

Implementasi Algoritma *K-Nearest Neighbor* (KNN) Untuk Klasifikasi Status Gizi Balita

Fadhillah Rashidatul A'la*¹, Ahmad Homaidi², Lukman Fakhid Lidimillah³

^{1,2,3}Universitas Ibrahimy, Situbondo, Jawa Timur, Indonesia

Email: ¹fadhillah5@gmail.com, ²ahmadhomaidi@ibrahimy.ac.id, ³luky.lukman7@gmail.com

*Penulis Korespondensi

Abstrak

Pemantauan status gizi balita merupakan salah satu upaya penting dalam mencegah masalah pertumbuhan seperti kurang gizi, obesitas, maupun stunting. Pada penelitian ini bertujuan untuk mengklasifikasikan status gizi balita dengan algoritma *K-Nearest Neighbor* (KNN) berdasarkan data antropometri seperti umur, jenis kelamin, dan tinggi badan. Implementasi algoritma KNN untuk klasifikasi status gizi balita ini menggunakan data yang diperoleh dari platform *Kaggle* dengan entri berjumlah 120.999. Data tersebut kemudian diolah melalui platform *Google Colab* menggunakan bahasa pemrograman *Python* dan perangkat lunak *RapidMiner* untuk mendukung proses data *preprocessing*, pelatihan, dan evaluasi model. Evaluasi model ini dilakukan untuk menghasilkan metrik evaluasi utama pada setiap kelas seperti *confusion matrix*, presisi, *recall*, dan *F1-Score*. Melalui evaluasi model yang sudah dilakukan, hasil klasifikasi menggunakan algoritma KNN mampu mengklasifikasikan status gizi balita dengan akurasi tinggi, yakni 97%. Penelitian ini menunjukkan bahwa klasifikasi status gizi balita, efektif menggunakan algoritma KNN dan dapat membantu dalam pengambilan keputusan di bidang kesehatan masyarakat.

Kata kunci: Balita, Data Mining, Gizi, *K-Nearest Neighbor*, Klasifikasi

Abstract

Monitoring the nutritional status of toddlers is an important measure in preventing growth problems such as malnutrition, obesity, and stunting. This study aims to classify the nutritional status of infants using the *K-nearest Neighbor* (KNN) algorithm based on anthropometric data such as age, gender, and height. The implementation of the KNN algorithm for classifying the nutritional status of infants uses data obtained from the *Kaggle* platform with 120.999 entries. The data was then processed through the *Google Colab* platform using the *Python* programming language and *RapidMiner* software to support the data preprocessing, training, and model evaluation. The model evaluation was conducted to generate key evaluation metrics for each class, such as *confusion matrix*, precision, recall, and *F1-Score*. Through the model evaluation that has been conducted, the classification results using the KNN algorithm were able to classify the nutritional status of toddlers with high accuracy, namely 97%. This study shows that the classification of toddler nutritional status is effective using the KNN algorithm and can assist in decision-making in the field of public health.

Keywords: Classification, Data Mining, *K-Nearest Neighbor*, Nutrition, Toddlers.

I. PENDAHULUAN

Salah satu langkah penting dalam upaya pencegahan malnutrisi yang dapat mengganggu pertumbuhan anak dan kualitas hidup jangka panjang adalah dengan mendeteksi dini status gizi balita. Di Indonesia, yang masih menjadi permasalahan kesehatan masyarakat itu gizi buruk dan kurang gizi yang signifikan. Selain angka stunting yang tinggi, kasus obesitas dan kelebihan berat badan juga meningkat. Berdasarkan data dari Kementerian Kesehatan, persentase stunting mencapai angka 24,4% pada balita di tahun 2020. Hal ini menunjukkan bahwa satu dari empat anak mengalami gangguan pertumbuhan kronis[1].

Suatu kondisi ketika panjang atau tinggi badan balita tidak sesuai dengan usianya didefinisikan sebagai stunting[2]. Pengukuran antropometri diperlukan untuk mengetahui status gizi anak yang kemudian dibandingkan dengan ketetapan standar pertumbuhan. Maka dari itu, sebelum kondisi gizi berdampak lebih lanjut pada kesehatan anak, deteksi dini sangat penting agar tindakan pencegahan dapat dilakukan.

Teknologi *machine learning* telah digunakan untuk membantu pengambilan keputusan di bidang kesehatan dalam beberapa tahun terakhir, termasuk dalam klasifikasi status gizi[3]. Algoritma *K-Nearest Neighbor* (KNN) yang termasuk algoritma *lazy learning* berbasis jarak merupakan salah satu algoritma yang banyak digunakan. Cara kerja KNN adalah dengan mengklasifikasikan kedekatan antara data baru dengan data yang sudah ada[4].

Algoritma KNN memiliki performa akurasi tinggi berdasarkan beberapa penelitian terdahulu dalam mendeteksi status gizi balita. Sebuah studi melakukan penelitian dengan membandingkan KNN dan Naïve Bayes. Perbandingan dua algoritma tersebut mencapai 96,10% dengan KNN yang memiliki akurasi lebih tinggi. Meski demikian, dalam penerapan KNN masih ditemukan tantangannya, antara lain, ketidakseimbangan data, pemilihan fitur yang relevan, serta sensitivitas terhadap pemilihan parameter k [5].

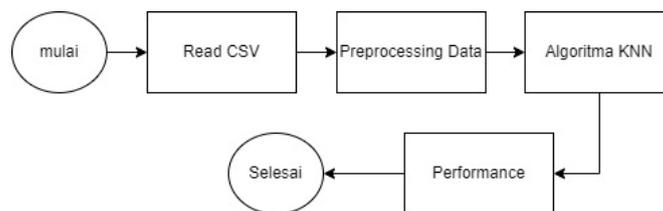
Penelitian ini bertujuan mengimplementasikan algoritma KNN untuk memperluas pemanfaatan teknologi di bidang kesehatan dalam mengklasifikasikan status gizi balita berdasarkan data antropometri yaitu, umur, jenis kelamin, dan tinggi badan. Data yang digunakan diperoleh dari repositori *Stunting Toddler Detection* di platform Kaggle dengan 120.999 entri. Proses pengolahan dan pelatihan model menggunakan RapidMiner untuk visualisasi dan juga menggunakan *Python* pada *Google Colaboratory*. [6]

II. METODE PENELITIAN

Metode penelitian pada dasarnya merupakan pendekatan sistematis untuk mengumpulkan data guna mencapai tujuan tertentu. Proses ilmiah dilakukan dengan tujuan menyelesaikan suatu permasalahan dengan suatu metode yang dirancang secara teliti, terencana, dan sistematis dengan tujuan menemukan fakta atau prinsip yang mendasarinya. Penelitian untuk klasifikasi status gizi balita ini termasuk jenis penelitian kuantitatif dengan pendekatan data mining menggunakan algoritma *K-Nearest Neighbor* (KNN)[4].

Metode Algoritma *K-Nearest Neighbor* (KNN)

Algoritma *K-Nearest Neighbor* (KNN) adalah sebuah metode untuk melakukan klasifikasi terhadap objek berdasarkan data pembelajaran yang jaraknya paling dekat dengan objek tersebut [7]. Penelitian ini dilakukan melalui beberapa tahapan utama untuk mengimplementasikan performa algoritma KNN dalam mengklasifikasikan staus gizi balita. Setiap tahapan dirancang secara sistematis agar proses pengujian dapat berjalan dengan optimal. Alur proses penelitian ditunjukkan pada gambar 1 berikut:



Gambar 1. Flowchart Tahapan Penelitian

Gambar tersebut menunjukkan alur kerja penelitian dimulai dengan fitur *Read CSV* sebagai input utama. Kemudian dilanjutkan dengan tahapan *preprocessing* dimana tahap tersebut untuk membersihkan, menyiapkan data, serta membagi data menjadi data *training* dan data *testing*. Setelah data sudah disiapkan, selanjutnya adalah evaluasi algoritma KNN untuk mengklasifikasikan data, kinerja algoritma kemudian diukur melalui metrik performa meliputi akurasi, presisi, *recall*, dan *F1-Score*.

1. Pengumpulan dan Preprocessing Data

Data *Stunting Toddler Detection* dengan jumlah entri 120.999 tersebut didapatkan dari repositori online *Kaggle* yang bisa diakses dengan tautan berikut <https://www.kaggle.com/datasets/rendiputra/stunting-balita-detection-121k-rows/data> dengan format CSV (*Comma Separated Values*) berisi atribut umur, jenis kelamin, tinggi badan, dan status.

Tabel 1. Atribut Pada Data Status Gizi Balita

Sumber: Kaggle – *Stunting Toddler Detection*

No.	Umur	Jenis Kelamin	Tinggi Badan (centimeter)	Status
1	0 Bulan	Laki-laki	44	Normal
2	1 Bulan	Perempuan	63	Tinggi
3	2 Bulan	Laki-laki	54	Stunted
4	3 Bulan	Perempuan	53	Severely Stunted
5	57 Bulan	Laki-laki	118	Normal
6	58 Bulan	Perempuan	123	Tinggi
7	59 Bulan	Laki-laki	98	Stunted

8	60 Bulan	Perempuan	91	Severely Stunted
---	----------	-----------	----	------------------

Langkah pertama setelah data berhasil dikumpulkan adalah *preprocessing* data untuk memastikan kualitas data. Langkah ini mencakup pemeriksaan data kosong, penghapusan data yang tidak lengkap, serta mengubah data kategorikal menjadi data numerikal. Kolom yang dikodekan adalah atribut jenis kelamin dan atribut status. Data numerik dinormalisasi menggunakan metode *Min-Max Scalling* agar skala serupa antara 0-1. Normalisasi ini dilakukan agar penghitungan jarak antar titik dalam algoritma KNN tidak dipengaruhi oleh perbedaan skala antar fitur [8], [9].

2. Pemisahan Data

Setelah data selesai dinormalisasi, langkah selanjutnya adalah pembagian data menjadi dua bagian, yaitu data latih dan data uji. Data latih sebesar 80% dan data uji sebesar 20%. Lalu, dilanjutkan dengan pemisahan data antara fitur (input) dan target (output/label) [10].

3. Implementasi KNN

Setelah data selesai dilatih dan digunakan untuk memprediksi data uji, langkah selanjutnya adalah melakukan implementasi algoritma [11]. Tahapan-tahapan untuk mengimplementasikan algoritma KNN adalah sebagai berikut:

- a. Menentukan Data Uji
- b. Menyusun Data Latih
- c. Menghitung Jarak Euclidean
- d. Menentukan Parameter Nilai K
- e. Prediksi Akhir

III. HASIL DAN PEMBAHASAN

Implementasi Algoritma *K-Nearest Neighbor* (KNN)

Algoritma *K-Nearest Neighbor* (KNN) adalah sebuah metode untuk melakukan klasifikasi terhadap objek berdasarkan data pembelajaran yang jaraknya paling dekat dengan objek tersebut. KNN memiliki beberapa kelebihan, yaitu ketangguhan terhadap training data yang memiliki banyak noise dan efektif apabila training datanya besar. Selain itu, KNN juga merupakan algoritma yang menggunakan seluruh data latih untuk melakukan proses klasifikasi. Proses klasifikasi pada KNN merupakan teknik yang melakukan prediksi tegas pada data uji berdasarkan perbandingan K tetangga terdekat. K adalah jumlah tetangga terdekat yang dilibatkan dan mempunyai pengaruh dalam penentuan hasil prediksi. Dekat atau jauhnya tetangga biasanya dihitung berdasarkan jarak *Euclidean*. Secara umum, nilai K yang tinggi akan mengurangi efek *noise* pada klasifikasi [7].

Setelah menghitung jarak *Euclidean*, langkah selanjutnya adalah menentukan K *neighbors*nya dengan cara mengurutkan nilai dari yang terkecil sampai yang terbesar. Dari K *neighbors* terdekat, tentukan label berdasarkan mayoritas tetanga terdekat untuk mengevaluasi model dari algoritma KNN tersebut. Setelah evaluasi, langkah berikutnya adalah menentukan kelas dataset. Dari seluruh penghitungan tersebut, terbagi menjadi penghitungan dengan *class* label ‘Normal’, ‘Tinggi’, ‘*Stunting*’, dan ‘*Severely Stunting*’. Kemudian, hasil yang paling besar dari perbandingan keempat label tersebut merupakan hasil dari prediksi algoritma *K-Nearest Neighbor* tersebut. Untuk menggambarkan bagaimana algoritma KNN bekerja dalam mengklasifikasikan, berikut adalah penghitungan jarak Euclidean antara satu data uji dengan beberapa data latih yang telah dinormalisasi.

3.1 Menentukan Data Uji

Setelah data dinormalisasi, langkah pertama yang dilakukan adalah memilih satu data uji yang akan diklasifikasikan. Data ini akan dibandingkan jaraknya dengan data latih yang telah tersedia.

Data uji memiliki:

Umur: 24 bulan

Tinggi Badan: 85 cm. Kemudian dilakukan normalisasi dengan rumus Min-Max berikut:

$$x_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}} \quad [2]$$

Dengan nilai berikut:

$$\text{Umur Min: 0. Umur Max: } 60 = \frac{24 - 0}{60 - 0} = 0.4 \quad [3]$$

$$\text{Tinggi Min: 40. Tinggi Max: } 128 = \frac{85 - 40}{88} = 0.511 \quad [4]$$

Sehingga hasil data uji setelah dinormalisasi adalah $x = [0.4, 0.511]$

3.2 Menyusun Data Latih

Langkah berikutnya adalah memilih beberapa data latih yang akan dihitung jaraknya terhadap data uji. Data latih juga harus sudah dinormalisasi. Contoh tiga data latih yang digunakan:

Tabel 2. Data Latih

Data Latih	Umur (bulan)	Tinggi	Umur_norm	Tinggi_norm	Label
y1	30	90	0.5	0.568	Kelas 0 (Normal)
y2	18	80	0.3	0.455	Kelas 2 (Stunted)
y3	24	85	0.4	0.511	Kelas 0 (Normal)

3.3 Menghitung Jarak Euclidean

Jarak antara data uji dan masing-masing data latih dihitung menggunakan rumus *Euclidean Distance* berikut:

$$d(x, y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2} \quad [5]$$

a. Jarak ke y1

Data latih ini memiliki umur dan tinggi sedikit lebih besar dari data uji.

$$d = \sqrt{(0.4 - 0.5)^2 + (0.511 - 0.568)^2} = \sqrt{0.01 + 0.0032} = \sqrt{0.0132} = 0.115 \quad [6]$$

b. Jarak ke y2

Data ini memiliki umur dan tinggi sedikit lebih rendah dibanding data uji.

$$d = \sqrt{(0.4 - 0.3)^2 + (0.511 - 0.455)^2} = \sqrt{0.01 + 0.0031} = \sqrt{0.0131} = 0.114 \quad [7]$$

c. Jarak ke y3

Data ini memiliki nilai yang identik dengan data uji, sehingga jaraknya adalah nol.

$$d = \sqrt{(0.4 - 0.4)^2 + (0.511 - 0.511)^2} = \sqrt{0 + 0} = \sqrt{0} = 0 \quad [8]$$

3.4 Menentukan Parameter K

Setelah seluruh jarak dihitung, langkah berikutnya adalah mengurutkan jarak dari yang terkecil, lalu memilih K data latih terdekat. Jika K = 3, maka tiga jarak terdekat beserta labelnya adalah

Tabel 3. Hasil Penghitungan Data Latih

Data Latih	Jarak	Label
y3	0	Kelas 0 (Normal)
y2	0.114	Kelas 2 (Stunted)
y1	0.115	Kelas 0 (Normal)

3.5 Prediksi Akhir

Dari tiga tetangga terdekat, akan dilakukan voting untuk menentukan prediksi kelas dari data uji berdasarkan label mayoritas.

Tabel 4. Prediksi Akhir

Label	Jumlah Muncul
Kelas 0 (Normal)	2 kali
Kelas 2 (Stunted)	1 kali

Maka, hasil prediksi adalah **Normal** (kelas 0)

3.6 Evaluasi Model

Setelah proses klasifikasi dilakukan menggunakan algoritma K-Nearest Neighbor (KNN), langkah selanjutnya adalah mengevaluasi seberapa baik model dalam memprediksi status gizi balita. Evaluasi ini dilakukan untuk mengetahui keakuratan dan keandalan model dalam konteks data nyata.

Penilaian performa dilakukan dengan menggunakan beberapa metrik evaluasi standar dalam klasifikasi, yaitu akurasi, precision, recall, dan *F1-Score* yang masing-masing evaluasi memberikan gambaran dari sudut pandang berbeda terhadap kinerja model. Selain itu, digunakan juga confusion

matrix untuk melihat distribusi kesalahan prediksi secara lebih detail antar kelas. Berikut adalah penjelasan masing-masing metrik yang digunakan dalam penelitian ini.

a. Akurasi

Mengukur proporsi data yang diklasifikasikan dengan benar dari seluruh data uji.

$$Akurasi = \frac{TP+TN}{TP+TN+FP+FN} \quad [9]$$

b. Precision

Mengukur ketepatan prediksi untuk masing-masing kelas, yakni seberapa banyak prediksi positif yang benar.

$$Precision = \frac{TP}{TP+FP} \quad [10]$$

c. Recall

Mengukur sejauh mana model berhasil menangkap semua data yang benar-benar termasuk dalam suatu kelas.

$$Recall = \frac{TP}{TP+FN} \quad [11]$$

d. F1-Score

Merupakan rata-rata harmonis dari *precision* dan *recall*, digunakan saat distribusi kelas tidak seimbang.

$$F1 - Score = 2 * \frac{Precision*Recall}{Precision+Recall} \quad [12]$$

Penelitian ini menggunakan dataset yang berisi atribut antropometri pada status gizi diatas. Dataset tersebut dibagi menjadi data latih sebesar 80% dan data uji sebesar 20%. Dengan menggunakan algoritma KNN yang diimplementasikan menggunakan bahasa pemrograman *Python* pada platform Google Colaboratory. Hasil klasifikasi dari algoritma KNN, di evaluasi menggunakan metrik meliputi akurasi, presisi, recall, dan F1-Score. Tabel 6. Berikut menunjukkan hasil evaluasi berdasarkan confusion matrix yang dihasilkan.

Tabel 6. Hasil Evaluasi Model KNN

Sumber: Hasil Pengujian Metode Menggunakan Google Colab

Kelas	Akurasi	Presisi	Recall	F1-Score
0	0.97	0.98	0.99	0.98
1		0.97	0.97	0.97
2		0.96	0.94	0.95
3		0.97	0.97	0.97

Hasil evaluasi kinerja model menunjukkan bahwa algoritma KNN memiliki performa yang sangat baik dalam mengklasifikasikan status gizi balita. Untuk semua kelas, model mencapai akurasi 97%. Untuk kelas 0 presisinya mencapai 98%, recall 99%, dan F1-Score 98% yang menandakan kemampuan model sangat tinggi dalam mengenali dan memprediksi kelas ini secara tepat. Kelas 1 dan kelas 3 juga menunjukkan kinerja yang seimbang dengan nilai presisi, recall, dan F1-Score masing-masing sebesar 97% yang mencerminkan prediksi konsisten dan akurat. Sementara untuk kelas 2, memiliki kinerja sedikit lebih rendah dibandingkan kelas lainnya dengan presisi 96%, recall 94%, dan F1-Score 95%, yang menunjukkan bahwa masih terdapat beberapa kasus pada kelas 2 yang kurang terdeteksi secara optimal. Secara keseluruhan, nilai metrik yang tinggi pada keempat kelas membuktikan bahwa model KNN efektif dalam tugas klasifikasi status gizi berdasarkan data antropometri.

IV. KESIMPULAN

Berdasarkan hasil evaluasi yang diperoleh, dapat disimpulkan bahwa implementasi algoritma *K-Nearest Neighbor* (KNN) untuk klasifikasi status gizi balita dengan akurasi tinggi sebesar 97%. Kinerja model yang konsisten di tiap kelas membuktikan efektivitas KNN dalam tugas ini. Untuk selanjutnya, penelitian ini

dapat dikembangkan dengan optimasi parameter K, pemilihan fitur otomatis, atau integrasi ke dalam sistem aplikasi kesehatan untuk deteksi dini yang lebih luas dan praktis.

REFERENSI

- [1] UNICEF Indonesia, “Perubahan Iklim dan Gizi di Indonesia,” UNICEF, 25 November 2024 [Online]. Available: <https://www.unicef.org/indonesia/id/gizi/laporan/perubahan-iklim-dan-gizi-di-indonesia> [Diakses: 25 Desember 2024]
- [2] M. Neherta and M. Novita Asri, *Intervensi Pencegahan Stunting (Pendekatan Terpadu untuk Mencegah Gangguan Pertumbuhan pada Anak)*. Jawa Barat: Penerbit Adab, 2023.
- [3] A. Mutoi Siregar and A. Puspabhuana, *DATA MINING (Pengolahan Data Menjadi Informasi dengan RapidMiner)*. Surakarta: CV Kekata Group. Accessed: Jan. 21, 2025. [Online]. Available: https://www.google.co.id/books/edition/DATA_MINING/rTImDwAAQBAJ?
- [4] R. Habibi and R. Aprilian, *ALGORITMA KNN DALAM MEMPREDIKSI CUACA UNTUK MENENTUKAN TANAMAN YANG COCOK UNTUK SEMUA MUSIM*. Bandung: Kreatif Industri Nusantara, 2019. Accessed: Jan. 24, 2025. [Online]. Available: https://www.google.co.id/books/edition/Algoritma_KNN_dalam_memprediksi_cuaca_un/69f9DwAAQBAJ?
- [5] J. Homepage, S. Kenia, P. Loka, and A. Marsal, “MALCOM: Indonesian Journal of Machine Learning and Computer Science Comparison Algorithm of K-Nearest Neighbor and Naïve Bayes Classifier for Classifying Nutritional Status in Toddlers Perbandingan Algoritma K-Nearest Neighbor dan Naïve Bayes Classifier Untuk Klasifikasi Status Gizi Pada Balita,” vol. 3, pp. 8–14, 2023.
- [6] G. Purnama Insany, I. Yustiana, and S. Rahmawati, “Penerapan KNN dan ANN pada Klasifikasi Status Gizi Balita Berdasarkan Indeks Antropometri,” *Jurnal Computer Science and Information Technology (CoSciTech)*, vol. 4, no. 2, pp. 385–393, Aug. 2023.
- [7] J. Indriyanto, *Algoritma K-Nearest Neighbor untuk Prediksi Nasabah Asuransi*. Jawa Tengah: Penerbit NEM, 2021. Accessed: Jun. 11, 2025. [Online]. Available: https://www.google.co.id/books/edition/ALGORITMA_K_NEAREST_NEIGHBOR_UNTUK_PREDI/EE0tEAAAQBAJ?
- [8] S. Maesaroh *et al.*, *Bahasa Pemrograman Python*. Banten: PT Sada Kurnia Pustaka, 2024. Accessed: Apr. 20, 2025. [Online]. Available: https://www.google.co.id/books/edition/Bahasa_Pemrograman_Python/bOIKEQAAQBAJ?
- [9] G. Maulani *et al.*, *Machine Learning*. Jawa Barat: CV. Mega Press Nusantara, 2025. Accessed: Jun. 03, 2025. [Online]. Available: https://www.google.co.id/books/edition/Machine_Learning/RblPEQAAQBAJ?
- [10] A. Maulida Argina, “Penerapan Metode Klasifikasi K-Nearest Neighbor pada Dataset Penderita Penyakit Diabetes,” *Indonesian Journal of Data and Science*, vol. 1, no. 2, p. 2933, Jul. 2020.
- [11] J. Indriyanto, *Algoritma K-Nearest Neighbor untuk Prediksi Nasabah Asuransi*. Jawa Tengah: Penerbit NEM, 2021. Accessed: Jan. 24, 2025. [Online]. Available: https://www.google.co.id/books/edition/ALGORITMA_K_NEAREST_NEIGHBOR_UNTUK_PREDI/EE0tEAAAQBAJ?