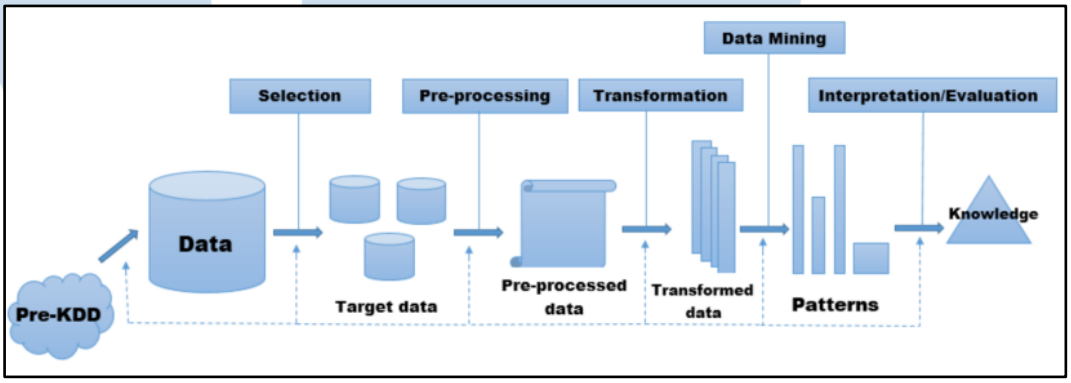
2. *Data understanding*: Mengumpulkan data awal untuk memahami tujuan dan mendapatkan wawasan mengenai data. Proses ini mencakup identifikasi kualitas data, pencarian wawasan dari data awal, dan pendeteksian subset data yang menarik.
3. *Data preparation*: Kemudian, setelah data terkumpul, data tersebut perlu dipersiapkan untuk membantuk dataset akhir. Proses data preparation ini mencakup berbagai aktivitas seperti pemilihan tabel, atribut, transformasi, *record* dan pembersihan data.
4. *Modelling*: Pemilihan dan penerapan berbagai teknik pemodelan pada proyek dilakukan, dengan pengukuran parameter pada model-model tersebut untuk mencapai hasil optimal. Tahap ini sering kali mengarahkan kita kembali ke tahap persiapan data, karena setiap model membutuhkan dataset yang berbeda.
5. *Evaluation*: Evaluasi dari model tentunya diperlukan untuk memastikan hasil akhir dari model tersebut sesuai dengan tujuan bisnis yang telah ditetapkan sejak awal. Proses evaluasi dilakukan secara menyeluruh untuk menentukan hasil model yang paling optimal.
6. *Deployment*: Pada tahap ini, model akhir akan diimplementasikan. Bergantung pada kebutuhan bisnis, *deployment* dapat dilakukan secara sederhana, seperti mengintegrasikan model ke dalam sistem operasi, atau bisa juga dilakukan dengan cara yang lebih kompleks.

2.3.2 KDD (Knowledge Discovery in Database Process)

Knowledge Discovery in Databases (KDD) merupakan sebuah proses mengekstrak pengetahuan yang berguna dari basis data. KDD berfokus pada penemuan pengetahuan dari data, termasuk cara penyimpanan dan akses data, pelaksanaan algoritma pada dataset besar secara efisien, serta interpretasi dan visualisasi hasilnya dengan baik.

Knowledge Discovery in Database Process atau KDD merupakan salah satu metode *data mining* yang biasa digunakan untuk memanfaatkan teknik-teknik *data mining* dalam rangka menemukan informasi penting dan pola tersembunyi dalam data melalui penerapan berbagai algoritma

yang bertujuan untuk mengenali pola tersebut. Proses KDD ini juga terdiri dari beberapa tahapan kunci, yaitu: pemilihan data, pra-pengolahan data, transformasi data, proses penambangan data itu sendiri, serta tahap interpretasi dan evaluasi hasil. Tujuan utama dari proses *Knowledge Discovery in Database* (KDD) adalah untuk mengeksplorasi potensi data yang diambil dari database. Data tersebut akan dianalisis untuk mengidentifikasi pola yang kemudian akan dianalisis lebih lanjut dan divisualisasikan. Hal ini memungkinkan informasi yang diperoleh menjadi lebih mudah dimengerti oleh pengguna. Tahapan KDD adalah proses berulang seperti yang dijelaskan dalam gambar … di bawah ini:



Gambar 2. 2 Tahapan KDD [8]

Berikut ini adalah tahapan pada KDD:

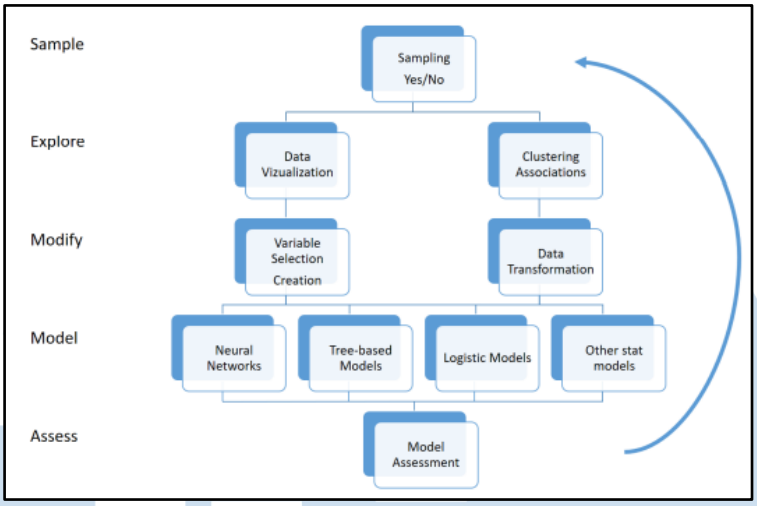
1. *Pre-KDD*: Penelitian dilakukan untuk memahami domain proyek yang akan dikembangkan dan menentukan langkah-langkah yang diperlukan untuk mendapatkan pengetahuan yang relevan. Hasil akhirnya adalah penetapan tujuan proyek dari perspektif pengguna akhir.
2. *Selection*: Penyusunan dataset target berdasarkan data yang telah dikumpulakan pada tahap sebelumnya. Data yang diperoleh akan diintegrasikan menjadi satu dataset yang terfokus pada sejumlah variabel atau sampel data tertentu.
3. *Pre-processing*: Tahap ini adalah tahap pembersihan data yang melibatkan penghapusan *noise*, pengumpulan informasi yang diperlukan untuk pemodelan, penanganan data yang hilang, serta

mempertimbangkan informasi urutan waktu dan perubahan yang telah diketahui.

4. *Transformation*: Tahap ini adalah tahap persiapan data untuk proses *data mining*. Pada tahap ini, fitur yang mewakili data akan diidentifikasi. Metode yang sering digunakan dalam tahap ini mencakup pengurangan dimensi, seperti pemilihan dan ekstraksi fitur.
5. *Data mining*: Proses ini melibatkan beberapa langkah, yaitu memilih metode *data mining* sesuai dengan tujuan KDD pada langkah awal, memilih algoritma dan metode untuk menemukan pola-pola penting dalam data, serta mengulangi implementasi algoritma *data mining* untuk menemukan pola-pola menarik dalam dataset guna mencapai hasil yang optimal.
6. *Interpretation/Evaluation*: Interpretasi atau evaluasi pola-pola yang ditemukan pada langkah sebelumnya untuk memastikan bahwa hasilnya sesuai dengan tujuan KDD yang telah ditetapkan di awal. Mengulangi tahapan sebelumnya juga mungkin dilakukan untuk menghasilkan beberapa perubahan.
7. *Post-KDD*: Mengambil langkah berdasarkan hasil akhir yang telah diperoleh. Pengetahuan yang dihasilkan dapat digunakan secara langsung, diimplementasikan ke dalam sistem, dan didokumentasikan atau dilaporkan.

2.3.3 SEMMA (*Sample, Explore, Modify, Model, Assess*)

Sample, Explore, Modify, Model, Assess (SEMMA) adalah seperangkat alat yang digunakan untuk menjalankan tugas-tugas utuama dalam pengemabngan model SEMMA, yang banyak digunakan dalam perangkat lunak SAS Enterprise Miner, berfokus pada pengembangan model. Tahapan SEMMA bersifat fleksibel, memungkinkan perpindahan langkah maji atau mundur sesuai kebutuhan. SEMMA terdiri dari lima tahapan seperti yang ditunjukkan pada Gambar …:



Gambar 2. 3 Tahapan SEMMA [8]

Berikut ini adalah tahapan pada SEMMA:

1. *Sample*: Pengambilan sampel data untuk membangun model. Pada tahap ini, data yang dikumpulkan juga dibagi menjadi sampel *training*, *validation*, dan *testing*.
2. *Explore*: Tahap eksplorasi pola dan hubungan menarik untuk memahami data. Pemahaman ini berguna untuk memperoleh ide dan membantu dalam pengambilan kesimpulan. Tahap ini dapat dilakukan melalui visualisasi atau analisis statistik.
3. *Modify*: Memodifikasi data yang diperoleh dari eksplorasi agar sesuai dengan model yang akan dibangun. Tahap ini juga melibatkan segmentasi tambahan dan pembuatan variabel baru.
4. *Model*: Membangun model dengan menerapkan teknik pemodelan pada data dan variabel yang ada untuk menghasilkan model yang dapat dipercaya untuk melakukan prediksi atau klasifikasi data.
5. *Assess*: Mengevaluasi hasil dan kinerja model menggunakan sampel *validation* dan *testing*. Evaluasi ini bertujuan untuk menentukan apakah model tersebut bermanfaat dan bisa diandalkan [8].

Pemilihan metode dalam penelitian harus disesuaikan dengan tujuan, konteks bisnis, dan jenis data yang digunakan. KDD adalah pilihan yang baik jika fokus utama penelitian adalah penemuan pengetahuan yang bermakna dari data, terutama

jika penelitian melibatkan berbagai jenis data dan memerlukan fleksibilitas dalam pendekatan. Maka dari itu, pada penelitian kali ini, metode yang digunakan dan dipilih adalah metode KDD.

2.4 Algoritma yang digunakan

2.3.1 *K-Nearest Neighbors* (KNN)

Metode *K-Nearest Neighbor* adalah teknik klasifikasi yang sudah lama dikenal dan sering digunakan. Dalam metode ini, nilai K merepresentasikan jumlah tetangga terdekat yang berperan dalam menentukan label kelas untuk data uji yang sedang diprediksi. KNN jug amerupakan salah satu algoritma *machine learning* yang paling sederhana dan paling sering digunakan untuk masalah klasifikasi dan regresi. Algoritma ini didasarkan pada prinsip bahwa objek yang serupa cenderung berada di dekat satu sama lain dalam ruang fitur.

KNN bekerja dengan mengidentifikasi 'k' tetangga terdekat dari sebuah sampel data yang ingin diklasifikasikan atau diprediksi nilainya. Tetangga-tetangga ini dipilih berdasarkan jarak terdekat dalam ruang fitur, biasanya menggunakan metrik jarak seperti Euclidean, Manhattan, atau Minkowski.

Jarak Euclidian:

$$\sqrt{\sum_{i=1}^{n}(xi - yi)^2}$$

Rumus 1 Rumus Jarak Eucladian KNN

Di mana xi adalah koordinat fitur dari sampel baru dan yi adalah koordinat fitur dari sampel dalam dataset pelatihan.

Dalam penelitian deteksi penyakit pada hewan ternak kambing berdasarkan suhu tubuh, KNN dapat digunakan untuk mengklasifikasikan kondisi kesehatan kambing berdasarkan suhu tubuh yang diukur. Dengan membandingkan suhu tubuh yang terukur dengan data suhu tubuh dari dataset pelatihan, KNN dapat membantu

mengidentifikasi apakah kambing tersebut dalam kondisi sehat atau mengalami penyakit tertentu.

2.3.2 *Decision Tree*

Decision Tree adalah metode pengolahan data yang digunakan untuk memprediksi hasil masa depan dengan membuat model klasifikasi atau regresi dalam format struktur pohon. Proses ini melibatkan pembagian berkelanjutan data menjadi subset yang lebih kecil, sambil secara bertahap membangun struktur pohon Keputusan. Struktur ini terdiri dari node Keputusan, seperti cuaca/outlook, yang membawa ke cabang-cabang yang menunjukkan pilihan seperti panas, berawan, dan hujan, serta node daun yang menandai hasil.

Selain itu, *Decision Tree* sangat bermanfaat untuk eksplorasi data dan untuk mengungkapkan hubungan antara berbagai variabel input potensial dan sebuah variabel target. Ini sering dianggap sebagai langkah awal yang efektif dalam pemodelan data yang dapat diikuti atau ditingkatkan dengan teknik lain untuk memperoleh model akhir.

Salah satu keuntungan utama dari menggunakan *Decision Tree* adalah kemampuannya untuk mengabaikan data yang tidak relevan, mengurangi kebutuhan untuk mengolah sampel yang tidak memenuhi kriteria tertentu. Ini membuat metode ini sangat efisien dalam menangani data dan membuat prediksi.

Algoritma Decision Tree bekerja dengan membagi dataset menjadi subset berdasarkan fitur-fitur tertentu, mengikuti aturan yang memaksimalkan pemisahan kelas atau nilai target. Proses ini berlanjut secara rekursif sampai setiap subset hanya berisi satu kelas atau tidak ada fitur yang tersisa untuk dibagi.

Berikut ini adalah struktur dari *Decision Tree*:

1. *Root Node*: Node pertama yang memulai pembagian data.
2. *Decision Nodes*: Node internal yang mewakili tes atau keputusan berdasarkan atribut tertentu.

3. *Leaf Nodes*: Node terminal yang mewakili hasil akhir atau klasifikasi.
4. *Branches*: Jalur dari satu node ke node lainnya yang menunjukkan hasil dari tes atau keputusan.

Berikut ini adalah langkah-langkah dari *Decision Tree*:

1. Pemilihan Atribut: Memilih atribut yang paling efektif untuk membagi data. Kriteria pemilihan dapat berdasarkan:
   A. Gini Impurity: Mengukur seberapa sering elemen yang dipilih secara acak akan salah klasifikasi jika elemennya secara acak berlabel menurut distribusi kelas data set.

$$Gini(D) = 1 - \sum_{i=1}^{n} Pi^2$$

Rumus 2 Rumus Gini Impurity

   B. Information Gain: Mengukur pengurangan ketidakpastian atau entropi dari data setelah dibagi berdasarkan atribut tertentu.

$$IG(D, A) = Entropy(D) - \sum_{vEValues(A)} \frac{|D_v|}{|D|} Entropy(D_v)$$

Rumus 3 Rumus Informasi Gain

   C. Entropy: Mengukur ketidakpastian dalam data.

$$Entropy(D) = -\sum_{i=1}^{n} p_i \ log_2(p_i)$$

Rumus 4 Rumus Entropy

2. Pembagian dataset: Membagi dataset berdasarkan nilai dari atribut yang dipilih.
3. Rekursi: Mengulangi proses pemilihan atribut dan pembagian dataset untuk setiap subset hingga kondisi penghentian tercapai.

4. Pembangunan pohon: Proses ini berlanjut sampai setiap daun pada pohon mewakili kelas target atau hingga kriteria penghentian lainnya terpenuhi.

Berikut ini adalah kelebihan dan kekurangan dari *Decision Tree*:

Kelebihan:

A. Sederhana dan interpretative: Mudah dipahami dan diinterpretasikan. Hasilnya dapat divisualisasikan dalam bentuk pohon keputusan yang intuitif.
B. Non-parametrik: Tidak mengasumsikan distribusi data tertentu.
C. Fleksibel: Dapat menangani data numerik dan kategorikal.

Kekurangan:

A. *Overfitting*: Decision Tree cenderung mengalami *overfitting* terutama jika pohonnya terlalu dalam dan kompleks.
B. Perubahan kecil dalam data dapat menghasilka pohon yang sangat berbeda.
C. Atribut dengan banyak nilai unik cenderung dipilih sebagai pembagi karena mereka dapat membagi data lebih baik daripada atribut dengan sedikit nilai.

Dalam konteks penelitian deteksi penyakit pada hewan ternak kambing berdasarkan suhu tubuh, *Decision Tree* dapat digunakan untuk membangun model yang memprediksi kondisi kesehatan kambing. Dengan menggunakan data suhu tubuh dan fitur lainnya, Decision Tree dapat membantu mengidentifikasi pola dan aturan yang menunjukkan apakah kambing tersebut sehat atau terkena penyakit.

2.3.3 *Naive Bayes*

Naive Bayes adalah sebuah algoritma klasifikasi yang dibangun berdasarkan teorema Bayes dan dilakukan dengan asumsi bahwa semua prediktor dalam model adalah independent satu sama lain. Dengan kata lain, *classifier Naive Bayes* beroperasi di bawah asumsi

bahwa kehadiran suatu fitur dalam sebuah kelas tidak dipengaruhi oleh kehadiran fitur lainnya.

Teorema Bayes memberikan cara untuk menghitung probabilitas posterior $P(C|X)$ dari sebuah kelas C mengingat fitur X, menggunakan probabilitas prior $P(C)$, probabilitas likelihood $P(X|C)$, dan probabilitas evidence $P(X)$. Rumus Teorema Bayes adalah sebagai berikut:

$$P(C|X) = \frac{P(X|C) \cdot P(C)}{P(X)}$$

Rumus 5 Rumus Teorema Bayes

Di mana:

- $P(C|X)$ adalah probabilitas posterior bahwa sampel X termasuk dalam kelas C.
- $P(X|C)$ adalah probabilitas likelihood dari X mengingat kelas C.
- $P(C)$ adalah probabilitas prior dari kelas C.
- $P(X)$ adalah probabilitas evidence dari fitur X.

Berikut ini adalah jenis-jenis dari Naïve Bayes:

A. Gaussian Naïve Bayes: Digunakan ketika fitur-fitur kontinu dan diasumsikan mengikuti distribusi normal (Gaussian).
B. Multinomial Naïve Bayes: Cocok untuk data dengan fitur diskrit, seperti frekuensi kata dalam dokumen teks.
C. Bernoulli Naïve Bayes: Digunakan untuk data biner, di mana fitur-fitur hanya memiliki dua nilai (0 dan 1).

Berikut ini adalah kelebihan dan kekurangan Naïve Bayes

Kelebihan:

A. Sederhana dan Cepat: Algoritma ini mudah diimplementasikan dan cepat dalam komputasi, terutama untuk dataset besar.

B. Efektif pada Data Kecil: Seringkali memberikan kinerja yang baik bahkan dengan jumlah data pelatihan yang kecil.
C. Skalabilitas: Dapat menangani sejumlah besar fitur.

Kekurangan:

A. Asumsi Kemandirian: Asumsi bahwa semua fitur independen satu sama lain jarang terjadi dalam kenyataan, yang dapat mempengaruhi akurasi.
B. Estimasi Probabilitas Nol: Jika nilai probabilitas likelihood suatu fitur adalah nol, maka probabilitas posterior juga menjadi nol. Hal ini dapat diatasi dengan teknik smoothing seperti Laplace smoothing.

Dalam konteks penelitian deteksi penyakit pada hewan ternak kambing berdasarkan suhu tubuh, algoritma Naive Bayes dapat digunakan untuk mengklasifikasikan kondisi kesehatan kambing. Dengan menggunakan data suhu tubuh dan fitur lainnya, Naive Bayes dapat membantu memprediksi apakah kambing tersebut dalam kondisi sehat atau mengalami penyakit tertentu berdasarkan probabilitas.

2.3.4 *Random Forest*

Random Forest adalah salah satu algoritma *machine learning* yang termasuk dalam kategori *ensemble learning*, dimana *random forest* menggabungkan prediksi dari beberapa model untuk membuat prediksi yang lebih akurat daripada model individual. Algoritma ini menggunakan banyak *decision tree* untuk membuat Keputusan akhir. Konsep utama dari *random forest* adalah bahwa kumpulan dari banyak model sederhana, masing-masing dilatih dengan subset data yang sedikit berbeda, dapat menghasilkan prediksi yang lebih akurat dan robust daripada satu pohon keputusan yang dilatih dengan seluruh dataset.

Proses pembuatan model *random forest* dimulai dengan teknik yang disebut "*bootstrap aggregating*" atau "*bagging*". Dalam proses ini,

beberapa subset data dibuat dari dataset asli dengan pengambilan sampel ulang dengan penggantian, artinya beberapa sampel dapat muncul lebih dari sekali dalam satu subset, dan beberapa mungkin tidak muncul sama sekali. Setiap subset ini kemudian digunakan untuk melatih pohon keputusan yang berbeda. Karena setiap pohon yang dilatih dengan subset data yang berbeda, mereka cenderung membuat prediksi yang berbeda-beda. Dengan menggabungkan prediksi dari semua pohon tersebut, *random forest* mengurarngi risiki *overfitting* yang sering dihadapi oleh model pohon keputusan tunggal.

Dalam prakteknya, prediksi akhir dari Random Forest dihasilkan dengan mengambil rata-rata dari prediksi semua pohon dalam kasus regresi, atau dengan mengambil modus (nilai yang paling sering muncul) dari prediksi semua pohon dalam kasus klasifikasi.

Dengan cara ini, *random forest* berhasil menggabungkan kekuatan prediktif dari banyak pohon keputusan, sambil mengurangi variabilitas dan kesalahan prediksi yang sering muncul ketika hanya menggunakan satu model. Random Forest bekerja dengan membangun sejumlah besar pohon keputusan selama fase pelatihan dan menggabungkan hasilnya untuk membuat prediksi yang lebih akurat dan stabil. Inti dari Random Forest adalah mengurangi overfitting yang sering terjadi pada pohon keputusan individual dengan cara membangun berbagai model dan menggabungkan hasilnya.

Berikut ini adalah tahapan dari *Random Forest*:

1. Bagging (Bootstrap Aggregating): Membuat beberapa subset dari dataset pelatihan dengan menggunakan teknik bootstrapping. Bootstrapping adalah proses sampling dengan penggantian, di mana sampel diambil dari dataset asli untuk membentuk beberapa subset.

2. Pembangunan Pohon Keputusan: Membangun pohon keputusan untuk setiap subset. Selama pembentukan setiap pohon, hanya sebagian acak dari fitur yang dipilih untuk menentukan split terbaik di setiap node, yang membantu dalam mengurangi korelasi antar pohon.
3. Agregasi Hasil: Setelah semua pohon keputusan dibangun, prediksi untuk setiap pohon dikombinasikan untuk menghasilkan prediksi akhir. Untuk masalah klasifikasi, prediksi akhir biasanya ditentukan dengan voting mayoritas (majority voting), sementara untuk regresi, prediksi akhir adalah rata-rata dari semua prediksi pohon.

Berikut ini adalah kelebihan dan kekurangan *Random Forest*:

Kelebihan:

A. Akurasi Tinggi: Random Forest seringkali menghasilkan prediksi yang sangat akurat dengan mengurangi variabilitas dan overfitting.
B. Robustness: Algoritma ini sangat tahan terhadap outliers dan data noise.
C. Fleksibilitas: Dapat digunakan untuk masalah klasifikasi dan regresi.
D. Estimasi Fitur Penting: Random Forest dapat digunakan untuk memperkirakan pentingnya fitur dalam dataset.

Kekurangan:

A. Kompleksitas dan Waktu Komputasi: Membangun sejumlah besar pohon keputusan bisa memakan waktu dan memori yang signifikan.
B. Kurangnya Interpretabilitas: Meskipun setiap pohon keputusan individual dapat diinterpretasikan, menggabungkan sejumlah besar pohon membuat model akhir menjadi kurang transparan.

Berikut ini adalah proses pembentukan *Random Forest*

A. Inisialisasi: Tentukan jumlah pohon (n_estimators) yang akan dibangun dan jumlah fitur maksimum (max_features) yang akan dipertimbangkan untuk setiap split.
B. Pembentukan Subset: Untuk setiap pohon, buat subset data dengan teknik bootstrapping dari dataset pelatihan.
C. Pembentukan Pohon: Buat pohon keputusan dari setiap subset dengan memilih split terbaik berdasarkan subset fitur acak yang dipilih pada setiap node.
D. Prediksi: Untuk membuat prediksi, gunakan voting mayoritas untuk masalah klasifikasi atau rata-rata prediksi untuk masalah regresi.

Dalam konteks penelitian deteksi penyakit pada hewan ternak kambing berdasarkan suhu tubuh, algoritma Random Forest dapat digunakan untuk membangun model prediksi yang andal. Dataset suhu tubuh kambing dan fitur-fitur lainnya dapat digunakan untuk melatih beberapa pohon keputusan. Random Forest akan membantu mengurangi risiko overfitting dan meningkatkan akurasi prediksi dengan menggabungkan hasil dari berbagai pohon keputusan.

Algoritma Random Forest menawarkan pendekatan yang kuat dan fleksibel untuk tugas-tugas klasifikasi dan regresi. Dengan menggabungkan prediksi dari berbagai pohon keputusan, Random Forest mampu menghasilkan model yang lebih stabil dan akurat. Dalam penelitian ini, algoritma Random Forest dapat diandalkan untuk mengidentifikasi pola dalam data suhu tubuh kambing dan memprediksi kondisi kesehatan dengan presisi tinggi.

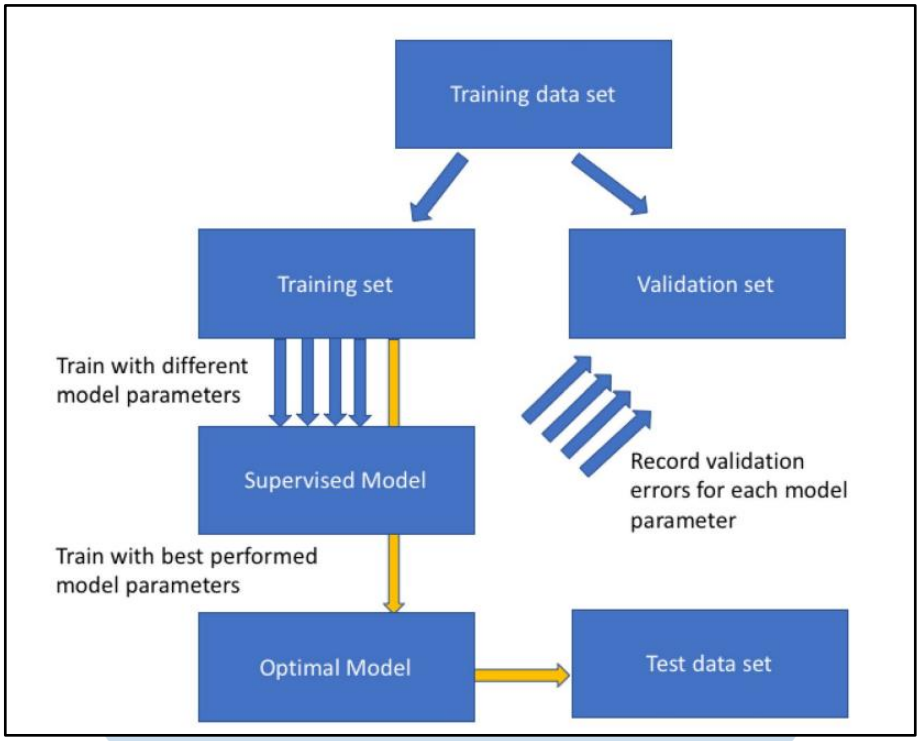
2.4 Data *Training*, Data Validasi, dan Data *Testing*

Dalam penelitian untuk mendeteksi penyakit pada hewan ternak kambing berdasarkan suhu tubuh, dataset dibagi menjadi tiga bagian: data *training*, data validasi, dan data *testing*. Data *training* digunakan untuk mengajari model agar

dapat mengidentifikasi objek tertentu dalam dataset. Selama proses pelatihan, model dipersiapkan untuk mengenali pola atau fitur khusus yang terdapat dalam dataset. Setelah pelatihan selesai, model akan diuji menggunakan dua kelompok data yang berbeda, yaitu data validasi dan data testing [9].

Penggunaan data validasi adalah untuk mengevaluasi kinerja model prediksi yang dilatih. Evaluasi dengan menggunakan data validasi melibatkan perbandingan model prediksi dengan label sebenarnya pada data validasi. Tujuan evaluasi ini adalah untuk mengukur sejauh mana model dapat mengidentifikasi objek tertentu yang tidak diketahui saat proses pelatihan. Selama proses ini, dapat terlihat apakah model mengalami *overfitting* atau *underfitting* [10].

Evaluasi menggunakan data *testing* bertujuan untuk menguji akurasi dan kinerja keseluruhan model prediksi. Data yang digunakan dalam tahap ini harus berbeda dengan data *training* maupun data validasi. Hasil evaluasi ini memberikan gambaran yang akurat tentang kemampuan model dalam mendeteksi objek pada dataset yang digunakan [9]. Gambaran tentang proses pembuatan dan pengujian model prediksi dengan menggunakan data *training*, data validasi, dan data *testing* dapat ditemukan pada Gambar … .



Gambar 2. 4 Alur Pemilihan Model Deteksi [9]

2.5 Evaluasi Model

2.5.1 Confusion matrix

Confusion matrix biasa digunakan untuk klasifikasi jumlah data *train* yang benar dan yang salah [11]. Biasanya, *confusion matrix* berbentuk tabel 2x2 yang memuat dua kelas, yaitu positif dan negatif. Dari *confusion matrix*, dapat diperoleh informasi mengenai prediksi yang benar yang dihasilkan oleh model yang telah dilatih. Struktur *confusion matrix* dapat ditemukan pada Tabel 2.1.

Tabel 2. 1 Struktur Confusion Matrix [12]

Class	***Classified as Positive***	***Classified as Negative***
Positive	*True Positive (TP)*	*False Negative (FN)*
Negative	*False Positive (FP)*	*True Negative (TN)*

2.6 Teori tentang Tools / Software yang digunakan

2.6.1 *Phyton*

Python adalah bahasa pemrograman tingkat tinggi yang dikenal dengan sintaksnya yang jelas dan mudah dibaca, membuatnya populer di kalangan pemula dan juga profesional. Diciptakan oleh Guido van Rossum dan pertama kali dirilis pada tahun 1991, Python mendukung berbagai paradigma pemrograman, termasuk pemrograman prosedural, objek-orientasi, dan fungsional. Python dirancang untuk menjadi mudah digunakan dan dapat dipelajari dengan cepat, tetapi juga kuat dan serbaguna, memungkinkan penggunaannya dalam pengembangan web, analisis data, kecerdasan buatan, ilmu komputer, dan banyak lagi.

Salah satu kekuatan utama Python adalah komunitas besar dan aktif yang terus mengembangkan dan memelihara berbagai pustaka dan kerangka kerja. Pustaka-pustaka ini, seperti NumPy untuk komputasi numerik, Pandas untuk manipulasi data, dan TensorFlow serta PyTorch untuk pembelajaran mesin, memudahkan para pengembang untuk melakukan tugas-tugas kompleks tanpa perlu menulis banyak kode dari awal. Selain itu, Python memiliki ekosistem yang kaya dengan alat pengembangan, seperti IPython dan Jupyter Notebook, yang mendukung eksplorasi interaktif dan visualisasi data.

Python juga sangat fleksibel dan dapat dijalankan di hampir semua sistem operasi dengan sedikit atau tanpa modifikasi kode. Hal ini menjadikannya pilihan yang populer untuk pengembangan perangkat lunak lintas platform. Python diterjemahkan menggunakan interpreter, yang berarti kode dapat dijalankan segera setelah ditulis, memudahkan pengujian dan debug. Kecepatan eksekusi yang lambat relatif terhadap bahasa yang dikompilasi seperti C++ sering diatasi dengan menggunakan ekstensi yang ditulis dalam bahasa lain seperti C, yang dapat diintegrasikan dengan mudah. Dengan kelebihan-kelebihan ini, Python terus memperkuat posisinya sebagai salah satu

bahasa pemrograman yang paling berpengaruh dan banyak digunakan di dunia.

2.6.2 *Jupyter Notebook*

Jupyter Notebook adalah aplikasi web open-source yang memungkinkan pengguna untuk membuat dan berbagi dokumen yang mengandung kode live, persamaan, visualisasi, dan teks naratif. Alat ini sangat populer di kalangan ilmuwan data, peneliti, dan pendidik untuk melakukan analisis data, pemodelan statistik, visualisasi data, machine learning, dan banyak lagi. Dengan Jupyter Notebook, pengguna dapat menulis dan menjalankan kode dalam berbagai bahasa pemrograman seperti Python, R, Julia, dan Scala, yang disatukan dalam sebuah lingkungan yang mudah diakses melalui browser web.