BAB II
TINJAUAN PUSTAKA

2.1 Penelitian Terkait

Penelitian yang merujuk pada [4] oleh Kharis Hudaiby Hanif dan Novita Ranti Muntiari pada tahun 2024 dengan judul "Penerapan Algoritma *Decision Tree, SVM, Naïve Bayes* Dalam Deteksi Stunting Pada Balita" Hasil dari pengujian dengan data *testing* 30 % dan data *training* 70 % menggunakan algoritma *decision tree, naïve bayes, dan SVM*. Hasil uji tingkat akurasi *decision tree* sebesar 99%, *naïve bayes* sebesar 48%, dan *SVM* sebesar 95%. Jadi, algoritma yang tingkat akurasi paling tinggi yaitu *decision tree* sebesar 99%. Maka algotitma *decision tree* lebih baik untuk deteksi stunting pada balita.

Penelitian dengan judul "Perbandingan Metode Klasifikasi *Naive Bayes, Decision Tree, Random Forest* Terhadap Analisis Sentimen Kenaikan Biaya Haji 2023 Pada Media Sosial Youtube" oleh Muhammad Yasir, Robertus Suraji pada tahun 2023 Penelitian ini melakukan perbandingan hasil akurasi terhadap beberapa metode klasifikasi seperti *naïve bayes, decision tree dan random forest*. Dari penelitian ini didapatkan hasil akurasi *naïve bayes* sebesar 90%, *Decision Tree* sebesar 83%, *Random Forest* sebesar 87%. [5]

Penelitian yang dilakukan oleh Mario Utomo, Rastri Prathivi pada tahun 2024 dengan judul "Perbandingan Algoritma Support Vector Machine dan Decision Tree untuk Klasifikasi Performa Perusahaan Mario Berdasarkan Kualitas Tidur" Penelitian ini membandingkan *Support Vector Machine* dan *Decision Tree* dengan pendekatan *One Against All* dalam mengklasifikasikan performa perusahaan. Feature yang digunakan untuk klasifikasi performa perusahaan ini

terdiri dari 3 rasio keuangan yaitu rasio profitabilitas (ROA), likuiditas (CR), dan *leverage* (DER). Labelling atau target dalam klasifikasi dibagi menjadi 3 kategori yaitu normal, baik, dan kurang baik. Pada penelitian ini akan mempertimbangkan evaluasi seperti *accuracy*, cross *validation*, dan *confusion matrix*. Hasil kinerja algoritma *Support Vector Machine* menghasilkan akurasi sebesar 86,67%, sedangkan pada algoritma *Decision Tree* menghasilkan akurasi sebesar 93,33%.[6]

Penelitian yang merujuk pada [7] oleh Deo Haganta Depari, Yuni Widiastiwi, Mayanda Mega Santoni pada tahun 2022 dengan judul "Perbandingan Model Decision Tree, Naive Bayes dan Random Forest untuk Prediksi Klasifikasi Penyakit Jantung" Tujuan dari penelitian ini adalah untuk bagaimana mengolah dan melakukan analisa data, bagaimana penerapan metode Decision Tree, Naive Bayes dan Random Forest pada klasifikasi penyakit jantung, kemudian bagaimana hasil akurasi metode-metode yang digunakan tersebut, bagaimana hasil perbandingan antara Decision Tree, Naive Bayes dan Random Forests yang digunakan dan metode apa yang merupakan terbaik dari klasifikasi penyakit jantung. Hasil dari penelitian ini adalah evaluasi performa metode klasifikasi Decision Tree, Naive Bayes dan Random Forest. Dimana nilai akurasi metode Decision Tree sebesar 0.71%, Naive Bayes sebesar 0.72% dan Random Forest sebesar 0.75%.

Tabel 2.1 Penelitian Terkait

No.	Judul Penelitian	Metode	Akurasi	Jumlah Dataset
1.	Penerapan Algoritma *Decision Tree, SVM, Naïve Bayes* Dalam Deteksi Stunting Pada	*Decision Tree, SVM, Naïve Bayes*	*Decision Tree* (99%), *SVM* (95%), *Naïve Bayes* (48%)	120999 Data

	Balita			
2.	Perbandingan Metode Klasifikasi *Naive Bayes, Decision Tree, Random Forest* Terhadap Analisis Sentimen Kenaikan Biaya Haji 2023 Pada Media Sosial Youtube	*Support Vector Machine* dan *Decision Tree*	*Naive Bayes (90%), Decision Tree (83%), Random Forest (87%)*	1.014 Data
3.	Perbandingan Algoritma *Support Vector Machine* dan *Decision Tree* untuk Klasifikasi Performa Perusahaan Mario Berdasarkan Kualitas Tidur	*Support Vector Machine* dan *Decision Tree*	*Support Vector Machine* (86,67%), *Decision Tree (93,33%)*	150 Data
4.	Perbandingan Model *Decision Tree, Naive Bayes* dan *Random Forest* untuk Prediksi Klasifikasi Penyakit Jantung	*Decision Tree, Naive Bayes* dan *Random Forest*	*Decision Tree* (0.71%) , *Naive Bayes* (0.72%)dan *Random Forest* (0.75%)	54746 Data

2.2 Landasan Teori

a. Klasifikasi

Klasifikasi adalah teknik data mining yang menetapkan kategori pada kumpulan data untuk membantu dalam memprediksi dan analisis yang lebih akurat. Oleh karena itu tiga algoritma klasifikasi machine learning yaitu *Decision Tree, Naive Bayes dan Suport Vector Machine* digunakan dalam percobaan ini untuk mendeteksi diabetes secara dini.[8]

b. Stunting

Stunting adalah permasalahan gizi kronis disebabkan kurangnya asupan gizi dengan rentang waktu yang cukup lama yang berdampak pada tumbuh kembangnya secara fisik seperti tinggi yang kurang, dan berat badan yang kurang dari standar pertumbuhan anak yang dikeluarkan oleh WHO. Tidak hanya mempengaruhi perkembangan secara fisik stunting juga mempengaruhi perkembangan otak hal ini mempengaruhi kemampuan mental dan belajar tidak maksimal.[9]

Berbagai faktor mempengaruhi terjadinya stunting, antara lain kekurangan gizi dalam periode 1.000 hari pertama kehidupan, infeksi berulang, serta kurangnya akses terhadap layanan kesehatan yang memadai. Selain itu, faktor sosial ekonomi, seperti tingkat pendidikan ibu, pendapatan keluarga, dan kondisi sanitasi lingkungan, juga berperan penting. Anak-anak yang dibesarkan dalam lingkungan dengan akses terbatas terhadap makanan bergizi, perawatan kesehatan, dan edukasi cenderung memiliki risiko lebih tinggi untuk mengalami stunting.

Penelitian menunjukkan bahwa stunting tidak hanya memengaruhi pertumbuhan fisik, tetapi juga berhubungan erat dengan perkembangan kognitif, yang dapat berdampak pada kemampuan belajar dan kesehatan

mental di masa depan. Organisasi Kesehatan Dunia (WHO) memperingatkan bahwa stunting dapat menyebabkan konsekuensi jangka panjang, seperti peningkatan risiko penyakit kronis dan kehilangan potensi ekonomi. Oleh karena itu, penanganan dan pencegahan stunting memerlukan pendekatan komprehensif yang mencakup intervensi gizi, akses terhadap layanan kesehatan, dan perbaikan kondisi sosial ekonomi di tingkat komunitas.

c. *Machine Learning*

Machine learning adalah suatu metode pembelajaran mesin yang mengacu pada teknik yang berhubungan dengan pola berdasarkan model untuk kalsifikasi dan prediksi data baru. Machine learning merupakan cabang dari kecerdasan buatan (*artificial intelligence*) yang berfokus pada pengembangan sistem atau algoritma yang memungkinkan komputer untuk belajar dan membuat keputusan tanpa diprogram secara eksplisit. Dalam *machine learning*, model dibangun dengan menganalisis data historis (data latih) untuk menemukan pola atau hubungan tertentu yang kemudian digunakan untuk membuat prediksi atau keputusan pada data baru. Proses pembelajaran ini dapat bersifat *supervised* (dengan data berlabel), *unsupervised* (tanpa label), atau reinforcement learning (berbasis umpan balik).[10]

Machine learning digunakan dalam berbagai aplikasi seperti klasifikasi teks, pengenalan wajah, deteksi anomali, dan analisis prediktif, dengan algoritma populer seperti *decision tree, random forest, support vector machine* (SVM), dan *neural network*. Keunggulannya adalah kemampuannya untuk menangani dataset besar dan kompleks, tetapi

keberhasilannya sangat bergantung pada kualitas data, pemilihan algoritma, serta parameter yang digunakan.

d. *Decision Tree*

Decision Tree adalah struktur *flowchart* yang menyerupai *Tree* (pohon), dimana setiap simpul internal menandakan suatu tes pada atribut, setiap cabang merepresentasikan hasil tes,dan simpul daun merepresentasikan kelas atau distribusi kelas [11]. Decision tree merupakan salah satu jenis algoritma penambangan data yang paling populer untuk klasifikasi dan prediksi. Dalam decision tree ini data yang berupa fakta dirubah menjadi sebuah pohon keputusan yang berisi aturan dan decision tree mengatur catatan dalam struktur pohon yang terdiri dari simpul akar, cabang, dan simpul daun. Node akar berada di bagian atas struktur pohon. Node mewakili atribut, cabang mewakili hasil, lalu daun mewakili keputusan.[12]

Decision Tree merupakan suatu struktur yang mempunyai dasar dari proses dimana sifatnya sekuensial, dalam decision tree ini data yang berupa fakta dirubah menjadi sebuah pohon keputusan yang berisi aturan dan tentunya dapat lebih mudah dipahami dengan bahasa alami. Model pohon keputusan banyak digunakan pada kasus data dengan output yang bernilai diskrit. Walaupun tidak menutup kemungkinan dapat juga digunakan untuk kasus data dengan atribut numeric.[7]

Decision Tree memiliki kelebihan yaitu dimana dalam daerah pengambilan keputusan yang sebelum nya kompleks dan sangat global, dapat diubah menjadi lebih simpel dan spesifik. Dengan jumlah kriteria yang lebih sedikit pada setiap node internal metode *decision tree* dapat mencegah munculnya permasalahan tersebut tanpa mengurangi kualitas sebuah

keputusan yang dihasilkan, Salah satu kelebihan *decision tree* lainnya adalah *decision tree* memiliki model yang sederhana dan mudah dipahami karena ditampilkan dalam bentuk pohon yang bercabang, sehingga memudahkan dalam interpretasi, selain itu decision tree berdasarkan penelitian sebelumnya juga mendapat nilai akurasi yang termasuk kategori tinggi.

Rumus dasar *Decision Tree :*

1) Entropy untuk mengukur impurity dataset:

$$H(S) = -\sum_{i=1}^{n} P_i \log_2(P_i)$$

H(S) : Entropy dataset

Pi : probabilitas data berada di kelas i

2) Information Gain (IG) untuk memilih atribut terbaik (berdasarkan pengurangan Entropy):

$$IG(S, A) = H(S) - \sum_{v \in \text{Values}(A)} \frac{|S_v|}{|S|} H(S_v)$$

IG(S,A) : Information Gain untuk atribut A.

H(S) : Entropy dataset awal.

H(Sv) : Entropy subset Sv setelah dataset dibagi berdasarkan atribut A.

3) Gini Impurity untuk mengukur impurity dataset (alternatif dari Entropy):

$$Gini(S) = 1 - \sum_{i=1}^{n} P_i^2$$

Gini(S) : Gini Impurity dataset S.

Pi : probabilitas data berada di kelas i.

e. *Naïve Bayes*

Naïve Bayes merupakan sebuah model klasifikasi statistik yang dapat digunakan untuk memprediksi probabilitas keanggotaan suatu kelas, dimana Metode *Naive Bayes* merupakan suatu metode yang digunakan untuk memprediksi suatu keputusan berdasarkan dengan kriteria yang sudah ditentukan. Metode ini juga dapat digunakan untuk meningkatkan probabilitas klasifikasi pada variabel dan faktor kondisi [13]. Algoritma ini mengasumsikan bahwa setiap fitur dalam data bersifat independen satu sama lain (*naive assumption*), meskipun dalam kenyataannya, fitur-fitur tersebut seringkali saling bergantung. Dengan menggunakan *teorema Bayes*, algoritma ini menghitung probabilitas posterior suatu kelas dengan mempertimbangkan distribusi nilai-nilai fitur, lalu memilih kelas dengan probabilitas tertinggi sebagai hasil prediksi. *Naive Bayes* sangat cocok untuk tugas klasifikasi, terutama untuk data dengan dimensi tinggi, karena algoritma ini tidak membutuhkan banyak sumber daya komputasi dan memiliki waktu pelatihan yang relatif cepat.[14]

Naive Bayes sering diterapkan dalam berbagai aplikasi, seperti deteksi spam pada email, analisis sentimen dalam teks, dan klasifikasi dokumen. *Naive bayes* memanfaatkan *teorema bayes* kemungkinan masa depan berdasarkan data atau pengalaman sebelumnya, *Teorema Bayes* adalah prinsip fundamental dalam teori probabilitas dan inferensi statistik yang digunakan untuk memperbarui keyakinan suatu hipotesis berdasarkan data baru [15]. Meskipun algoritma ini sederhana, asumsi independensinya memungkinkan perhitungan probabilitas menjadi lebih efisien, meskipun kinerjanya dapat menurun jika fitur memiliki korelasi yang tinggi atau jika data tidak seimbang. *Naive Bayes* juga memiliki beberapa varian, seperti Gaussian *Naive Bayes* untuk data kontinu, Multinomial *Naive Bayes* untuk

data diskrit, dan Bernoulli *Naive Bayes* untuk data biner, sehingga fleksibel untuk berbagai jenis dataset. Keunggulan utama algoritma ini adalah kesederhanaannya, efisiensinya dalam menangani dataset besar, dan kemampuannya memberikan hasil prediksi yang kompetitif pada data dengan jumlah latih yang terbatas.[16]

Rumus dasar *Teorema Bayes* :

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

P(C|X) : Probabilitas bahwa kelas C terjadi diberikan fitur X (probabilitas posterior).
P(X|C) : Probabilitas fitur X muncul jika kelas C diketahui (likelihood).
P(C) : Probabilitas awal kelas C (prior probability).
P(X) : Probabilitas keseluruhan fitur X (evidence).

f. *Support Vector Machine*

SVM adalah algoritma klasifikasi yang berfungsi untuk melakukan klasifikasi data non linier dan data liner. SVM dikenal karena kemampuannya dalam menangani data berdimensi tinggi dan bekerja dengan baik pada dataset yang relatif kecil namun dengan jumlah fitur yang besar[3]. Karena kesederhanaan dan fleksibilitasnya yang relatif untuk mengatasi berbagai masalah klasifikasi, SVM secara khusus memberikan kinerja prediksi yang seimbang Dimana algoritma ini bekerja dengan mencari hyperplane atau batas keputusan yang dapat memisahkan data dari dua kelas dengan margin terbesar, yaitu jarak maksimum antara data titik terdekat (*support vectors*) dari masing-masing kelas dengan hyperplane. Jika data tidak dapat dipisahkan secara linier, SVM menggunakan fungsi kernel untuk memetakan data ke dimensi yang lebih tinggi agar menjadi

linier. Fungsi kernel yang sering digunakan meliputi linear, polynomial, radial basis function (RBF), dan sigmoid.[17]

SVM memiliki 2 metode yaitu regresi (*Support Vector Regression*) dan klasifikasi (*Support Vector Classification*) Untuk menangani data yang tidak sepenuhnya terpisah, SVM menerapkan konsep soft margin dengan parameter regularisasi C yang mengontrol keseimbangan antara margin yang lebar dan kesalahan klasifikasi. SVM dikenal efektif untuk data berdimensi tinggi, fleksibel dalam menangani data non-linear, dan robust terhadap overfitting, tetapi memiliki keterbatasan pada dataset besar karena waktu komputasi yang tinggi. Algoritma ini banyak diterapkan dalam berbagai bidang, seperti deteksi spam, pengenalan pola pada citra, analisis bioinformatika, dan analisis sentiment.[18]

Rumus dasar *Support Vector Machine* :

$$w \cdot x + b = 0$$

w : vektor bobot (parameter model).

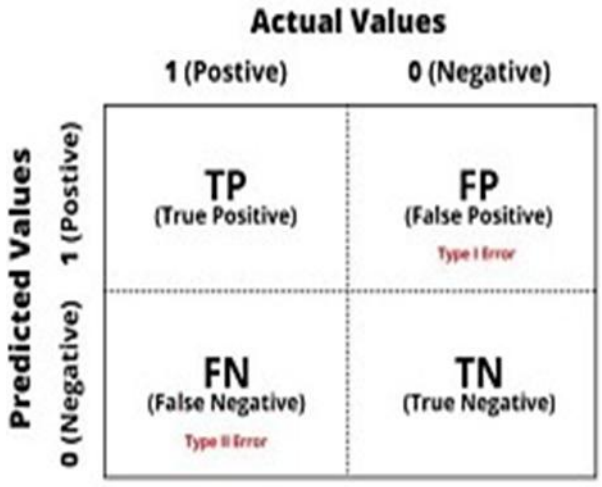
x : vektor fitur (input data).

b : bias (intersep).

g. Metrik Evaluasi

Pada tahap ini, pengujian akurasi hasil dilakukan dengan confusion matrix yang digunakan untuk mengukur performa model *Machine Learning*. *Confusion Matrix* adalah tabel yang terdiri dari jumlah baris data uji yang diprediksi benar dan salah dengan menggunakan model klasifikasi yang digunakan [19]. *Confusion matrix* atau Matriks kebingungan adalah ukuran yang sangat populer digunakan saat memecahkan masalah klasifikasi,

Matriks ini menyajikan ringkasan hasil prediksi model pada sebuah dataset untuk mengevaluasi seberapa akurat atau salah model dalam mengklasifikasikan kumpulan data.[20]



Keterangan :

1) TP (True Positives) : Jumlah kasus yang benar-benar positif dan diprediksi sebagai positif oleh model.

2) FP (False Positives) : Jumlah kasus yang sebenarnya negatif tetapi diprediksi sebagai positif oleh model.

3) FN (False Negatives): Jumlah kasus yang sebenarnya positif tetapi diprediksi sebagai negatif oleh model.

4) TN (True Negatives) : Jumlah kasus yang benar-benar negatif dan diprediksi sebagai negatif oleh model.

Berdasarkan Confusion Matrix, Berikut Rumus untuk menghitung matrik evaluasi klasifikasi:

Pada Tabel 1 TP adalah True Positive, TN adalah True Negative, FP adalah *False Positive* dan FN adalah *False Negative*. Pada penelitian ini

performa klasifikasi yang akan dihitung adalah *accuracy*, *precision*, *recall* dan F1 *score*. rumus perhitungan *accuracy*, *precision*, *recall* dan F1 *score*.

a) Accuracy merupakan perhitungan untuk mengukur persentase prediksi yang benar dari keseluruhan data.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

b) Precision merupakan perhitungan untuk mengukur akurasi dan prediksi positif.

$$\mathbf{Precision} = \frac{\mathbf{TP}}{\mathbf{TP} + \mathbf{FP}}$$

c) Recall adalah mengukur sensitivitas dari model, yaitu seberapa baik model dalam mendeteksi kelas positif.

$$\mathbf{Recall} = \frac{\mathbf{TP}}{\mathbf{TP} + \mathbf{FN}}$$

d) F1-Score adalah perhitungan rata-rata harmonis dari precision dan recall, yang berguna jika terdapat ketidakseimbangan kelas.

$$\mathbf{F1\ Score} = \mathbf{2} \times \frac{\mathbf{Precision} \times \mathbf{Recall}}{\mathbf{Precision} + \mathbf{Recall}}$$

h. Google Colaboratory

Google Collaboratory atau Google Colab adalah platform berbasis cloud untuk menulis, menjalankan, dan berbagi kode Python melalui web browser. Platform. Platform ini dirancang bagi analyst, developer, peneliti, dan pendidik yang bekerja di bidang data science dan machine learning dengan menyediakan environment komputasi yang fleksibel dan mudah diakses tanpa biaya. Google Colab juga menawarkan kemampuan untuk menjalankan Jupyter Notebook (web app open- source untuk kombinasi kode, teks terformat, dan visualisasi data) langsung dari web browser tanpa perlu konfigurasi apa pun.[21]