

Klasifikasi Stunting Pada Balita Menggunakan Algoritma Decision Tree Pada Machine Learning

La Denna Hasri Monasari¹, Nunik Pratiwi^{1*}

¹Program Studi Teknik Informatika, Universitas Muhammadiyah Prof. DR. HAMKA, Indonesia.

Artikel Info

Kata Kunci:

Balita;
Decision Tree;
Klasifikasi;
Machine Learning;
Status Gizi;
Stunting.

Keywords:

Toddler;
Decision Tree;
Classification;
Machine Learning;
Nutritional Status;
Stunting.

Riwayat Artikel:

Submitted: 19 Juni 2025
Accepted: 17 Juli 2025
Published: 24 Juli 2025

Abstrak: *Stunting* adalah gangguan pertumbuhan yang dialami balita akibat kekurangan gizi sehingga menghambat pertumbuhan fisik dan perkembangan kognitif. Faktor penyebabnya meliputi kurangnya asupan nutrisi, rendahnya pendidikan orang tua, serta lingkungan kesehatan yang tidak memadai. Penelitian ini bertujuan untuk mengklasifikasi faktor penyebab terjadinya *stunting* sehingga dapat dilakukan intervensi dini, serta dapat mengembangkan model klasifikasi *stunting* menggunakan algoritma *Decision Tree*. Untuk mendeteksi risiko *stunting* secara dini, teknologi *machine learning* dimanfaatkan guna mengklasifikasikan data secara lebih akurat dan efisien. Penelitian ini menggunakan algoritma *Decision Tree* untuk mengkategorikan status gizi balita berdasarkan usia (bulan), jenis kelamin, dan tinggi badan (cm). Data diperoleh dari platform Kaggle, terdiri atas 121.000 data. Penelitian mencakup proses pengumpulan data, *pre-processing* data (*cleaning* data, deteksi *outlier*, dan *label encoding*), eksplorasi data, pemisahan data latih dan uji, serta pelatihan model. Penelitian ini menerapkan *confusion matrix* sebagai metode evaluasi untuk menilai kinerja model dalam klasifikasi *stunting* pada balita. Hasil evaluasi *model Decision Tree* memperoleh akurasi sebesar 95.73% dalam mengklasifikasi status gizi ke dalam empat kategori yaitu normal, *stunted*, *severely stunted*, dan tinggi. Temuan menunjukkan bahwa algoritma *Decision Tree* efektif dalam mengidentifikasi kondisi *stunting* serta mendukung sistem pemantauan gizi anak balita secara digital.

Abstract: *Stunting* is a growth disorder experienced by toddlers due to malnutrition, which inhibits physical growth and cognitive development. The causes include inadequate nutritional intake, low parental education levels, and inadequate health environments. The study aims to classify the factors causing *stunting* so that early intervention can be carried out, as well as to develop a *stunting* classification model using the *Decision Tree* algorithm. To detect the risk of *stunting* early, machine learning is used to classify data more accurately and efficiently. This study uses the *Decision Tree* algorithm to classify infants' nutritional status based on age (months), gender, and height (cm). Data was obtained from the Kaggle platform, consisting of 121,000 data points. This study includes data collection, data *pre-processing* (data cleaning, outlier detection, and label encoding), data exploration, training and testing data separation, and model training. This study used a *confusion matrix* as an evaluation method to assess the model's performance in *stunting* classification. The *Decision Tree* model achieved an accuracy of 95.73% in classifying nutritional status into four categories: normal, *stunted*, *severely stunted*, and tall. These findings indicate that the *Decision Tree* algorithm is effective in identifying *stunting* conditions and supports digital monitoring systems for infant nutrition.

Corresponding Author:

Nunik Pratiwi

Email: npratiwi@uhamka.ac.id

PENDAHULUAN

Stunting adalah kondisi dimana balita mengalami kekurangan gizi kronis, bahkan dapat memengaruhi pertumbuhan fisik dan kemampuan kognitif anak. Banyak faktor yang dapat menyebabkan kondisi tersebut yaitu kurangnya asupan gizi selama kehamilan dan pasca kelahiran, asupan vitamin dan mineral yang buruk, kehamilan yang terlalu dekat, dan tingkat pendidikan orang tua. Tinggi badan juga merupakan penyebab utama *stunting*. Balita dengan tinggi badan pendek biasanya mengalami *stunting* dengan 25% pada usia 12 hingga 59 bulan (Hardiani & Putri, 2024). *Stunting* yang terjadi sejak kandungan hingga anak usia dua tahun, dapat melemahkan fungsi sistem kekebalan tubuh serta meningkatkan risiko kematian bayi dan anak (Septian et al., 2024). Dengan begitu jika faktor risiko *stunting* ditemukan lebih awal atau kurang optimal, kemungkinan masalah akan muncul lebih besar. Pada tahun 2022, tercatat sebanyak 149 juta anak di bawah usia lima tahun mengalami *stunting*, yaitu kondisi yang terlalu pendek dibandingkan usia mereka. Selain itu, sekitar 45 juta anak mengalami *wasting* yang merupakan kondisi terlalu kurus untuk badan yang tinggi, dan 37 juta anak lainnya mengalami berat badan yang lebih (World Health Organization, 2024).

Dengan kemajuan teknologi informasi, data *science* sangat dibutuhkan dalam mengolah data untuk menjadi hal yang bermanfaat sebagai pendukung keputusan dalam berbagai bidang. Data *science* merupakan komponen dari *machine learning* dan telah berkembang ke berbagai bidang kehidupan. Dengan memberikan insight yang berjalan secara otomatis dan prediktif. *Machine learning* dapat membantu mengoptimalkan proses pengambilan keputusan. *Stunting* memiliki dampak yang serius dan beragam, jadi sangat penting untuk mendeteksi sejak dini. *Machine learning* menghadirkan solusi inovatif untuk mendeteksi *stunting* sejak awal, memungkinkan untuk memprediksi kondisi *stunting* dengan menggunakan data berbagai karakteristik balita (Lonang et al., 2023). Salah satu contohnya adalah dengan menggunakan metode *Decision Tree*, sebuah metode pembelajaran yang membangun model prediksi dalam bentuk pohon. Pohon ini terdiri dari node yang mewakili keputusan berdasarkan nilai-nilai atribut, dan cabang-cabang yang mewakili hasil dari pengambilan keputusan.

Dalam proses pengkategorian, tujuan utama proses klasifikasi adalah kelas label dimana tahap klasifikasi digunakan untuk menemukan kelas data yang tidak memiliki label kelas yang diketahui (Jalil et al., 2024). Salah satu metode pembelajaran mesin yang dikenal sebagai klasifikasi bertujuan untuk mengelompokkan atau mengidentifikasi item berdasarkan fitur tertentu. Metode ini mirip dengan cara orang membedakan benda satu sama lain. Setelah itu, klasifikasi dilakukan untuk mengetahui ketepatan klasifikasi dengan data pengujian. Algoritma *machine learning* dapat membantu klasifikasi *stunting* pada balita lebih cepat, akurat, dan efektif dibandingkan dengan metode konvensional. Algoritma klasifikasi yang digunakan dalam penelitian ini adalah *Decision Tree*.

Stunting telah dibahas dalam sejumlah penelitian sebelumnya menggunakan beragam pendekatan dengan memanfaatkan metode selain *Decision Tree*, seperti C4.5, *Naïve Bayes*, dan *Fuzzy K-Nearest Neighbor* (FKNN), serta *K-Nearest Neighbor* (KNN). Devy Fatmawati, Wulan Trisnawati dan tim peneliti lainnya menampilkan pendekatan dengan melibatkan dua metode sekaligus, yaitu *Decision Tree* dan *Naïve Bayes* dengan tujuan perbandingan. Dengan menggunakan teknik *Cross Validation* dimana jumlah 'k' pada *k-fold cross validation* adalah 5. Hasil menunjukkan akurasi *Decision Tree* sebesar 98,10% sedangkan *Naïve Bayes* hanya 97,95%, dan nilai *recall Decision Tree* sebesar 99,51% dan *Naïve Bayes* sebesar 98,78%. Temuan dalam penelitian ini mengindikasikan bahwa metode *Decision Tree* sangat efektif dalam mengklasifikasi data dengan persentase akurasi yang sangat baik (Fatmawati et al., 2023). Sedangkan hasil riset yang melibatkan penulis Fanny Ramadhani, Andy Satria, dan Indah Purnama Sari menampilkan pendekatan yang berbeda dengan menggunakan metode *Fuzzy K-Nearest Neighbor* (FKNN) pada klasifikasi penyakit DBD, dengan menggunakan pembagian dataset 70% data training serta 30% data testing pada 300 jumlah dataset mendapatkan hasil akurasi yang optimal pada

K=13 dengan hasil akurasi sebesar 85.01%, dengan demikian pendekatan ini mampu memberikan hasil klasifikasi yang sangat akurat (Ramadhani et al., 2023).

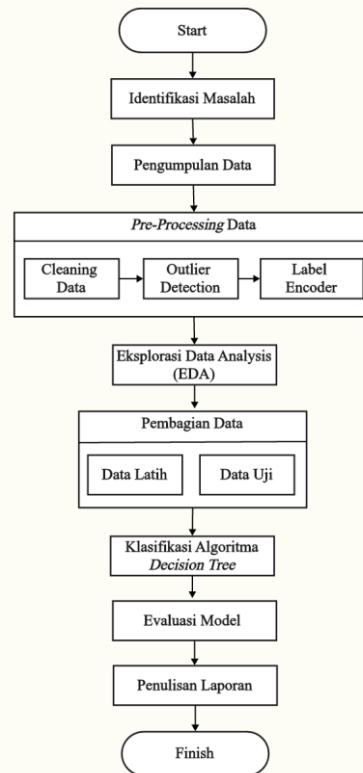
Sedangkan penelitian yang melibatkan Agung Rachmat Raharja, Jayadi dan rekannya yang menggunakan metode *Decision Tree*, mengungkapkan bahwa untuk memprediksi risiko penyakit jantung berdasarkan dataset *Framingham*. Model ini mencapai akurasi 74% pada data pengujian. Hasil ini membuktikan bahwa metode *Decision Tree* merupakan pendekatan yang efektif dalam mendukung prediksi risiko penyakit jantung (Raharja et al., 2024). Sedangkan penelitian lainnya yang dilakukan oleh Nadhea Ovella Syaahasdy dan rekannya dengan metode *Decision Tree* yang menggunakan algoritma C4.5 dan Arell S. Biyantoro, Budi Prasetyo yang menggunakan pendekatan *Decision Tree*, *K-Nearest Neighbor* (KNN), serta algoritma *Naïve Bayes*, keduanya lebih fokus dalam memprediksi faktor atau karakteristik yang paling signifikan dalam menentukan tingkat kesehatan seseorang (Biyantoro & Prasetyo, 2024; Syaahasdy et al., 2023).

Beberapa penelitian bertujuan yaitu menggunakan teknik *machine learning* untuk mengevaluasi seberapa efektif metode yang digunakan dalam melakukan klasifikasi, sementara yang lain lebih menekankan dalam mengidentifikasi faktor risiko yang diteliti. Berbeda dengan pendekatan tersebut, penelitian ini secara komprehensif menggabungkan kedua aspek tersebut. Penelitian yang dilakukan ini tidak hanya membahas performa algoritma *Decision Tree* dalam mengklasifikasi, namun juga membahas terkait masalah *stunting* sebagai isu sosial. Penelitian ini memberikan hasil berupa akurasi *Decision Tree* sebagai ukuran kinerja, sekaligus merinci dalam mengidentifikasi terhadap fenomena *stunting*. Oleh karena itu, penelitian ini dapat menjadi dasar dalam pengambilan keputusan, khususnya dalam upaya pencegahan dan penanganan *stunting* pada balita. Melalui penerapan digital berbasis *machine learning*, hal ini sejalan dengan perkembangan teknologi informasi yang semakin maju di era digital dalam mendukung pelayanan kesehatan yang responsif.

Dengan membatasi ruang lingkup penelitian, peneliti mengarahkan fokus pada beberapa aspek tertentu. Kaggle.com adalah *platform* yang digunakan untuk mengumpulkan data sebagai penelitian ini dengan memanfaatkan *tool Google Colab*. Penelitian ini mengklasifikasikan *stunting* dengan menggunakan algoritma *Decision Tree* karena kemampuannya untuk memberikan pemahaman yang intuitif tentang komponen faktor yang berkontribusi terhadap risiko *stunting* pada balita. Data yang didapat terkait *stunting* pada balita oleh peneliti adalah sebanyak 121000 data.

METODE

Diagram alir ini memperlihatkan langkah-langkah penelitian yang akan dilakukan. Tahapan yang dilakukan yaitu untuk mengklasifikasi *stunting* pada balita dengan menerapkan algoritma *Decision Tree*, terlihat pada Gambar 1. Berikut:



Gambar 1. Tahapan Penelitian

Pengumpulan Data

Pada tahap ini, melakukan langkah awal dengan mengumpulkan data melalui analisis literatur yang relevan, tahap pengumpulan data dimulai dengan mengimpor file CSV dengan berisikan data mengenai stunting pada balita yang diperoleh dari dataset Kaggle.com sejumlah 121000 data. Data yang digunakan penelitian ini berupa dataset *stunting* berdasarkan usianya (bulan), jenis kelamin, tinggi badannya (cm), serta status gizi balita dengan rentang usia adalah 0 hingga 60 bulan.

	Umur (bulan)	Jenis Kelamin	Tinggi Badan (cm)	Status Gizi
0	0	laki-laki	44.591973	stunted
1	0	laki-laki	56.705203	tinggi
2	0	laki-laki	46.863358	normal
3	0	laki-laki	47.508026	normal
4	0	laki-laki	42.743494	severely stunted
...
120994	60	perempuan	100.600000	normal
120995	60	perempuan	98.300000	stunted
120996	60	perempuan	121.300000	normal
120997	60	perempuan	112.200000	normal
120998	60	perempuan	109.800000	normal

120999 rows × 4 columns

Gambar 2. Tampilan Data *Stunting* Pada Google Colab

Pada Gambar 2. data yang di dapatkan sebanyak 121000 dataset *stunting* yang berisikan umur (bulan), jenis kelamin, tinggi badan (cm), serta status gizi yang telah diimpor ke dalam library Python.

Pre-Processing Data

Pada langkah ini, data yang telah dikumpulkan perlu melalui proses *pre-processing* sebelum dapat digunakan untuk analisis lebih lanjut. Proses *pre-processing* yang meliputi *cleaning* data untuk menghilangkan noise dan inkonsistensi, *outlier detection* untuk mengidentifikasi data yang menyimpang, serta *label encoding* untuk mengubah data kategorial menjadi format numerik.

a. *Cleaning Data*

Cleaning data merupakan proses penting untuk memastikan kualitas data yang akan dianalisis. Proses ini meliputi, penanganan nilai yang hilang (*missing value*), *handling inconsistent data*, dan *duplicated data*.

b. *Outlier Detection*

Mengidentifikasi *outlier* sangat penting karena kehadiran nilai-nilai ini dapat memengaruhi hasil analisis serta model yang dikembangkan ini dapat menyebabkan estimasi statistik yang tidak akurat serta dapat mengurangi keakuratan model. Dengan demikian, proses identifikasi *outlier* merupakan bagian penting untuk memastikan integritas data. (Samosir et al., 2024).

c. *Label Encoder*

Dalam *machine learning*, *label encoder* adalah teknik *pre-processing* data yang penting. *Label Encoder* memungkinkan algoritma *machine learning* untuk memproses data secara efisien dengan mengubah data kategorikal menjadi representasi numerik. Setiap kategori menerima nilai numerik unik selama proses ini, yang memungkinkan integrasi data kategori dengan data numerik lainnya. Dengan demikian, model dapat memanfaatkan data secara optimal (Herdian et al., 2024).

Exploratory Data Analysis (EDA)

Salah satu langkah penting dalam proses analisis data adalah *Exploratory Data Analysis* (EDA). Sebelum memasuki tahap pemodelan, langkah ini berfokus dalam memperoleh pemahaman mendalam tentang struktur dan karakteristik dataset. Dengan demikian, proses ini memberikan wawasan penting yang menjadi dasar untuk analisis berikutnya tetapi juga memastikan bahwa prosedur analisis lanjutan, seperti pemodelan statistik (Radhi et al., 2021).

Pembagian Data

Pembagian data atau *splitting data* merupakan tahapan krusial dalam proses analisis data yang bertujuan untuk membagi dataset menjadi beberapa kelompok kecil di mana setiap bagian memiliki peran yang berbeda. Data pelatihan dan data pengujian adalah dua komponen utama dalam model klasifikasi. Data latih dimanfaatkan untuk membangun model, sementara data uji digunakan guna menilai performa model yang telah dibuat. Pembagian yang tepat dapat melatih model untuk memahami pola klasifikasi dan membuat prediksi yang akurat (Marsya Finda & Wahyu Utomo, 2024).

Klasifikasi Algoritma Decision Tree

Untuk membuat model prediksi dalam klasifikasi, metode *Decision Tree* merupakan salah satu metode penting yang sering digunakan dalam *machine learning*. Pengelompokan data ke dalam kategori yang berbeda berdasarkan atributnya dikenal sebagai klasifikasi sendiri. Untuk memahami hubungan antar atribut dalam data, *Decision Tree* digunakan untuk mengidentifikasi aturan-aturan yang mendasari pemisahan antara kelas (Antika et al., 2023).

Evaluasi Model

Proses evaluasi model dilakukan untuk menentukan kombinasi model yang paling efisien dengan melakukan evaluasi melalui metrik seperti akurasi, presisi, recall, dan skor F1. Dalam *machine learning*, Untuk menilai performa suatu model, teknik evaluasi yang umum digunakan adalah *confusion matrix*. Pada *machine learning*, *confusion matrix* juga digunakan untuk mendeteksi permasalahan dalam klasifikasi, terutama ketika *output* mencakup dua kelas atau lebih (Melyani & Zuhri Harahap, 2024).

HASIL DAN PEMBAHASAN***Pre-processing***

Dalam tahap ini, data yang telah dikumpulkan perlu melalui proses *preprocessing*, sebelum dapat digunakan untuk analisis lebih lanjut yang bertujuan untuk mengolah data dengan mendeteksi dan menangani anomali strukturnya melalui sejumlah proses tertentu (Asno Azzawagama Firdaus et al., 2023). Proses *preprocessing* yang meliputi *cleaning* data untuk menghilangkan *noise* dan inkonsistensi,

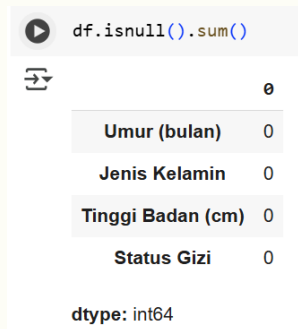
outlier detection untuk mengidentifikasi data yang menyimpang, serta *label encoding* untuk mengubah data kategorial menjadi format numerik.

a. *Cleaning Data*

Pada tahap *cleaning* data dilakukan proses penting untuk memastikan kualitas data yang meliputi penanganan nilai hilang, inkonsistensi, dan kesalahan lainnya (Fitriah et al., 2023). Proses ini melibatkan tahap *Missing Value*, *Handling Imconsistent Data*, dan *Duplicated Data*.

1. *Missing Value*

Kondisi di mana nilai tidak lengkap atau hilang pada salah satu kriteria dalam dataset dikenal sebagai *missing data* atau *missing value*. Ini sering menjadi masalah dalam analisis data karena dapat mempengaruhi hasil analisis dan interpretasi data secara signifikan (Sudrajat & Cholid, 2023).



```
df.isnull().sum()
```

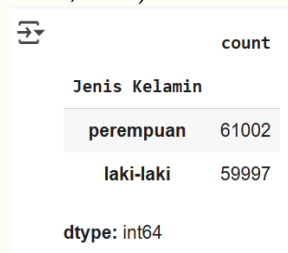
	0
Umur (bulan)	0
Jenis Kelamin	0
Tinggi Badan (cm)	0
Status Gizi	0
dtype: int64	

Gambar 3. Hasil *Missing Value*

Pada Gambar 3. hasil menunjukkan bahwa semua nilai dikolom data adalah tipe integer yang konsisten, dengan begitu tidak ditemukan adanya indikasi nilai yang hilang atau tidak valid.

2. *Handling Inconsistent Data*

Proses ini melibatkan identifikasi dan perbaikan data yang memiliki format tidak seragam, nilai yang tidak valid, atau kesalahan pencatatan, sehingga memastikan dataset yang digunakan konsisten (Syam & Saharuna, 2024).



```
df['Jenis Kelamin'].value_counts()
```

	count
Jenis Kelamin	
perempuan	61002
laki-laki	59997
dtype: int64	

Gambar 4. Hasil *Inconsistent Data*

Pada Gambar 4. Hasil dari tahap ini menunjukkan bahwa tidak terdapat entri yang berulang, dengan demikian data yang digunakan telah divalidasi dan layak untuk digunakan dalam tahap pemrosesan selanjutnya.

3. *Duplicated Data*

Dalam *cleaning* data, data duplikat merupakan data yang memiliki nilai identik dan sering muncul lebih dari satu kali. Proses *duplicated data* merupakan tahap pemeriksaan apakah data terindikasi adanya duplikat data atau tidak (Guo et al., 2023).


```
[ ] df = df.drop_duplicates()

[ ] df.duplicated().sum()

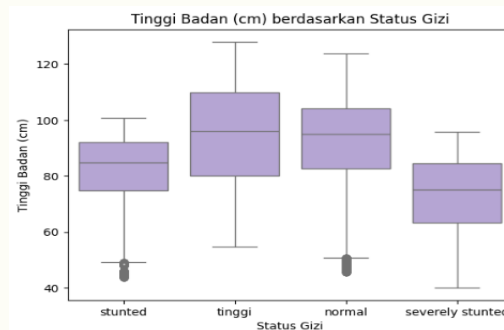
0
```

Gambar 5. Hasil *Duplicated Data*

Berdasarkan Gambar 5. hasil dari tahap ini menunjukkan bahwa tidak terdapat entri yang berulang. Oleh karena itu, data yang telah melalui tahap *cleaning* dengan siap dapat digunakan dalam proses berikutnya.

b. *Outlier Detection*

Tahap pemeriksaan outlier ini dilakukan untuk menemukan berbagai pola berbeda dari data yang tersedia. Tujuannya agar dataset yang digunakan merupakan data yang berkualitas dan bersih, sehingga analisis yang dilakukan menjadi lebih akurat.

Gambar 6. Hasil *Outlier*

Berdasarkan Gambar 6. yang menyajikan diagram *bloxpot* tinggi badan (cm) terhadap status gizi. Distribusi *outlier* pada setiap kategori status gizi menunjukkan bahwa data memiliki pola-pola yang berbeda. Pada kategori “*stunted*” terlihat *outlier* berada di bagian bawah grafik yang menunjukkan bahwa tinggi badan pada balita jauh lebih pendek dibandingkan dengan kategori “*tinggi*” dan “*normal*”. Dilihat pada kategori “*normal*” terdapat beberapa outlier dibagian bawah, hal ini menunjukkan bahwa meskipun secara umum balita masuk kedalam kategori tubuh normal, tetapi masih ada Sebagian yang tinggi badannya jauh lebih rendah dari kebanyakan balita yang normal. Pada kategori “*severely stunted*” terdapat outlier yang berada dibawah whisker yang menunjukkan kasus ekstrem kekurangan gizi kronis.

c. *Label Encoder*

Pada tahap *label encoder* data dengan kategori “Jenis Kelamin” dan “Status Gizi” yang semula data kategori diubah menjadi representasi numerik. Proses ini mengubah nilai tekstual atau simbolik menjadi angka yang unik sehingga dapat dengan mudah digunakan dalam model *machine learning*.

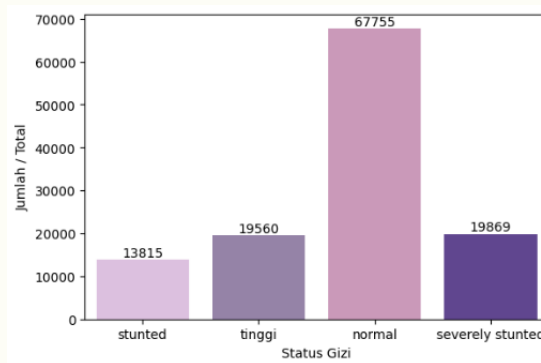
	Umur (bulan)	Jenis Kelamin	Tinggi Badan (cm)	Status Gizi
0	0	0	44.591973	2
1	0	0	56.705203	3
2	0	0	46.863358	0
3	0	0	47.508026	0
4	0	0	42.743494	1

Gambar 7. Hasil *Label Encoder*

Pada Gambar 7. hasil menunjukkan bahwa data dengan kategori “Jenis Kelamin” dan “Status Gizi” berhasil dikonversi ke dalam bentuk data numerik. Proses ini sangat diperlukan karena algoritma *Decision Tree* dapat dengan mudah melakukan klasifikasi secara akurat. Jenis kelamin dengan nilai 0 adalah sebagai laki-laki, sedangkan nilai 1 merupakan perempuan. Pada status gizi dengan nilai 0 yaitu sebagai kategori normal, nilai 1 untuk kategori *severely stunted*, nilai 2 sebagai kategori *stunted*, serta nilai 3 untuk menunjukkan kategori tinggi.

Exploratory Data Analysis (EDA)

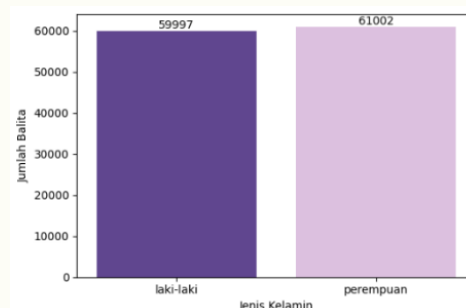
Melalui Teknik visualisasi data dan statistik deskriptif, *exploratory data analysis* membantu memahami data dengan baik seperti tren, serta hubungan antar variable yang jelas, sehingga memudahkan dalam menganalisis.



Gambar 8. Hasil Visualisasi Berdasarkan Tinggi Badan

Pada Gambar 8. adalah diagram yang memperlihatkan status gizi balita dikelompokkan berdasarkan tinggi badan, yang merupakan salah satu variable utama dalam penelitian. Berikut penjelasan untuk setiap kategori:

- Severely Stunted*: Pada kategori ini balita dengan kondisi stunting yang sangat kronis memiliki tinggi badan di bawah rata-rata. Jumlah kategori ini lebih banyak setelah kategori “Tinggi”.
- Stunted*: Kategori ini adalah balita dengan tinggi badan di bawah standar di usianya yang menunjukkan masalah pertumbuhan. Jumlahnya paling sedikit dibandingkan dengan kategori yang lain.
- Normal: Kategori ini berisi balita dengan tinggi badan normal. Jumlah kategori ini paling banyak dibandingkan kategori lainnya.
- Tinggi: Pada kategori ini menunjukkan balita dengan tinggi badan di atas rata-rata. Jumlahnya lebih banyak dibandingkan dengan kategori “Normal”.

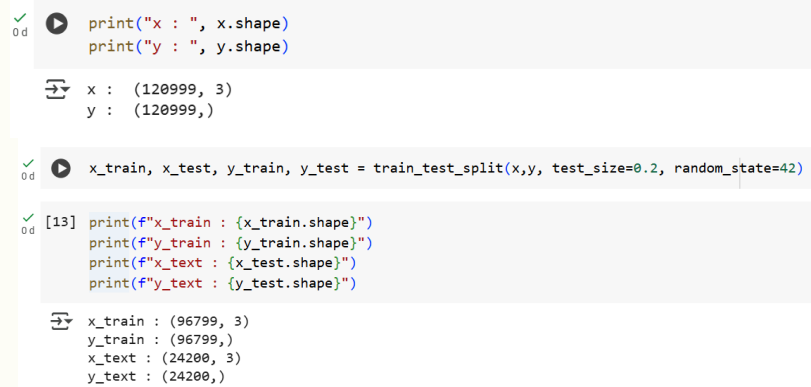


Gambar 9. Hasil Visualisasi Jenis Kelamin

Sebagaimana ditunjukkan pada Gambar 9. hasil grafik tersebut jumlah balita pada kategori perempuan lebih banyak dibandingkan dengan laki-laki mencapai lebih dari 61000 orang, sedangkan jumlah laki-laki mencapai 50000 orang.

Pembagian Data

Setelah tahap *preprocessing*, data dibagi ke dalam dua bagian yaitu data pelatihan dan pengujian. Pada penelitian ini dataset dipisahkan dengan rasio 80:20 sebagai data pelatihan dan data pengujian. Tujuan dari proses ini adalah untuk mengevaluasi sejauh mana model mampu mengklasifikasi data secara akurat.



```

0d print("x : ", x.shape)
   print("y : ", y.shape)

x : (120999, 3)
y : (120999,)

0d x_train, x_test, y_train, y_test = train_test_split(x,y, test_size=0.2, random_state=42)

0d [13] print(f"x_train : {x_train.shape}")
      print(f"y_train : {y_train.shape}")
      print(f"x_test : {x_test.shape}")
      print(f"y_test : {y_test.shape}")

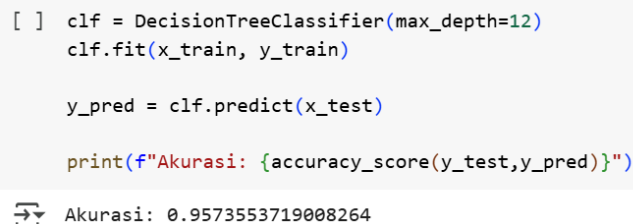
x_train : (96799, 3)
y_train : (96799,)
x_test : (24200, 3)
y_test : (24200,)
  
```

Gambar 10. Pembagian Data Pelatihan dan Pengujian

Seperti yang disajikan pada Gambar 10. dataset telah dibagi menjadi dua bagian utama yaitu sebesar 80% untuk proses pelatihan dan sebesar 20% untuk pengujian model. Dari data training dihasilkan 96799 sampel data dan setiap sampel memiliki 3 fitur, sementara itu 24200 sampel data dan memiliki 3 fitur diperoleh dari data testing.

Model Decision Tree

Pada proses ini algoritma *Decision Tree* digunakan dalam mengklasifikasi data dimulai dengan melatih model. Proses pelatihan *model Decision Tree* akan belajar dari data pelatihan untuk memahami bagaimana fitur-fitur berhubungan dengan label.



```

[ ] clf = DecisionTreeClassifier(max_depth=12)
    clf.fit(x_train, y_train)

    y_pred = clf.predict(x_test)

    print(f"Akurasi: {accuracy_score(y_test,y_pred)}")

Akurasi: 0.9573553719008264
  
```

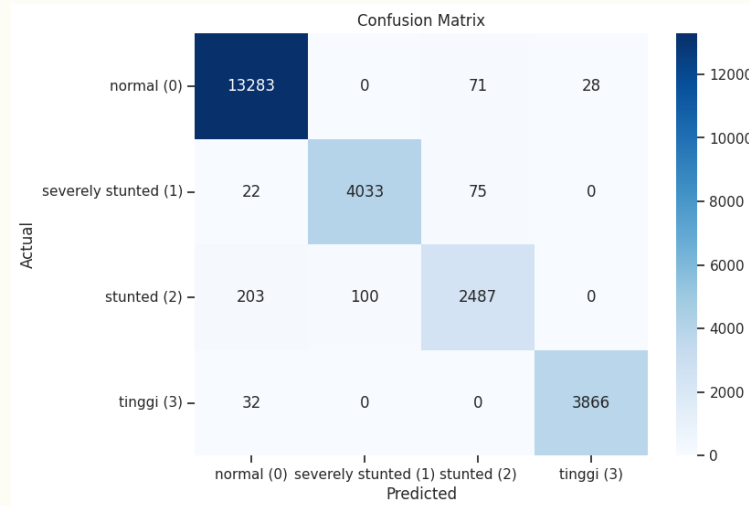
Gambar 11. Model Klasifikasi *Decision Tree*

Pada Gambar 11. dapat dilihat model *Decision Tree* menggunakan nilai *max_depth* = 12 sebagai parameter. Penggunaan *maxdepth* diharapkan dapat melakukan evaluasi terhadap data dengan tujuan untuk menghasilkan model dengan tingkat akurasi yang tinggi dalam mengklasifikasi data. Dari hasil tersebut didapatkan akurasi model dengan algoritma *Decision Tree* sebesar 0.957 atau 95.73%. Dengan kata lain, setelah dilakukan pembagian data latih dan uji model *Decision Tree* mampu memprediksi dengan tepat sekitar 95.73% dari seluruh data uji.

Evaluasi Model

Penelitian ini menerapkan *confusion matrix* sebagai metode evaluasi untuk menilai kinerja model *Decision Tree*. Metode ini efektif dalam menilai model klasifikasi karena mampu menghitung matrik penting seperti nilai *accuracy*, *precision*, *recall*, serta *f1-score*, serta memberikan ilustrasi mengenai jumlah

prediksi yang tepat dan keliru dalam setiap kategori. Model yang diuji adalah hasil dari proses pelatihan dengan menggunakan *Decision Tree* dengan akurasi training 96.22% dan akurasi testing mencapai 95.74%. Hasil pengujian menunjukkan kinerja model yang cukup optimal seperti yang ditunjukkan Gambar 12. Berikut:



Gambar 12. Hasil *Confusion Matrix*

Pada Gambar 12. memperlihatkan hasil penilaian model melalui *confusion matrix*. Model *Decision Tree* berhasil digunakan untuk mengklasifikasi *stunting* pada balita berdasarkan parameter tinggi badan. Model yang dikembangkan mampu memprediksi hasil klasifikasi secara tepat yaitu sejumlah 23669 data dari total 24200 data uji, diperoleh tingkat akurasi rata-rata sebesar 97%. Berdasarkan hasil ini, dapat disimpulkan bahwa model pelatihan yang digunakan berjalan dengan efektif.

Confusion matrix juga digunakan untuk melakukan berbagai perhitungan seperti nilai *accuracy*, *recall*, *precision* dan *f1-score*. Perhitungan *accuracy* bertujuan untuk menilai sejauh mana algoritma *Decision Tree* dalam mengklasifikasi data pada seluruh dataset pengujian. *Recall* bertujuan untuk menunjukkan secara akurat dalam mengidentifikasi seluruh sentiment positif. Sementara *precision* yaitu untuk mengukur tingkat akurasi algoritma dalam memberikan prediksi positif sebagaimana ditampilkan pada Tabel 1. Berikut:

Tabel 1. Hasil *Classification Report* Status Gizi

Status Gizi	Precision	Recall	F1-Score
Normal	0.96	0.99	0.97
Severely Stunted	0.97	0.94	0.95
Stunted	0.89	0.86	0.87
Tinggi	0.99	0.95	0.97

Berdasarkan pada Tabel 1. Pada kategori normal menunjukkan performa yang sangat baik dengan *precision* 0.96, *recall* 0.99, dan *F1-score* 0.97, dimana *recall* memiliki nilai tertinggi yang menunjukkan efektifitas model dengan baik dalam mengklasifikasi *stunting* dengan sedikit *false negative*. Kategori *severely stunted* juga menunjukkan kinerja model yang baik yaitu *precision* sebesar 0.97, *recall* 0.94, serta *F1-score* 0.95. Pada kategori *stunted* performa model dalam mengidentifikasi status gizi ini kurang optimal karena *recall* yang relatif rendah yaitu 0.86 yang menunjukan kategori tersebut tidak terdeteksi dengan baik (*false negative*). Sementara itu, pada kategori tinggi menunjukkan performa model memiliki keseimbangan yang baik antara *precision* dan *recall*, sehingga model dengan baik mengklasifikasikan status gizi dengan sedikit kesalahan *false positive* dan *false negative* dengan *precision* sebesar 0.99, *recall* 0.95, *F1-score* 0.97.

Proses klasifikasi pada model tersebut digunakan untuk menguji peforma model dalam mengklasifikasi *stunting* pada balita. Hasil dari klasifikasi tersebut akan menentukan kategori pada

status gizi balita untuk memastikan bahwa model memiliki kemampuan untuk mengelompokkan data dengan sangat akurat.

```
[ ] testing = {'Jenis Kelamin' : [0],
              'Umur (bulan)' : [2],
              'Tinggi Badan (cm)' : [45.07]}

testing = pd.DataFrame(testing)
testing

[ ] pred_coba = dt.predict(testing)
print("Hasil Prediksi dari Balita Baru")
print(pred_coba)
```

Hasil Prediksi dari Balita Baru
[1]

Gambar 13. Hasil Uji Coba

Pada Gambar 13. merupakan hasil klasifikasi stunting yang diambil dari beberapa sampel menggunakan model *Decision Tree* menunjukkan bahwa balita tersebut diklasifikasikan ke dalam kategori '3'. Terdapat empat klasifikasi dalam status gizi yaitu 0 untuk normal, 1 untuk *severely stunted*, 2 untuk *stunted*, dan 3 untuk tinggi. Dengan demikian hasil klasifikasi tersebut berdasarkan atribut penting seperti jenis kelamin, umur (bulan) dan tinggi dadan (cm) yang diberikan merupakan kategori tinggi. Berikut ini adalah hasil klasifikasi *stunting* pada balita berdasarkan tinggi badan menggunakan algoritma *Decision Tree* dapat dilihat pada Tabel 2.

Tabel 2. Hasil Klasifikasi Status Gizi

Jenis Kelamin	Umur (bulan)	Tinggi (cm)	Hasil Kalsifikasi	Status Gizi
Perempuan	3	52	1	<i>Severely stunted</i>
Laki-laki	27	82	2	<i>Stunted</i>
Laki-laki	39	115	3	Tinggi
Perempuan	47	80	1	<i>Severely stunted</i>
Laki-laki	15	71	2	<i>Stunted</i>
Perempuan	9	80	3	Tinggi
Laki-laki	49	100	0	Normal
Perempuan	60	95	2	<i>Stunted</i>
Perempuan	23	95	0	Normal
Laki-laki	0	60	3	Tinggi

KESIMPULAN

Berdasarkan temuan penelitian, penggunaan algoritma *Decision Tree* terbukti mampu mengklasifikasikan status *stunting* pada balita secara efektif. Tahapan klasifikasi ini memungkinkan identifikasi kondisi gizi secara tepat ke dalam beberapa kategori yaitu normal, *severely stunted*, *stunted*, dan tinggi dengan mempertimbangkan parameter seperti umur (bulan), jenis kelamin, dan tinggi badan (cm). Hasil klasifikasi berhasil mendiagnosis 15 balita berdasarkan atribut yang diberikan, dan mengindikasikan bahwa balita dapat dikategorikan sebagai *stunting* berdasarkan parameter tinggi badan. Hal ini mencerminkan bahwa model memiliki performa yang baik dalam mengidentifikasi status gizi serta memberikan kontribusi signifikan dalam mendukung pengambilan keputusan dalam tindakan preventif dan penanggulangan kasus *stunting* pada anak usia balita. Berdasarkan 121000 data yang diperoleh dari Kaggle.com, model berhasil mencapai tingkat *accuracy* sebesar 95.73%. Ini

mengindikasikan bahwa algoritma *Decision Tree* terbukti efektif dan akurat dalam proses identifikasi *stunting*, sehingga dapat menjadi alat bantu yang potensial dalam sistem *monitoring* kesehatan anak baik tingkat individu maupun masyarakat. Sebagai landasan untuk penelitian selanjutnya, disarankan algoritma *Decision Tree* diintegrasikan lebih luas dalam sistem pemantauan gizi di layanan kesehatan untuk mendeteksi dan menangani *stunting* pada balita secara lebih cepat, serta memperhatikan aspek etika terkait dengan pengumpulan dan pemrosesan data sebagai arah penelitian di masa depan. Dengan begitu, penelitian selanjutnya dapat memberikan solusi dan wawasan terkait pencegahan dan penanggulangan *stunting* secara lebih efisien dan berkelanjutan di Indonesia.

DAFTAR PUSTAKA

- Antika, R., Ahmad, R. I., Dikananda, F., Efendi, D. I., & Narasati, R. (2023). Penerapan Algoritma Decision Tree Berbasis Pohon Keputusan Dalam Klasifikasi Penyakit Jantung. *JATI (Jurnal Mahasiswa Teknik Informatika)*, 7(6), 3688-3692. <https://doi.org/https://doi.org/10.36040/jati.v7i6.8264>
- Biyantoro, A. S., & Prasetyo, B. (2024). Penerapan Decision Tree untuk Klasifikasi Status Kesehatan dengan Perbandingan KNN dan Naive Bayes. *Indonesian Journal of Informatic Research and Software Engineering*, 4(1), 47–55.
- Fatmawati, D., Trisnawati, W., Jumaryadi, Y., & Triyono, G. (2023). Klasifikasi Tingkat Kepuasan Penggunaan Layanan Teknologi Informasi Menggunakan Decision Tree. 3(6), 1056–1062. <https://doi.org/10.30865/klik.v3i6.803>
- Finda, S. M., & Utomo, D. W. (2024). Klasifikasi stunting balita menggunakan metode ensemble learning dan random forest. *Jl. Imam Bonjol No, 15(02)*. <https://doi.org/10.35970/infotekmesin.v15i2.2326>
- Fitriah, Imam Riadi, & Herman. (2023). Analisis Data Mining Sistem Inventory Menggunakan Algoritma Apriori. *Decode: Jurnal Pendidikan Teknologi Informasi*, 3(1), 118–129. <https://doi.org/10.51454/decode.v3i1.132>
- Firdaus, A. A., Yudhana, A., & Riadi, I. (2023). Analisis Sentimen Pada Proyeksi Pemilihan Presiden 2024 Menggunakan Metode Support Vector Machine. *Decode: Jurnal Pendidikan Teknologi Informasi*, 3(2), 236-245. <https://doi.org/10.51454/decode.v3i2.172>
- Guo, M., Wang, Y., Yang, Q., Li, R., Zhao, Y., Li, C., Zhu, M., Cui, Y., Jiang, X., Sheng, S., Li, Q., & Gao, R. (2023). Normal Workflow and Key Strategies for Data Cleaning Toward Real-World Data: Viewpoint. *Interactive Journal of Medical Research*, 12, e44310. <https://doi.org/10.2196/44310>
- Hardiani, T., & Putri, R. N. (2024). Implementasi Metode Naïve Bayes Classifier Untuk Klasifikasi Stunting Pada Balita. *Digital Transformation Technology*, 4(1), 621–627. <https://doi.org/10.47709/digitech.v4i1.4481>
- Herdian, C., Kamila, A., & Agung Musa Budidarma, I. G. (2024). Studi Kasus Feature Engineering Untuk Data Teks: Perbandingan Label Encoding dan One-Hot Encoding Pada Metode Linear Regresi. *Technologia : Jurnal Ilmiah*, 15(1), 93. <https://doi.org/10.31602/tji.v15i1.13457>
- Jalil, A., Homaidi, A., & Fatah, Z. (2024). a Implementation of the Support Vector Machine Algorithm for Classifying Stunting Status in Toddlers. 8(3), 2070–2079. <https://doi.org/https://doi.org/10.33379/gtech.v8i3.4811>
- Lonang, S., Yudhana, A., & Biddinika, M. K. (2023). Analisis Komparatif Kinerja Algoritma Machine Learning untuk Deteksi Stunting. 7, 2109–2117. <https://doi.org/10.30865/mib.v7i4.6553>
- Melyani, S., Harahap, S. Z., & Irmayanti, I. (2024). Prediction of Stunting in Toddlers Combining the Naive Bayes Method and the C4. 5 Algorithm. *Sinkron: jurnal dan penelitian teknik informatika*, 8(2), 1160-1168. <https://doi.org/10.33395/v8i2.13651>

- Putri, I. S. I., Pradini, R. S., & Anshori, M. (2024). Decision Tree Regression untuk Prediksi Prevalensi Stunting di Provinsi Nusa Tenggara Timur. *Jurnal Teknologi Informatika dan Komputer*, 10(2), 413-427. <https://doi.org/https://doi.org/10.37012/jtik.v10i2.2179>
- Radhi, M., Amalia, A., Sitompul, D. R. H., Sinurat, S. H., & Indra, E. (2021). Analisis Big Data Dengan Metode Exploratory Data Analysis (Eda) Dan Metode Visualisasi Menggunakan Jupyter Notebook. *Jurnal Sistem Informasi dan Ilmu Komputer*, 4(2), 23-27. <https://doi.org/https://doi.org/10.34012/jurnalsisteminformasidanilmukomputer.v4i2.2475>
- Raharja, A. R., Pramudianto, A., & Muchsam, Y. (2024). Penerapan algoritma decision tree dalam klasifikasi data 'Framingham' untuk menunjukkan risiko seseorang terkena penyakit jantung dalam 10 tahun mendatang. *Technol. J*, 1(1). <https://doi.org/https://doi.org/10.62872/cwgzp962>
- Ramadhani, F., Satria, A., & Sari, I. P. (2023). Implementasi Metode Fuzzy K-Nearest Neighbor dalam Klasifikasi Penyakit Demam Berdarah. *Hello World Jurnal Ilmu Komputer*, 2(2), 58-62. <https://doi.org/10.56211/helloworld.v2i2.253>
- Samosir, V. B., Widodo, A. M., Anwar, N., Sekti, B. A., & Erzed, N. (2024). Identifikasi Outlier Menggunakan Teknik Data Mining Clustering Untuk Analisis Data Tracer Study Pada Fakultas Ilmu Komputer Universitas Esa Unggul. *IKRA-ITH Informatika: Jurnal Komputer dan Informatika*, 8(1), 162-174. <https://doi.org/10.37817/ikraith-informatika.v8i1>
- Sudrajat, W., & Cholid, I. (2023). K-Nearest Neighbor (K-Nn) Untuk Penanganan Missing Value Pada Data Umkm. *Jurnal Rekayasa Sistem Informasi dan Teknologi*, 1(2), 54-63. <https://doi.org/https://doi.org/10.47065/jieee.v2i3.891>
- Syam, N. T., & Saharuna, Z. (2024). Penerapan Machine Learning Untuk Mengatasi Ketimpangan Data Dalam Menentukan Klasifikasi Uang Kuliah Tunggal (UKT). *Journal of Informatics and Computer Engineering Research*, 1(1), 7-14.
- Syaqhasdy, N. O., Martha, Z., Amalita, N., & Fitria, D. (2023). Classification of Nutrition Problems for Indonesian Toddler with Decision Tree Algorithm C4. 5. *UNP Journal of Statistics and Data Science*, 1(5), 413-419. <https://doi.org/https://doi.org/10.24036/ujsds/vol1-iss5/98>
- World Health Organization. (2024). *Malnutrition*. <https://www.who.int/news-room/fact-sheets/detail/malnutrition>