

# Analisis Komparatif Kinerja Algoritma Machine Learning untuk Deteksi Status Gizi Balita

Della Sabrina, Defri Kurniawan\*

Fakultas Ilmu Komputer, Program Studi Teknik Informatika, Universitas Dian Nuswantoro, Semarang, Indonesia

Email: <sup>1</sup>111202214019@mhs.dinus.ac.id, <sup>2,\*</sup>defri.kurniawan@dsn.dinus.ac.id

Email Penulis Korespondensi: 111202214019@mhs.dinus.ac.id

Submitted: 09/11/2025; Accepted: 06/12/2025; Published: 08/12/2025

**Abstrak**—Status gizi pada anak usia balita berperan sebagai indikator utama dalam menilai kondisi kesehatan, pertumbuhan, dan perkembangan anak secara menyeluruh. Penentuan status gizi secara konvensional biasanya dilakukan melalui pengukuran manual dan interpretasi tabel antropometri, yang memerlukan waktu cukup lama serta berpotensi menimbulkan kesalahan manusiawi. Seiring dengan kemajuan teknologi, pendekatan berbasis *machine learning* dapat dimanfaatkan untuk membantu proses klasifikasi status gizi secara lebih cepat, objektif, dan akurat, sehingga dapat mendukung proses pengambilan keputusan di bidang kesehatan masyarakat. Penelitian ini difokuskan untuk menganalisis serta membandingkan terhadap kinerja tiga algoritma *machine learning*, yaitu *Support Vector Machine* (SVM), *K-Nearest Neighbors* (KNN), dan *Decision Tree* (DT) dalam mengelompokkan tingkat status gizi balita dengan memanfaatkan data antropometri yang mencakup variabel umur, jenis kelamin, berat badan, dan tinggi badan. Dalam penelitian ini, kategori status gizi yang diklasifikasikan untuk dataset berat balita meliputi: Severely Underweight, Underweight, Normal, dan Overweight. Adapun kategori untuk dataset tinggi badan meliputi Severely Stunted, Stunted, Normal, dan Tinggi. Tahapan penelitian meliputi proses *data preprocessing*, pemecahan data ke dalam *training* serta *testing*, serta penilaian performa model melalui matrik akurasi, presisi, *recall*, dan *F1-score*. Berdasarkan hasil evaluasi terhadap dataset tinggi balita, algoritma *K-Nearest Neighbors* (KNN) terbukti menjadi model dengan performa paling unggul, dengan akurasi mencapai 99,91%. Nilai tersebut melampaui *Decision Tree* yang memperoleh akurasi 99,89%, serta algoritma SVM (RBF) yang mencapai 98,48%.

**Kata Kunci:** Pembelajaran Mesin; Klasifikasi; Support Vector Machine; K-Nearest Neighbors; Decision Tree

**Abstract**—Nutritional status in children under five years of age serves as a key indicator in assessing the overall health, growth, and development of children. Conventionally, nutritional status is determined through manual measurements and interpretation of anthropometric tables, which is time-consuming and prone to human error. With advances in technology, machine learning-based approaches can be used to help classify nutritional status more quickly, objectively, and accurately, thereby supporting decision-making in public health. This study focuses on analyzing and comparing the performance of three machine learning algorithms, namely Support Vector Machine (SVM), K-Nearest Neighbors (KNN), and Decision Tree (DT) in classifying the nutritional status of toddlers using anthropometric data that includes variables such as age, gender, weight, and height. In this study, the nutritional status categories classified for the toddler weight dataset include: Severely Underweight, Underweight, Normal, and Overweight. The categories for the height dataset include Severely Stunted, Stunted, Normal, and Tall. The research stages included data preprocessing, data splitting into training and testing, and model performance assessment through accuracy, precision, recall, and F1-score matrices. Based on the evaluation results of the toddler height dataset, the K-Nearest Neighbors (KNN) algorithm proved to be the model with the best performance, with an accuracy of 99.91%. This value exceeded that of the Decision Tree, which achieved an accuracy of 99.89%, and the SVM (RBF) algorithm, which achieved 98.48%.

**Keywords:** Machine Learning; Classification; Support Vector Machine; K-Nearest Neighbors; Decision Tree

## 1. PENDAHULUAN

Status gizi merupakan indikator yang menunjukkan sejauh mana kebutuhan pangan tubuh terpenuhi untuk mendukung kesehatan[1]. Gizi yang tidak seimbang dapat menyebabkan berbagai permasalahan, mulai dari stunting, gizi kurang, hingga obesitas, yang semuanya menimbulkan pengaruh terhadap aspek perkembangan tubuh secara fisik maupun kemampuan kognitif dan tingkat efektivitas serta kinerja individu pada waktu yang akan datang. Dengan demikian, diperlukan pemantauan status gizi anak secara berkala dan akurat.

Penentuan status gizi sering kali dilakukan secara manual melalui perbandingan data berat dan tinggi badan[2]. Seiring berjalannya waktu metode konvensional tersebut memiliki keterbatasan dalam hal kecepatan, akurasi, serta ketergantungan terhadap kemampuan tenaga medis. Dengan semakin berkembangnya teknologi kecerdasan buatan (*Artificial Intelligence*). Pendekatan berbasis *machine learning* menjadi alternatif yang potensial untuk membantu proses klasifikasi status gizi secara efisien[3]. Beberapa algoritma yang digunakan seperti *Decision Tree* (DT), *K-Nearest Neighbors* (KNN), dan *Support Vector Machine* (SVM)[4]. Ketiga algoritma ini memiliki karakteristik dan performa yang berbeda dalam hal akurasi, sehingga penting untuk dilakukan analisis komparatif guna menentukan algoritma terbaik dalam mengklasifikasikan status gizi berdasarkan dataset berat balita dan tinggi balita.

Beberapa penelitian sebelumnya telah mengimplementasikan pendekatan *machine learning* terkait klasifikasi status gizi. Pada penelitian oleh Akmad[5] yang menggunakan dua dataset data antropometri berupa berat dan tinggi badan balita dengan menerapkan algoritma SVM. Hasil dari penelitian tersebut menunjukkan jika metode SVM dapat menghasilkan kinerja klasifikasi yang optimal, pada dataset tinggi balita akurasi mencapai 99.50% dan 94.92% pada dataset berat balita. Hasil penelitian yang dilakukan oleh Andi Widya Mufila Gaffar, Andi Muhammad Halis, Purnawansyah, dan Sitti Rahmah Jabir[6] menunjukkan bahwa algoritma *Support Vector Machine* (SVM) dengan kernel *polynomial* menunjukkan performa yang sangat optimal dalam membedakan dan memetakan status stunting terhadap anak usia balita di Kabupaten Enrekang. Uji coba yang dilakukan dengan metode 10-Fold Cross Validation

menunjukkan tingkat akurasi tertinggi sebesar 99,13% serta tingkat akurasi rata-rata diperoleh sebesar 96,98%, dengan nilai *precision* 96,99%, *recall* 96,98%, dan *f1-score* 96,94%. Nilai akurasi yang tinggi pada dua penelitian tersebut menunjukkan bahwa algoritma SVM memiliki kapasitas generalisasi yang baik dalam mengenali pola data gizi anak. Namun, penelitian ini masih memiliki kekurangan karena hanya menitikberatkan pada satu algoritma, yaitu SVM, tanpa dilakukan perbandingan dengan algoritma lain untuk menilai stabilitas dan efektifitas model terhadap variasi data.

Menurut penelitian yang dilakukan oleh Alfath, Cipta, dan Abdullah[7] Salah satu metode yang sering diaplikasikan serta menunjukkan tingkat penerapan yang relatif sederhana adalah *K-Nearest Neighbor* (KNN), yaitu algoritma yang memperlihatkan performa unggul dengan capaian nilai akurasi dan *precision* serta *recall* masing-masing sebesar 94.64%, yang mengindikasikan kemampuan algoritma ini dalam mengklasifikasikan data secara tepat dan konsisten[8]. Keterbatasan penelitian ini adalah penggunaan nilai *k* yang tetap tanpa eksplorasi atau validasi, sehingga model rentan terhadap *overfitting* (jika *k* terlalu kecil) atau *underfitting* (jika *k* terlalu besar). Sementara itu, penelitian yang dilakukan oleh Gina Purnama Insany, Indra Yustiana, dan Sri Rahmawati[9] juga menggunakan algoritma *KNN* dalam klasifikasi status gizi balita berdasarkan variabel umur, jenis kelamin, berat badan, dan tinggi badan. Hasilnya menunjukkan bahwa *KNN* dengan nilai *k* tertentu (3, 5, dan 7) mampu menghasilkan akurasi tinggi sebesar 99% dengan memiliki tingkat nilai error rendah (0,007). Namun, kekurangan dari penelitian ini adalah belum dilakukan analisis mendalam terhadap pengaruh variasi nilai *k* dan jumlah data pelatihan, sehingga performa model masih berpotensi menurun ketika diterapkan pada dataset yang lebih kompleks atau berukuran besar.

Berbeda dengan dua algoritma sebelumnya, implementasi algoritma *machine learning* pada proses pengelompokan kategori status gizi pada anak dengan pendekatan *Decision Tree* memperlihatkan kinerja yang tergolong baik, dengan rata-rata akurasi mencapai 81.25%, yang menandakan kemampuan model dalam mengelompokkan data gizi secara tepat dan konsisten[10]. *Decision Tree* termasuk salah satu pendekatan metode *machine learning* yang umum diterapkan atas dasar kemampuannya dalam merepresentasikan serta memahami hubungan yang kompleks antar variabel yang memengaruhi proses pengambilan keputusan[11]. Namun, *Decision Tree* peka terhadap perubahan kecil pada data, sehingga struktur pohonnya bisa berubah cukup besar meskipun datanya sedikit berbeda[12]. Dalam kasus klasifikasi status gizi anak yang memiliki banyak variasi, hal ini bisa membuat hasil prediksi menjadi kurang stabil.

Oleh karena itu, penelitian ini difokuskan pada analisis komparatif terhadap tiga algoritma klasifikasi, yaitu *Decision Tree* (DT), *K-Nearest Neighbors* (KNN), dan *Support Vector Machine* (SVM). Pemilihan ketiga algoritma tersebut didasarkan bahwa metode-metode ini merupakan pendekatan klasifikasi yang banyak digunakan dalam penelitian berbasis data antropometri. Selain itu, ketiganya memiliki karakteristik pemodelan yang berbeda DT menawarkan interpretabilitas melalui struktur pohon keputusan, KNN mengandalkan kedekatan antar data sebagai dasar pengelompokan, sedangkan SVM memanfaatkan konsep pemisahan ruang fitur melalui *hyperplane*. Perbedaan fundamental pada pendekatan dan mekanisme kerja inilah yang menjadikan ketiga algoritma tersebut layak untuk dibandingkan secara sistematis, sehingga dapat diidentifikasi metode yang memberikan akurasi tertinggi dan performa paling optimal dalam klasifikasi multikategori status gizi balita berdasarkan data berat badan (WFA) dan tinggi badan (HFA).

## 2. METODOLOGI PENELITIAN

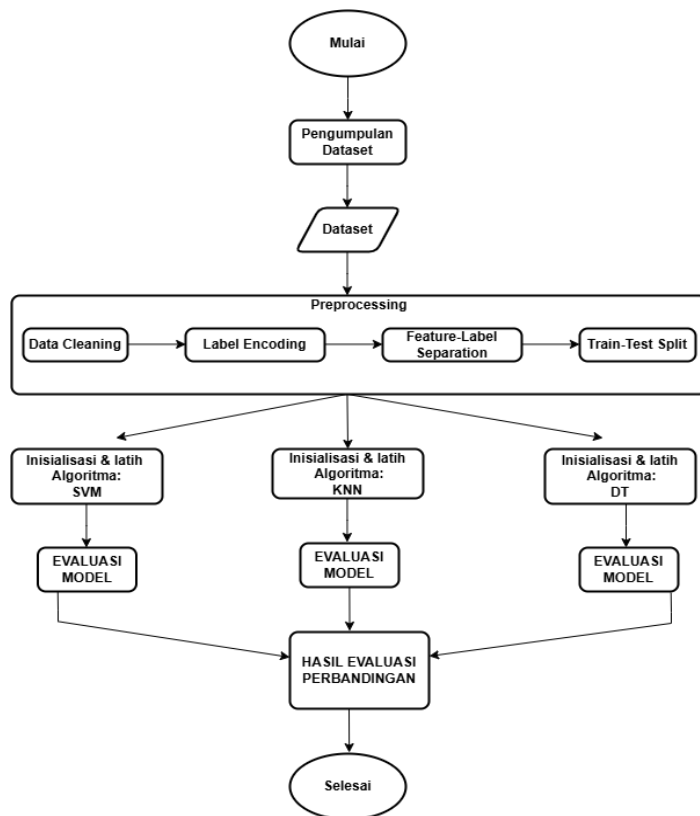
### 2.1 Tahapan Penelitian

Tahapan penelitian ini disusun untuk memberikan penjelasan yang menyeluruh mengenai proses analisis yang dilakukan, dimulai dari kegiatan awal berupa pengumpulan data hingga tahap akhir berupa evaluasi performa model klasifikasi. Penyusunan tahapan dilakukan secara bertahap dan terstruktur agar setiap proses yang dijalankan memiliki kejelasan tujuan serta kesinambungan dengan langkah sebelumnya. Penjelasan mengenai tahapan ini penting untuk memastikan bahwa keseluruhan proses penelitian dapat dipahami dengan baik, terutama terkait landasan metodologis yang digunakan dan alasan pemilihan pendekatan pada setiap tahap.

Penelitian ini mengikuti alur kerja yang diawali dengan proses identifikasi dan pengumpulan dataset dari sumber yang relevan, dilanjutkan dengan pemeriksaan awal untuk memastikan kelayakan data sebelum masuk pada tahap preprocessing. Pada tahap ini, berbagai langkah seperti pembersihan data, penanganan nilai hilang, normalisasi, dan pemisahan fitur serta label dilakukan guna meningkatkan kualitas data yang akan digunakan dalam pemodelan. Pengolahan awal tersebut menjadi tahap penting karena kualitas data sangat menentukan kemampuan model dalam menghasilkan prediksi yang akurat. Penjelasan mengenai tahapan ini disampaikan untuk menegaskan bahwa proses analisis tidak dapat dilepaskan dari keandalan data yang digunakan.

Untuk memberikan pemahaman yang lebih jelas mengenai hubungan antarproses, Gambar 1 menyajikan diagram alur yang merangkum keseluruhan tahapan penelitian.

Melalui penyusunan tahapan yang sistematis dan penjelasan yang rinci, seluruh proses penelitian dapat memberikan gambaran yang kuat mengenai pendekatan analitis yang digunakan. Selain itu, tahapan tersebut menjadi dasar untuk menilai keandalan temuan penelitian, khususnya dalam menentukan model klasifikasi yang paling sesuai untuk mengidentifikasi status gizi balita berdasarkan data antropometri.



**Gambar 1.** Flowchart Penelitian

Gambar 1 menggambarkan urutan proses penelitian secara menyeluruh. Tahap pertama dimulai dengan pengumpulan dataset yang kemudian masuk ke proses preprocessing, mencakup data cleaning, label encoding, pemisahan fitur dan label, serta pembagian data menjadi training dan testing. Selanjutnya, masing-masing algoritma—SVM, KNN, dan Decision Tree—diinisialisasi dan dilatih menggunakan data latihan yang telah diproses. Setiap model kemudian dievaluasi menggunakan metrik performa tertentu untuk memperoleh hasil akurasi dan kemampuan klasifikasinya. Seluruh hasil evaluasi tersebut dikompilasi pada tahap akhir untuk dilakukan analisis perbandingan, sehingga dapat ditentukan algoritma yang memberikan performa paling optimal terhadap klasifikasi status gizi balita.

## 2.2 Pengumpulan Dataset

Tahap pengumpulan dataset merupakan proses awal dalam penelitian yang bertujuan untuk memperoleh data sebagai dasar dalam proses pelatihan dan pengujian model. Pada tahap ini, dataset diperoleh dari penelitian sebelumnya oleh Akhmad[5] yang relevan dengan topik penelitian, sehingga dataset yang digunakan telah melewati proses pengumpulan, verifikasi, dan validasi dalam penelitian sebelumnya.

Dataset tersebut terdiri atas 122.123 data mengenai berat badan balita serta 121.000 data tinggi badan balita, yang memuat informasi mengenai status gizi anak berdasarkan tinggi badan (cm), berat badan (kg), umur (bulan), dan jenis kelamin. Seluruh data disajikan dalam bentuk file Comma-Separated Values (CSV) guna memudahkan proses pengolahan dan analisis pada tahap *preprocessing* data.

## 2.3 Pra-pemrosesan Data (Data Processing)

Tahap pra-proses data adalah tahapan yang berfungsi untuk mempersiapkan serta menyempurnakan data mentah sebelum dianalisis atau digunakan untuk pelatihan model. Tujuan utama dari preprocessing yaitu melakukan pembersihan, normalisasi, serta menyiapkan data supaya dapat diolah secara optimal oleh algoritma analisis maupun model machine learning[13]. Proses ini memiliki peran yang krusial dalam proses klasifikasi, sebab dapat meningkatkan tingkat akurasi model dengan mengurangi ketidakteraturan dan kompleksitas dalam data[14]. Beberapa bagian dari *Preprocessing* dapat dilihat pada step berikut:

### 2.3.1 Data Cleaning

Merupakan proses penting dalam tahapan pengolahan data yang ditujukan untuk memastikan kualitas dataset agar layak dan siap digunakan untuk dianalisis seperti pada Gambar 2 dan Gambar 4, dan juga terdapat proses Data Transformation untuk menyamakan format atribut jenis kelamin serta status gizi, yang semula tercatat dalam format teks, menjadi format numerik, hasilnya seperti pada Gambar 3 dan Gambar 5. Gambar 2 menyajikan hasil proses *data cleaning* pada dataset berat badan balita, yang meliputi penampilan lima data pertama, informasi struktur dataset,

serta pengecekan nilai hilang (*missing values*) untuk memastikan kualitas data sebelum memasuki tahap preprocessing lebih lanjut.

```

===== 5 Data Pertama =====
   Umur (bulan)  Berat Badan (kg)  Jenis Kelamin  Status Gizi
0              0              1  perempuan  severely  underweight
1              0              6  perempuan  overweight
2              0              1  perempuan  severely  underweight
3              0              6  perempuan  overweight
4              0              4  perempuan  normal

===== Informasi Dataset =====
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 122122 entries, 0 to 122121
Data columns (total 4 columns):
#   Column                Non-Null Count  Dtype
---  ---
0   Umur (bulan)          122122 non-null  int64
1   Berat Badan (kg)     122122 non-null  int64
2   Jenis Kelamin        122122 non-null  object
3   Status Gizi          122122 non-null  object
dtypes: int64(2), object(2)
memory usage: 3.7+ MB
None

===== Cek Missing Values =====
Umur (bulan)      0
Berat Badan (kg)  0
Jenis Kelamin     0
Status Gizi       0
dtype: int64
    
```

**Gambar 2.** Data cleaning dataset berat balita

Berdasarkan hasil pada Gambar 2, seluruh variabel pada dataset teridentifikasi tanpa nilai hilang serta memiliki tipe data yang sesuai, sehingga dataset dinyatakan dalam kondisi bersih dan siap digunakan untuk tahapan pemrosesan berikutnya, seperti *label encoding*, *feature-label separation*, dan *train-test split*.

Pada Gambar 3 tahap data transformation dilakukan untuk mengubah nilai kategorikal menjadi bentuk numerik melalui proses label encoding, sehingga seluruh variabel dalam dataset dapat diproses oleh algoritma machine learning secara optimal.

Umur (bulan)	Berat Badan (kg)	Jenis Kelamin	Status Gizi
0	0	1	2
1	0	6	1
2	0	1	2
3	0	6	1
4	0	4	0
5	0	4	1
6	0	7	1
7	0	8	1
8	0	5	1

**Gambar 3.** Data transformation dataset berat balita

Gambar 3 menunjukkan hasil akhir proses transformasi, di mana setiap kategori pada variabel jenis kelamin dan status gizi telah direpresentasikan dalam bentuk angka. Transformasi ini memastikan konsistensi format data dan mempermudah model dalam mempelajari pola hubungan antarvariabel.

Tahap data cleaning pada dataset tinggi balita pada Gambar 4 dilakukan untuk memastikan bahwa seluruh variabel bebas dari nilai hilang (*missing values*), inkonsistensi, maupun tipe data yang tidak sesuai. Proses ini penting agar dataset berada dalam kondisi optimal sebelum memasuki tahap preprocessing lanjutan.

```

===== 5 Data Pertama =====
   Umur (bulan)  Jenis Kelamin  Tinggi Badan (cm)  Status Gizi
0              0      laki-laki      44.5          stunted
1              0      laki-laki      56.7          tinggi
2              0      laki-laki      46.8          normal
3              0      laki-laki      47.5          normal
4              0      laki-laki      42.7  severely  stunted

===== Informasi Dataset =====
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 120999 entries, 0 to 120998
Data columns (total 4 columns):
#   Column                Non-Null Count  Dtype
---  ---
0   Umur (bulan)          120999 non-null  int64
1   Jenis Kelamin        120999 non-null  object
2   Tinggi Badan (cm)    120999 non-null  float64
3   Status Gizi          120999 non-null  object
dtypes: float64(1), int64(1), object(2)
memory usage: 3.7+ MB
None

===== Cek Missing Values =====
Umur (bulan)      0
Jenis Kelamin     0
Tinggi Badan (cm)  0
Status Gizi       0
dtype: int64
    
```

**Gambar 4.** Data cleaning dataset tinggi balita

Gambar 4 menampilkan hasil proses data cleaning pada dataset tinggi balita, yang meliputi pemeriksaan lima data pertama, struktur dataset, tipe data setiap kolom, serta verifikasi bahwa tidak ditemukan nilai kosong pada seluruh variabel. Gambar 5 menyajikan hasil proses data cleaning pada dataset tinggi badan balita, yang meliputi penampilan lima data pertama, informasi struktur dataset, serta pengecekan nilai hilang (missing values) untuk memastikan kualitas data sebelum memasuki tahap preprocessing lebih lanjut.

Umur (bulan)	Jenis Kelamin	Tinggi Badan (cm)	Status Gizi
0	0	0	44.5
1	0	0	56.7
2	0	0	46.8
3	0	0	47.5
4	0	0	42.7
5	0	0	44.2
6	0	0	59.5
7	0	0	42.7
8	0	0	45.2
9	0	0	57.0

Gambar 5. Data transformation dataset tinggi balita

Berdasarkan hasil pada Gambar 5, seluruh variabel pada dataset teridentifikasi tanpa nilai hilang serta memiliki tipe data yang sesuai, sehingga dataset dinyatakan dalam kondisi bersih dan siap digunakan untuk tahapan pemrosesan berikutnya, seperti *label encoding*, *feature-label separation*, dan *train-test split*.

### 2.3.2 Label Encoding

Algoritma machine learning hanya dapat memproses data dalam bentuk numerik, sehingga diperlukan penerapan teknik label encoding yang digunakan untuk mengonversi variabel kategorikal menjadi nilai numerik agar dapat diolah secara efektif. Tahapan ini memiliki tujuan untuk memberikan representasi numerik pada setiap kategori tanpa mengubah makna atau karakteristik data aslinya. Pada penelitian ini, proses label encoding dilakukan terhadap atribut jenis kelamin dan status gizi pada dua dataset, yaitu dataset tinggi balita dan berat balita. Hasil konversi label encoding tersaji dalam Tabel 1, Tabel 2, dan Tabel 3:

Tabel 1. Data Numerik Jenis Kelamin

No	Jenis Kelamin	Data Numerik
1	Perempuan	1
2	Laki – laki	0

Tabel 2. Data Numerik Status Gizi pada Dataset Berat Balita

No	Status Gizi	Data Numerik
1	Overweight	1
2	Normal	0
3	Underweight	3
4	Severely Underweight	2

Tabel 3. Data Numerik Status Gizi pada Dataset Tinggi Balita

No	Status Gizi	Data Numerik
1	Tinggi	3
2	Normal	0
3	Stunted	2
4	Severely Stunted	1

### 2.3.3 Feature-Label Separation

Merupakan proses pemisahan antara variabel fitur (feature) dan label (target) dalam suatu dataset sebelum dilakukan pelatihan model machine learning. Pada penelitian ini, fitur merupakan sekumpulan variabel independen yang berfungsi sebagai input bagi model untuk mempelajari pola atau hubungan antar data, sedangkan label merupakan variabel terikat yang diprediksi oleh model.

Proses *Feature-Label Separation* pada dataset berat balita dilakukan sebagai langkah pemisahan antara variabel fitur dan variabel target sebelum model dilatih. Pada tahap ini, variabel **X** terdiri atas kolom “*Umur (bulan)*”, “*Jenis Kelamin*”, dan “*Berat Badan (kg)*” yang berfungsi sebagai fitur atau variabel independen yang digunakan untuk membentuk pola pembelajaran model. Sementara itu, variabel **y** memuat kolom “*Status Gizi*” yang berperan sebagai label atau variabel terikat yang akan menjadi target prediksi.

Proses *Feature-Label Separation* pada dataset tinggi balita dilakukan dengan prinsip yang sama, yaitu memisahkan variabel fitur dari variabel target sebelum proses pelatihan model. Pada tahap ini, variabel **X** mencakup kolom “*Umur (bulan)*”, “*Jenis Kelamin*”, dan “*Tinggi Badan (cm)*” yang berfungsi sebagai variabel independen pembentuk pola pembelajaran model. Adapun variabel **y** berisi kolom “*Status Gizi*” sebagai label atau variabel dependen yang akan diprediksi oleh model.

Pemisahan fitur dan label pada kedua dataset tersebut bertujuan agar model machine learning hanya mempelajari hubungan antar variabel input tanpa terpengaruh secara langsung oleh nilai output. Dengan demikian, proses pelatihan dapat menghasilkan model yang objektif, valid, serta mampu melakukan generalisasi dengan baik terhadap data baru.

### 2.3.4 Train-Test Split

Train-Test Split, adalah tahap pemisahan dataset menjadi dua subset, yaitu data latih (training data) dan data uji (testing data). Cara kerja *Train-Test Split* dilakukan dengan memanfaatkan fungsi `train_test_split()` dari library *scikit-learn*, yang digunakan untuk membagi dataset menjadi dua bagian, yaitu data fitur (**x**) dan label (**y**) dengan proporsi yang telah ditentukan. Parameter `test_size=0.2` menandakan bahwa 20% dari keseluruhan data digunakan sebagai data uji (*testing data*), sedangkan 80% sisanya digunakan sebagai data latih (*training data*). Parameter `random_state=42` berfungsi untuk menjaga konsistensi hasil pembagian data setiap kali kode dijalankan, sehingga proses pelatihan model dapat direplikasi dengan hasil yang sama. Selain itu, penggunaan `stratify=y` diterapkan untuk memastikan bahwa proporsi kelas pada data latih dan data uji tetap seimbang sesuai dengan distribusi label aslinya, sehingga hasil evaluasi model menjadi lebih representatif dan tidak menimbulkan bias terhadap kelas tertentu. Tahapan ini memegang peranan penting dalam memastikan objektivitas evaluasi model serta kemampuan generalisasi model *machine learning*. Melalui pembagian data yang terencana, model dapat menghasilkan prediksi yang konsisten, akurat.

## 2.4 Pelatihan Model (Model Training)

Setelah semua data terkumpul, dilakukan tahap preprocessing data yang mencakup serangkaian tahapan krusial guna menjamin bahwa data telah siap dimanfaatkan dalam proses pelatihan model. Langkah pertama yaitu data cleaning, yang berfungsi untuk menghapus atau memperbaiki data yang mengandung nilai hilang (*missing values*), data duplikat, atau kolom yang tidak memiliki keterkaitan langsung dengan tujuan penelitian. Proses ini dilakukan untuk memastikan bahwa data yang digunakan memiliki kualitas tinggi dan tidak menimbulkan bias terhadap hasil klasifikasi.

Tahap berikutnya adalah label encoding, yaitu proses mengubah data kategorikal seperti jenis kelamin atau kategori status gizi menjadi bentuk numerik. Langkah ini diperlukan sebab mayoritas metode *machine learning* hanya dapat mengolah data numerik. Setelah itu dilakukan *feature-label separation*, yakni pemisahan antara fitur (variabel input seperti tinggi badan, berat badan, dan umur) dengan label (label keluaran berupa klasifikasi multikategori status gizi, yaitu *Severely Underweight*, *Underweight*, *Normal*, dan *Overweight*, yang ditentukan berdasarkan nilai BB (Berat Badan). Sementara itu, pada dataset tinggi balita, label keluaran mencakup kategori *Severely Stunted*, *Stunted*, *Normal*, dan *Tinggi*, yang ditentukan menggunakan indikator TB (Tinggi Badan)). Pemisahan ini penting untuk membedakan antara data yang dipakai dalam proses prediksi dan data yang berperan sebagai target prediksi.

Setelah tahap pemisahan fitur dan label, dilakukan *feature scaling* melalui metode normalisasi untuk menyeragamkan rentang nilai pada setiap fitur. Langkah ini sangat penting, terutama bagi algoritma berbasis jarak seperti *K-Nearest Neighbors* (KNN) dan algoritma berbasis margin seperti *Support Vector Machine* (SVM), agar performa model tetap stabil serta tidak bias terhadap fitur yang memiliki skala nilai lebih besar.

Tahap berikutnya adalah *train-test split*, yakni pembagian dataset menjadi dua bagian utama: data latih (*training data*) dan data uji (*testing data*). Pada penelitian ini, digunakan proporsi pembagian 80% data untuk pelatihan dan 20% data untuk pengujian (rasio 8:2). Pembagian ini bertujuan agar model dapat mempelajari pola dari mayoritas data, sementara akurasi dan kemampuan generalisasi dievaluasi menggunakan data yang tidak pernah dilihat sebelumnya, sehingga hasil prediksi yang diperoleh lebih objektif dan representatif.

## 2.5 Evaluasi Model (Model Evaluation)

Tahap selanjutnya adalah inisialisasi dan pelatihan model dengan menerapkan tiga metode klasifikasi yang berbeda, yaitu *Support Vector Machine* (SVM), *K-Nearest Neighbors* (KNN), dan *Decision Tree* (DT). Ketiga algoritma tersebut dipilih untuk dibandingkan sebab masing-masing memiliki karakteristik serta pendekatan yang bervariasi dalam melakukan klasifikasi. SVM beroperasi dengan menentukan *hyperplane* yang paling optimal untuk memisahkan antar kelas, KNN mengklasifikasikan data berdasarkan kedekatan jarak antar titik, sedangkan *Decision Tree* menggunakan struktur pohon keputusan untuk mengidentifikasi kelas suatu data melalui serangkaian aturan logis.

### 2.5.1 Support Vector Machine (SVM)

SVM memanfaatkan konsep kernel dalam menentukan hyperplane atau bidang pemisah yang paling optimal dalam memisahkan berbagai kelas yang terdapat pada suatu dataset[15]. Selanjutnya, untuk algoritma *Support Vector*

*Machine* (SVM), digunakan empat jenis kernel function di mana memiliki pendekatan berbeda dalam memisahkan data pada ruang dengan dimensi yang lebih tinggi. Pada penelitian ini, proses pelatihan model dilakukan menggunakan parameter default, yaitu nilai  $C = 1$ , nilai  $\gamma$  (gamma) default sesuai implementasi pada library, serta pendekatan One-vs-One (OVO) untuk menangani klasifikasi multikategori. Masing-masing kernel dirumuskan sebagai berikut:

### 2.5.1.1 Kernel Linear

$$K(x_i, x_j) = x_i \times x_j \quad (1)$$

Pada kernel ini, hasil perkalian titik (dot product) antara vektor data  $x_i$  dan  $x_j$  digunakan untuk menentukan tingkat kesamaan antar data. Kernel linear bekerja dengan baik pada data yang bersifat linear separable, yaitu data yang dapat dipisahkan menggunakan satu garis atau bidang lurus tanpa perlu transformasi ke ruang dimensi yang lebih tinggi.

### 2.5.1.2 Kernel Polynomial

$$K(x_i, x_j) = (\gamma \times x_j + r)^d \quad (2)$$

Rumus ini memungkinkan model untuk memetakan data ke dalam ruang dengan dimensi lebih tinggi sehingga hubungan non-linear antar fitur dapat dipelajari. Parameter  $\gamma$  berfungsi untuk mengatur skala data,  $r$  adalah konstanta bias yang menyesuaikan jarak antar data, dan  $d$  menunjukkan derajat polinomial yang menentukan kompleksitas kurva pemisah yang dihasilkan.

### 2.5.1.3 Kernel Radial Basis Function (RBF)

$$K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2) \quad (3)$$

Kernel ini bekerja dengan menghitung jarak Euclidean antara dua titik data, lalu mengubahnya menjadi nilai kesamaan melalui fungsi eksponensial. Semakin kecil jarak antar data, semakin besar nilai kernel. RBF banyak digunakan karena kemampuannya yang tinggi dalam menangani data non-linear dan mendeteksi pola kompleks pada dataset.

### 2.5.1.4 Sigmoid

$$K(x_i, x_j) = \tanh(\gamma x_i \times x_j + r) \quad (4)$$

Kernel ini terinspirasi dari fungsi aktivasi pada jaringan saraf tiruan (*neural network*), di mana fungsi hiperbolik tangen digunakan untuk menormalkan nilai kesamaan antar data. Parameter  $\gamma$  mengatur sensitivitas fungsi terhadap perubahan data, sedangkan  $r$  berfungsi sebagai bias. Kernel sigmoid cocok digunakan pada data yang memiliki hubungan non-linear moderat antar fitur.

## 2.5.2 K-Nearest Neighbors (K-NN)

Algoritma KNN termasuk ke dalam metode klasifikasi sederhana yang beroperasi dengan membandingkan jarak antara data uji dan data latih, kemudian menentukan kelas dari data uji berdasarkan kedekatannya dengan sejumlah  $k$  tetangga terdekat [16]. Pada penelitian ini, algoritma KNN diterapkan dalam proses pengelompokan tingkat status gizi balita dengan memanfaatkan data antropometri yang mencakup variabel umur, jenis kelamin, berat balita, dan tinggi balita. Prinsip kerja KNN didasarkan pada asumsi bahwa yang memiliki kesamaan karakteristik akan berada dalam kelompok yang sama, sehingga klasifikasi dilakukan berdasarkan mempertimbangkan tingkat kemiripan atau jarak paling dekat antara data uji dan sejumlah data latih terdekat ( $k$  tetangga terdekat) [17]. Dalam mengukur kedekatan atau perbedaan antar data, digunakan persamaan jarak Euclidean sebagai berikut:

$$d(x, y) = \sqrt{\sum_{i=0}^n (x_i - y_i)^2} \quad (5)$$

di mana  $x_i$  dan  $x_j$  adalah dua vektor data dengan masing-masing memiliki  $m$  fitur, seperti tinggi ataupun berat badan, umur, dan jenis kelamin. Semakin kecil nilai jarak, maka tingkat kesamaan antar data dianggap lebih tinggi antar data. Setelah proses klasifikasi selesai, tingkat kesalahan model dapat diukur menggunakan rumus *Error Rate*, yaitu:

$$\text{Error Rate (K)} = \frac{\text{Jumlah Prediksi Salah}}{\text{Total Data Uji}} \quad (6)$$

Rumus ini digunakan untuk menghitung persentase kesalahan prediksi model terhadap seluruh data uji. Nilai *error rate* yang lebih rendah menandakan bahwa performa model KNN semakin optimal dalam mengklasifikasikan data secara akurat. Dengan demikian, KNN menjadi metode yang sederhana namun cukup efektif dalam mengidentifikasi pola pada data status gizi balita.

## 2.5.3 Decision Tree

*Decision Tree* atau dapat disebut dengan Pohon Keputusan termasuk pendekatan yang banyak digunakan dalam proses klasifikasi maupun prediksi banyak digunakan karena mampu mengubah data yang kompleks menjadi struktur pohon

yang mudah dipahami, serta membantu mengidentifikasi pola dan hubungan antara variabel input dan variabel target guna mendukung proses analisis dan pengambilan keputusan secara lebih akurat[18]. Algoritma *Decision Tree* berfungsi untuk mengelompokkan data baru yang belum diketahui kategorinya ke dalam kelas yang telah ditentukan sebelumnya[19]. Dalam proses pembentukan pohon keputusan, penelitian ini menggunakan kriteria pemisah berupa *Entropy* untuk mengukur tingkat ketidakpastian atau ketidakteraturan data pada setiap node[20]. *Entropy* menggambarkan seberapa murni suatu himpunan data terhadap kelas target, jika seluruh data dalam satu node termasuk ke dalam satu kelas yang sama, maka nilai *Entropy* akan bernilai nol, yang berarti node tersebut sepenuhnya murni. Sebaliknya, semakin beragam distribusi kelas di dalam node, semakin tinggi pula nilai *Entropy* yang menunjukkan ketidakpastian yang besar[21]. Nilai *Entropy* ini kemudian digunakan untuk menentukan *Information Gain*, yang menunjukkan tingkat pengurangan ketidakpastian setelah data dipisahkan berdasarkan suatu atribut. Atribut dengan nilai *Information Gain* tertinggi akan dipilih sebagai pemisah terbaik, karena memberikan pembagian data yang paling informatif dan menghasilkan node-node dengan kemurnian yang lebih tinggi. Berikut rumus-rumus utama yang digunakan dalam proses perhitungan *Entropy* serta *Information Gain* pada algoritma *Decision Tree*:

$$Entropy(S) = - \sum_{i=1}^n p_i \log_2(p_i) \quad (7)$$

di mana  $p_i$  adalah probabilitas data masuk ke dalam kelas ke- $i$ , dan  $n$  merupakan jumlah kelas yang ada. Selanjutnya, untuk menentukan fitur terbaik yang digunakan dalam pemisahan data, digunakan ukuran *Information Gain* yang dihitung dengan rumus:

$$Entropy_{split}(A) = \sum_{v \in Values(A)} \frac{|S_v|}{|S|} \times Entropy(S_v) \quad (8)$$

$$Gain(S, A) = Entropy(S) - Entropy_{split}(A) \quad (9)$$

Pada rumus tersebut,  $S$  merupakan himpunan data awal,  $S_v$  adalah subset data dari  $S$  dengan nilai atribut tertentu, dan  $|S_v|/|S|$  adalah proporsi data dalam subset tersebut. Selain itu, pada penelitian ini tidak diterapkan teknik pruning, sehingga seluruh node yang dihasilkan mengikuti pemisahan berdasarkan nilai *Information Gain* tertinggi pada setiap tahap pembentukan pohon.

## 2.6 Analisis Perbandingan (Model Comparison)

Setelah ketiga model selesai dilatih, dilakukan tahap evaluasi model untuk menilai performansi setiap metode yang digunakan. Proses evaluasi ini menggunakan matrik konfusi (*confusion matrix*) sebagai alat ukur, yang menghasilkan beberapa metrik evaluasi seperti *accuracy*, *precision*, *recall*, dan *F1-score*. Keempat metrik tersebut digunakan untuk menilai sejauh mana model dapat mengelompokkan data secara akurat dan dengan maksimal, dan sesuai dengan label sebenarnya.

Tingkat akurasi (*accuracy*) digunakan untuk mengukur persentase total data yang dapat diidentifikasi dengan tepat oleh model. Nilainya dihitung menggunakan persamaan berikut:

$$Accuracy = \frac{\text{Jumlah prediksi benar}}{\text{Total jumlah data}} \quad (10)$$

Selanjutnya, *precision* menunjukkan seberapa tepat model dalam menghasilkan prediksi positif yang diperoleh dari model, yaitu proporsi merujuk pada rasio antara jumlah prediksi positif yang benar terhadap total keseluruhan prediksi positif yang dihasilkan oleh model. Persamaan *precision* dapat dinyatakan sebagai berikut:

$$Precision = \frac{TP}{TP+FP} \quad (11)$$

Sementara itu, *recall* digunakan untuk mengukur sejauh mana model mampu mengenali seluruh data positif yang sebenarnya ada di dalam dataset. Rumus *recall* dituliskan sebagai berikut:

$$Recall = \frac{TP}{TP+FN} \quad (12)$$

Terakhir, *F1-score* merupakan ukuran yang menggabungkan *precision* dan *recall* dalam satu nilai yang seimbang. Metrik ini sangat berguna ketika terdapat ketidakseimbangan jumlah data antar kelas. Rumus *F1-score* dituliskan sebagai berikut:

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (13)$$

## 3. HASIL DAN PEMBAHASAN

### 3.1 Inisialisasi dan Latih Model

Tahap inisialisasi model merupakan langkah penting yang dilakukan setelah proses *preprocessing* selesai. Dengan inisialisasi, pelatihan model dapat berjalan secara optimal, sehingga mampu menghasilkan kinerja prediksi yang memiliki tingkat akurasi tinggi serta stabilitas yang baik.

### 3.1.1 Support Vector Machine (SVM)

Algoritma Support Vector Machine (SVM) termasuk salah satu algoritma yang diterapkan dalam proses klasifikasi dan termasuk ke dalam kategori pembelajaran terawasi (supervised learning), dalam pendekatan ini, model dipelajari menggunakan data yang telah memiliki label guna membangun fungsi pemisah yang paling efektif dalam membedakan tiap kategori data pada suatu dataset[14]. Pada penelitian ini, Algoritma SVM diterapkan dalam mengklasifikasikan status gizi balita yang berdasarkan dataset berupa antropometri, yaitu umur, jenis kelamin, serta berat atau tinggi balita. Proses penelitian dilakukan dengan menginisialisasi model SVM menggunakan empat kernel, yaitu polynomial, linear, radial basis function (RBF), serta sigmoid, guna membandingkan performa masing-masing kernel dalam menghasilkan klasifikasi terbaik.

Tahap berikutnya mencakup proses perhitungan matrik evaluasi kinerja model klasifikasi. Secara umum, perhitungan matrik evaluasi seperti akurasi, presisi, recall, f1-score, macro avg, dan weight avg dapat ditentukan menggunakan persamaan (10), (11), (12), dan (13).

Berdasarkan hasil pengujian model SVM dengan kernel *Linear*, *Polynomial* (Poly), *Radial Basis Function* (RBF), dan *Sigmoid*, diperoleh nilai matrik evaluasi yang berbeda untuk setiap kernel. Dapat dilihat pada Tabel 4:

**Tabel 4.** Hasil Uji Kernel SVM dengan Classification Report pada Status Gizi berdasarkan Berat Balita

No	Kernel	Class	Precision	Recall	F1 Score	Accuracy
1	Linear	Normal	0.88	0.90	0.89	91.57%
		Overweight	0.96	0.95	0.96	
		Severely Underweight	0.93	0.96	0.95	
		Underweight	0.73	0.62	0.67	
2	Poly	Normal	0.91	0.94	0.92	93.79%
		Overweight	0.97	0.96	0.97	
		Severely Underweight	0.96	0.97	0.96	
		Underweight	0.78	0.71	0.74	
3	RBF	Normal	0.93	0.95	0.94	94.79%
		Overweight	0.98	0.97	0.98	
		Severely Underweight	0.96	0.97	0.97	
		Underweight	0.78	0.73	0.75	
4	Sigmoid	Normal	0.48	0.63	0.54	62.96%
		Overweight	0.73	0.72	0.72	
		Severely Underweight	0.77	0.65	0.70	
		Underweight	0.20	0.07	0.11	

Berdasarkan hasil pengujian model pada Tabel 4 dengan berbagai kernel, terlihat bahwa performa klasifikasi menunjukkan variasi yang cukup signifikan antarjenis kernel. Kernel *Linear* menghasilkan akurasi sebesar 91,57%, dengan nilai recall tertinggi pada kelas *severely underweight* sebesar 0,96, diikuti oleh *overweight* sebesar 0,95, dan *normal* sebesar 0,90. Sementara itu, kelas *underweight* memiliki nilai recall terendah yaitu 0,62. Secara keseluruhan, nilai rata-rata recall menunjukkan bahwa kernel *Linear* telah mampu mendeteksi status gizi dengan cukup baik, meskipun masih terdapat perbedaan kemampuan antar kelas.

Selanjutnya, kernel *Polynomial* (Poly) menunjukkan peningkatan performa dengan akurasi sebesar 93,79%. Nilai recall tertinggi dicapai oleh kelas *severely underweight* dan *overweight* sebesar 0,96 dan 0,97, diikuti kelas *normal* sebesar 0,94, sementara *underweight* memperoleh nilai terendah yaitu 0,71. Hasil ini mengindikasikan bahwa kernel *Poly* memiliki kestabilan dan kekuatan yang baik dalam mengklasifikasikan sebagian besar kelas. Secara keseluruhan, model dengan kernel ini mampu menangkap hubungan nonlinier antar fitur dengan cukup efektif, memberikan keseimbangan yang baik antara presisi dan sensitivitas deteksi.

Di sisi lain, pada kernel *Radial Basis Function* (RBF) mendapat hasil terbaik dengan akurasi tertinggi yaitu 94,79% dan nilai rata-rata recall sekitar 0,91. Kernel ini menunjukkan keseimbangan kinerja antar kelas, di mana kelas *severely underweight* dan *overweight* memperoleh recall 0,97, *normal* sebesar 0,95, serta *underweight* 0,73. Kinerja ini memperlihatkan kemampuan RBF dalam mengenali pola data yang kompleks serta mempertahankan akurasi tinggi pada seluruh kategori status gizi. Sebaliknya, kernel *Sigmoid* mencatat akurasi paling rendah sebesar 62,96%, menunjukkan bahwa kernel tersebut kurang efektif untuk pola nonlinier kompleks. Secara keseluruhan, kernel RBF dapat dikategorikan sebagai pilihan paling optimal karena dapat memberikan keseimbangan terbaik antara akurasi, presisi, dan recall di seluruh kelas.

**Tabel 5.** Hasil Uji Kernel SVM dengan Classification Report pada Status Gizi berdasarkan Tinggi Balita

No	Kernel	Class	Preci-sion	Recall	F1 Score	Accuracy
1	Linear	Normal	0.80	0.89	0.84	78.71%
		Severely Stunted	0.78	0.86	0.82	
		Stunted	0.63	0.32	0.42	
		Tinggi	0.79	0.69	0.74	

No	Kernel	Class	Preci-sion	Recall	F1 Score	Accuracy
2	Poly	Normal	0.96	0.98	0.97	95.86%
		Severely Stunted	0.98	0.98	0.98	
		Stunted	0.94	0.91	0.93	
		Tinggi	0.94	0.91	0.92	
3	RBF	Normal	0.99	0.99	0.99	98.48%
		Severely Stunted	0.97	0.99	0.98	
		Stunted	0.98	0.94	0.96	
		Tinggi	0.99	0.98	0.98	
4	Sigmoid	Normal	0.61	0.81	0.70	55.05%
		Severely Stunted	0.34	0.47	0.39	
		Stunted	0.00	0.00	0.00	
		Tinggi	0.74	0.13	0.23	

Berdasarkan hasil pengujian pada Tabel 5, kernel *Linear*, model memperoleh akurasi sebesar 78,71% dengan performa yang bervariasi pada setiap kelas. Nilai recall tertinggi terdapat pada kelas *Severely Stunted* sebesar 0.86, diikuti oleh kelas *Normal* sebesar 0.89, sedangkan kelas *Stunted* menunjukkan nilai recall terendah yaitu 0.32. Hasil ini menandakan bahwa kernel *Linear* masih kurang optimal dalam mendeteksi kategori dengan jumlah data tidak seimbang atau pola fitur yang kompleks. Meskipun demikian, kernel ini tetap mampu memberikan performa moderat pada kelas utama seperti *Normal* dan *Severely Stunted*.

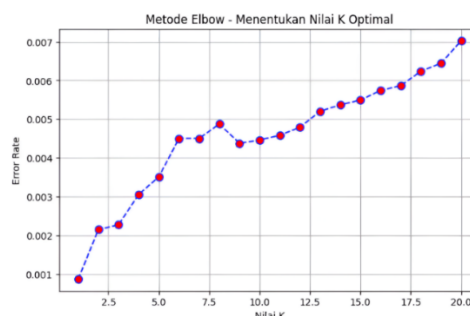
Kernel *Polynomial* (Poly) menunjukkan peningkatan kinerja signifikan dengan akurasi mencapai 95,86%. Nilai recall tertinggi diperoleh pada kelas *Normal* dan *Severely Stunted* sebesar 0.98, diikuti oleh kelas *Stunted* dan *Tinggi* yang masing-masing memiliki recall di angka 0.91. Secara keseluruhan, hasil ini mengindikasikan bahwa kernel Poly mampu mengklasifikasikan data dengan baik di seluruh kelas. Kinerja yang stabil ini mengindikasikan bahwa model dengan kernel *Polynomial* dapat mempelajari hubungan nonlinier antara fitur tinggi badan dan status gizi secara lebih efektif dibandingkan kernel *Linear*.

Sementara itu, kernel *Radial Basis Function* (RBF) menghasilkan kinerja paling unggul dengan tingkat akurasi tertinggi sebesar 98,48%. Kernel ini juga memperlihatkan nilai *precision*, *recall*, dan *F1-score* yang tinggi serta seimbang antar kelas, dengan nilai *recall* mencapai 0.99 pada kelas *Normal* dan *Severely Stunted*, serta 0.94 pada kelas *Stunted*. Capaian tersebut menegaskan bahwa kernel RBF sangat efektif dalam menangani pola data nonlinier dan kompleks pada klasifikasi status gizi berbasis tinggi badan. Sebaliknya, kernel *Sigmoid* hanya menghasilkan akurasi 55,05%, dengan performa yang sangat rendah pada kelas *Stunted* dan *Tinggi*, yang menunjukkan bahwa kernel ini kurang sesuai untuk mendeteksi variasi data gizi dengan karakteristik nonlinier.

Berdasarkan hasil evaluasi terhadap keempat jenis kernel, kernel RBF menunjukkan kinerja paling unggul dibandingkan kernel lainnya. Hasil confusion matrix memperlihatkan bahwa nilai *precision*, *recall*, dan *F1-score* pada kernel RBF relatif seimbang antar kelas, menandakan kemampuannya dalam mengenali pola fitur nonlinier secara konsisten. Dengan demikian, kernel RBF dapat dikategorikan sebagai pilihan paling efektif untuk klasifikasi status gizi, karena mampu memberikan keseimbangan optimal antara akurasi, presisi, dan keandalan deteksi pada setiap kategori data.

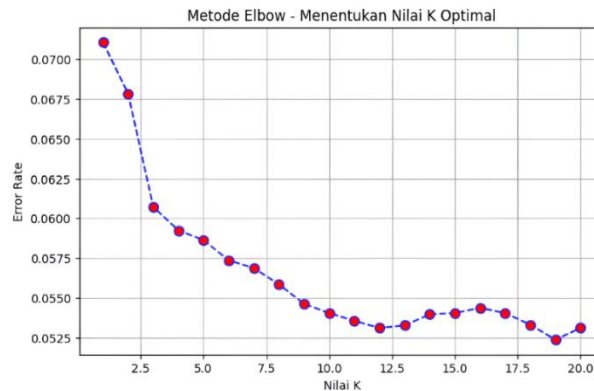
### 3.1.2 K-Nearest Neighbors (K-NN)

Algoritma KNN termasuk ke dalam metode klasifikasi sederhana yang beroperasi dengan membandingkan jarak antara data uji dan data latih, kemudian menentukan kelas dari data uji berdasarkan kedekatannya dengan sejumlah *k* tetangga terdekat [16]. Dalam penelitian ini, algoritma KNN diterapkan dalam proses pengelompokan status gizi balita berdasarkan data antropometri yang mencakup variabel umur, jenis kelamin, berat balita, dan tinggi balita. Prinsip kerja KNN didasarkan pada asumsi bahwa data memiliki kesamaan fitur atau sifat tertentu akan berada dalam kelompok yang sama, sehingga klasifikasi dilakukan berdasarkan mempertimbangkan tingkat kemiripan atau jarak paling dekat antara data uji dan sejumlah data latih terdekat (*k* tetangga terdekat) [17], yang dirumuskan pada (5). Setelah jarak dihitung, langkah berikutnya adalah menetapkan jumlah tetangga terdekat (*K*) yang akan dijadikan acuan. Nilai *K* optimal ditentukan menggunakan metode Elbow dengan rumus (6).



Gambar 11. Menentukan nilai K optimal berdasarkan dataset berat balita

Pada Gambar 11 dilakukan pengujian dengan menguji nilai  $K$  dari 1 hingga 20 untuk mengevaluasi pengaruh variasi jumlah tetangga terhadap performa klasifikasi. Berdasarkan hasil pengujian tersebut, terlihat bahwa peningkatan nilai  $K$  hingga titik tertentu berkontribusi pada penurunan error rate dan peningkatan akurasi model. Namun, setelah mencapai rentang  $K = 10$  hingga  $K = 20$ , perubahan nilai  $K$  tidak memberikan peningkatan akurasi yang signifikan. Dengan demikian, rentang nilai  $K$  antara 5 hingga 10 dapat dikategorikan sebagai parameter yang paling optimal karena mampu menghasilkan keseimbangan antara tingkat akurasi, kompleksitas model, dan kestabilan performa klasifikasi.



**Gambar 12.** Menentukan nilai  $K$  optimal berdasarkan dataset tinggi balita

Sementara itu, pada Gambar 12, terlihat bahwa *error rate* cenderung meningkat seiring dengan bertambahnya nilai  $K$ . Proses pengujian dilakukan dengan menguji nilai  $K$  dari 1 hingga 20 untuk menganalisis pengaruh variasi jumlah tetangga terhadap performa klasifikasi pada dataset tinggi badan. Pola yang ditunjukkan grafik mengindikasikan ketika bahwa nilai  $K$  cukup besar, model menjadi kurang sensitif terhadap variasi data, dan menghasilkan klasifikasi yang bersifat terlalu umum (*underfitting*), sehingga kehilangan kemampuan dalam mengenali variasi data secara detail. Sebaliknya, pada nilai  $K$  yang terlalu kecil (misalnya  $K = 1$ ) dapat menyebabkan model menjadi sangat sensitif terhadap *noise* (*overfitting*). Berdasarkan hasil pengujian tersebut, rentang nilai  $K$  yang paling optimal berada pada kisaran  $K = 2$  hingga  $K = 4$ , di mana tingkat kesalahan masih relatif rendah dan stabil, sehingga dinilai paling efektif dalam menghasilkan keseimbangan antara akurasi dan kompleksitas model. Setelah dilakukan penentuan nilai  $K$  optimal melalui penerapan metode Elbow, langkah selanjutnya adalah menguji performa model KNN dengan berbagai nilai  $K$  untuk memperoleh tingkat akurasi yang paling tinggi. Tabel berikut menunjukkan hasil perbandingan akurasi terhadap nilai  $K$  yang diterapkan selama proses klasifikasi:

**Tabel 6.** Hasil Uji Nilai  $K$  pada Status Gizi berdasarkan Dataset Berat Balita

No	Nilai K	Akurasi
1	1	0.9289
2	2	0.9322
3	3	0.9393
4	4	0.9408
5	5	0.9414
6	6	0.9426
7	7	0.9431
8	8	0.9442
9	9	0.9454
10	10	0.9460
11	11	0.9464
12	12	0.9469
13	13	0.9467
14	14	0.9460
15	15	0.9460
16	16	0.9456
17	17	0.9460
18	18	0.9467
19	19	0.9476
20	20	0.9469

Berdasarkan hasil yang ada pada Tabel 6 pengujian dataset berat balita, akurasi model KNN menunjukkan peningkatan bertahap seiring dengan bertambahnya nilai  $K$  hingga mencapai kestabilan pada kisaran  $K = 10$  hingga  $K = 20$ , dengan nilai akurasi tertinggi sebesar 0.9476 pada  $K = 19$ . Pola tersebut sejalan dengan hasil analisis menggunakan metode Elbow sebelumnya, di mana penurunan *error rate* mulai melambat setelah nilai  $K$  melebihi 10,

menandakan bahwa model telah mencapai kondisi stabil dan tidak terjadi peningkatan signifikan terhadap performa. Dengan demikian, nilai K optimal dalam penelitian ini berada pada kisaran  $K = 10-19$ , dan  $K = 19$  dipilih sebagai nilai terbaik karena memberikan hasil akurasi tertinggi dalam proses klasifikasi status gizi.

**Tabel 7.** Hasil Uji Nilai K pada Status Gizi berdasarkan Dataset Tinggi Balita

No	Nilai K	Akurasi
1	1	0.9991
2	2	0.9979
3	3	0.9977
4	4	0.9969
5	5	0.9965
6	6	0.9955
7	7	0.9955
8	8	0.9951
9	9	0.9956
10	10	0.9955
11	11	0.9954
12	12	0.9952
13	13	0.9948
14	14	0.9946
15	15	0.9945
16	16	0.9943
17	17	0.9941
18	18	0.9938
19	19	0.9936
20	20	0.9930

Pada Tabel 7 meskipun akurasi yang diperoleh pada nilai  $K=1$  mencapai 99,91%, nilai ini perlu diinterpretasikan dengan hati-hati karena performa yang terlalu tinggi berpotensi mengindikasikan risiko overfitting, terutama pada algoritma KNN yang sangat sensitif terhadap data latih ketika nilai K bernilai kecil. Dalam konteks penelitian ini, evaluasi model dilakukan menggunakan pembagian data train–test sebesar 80:20 tanpa menerapkan prosedur validasi tambahan seperti K-Fold Cross Validation, Repeated Validation, atau pengujian menggunakan dataset eksternal. Oleh karena itu, meskipun hasil akurasi menunjukkan performa yang sangat tinggi, nilai tersebut belum dapat dikonfirmasi sepenuhnya sebagai representasi kemampuan generalisasi model. Penelitian selanjutnya direkomendasikan untuk menerapkan metode validasi tersebut guna memastikan bahwa performa model tidak dipengaruhi oleh pola tertentu pada data latih dan benar-benar mampu bekerja secara stabil pada data baru.

Evaluasi kinerja algoritma KNN dalam penelitian ini dilakukan melalui penerapan *Confusion Matrix*, yang berfungsi untuk menggambarkan distribusi keluaran model terhadap nilai sebenarnya dari data uji. Melalui *Confusion Matrix*, diperoleh sejumlah metrik evaluasi utama yang digunakan untuk mengukur performa model, yaitu *Accuracy*, *Precision*, *Recall*, dan *F1-Score* yang di definisikan dalam persamaan (10), (11), (12), dan (13). Berikut hasil dari *Confusion Matrix* dapat dilihat pada Tabel 8:

**Tabel 8.** Hasil Uji Algoritma KNN dengan Classification Report pada Status Gizi berdasarkan Dataset Berat dan Tinggi Balita

Dataset	Class	Precision	Recall	F1 Score	Accuracy
Berat K = 19	Normal	0.94	0.93	0.94	94.76%
	Overweight	0.97	0.98	0.98	
	Severely Underweight	0.97	0.96	0.96	
	Underweight	0.75	0.78	0.76	
Tinggi K = 1	Normal	1.00	1.00	1.00	99.91%
	Severely Stunted	1.00	1.00	1.00	
	Stunted	1.00	1.00	1.00	
	Tinggi	1.00	1.00	1.00	

### 3.1.3 Decision Tree

*Decision Tree* (Pohon Keputusan) merupakan metode yang sering digunakan dalam proses klasifikasi maupun prediksi banyak digunakan karena mampu mengubah data yang kompleks menjadi struktur pohon yang mudah dipahami, serta membantu mengidentifikasi pola dan hubungan antara variabel input dan variabel target guna mendukung proses analisis dan pengambilan keputusan secara lebih akurat[18]. Dalam proses pembentukan pohon keputusan, konsep *Entropy* digunakan untuk mengukur tingkat ketidakpastian atau ketidakaturan data pada s setiap node[20]. *Entropy* menggambarkan seberapa murni suatu himpunan data terhadap kelas target, jika seluruh data dalam

satu node termasuk ke dalam satu kelas yang sama, maka nilai *Entropy* akan bernilai nol, yang berarti node tersebut sepenuhnya murni. Sebaliknya, semakin beragam distribusi kelas di dalam node, semakin tinggi pula nilai *Entropy* yang menunjukkan ketidakpastian yang besar[21]. Nilai *Entropy* ini kemudian digunakan untuk menentukan *Information Gain*, yang menunjukkan tingkat pengurangan ketidakpastian setelah data dipisahkan berdasarkan suatu atribut. Atribut yang memiliki nilai dengan *Information Gain* paling tinggi akan dipilih sebagai pemisah terbaik, karena memberikan pembagian data yang paling informatif dan menghasilkan node-node dengan kemurnian yang lebih tinggi. Rumus-rumus utama yang digunakan dalam perhitungan *Entropy* dan *Information Gain* pada algoritma *Decision Tree* terdapat pada persamaan (7), dan ketika data dibagi berdasarkan suatu atribut, *Entropy* total dihitung sebagai rata-rata tertimbang dari *Entropy* setiap subset hasil pemisahan (8), lalu *Information Gain* pada persamaan (9) digunakan menentukan seberapa besar penurunan ketidakpastian setelah data dipisahkan berdasarkan atribut tertentu.

Berdasarkan hasil pengolahan nilai pada *Entropy* serta *Information Gain* pada setiap atribut yang dianalisis, algoritma akan memilih atribut dengan nilai *Information Gain* paling besar sebagai simpul akar (*root node*). Proses pemilihan atribut ini kemudian diulang secara rekursif pada setiap cabang hingga semua data berhasil diklasifikasikan dengan sempurna atau memenuhi kriteria penghentian tertentu. Dengan cara ini, terbentuk struktur pohon yang merepresentasikan hubungan antar atribut dan kelas target. Evaluasi kinerja model dilakukan dalam mengukur tingkat performa model *Decision Tree* secara keseluruhan dalam mengklasifikasikan data dengan tepat. Pengujian dilakukan menggunakan data uji, dan hasilnya dinilai berdasarkan matrik akurasi, *precision*, *recall*, serta *F1-score*. Berikut nilai *confusion matrix* tersebut disajikan pada Tabel 9 dan Tabel 10:

**Tabel 9.** Hasil Uji Algoritma DT dengan Classification Report pada Status Gizi berdasarkan Dataset Berat Balita

No	Dataset	Class	Precision	Recall	F1 Score	Accuracy
1	Berat	Normal	0.95	0.93	0.94	94.80%
		Overweight	0.97	0.98	0.98	
		Severely Underweight	0.96	0.96	0.96	
		Underweight	0.75	0.77	0.76	

Berdasarkan hasil evaluasi yang disajikan pada Tabel 9, algoritma *Decision Tree* menunjukkan performa yang cukup baik dalam mengklasifikasikan status gizi berdasarkan dataset berat badan balita, dengan tingkat akurasi mencapai 94,80%. Nilai *precision*, *recall*, dan *F1-score* pada kelas *Normal* berada pada rentang 0.93–0.95, sedangkan kelas *Overweight* dan *Severely Underweight* menunjukkan performa lebih tinggi dengan nilai mencapai 0.96–0.98. Sementara itu, kelas *Underweight* memiliki nilai *F1-score* yang lebih rendah, yaitu berada pada kisaran 0.75–0.77, yang mengindikasikan bahwa model relatif lebih sulit membedakan kategori tersebut dibandingkan kelas lainnya. Secara keseluruhan, hasil ini menunjukkan bahwa *Decision Tree* mampu mengelompokkan sebagian besar kategori status gizi dengan tingkat kesalahan yang rendah dan performa yang konsisten pada sebagian besar kelas.

**Tabel 10.** Hasil Uji Algoritma DT dengan Classification Report pada Status Gizi berdasarkan Dataset Tinggi Balita

No	Dataset	Class	Precision	Recall	F1 Score	Accuracy
1	Tinggi	Normal	1.00	1.00	1.00	99.89%
		Severely Stunted	1.00	1.00	1.00	
		Stunted	1.00	0.99	1.00	
		Tinggi	1.00	1.00	1.00	

Sementara itu, Tabel 10 menyajikan performa algoritma *Decision Tree* pada dataset tinggi badan balita, yang menunjukkan akurasi keseluruhan sebesar 99,89%. Seluruh kategori *Normal*, *Severely Stunted*, *Stunted*, dan *Tinggi* memperoleh nilai *precision*, *recall*, dan *F1-score* yang sangat tinggi, berkisar antara 0.99 hingga 1.00. Meskipun capaian tersebut tampak sangat optimal, interpretasinya memerlukan kehati-hatian karena model yang digunakan tidak menerapkan teknik *pruning* maupun strategi pengendalian kompleksitas. *Decision Tree* tanpa *pruning* cenderung membentuk struktur pohon yang sangat dalam dan sangat spesifik terhadap pola pada data pelatihan, sehingga meningkatkan risiko terjadinya *overfitting*. Dengan demikian, meskipun metrik evaluasi tampak sangat tinggi, hasil ini belum dapat dianggap sebagai indikator yang solid terhadap kemampuan generalisasi model.

#### 4. KESIMPULAN

Penelitian ini bertujuan untuk menganalisis dan membandingkan performa tiga algoritma *machine learning*, yaitu *Support Vector Machine (SVM)*, *K-Nearest Neighbors (KNN)*, dan *Decision Tree (DT)*, dalam mengklasifikasikan status gizi balita berdasarkan data antropometri yang mencakup umur, jenis kelamin, berat badan, dan tinggi badan. Hasil evaluasi menunjukkan bahwa algoritma *KNN* memberikan performa terbaik pada dataset tinggi badan balita dengan akurasi sebesar 99,91%, diikuti oleh *KNN* yang mencapai akurasi 99,89% pada dataset tinggi badan, serta 98,48% pada *SVM* dengan kernel *Radial Basis Function (RBF)* pada dataset tinggi badan. Meskipun *KNN* dan *DT* unggul secara numerik, *SVM* tetap menunjukkan kinerja yang stabil dan efektif dalam menangani pola data nonlinier. Dengan demikian, pemilihan algoritma yang optimal sangat bergantung pada kebutuhan implementasi, apakah



menekankan akurasi tertinggi, stabilitas model, atau kemudahan interpretasi. Untuk penelitian selanjutnya, disarankan untuk menerapkan prosedur validasi tambahan seperti K-Fold Cross Validation, Repeated Cross Validation pada KNN, menerapkan mekanisme pengendalian kompleksitas seperti pruning pada DT, dan pengembangan pendekatan *ensemble learning* atau model hibrida direkomendasikan guna meningkatkan akurasi dan ketahanan model terhadap variasi data, serta memperluas variabel penelitian seperti pola asupan gizi, kondisi lingkungan, dan status kesehatan anak agar sistem klasifikasi yang dihasilkan lebih komprehensif dan aplikatif.

## REFERENCES

- [1] L. D. Anggraeni, Y. R. Toby, and S. Rasmada, "Analisis asupan zat gizi terhadap status gizi balita," *Faletehan Health Journal*, vol. 8, no. 02, pp. 92–101, 2021, doi: <https://doi.org/10.33746/fhj.v8i02.191>.
- [2] S. P. Raturan, A. Achadiyahani, and A. F. Khairani, "Metode Antropometri Untuk Menilai Status Gizi: Sebuah Studi Literatur," *Health Information: Jurnal Penelitian*, vol. 15, 2023, [Online]. Available: <https://myjurnal.poltekkes-kdi.ac.id/index.php/hijp/article/view/704>
- [3] R. Ratnasari, A. J. Wahidin, and T. H. Andika, "Deteksi Dini Stunting Pada Anak Berdasarkan Indikator Antropometri dengan Menggunakan Algoritma Machine Learning," *Jurnal Algoritma*, vol. 21, no. 2, pp. 378–387, 2024, doi: <https://doi.org/10.33364/algoritma/v.21-2.2122>.
- [4] S. Lonang, A. Yudhana, and M. K. Biddinika, "Analisis Komparatif Kinerja Algoritma Machine Learning untuk Deteksi Stunting," *J. Media Inform. Budidarma*, vol. 7, no. 4, p. 2109, 2023, doi: <https://doi.org/10.30865/mib.v7i4.6553>.
- [5] A. F. F. Nugroho, *Klasifikasi Stunting dan Status Gizi Balita dengan Metode SVM (Support Vector Machine)*. 2024. [Online]. Available: <https://repository.unissula.ac.id/id/eprint/35614>
- [6] A. W. M. Gaffar, A. M. Halis, and S. R. Jabir, "Penerapan Algoritma Support Vector Machine untuk Klasifikasi Stunting pada Balita di Kabupaten Enrekang," *Jurnal Minfo Polgan*, vol. 13, no. 1, pp. 286–292, 2024, doi: <https://doi.org/10.33395/jmp.v13i1.13620>.
- [7] A. Maulana, C. Ramadhani, and A. Zainuddin, "Penerapan Algoritma K-Nearest Neighbor(KNN) Untuk Penentuan Status Stunting Pada Balita," *Repository Universitas Mataram*, 2024, [Online]. Available: <https://eprints.unram.ac.id/50284/2/JURNAL%20TA%20ALFATH%20MAULANA%20%20%28F1B018005%29.pdf>
- [8] F. M. Sarimole, F. B. Pasaribu, Y. Akbar, and A. Z. Hidayat, "Penerapan Algoritma K-Nearest Neighbor Untuk Klasifikasi Status Gizi Balita Di Posyandu Nusa Indah 4," *TEKNIKA*, vol. 18, no. 2, pp. 489–â, 2024, doi: <https://doi.org/10.5281/zenodo.12703925>.
- [9] G. P. Insany, I. Yustiana, and S. Rahmawati, "Penerapan KNN dan ANN pada klasifikasi status gizi balita berdasarkan indeks antropometri," *Jurnal CoSciTech (Computer Science and Information Technology)*, vol. 4, no. 2, pp. 385–393, 2023, doi: <https://doi.org/10.37859/coscitech.v4i2.5079>.
- [10] M. Ula, A. F. Ulva, M. Mauliza, M. A. Ali, and Y. R. Said, "Application Of Machine Learning In Determining The Classification Of Children's Nutrition With Decision Tree," *Jurnal Teknik Informatika (Jutif)*, vol. 3, no. 5, pp. 1457–1465, 2022, doi: <https://doi.org/10.20884/1.jutif.2022.3.5.599>.
- [11] R. N. Ramadhon, A. Ogi, A. P. Agung, R. Putra, S. S. Febrihartina, and U. Firdaus, "Implementasi Algoritma Decision Tree untuk Klasifikasi Pelanggan Aktif atau Tidak Aktif pada Data Bank," *Karimah Tauhid*, vol. 3, no. 2, pp. 1860–1874, 2024, doi: <https://doi.org/10.30997/karimahtauhid.v3i2.11952>.
- [12] A. C. Darmawan, "Pengembangan aplikasi berbasis web dengan python flask untuk klasifikasi data menggunakan metode decision tree C4. 5," 2023, [Online]. Available: <https://dspace.uui.ac.id/handle/123456789/42602>
- [13] F. Putra, H. F. Tahiyat, R. M. Ihsan, R. Rahmaddeni, and L. Efrizoni, "Penerapan Algoritma K-Nearest neighbor Menggunakan wrapper Sebagai preprocessing Untuk Penentuan Keterangan Berat Badan Manusia: application of K-Nearest neighbor algorithm using wrapper as preprocessing for determination of human weight information," *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, vol. 4, no. 1, pp. 273–281, 2024, doi: <https://doi.org/10.57152/malcom.v4i1.1085>.
- [14] W. N. Fadhillah, R. Susetyoko, and I. U. Nadhori, "Penerapan Algoritma Binning pada Preprocessing Data untuk Meningkatkan Akurasi Klasifikasi Multi-Kelas: Studi Kasus Data SDG," *Jurnal Infomedia: Teknik Informatika, Multimedia, dan Jaringan*, vol. 10, no. 2, pp. 87–94, 2025, doi: <http://dx.doi.org/10.30811/jim.v10i2.7165>.
- [15] A. Jalil, A. Homaidi, and Z. Fatah, "Implementasi Algoritma Support Vector Machine Untuk Klasifikasi Status Stunting Pada Balita," *G-Tech: Jurnal Teknologi Terapan*, vol. 8, no. 3, pp. 2070–2079, 2024, doi: <https://doi.org/10.33379/gtech.v8i3.4811>.
- [16] M. N. Maskuri, H. Harliana, K. Sukerti, and R. M. H. Bhakti, "Penerapan Algoritma K-Nearest Neighbor (KNN) untuk Prediksi Penyakit Stroke," *Jurnal Ilmiah Intech: Information Technology Journal of UMUS*, vol. 4, no. 01, pp. 130–140, 2022, doi: <https://doi.org/10.46772/intech.v4i01.751>.
- [17] S. N. Bakri and L. S. Harahap, "Analisis klasifikasi Algoritma K-Nearest Neighbor (K-NN) pada struktur Daerah di Kota Medan," *Jurnal Ilmu Komputer dan Sistem Informasi*, vol. 4, no. 2, pp. 182–193, 2025, doi: <https://doi.org/10.70340/jirsi.v4i2.165>.
- [18] A. Lasarudin, H. Gani, and M. Tomayahu, "Perbandingan metode Naïve Bayes dan C4. 5 klasifikasi status gizi bayi balita," *SPECTA Journal of Technology*, vol. 6, no. 3, pp. 273–283, 2022, doi: <https://doi.org/10.35718/specta.v6i3.789>.
- [19] L. Qadrini, A. Seppewali, and A. Aina, "Decision tree dan adaboost pada klasifikasi penerima program bantuan sosial," *Jurnal inovasi penelitian*, vol. 2, no. 7, pp. 1959–1966, 2021, [Online]. Available: <https://www.neliti.com/publications/469883/decision-tree-dan-adaboost-pada-klasifikasi-penerima-program-bantuan-sosial>
- [20] A. H. Anshor and A. T. Zy, "Implementasi Metode Decision Tree pada Sistem Prediksi Status Kualitas Produk Minuman A," *Jurnal Ilmiah Informatika Global*, vol. 15, no. 1, pp. 17–22, 2024, doi: <https://doi.org/10.36982/jiig.v15i1.3778>.
- [21] J. Bramanda, "Klasifikasi Masyarakat Penerima Bantuan Sosial dari Pemerintah dengan Metode Algoritma C4. 5," *Jurnal Komputer Antartika*, vol. 3, no. 1, pp. 34–41, 2025, doi: <https://doi.org/10.70052/jka.v3i1.234>.