

Optimasi Klasifikasi Stunting Balita dengan Teknik Boosting pada Decision Tree

Nanda Tri Hastuti ^{1,*}, Fikri Budiman ¹

¹ Program Studi Teknik Informatika, Universitas Dian Nuswantoro, Indonesia

* Correspondence: 111202113818@mhs.dinus.ac.id

Copyright: © 2024 by the authors

Received: 27 Oktober 2024 | Revised: 2 November 2024 | Accepted: 16 November 2024 | Published: 19 Desember 2024

Abstrak

Stunting yaitu masalah kekurangan gizi pada pertumbuhan balita. Saat ini, masalah gizi masih jadi masalah yang serius sehingga harus diatasi, khususnya gizi pada balita. Dilihat dari angka prevalensi yang ditargetkan (14%) pada tahun 2024 serta seberapa bahayanya stunting di Indonesia maka perlu diatasi untuk masalah stunting ini. Tujuan dari penelitian ini untuk mengoptimasi algoritma *decision tree* pada klasifikasi stunting dengan menggunakan optimasi *teknik boosting*. *Teknik boosting* yang digunakan adalah metode *AdaBoost*, *XgBoost*, dan *gradient boosting*. *Teknik boosting* dipilih karena dapat meningkatkan kinerja pengklasifikasi dengan menggabungkan beberapa model yang dipelajari secara berurutan, sehingga menghasilkan prediksi yang lebih efektif. Penelitian ini menggunakan data balita dari Kaggle yang memiliki jumlah 10.000, 8 atribut, dan 2 kelas. Berdasarkan hasil penelitian ini, optimasi *decision tree* menggunakan metode *xgboost* mendapatkan hasil yang terbaik dengan akurasi 83,8%, presisi 82%, recall 83,8%, dan F1-score 81,2% yang menunjukkan potensi besar dalam meningkatkan klasifikasi stunting balita. *Teknik boosting* menjadi pilihan terbaik dibandingkan dengan teknik-teknik yang lain. Berdasarkan hasil yang didapat dari penelitian ini, *Teknik boosting* dapat memprediksi secara tepat, membuktikan tingkat presisi yang tinggi dalam menangani klasifikasi stunting.

Kata kunci: *adaboost; decision tree; gradient boosting; stunting; xgboost*

Abstract

Malnutrition in the growth of small children is known as stunting. Currently, nutrition is still a serious problem that needs to be addressed, especially the nutrition of children under five. Considering the target prevalence rate (14%) in 2024 and how dangerous stunting is in Indonesia, this stunting problem needs to be addressed. The purpose of this research is to optimize the decision tree algorithm in stunting classification using boosting technique optimization. The boosting techniques used are AdaBoost, XGBoost, and Gradient Boosting methods. The boosting technique was chosen because it can improve classifier performance by combining multiple models that are learned sequentially, resulting in more effective predictions. This research uses infant data from Kaggle, which has a total of 10,000 data points, 8 attributes, and 2 classes. Based on the results of this study, decision tree optimization using the XGBoost method achieved the best results with accuracy of 83.8%, precision of 82%, recall of 83.8%, and F1-score of 81.2%, which shows great potential in improving the classification of stunted infants. The boosting technique is the best choice compared to other techniques. Based on the results of this study, the boosting technique can accurately predict and demonstrate a high level of precision in handling stunting classification.

Keywords: *adaboost; decision tree; gradient boosting; stunting; xgboost*



PENDAHULUAN

Stunting adalah masalah kesehatan serius bagi balita yang mengalami kekurangan gizi yang terus-menerus selama masa tumbuh kembangnya (Masacgi & Rohman, 2023; Matdoan et al., 2022). *Stunting* dapat menyebabkan meninggalnya satu juta anak setiap tahun di seluruh dunia (Muche et al., 2021). Faktor penyebab *stunting* balita adalah kurangnya akses terhadap gizi seimbang yang diberikan selama masa kehamilan dan balita, kebutuhan pengetahuan ibu tentang kesehatan dan gizi selama kehamilan, kurangnya vitamin dan mineral (Damayanti & Jakfar, 2023). Gizi yang kurang pada balita akan mempengaruhi pertumbuhannya, sehingga perlu diperhatikan karena memerlukan gizi yang cukup untuk pertumbuhannya. Kekurangan gizi pada balita dapat menjadi masalah kesehatan yang serius (Anku & Duah, 2024; Bitew et al., 2022). Angka prevalensi *stunting* pada tahun 2020 Kementerian Kesehatan Republik Indonesia adalah 38,9% (Titimeidara & Hadikurniawati, 2021). Pada tahun 2022 angka prevalensi *stunting* di Indonesia mengalami penurunan. Hal ini dibuktikan dalam penelitian (Mulyanto et al., 2024) yang menyatakan Dinas Kesehatan melaporkan hasil Survei Status Kesehatan Indonesia (SGI) pada tahun 2021 dan 2022 prevalensi di Indonesia turun dari 24,4% menjadi 21,6%, perlu usaha setiap tahun guna menguranginya sebesar (3,8%) sehingga target (14%) pada tahun 2024 dapat terealisasi (Wahyuni & Kusumodestoni, 2024).

Saat ini, masalah gizi di Indonesia masih menjadi masalah serius sehingga harus ditangani, khususnya masalah gizi balita. Resiko jangka pendek *stunting* dapat mencakup berkurangnya ketahanan, terhambatnya perkembangan kognitif, serta masalah mental dan resiko jangka panjang seperti bentuk tubuh balita saat mereka dewasa cenderung memiliki bentuk tubuh yang kurang baik (Masacgi & Rohman, 2023). Dilihat dari angka prevalensi *stunting* yang masih perlu diturunkan serta seberapa bahayanya *stunting* di Indonesia maka perlu diatasi untuk masalah *stunting* ini. Oleh karena itu, salah satunya menggunakan teknik *data mining* untuk mendeteksi *stunting* balita dengan menerapkan algoritma *decision tree* yang dioptimasi menggunakan *teknik boosting* guna meningkatkan nilai akurasi.

Data mining merupakan proses komputasi dalam menggali informasi penting dan mengidentifikasi pola-pola tertentu dalam data yang berskala besar (Nabila et al., 2021; Pratama et al., 2023; Setiyani et al., 2020). *Data mining* mempunyai banyak teknik salah satunya klasifikasi (Rahayu et al., 2022; Setiawan & Suryono, 2024), yang merupakan proses untuk mencari sekumpulan model yang mengenali kelas-kelas data (Pramadhana, 2021; Sihombing et al., 2021; Takdirillah, 2020; Uska et al., 2020). Dengan menggunakan teknik klasifikasi ini, data *stunting* dapat diorganisir dengan lebih baik dan diberikan label yang menggambarkan karakteristiknya.

Permasalahan yang sering terjadi pada algoritma klasifikasi *data mining* adalah ketidakseimbangan data (*imbalanced data*), yang disebabkan oleh adanya kelas-kelas yang mempunyai jumlah sampel yang jauh lebih banyak. Dalam penerapan model klasifikasi yang masih menjadi masalah adalah ketidakseimbangan data (Anugrah et al., 2024). Untuk menyeimbangkan data diperlukan sebuah metode salah satunya menggunakan metode *SMOTE*. *SMOTE* merupakan metode yang digunakan untuk menyesuaikan kelas yang jumlahnya tidak sama dengan memanfaatkan *oversampling* (Sholihah & Hermawan, 2023). Penelitian ini, jumlah kelas *stunting* lebih besar daripada jumlah kelas tidak *stunting*.

Pada penelitian yang telah dilakukan oleh temuan sebelumnya telah memanfaatkan pendekatan *data mining* dengan menerapkan algoritma *decision tree* dalam melakukan klasifikasi dengan bantuan salah satu Teknik yang ada di *ensemble learning* untuk melakukan optimasi. Algoritma *decision tree* dipilih karena kemampuannya untuk menghasilkan model yang mudah dipahami dan diinterpretasikan, yang dapat membantu dalam mengidentifikasi variabel-variabel penting yang mempengaruhi status gizi balita (Harelvi et al., 2024)

Metode *Ensemble Learning* yang digunakan pada penelitian ini yaitu teknik *boosting*. Teknik *boosting* yang digunakan yaitu metode *Boosting Adaptif (AdaBoost)*, *Boosting Gradien*

Ekstrem (XGBoost), dan *Gradien Boosting (GB)*. Teknik *boosting* merupakan metode untuk meningkatkan algoritma belajar apapun (Pramadhana, 2021). Teknik *boosting* dipilih karena *boosting* dapat meningkatkan kinerja pengklasifikasi dengan menggabungkan beberapa model yang dipelajari secara berurutan, sehingga menghasilkan prediksi yang lebih akurat (Finda & Utomo, 2024). Dengan menggunakan optimasi Teknik *boosting* dalam algoritma *decision tree* dapat meningkatkan akurasi yang diperoleh dari metode tersebut.

Penelitian sebelumnya telah juga dilakukan dengan berbagai algoritma klasifikasi *stunting* salah satunya menggunakan algoritma *naïve bayes* mendapatkan akurasi 80% (Hardiani & Putri, 2024). Penelitian yang dilakukan oleh (Hanif & Muntiari, 2024) melakukan deteksi *stunting* pada balita mendapatkan akurasi sebesar 48% pada *naïve bayes*, 95% untuk *SVM*, dan 99% untuk *Decision Tree*. Penelitian lain dengan membandingkan algoritma *Decision Tree* dan *SVM* oleh Amirudin & Wowor (2023) yang menghasilkan akurasi 83% untuk *SVM* dan 78% untuk *decision tree* dalam klasifikasi balita beresiko *stunting*. Penelitian lainnya menggunakan data sebanyak 22.855 data menggunakan algoritma *naïve bayes* oleh Herliansyah et al. (2021) untuk prediksi *stunting* pada balita menghasilkan akurasi sebesar 64.02%. Namun beberapa temuan mereka belum menerapkan optimasi dengan *XGboost* untuk meningkatkan tingkat akurasi dalam melakukan klasifikasi untuk mendeteksi dini *stunting* pada balita.

Penelitian ini bertujuan untuk melakukan optimasi akurasi dengan menggunakan teknik *boosting* pada algoritma *decision tree* dalam pengklasifikasian *stunting* balita sehingga hasil akurasi meningkat. Penelitian ini akan dilakukan empat percobaan, yakni mengklasifikasikan algoritma *decision tree* tidak menggunakan teknik *boosting*, klasifikasi algoritma *decision tree* dioptimasi dengan *AdaBoost*, klasifikasi algoritma *decision tree* dioptimasi dengan *xgboost*, dan klasifikasi algoritma *decision tree* dioptimasi dengan metode *GB*. Penelitian ini penting dilakukan karena dapat mendorong ketepatan klasifikasi *stunting* balita sehingga bisa memberikan bantuan mencegahnya sejak dini.

METODE

Penelitian ini menggunakan pendekatan data mining menggunakan algoritma *decision tree* dalam mengklasifikasi *stunting* pada balita untuk mencapai hasil yang akurat. Alur tahapan penelitian ini yang pertama yaitu mengumpulkan data, selanjutnya *preprocessing* data, kemudian implementasi algoritma *decision tree* dan teknik *boosting*, dan yang terakhir evaluasi hasil. Pada tahapan mengumpulkan data, data memiliki jumlah 10.000 yang diambil pada laman <https://www.kaggle.com/>. Data yang akan digunakan dalam pengumpulan data terdiri dari 8 variabel dan 2 kelas. Data 8 variabel yang terdiri dari *gender*, *age*, *birth weight*, *birth length*, *body weight*, *body length*, *breastfeeding*, dan *stunting*. Setelah pengumpulan data selanjutnya *preprocessing* data, yang dilakukan untuk mempersiapkan data sebelum dilakukan klasifikasi. Tahapan dalam *preprocessing* data meliputi seleksi data untuk pengambilan variabel yang akan digunakan, pembersihan data untuk memastikan tidak ada data kosong dan menghapus data *duplicate*, transformasi data untuk mengkonversi type data teks ke numerik, dan *balancing* data untuk menyeimbangkan data dengan metode *oversampling* menggunakan teknik *SMOTE*.

Decision tree digunakan untuk menangani tipe data numerik, memproses variabel yang hilang, dan memperoleh aturan yang mudah dimengerti. Pada penelitian ini tahap klasifikasi *decision tree* yang pertama menghitung *Entropy* dan selanjutnya menghitung *gain*. Nilai *Entropy* dihitung dengan menggunakan persamaan (1) dan nilai *Gain* dihitung dengan menggunakan persamaan (2).

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

$$Gain(S.A) = \sum_{k=1}^n \frac{S_i}{S} * Entropy(S_i) \quad (2)$$

Pada *teknik boosting* mempunyai metode seperti *AdaBoost*, *XGBoost*, dan metode GB. *AdaBoost* merupakan *ensemble learning* yang sering digunakan pada metode *boosting*. Kelebihan dari metode *AdaBoost* adalah metode *AdaBoost* bisa menyesuaikan bobot data berdasarkan kesalahan yang dibuat oleh model sebelumnya. Model yang lebih baik akan diberikan bobot lebih tinggi, sementara model yang kurang baik akan diberi bobot lebih rendah.

XGBoost merupakan metode *machine learning* yang dapat berupa algoritma *regresi* serta klasifikasi dengan algoritma *ensemble* yang merupakan variasi dari algoritma *tree gradient boosting* yang dibuat 10 kali lebih cepat daripada *teknik boosting* lainnya. Kelebihan metode *XGBoost* adalah dapat menangani data besar dengan baik, menjadikannya pilihan populer dalam kompetisi data sains.

Metode GB sebanding dengan *AdaBoost* karena merupakan strategi pelatihan yang berurutan. Perbedaan antara *AdaBoost* dan GB adalah bahwa GB tidak memberikan bobot lebih pada hal-hal yang diklasifikasikan secara tidak akurat. Kelebihan GB mampu membangun model secara bertahap dengan mengurangi kesalahan residu dari model sebelumnya menggunakan metode optimasi berbasis gradien, sehingga memberikan performa yang kuat pada data yang beragam dan bisa disesuaikan dengan baik.

Confusion matrix merupakan cara untuk evaluasi model klasifikasi. *Confusion matrix* dimanfaatkan untuk mengukur sebuah algoritma dalam pembelajaran mesin, untuk menghitung sejauh mana algoritma tersebut bisa memprediksi target kelas secara akurat. Meliputi perhitungan *True Positives* (TP), *True Negatives* (TN), *False Positives* (FP), dan *False Negatives* (FN) untuk memperoleh nilai akurasi, yang membantu menganalisis sejauh mana metode memprediksi secara akurat dan mengenali kesalahan klasifikasi.

HASIL DAN PEMBAHASAN

Hasil

Data *stunting* dari Kaggle yang berjumlah 10.000 dengan kategori *stunting* berjumlah 7955 dan kategori tidak *stunting* 2045. Dalam data *gender* jumlah *gender male* 6204 dan *gender female* 3796. Pada data ini atribut *stunting* difokuskan dengan status Yes atau No. Kemudian untuk atribut *breastfeeding* difokuskan dengan atribut No. Gambar 1 merupakan data *stunting* yang akan digunakan.

| | Gender | Age | Birth Weight | Birth Length | Body Weight | Body Length | Breastfeeding | Stunting |
|------|--------|-----|--------------|--------------|-------------|-------------|---------------|----------|
| 0 | Male | 17 | 3.0 | 49 | 10.0 | 72.2 | No | No |
| 1 | Female | 11 | 2.9 | 49 | 2.9 | 65.0 | No | Yes |
| 2 | Male | 16 | 2.9 | 49 | 8.5 | 72.2 | No | Yes |
| 3 | Male | 31 | 2.8 | 49 | 6.4 | 63.0 | No | Yes |
| 4 | Male | 15 | 3.1 | 49 | 10.5 | 49.0 | No | Yes |
| ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 9995 | Male | 15 | 3.0 | 49 | 9.0 | 63.0 | No | Yes |
| 9996 | Female | 12 | 2.8 | 48 | 7.7 | 63.0 | No | No |
| 9997 | Male | 16 | 2.8 | 49 | 7.7 | 49.0 | No | No |
| 9998 | Male | 14 | 2.8 | 49 | 10.0 | 69.0 | No | Yes |
| 9999 | Female | 10 | 3.0 | 49 | 7.7 | 80.0 | No | Yes |

10000 rows × 8 columns

Gambar 1. Data *stunting* balita

Pada tahap *preprocessing* data dilakukan seleksi, pembersihan, transformasi, dan *balancing* data. Pada bagian seleksi data, semua variabel akan digunakan kecuali variabel

breastfeeding, karena kurang relevan dan tidak mempunyai keterkaitan langsung dengan penelitian ini. Kemudian pembersihan data, dalam penelitian ini akan membersihkan data duplicate atau data yang kosong yang kemudian akan dihapus dan tidak digunakan dalam penelitian ini seperti yang nampak pada gambar 2.

All Duplicate Rows:

| | Gender | Age | Birth Weight | Birth Length | Body Weight | Body Length | Stunting |
|------|--------|-----|--------------|--------------|-------------|-------------|----------|
| 5 | Female | 11 | 2.8 | 49 | 8.5 | 65.0 | No |
| 9 | Female | 16 | 2.8 | 49 | 8.5 | 65.0 | Yes |
| 11 | Male | 13 | 2.9 | 50 | 10.0 | 69.0 | Yes |
| 15 | Female | 13 | 2.8 | 49 | 10.5 | 72.2 | Yes |
| 16 | Male | 7 | 2.3 | 50 | 6.4 | 68.3 | No |
| ... | ... | ... | ... | ... | ... | ... | ... |
| 9985 | Male | 7 | 2.3 | 50 | 6.4 | 68.3 | No |
| 9989 | Female | 9 | 3.0 | 49 | 7.1 | 65.0 | Yes |
| 9993 | Male | 12 | 2.3 | 49 | 9.0 | 69.0 | Yes |
| 9995 | Male | 15 | 3.0 | 49 | 9.0 | 63.0 | Yes |
| 9998 | Male | 14 | 2.8 | 49 | 10.0 | 69.0 | Yes |

3816 rows × 7 columns

Gambar 2. Data *duplicate*

Langkah selanjutnya yaitu transformasi data, berdasarkan tabel 1 ada data teks pada kelas *gender* dan kelas *stunting*. Dengan mengubah data teks pada kelas *gender* dengan angka 1 untuk *gender male* dan angka 0 untuk *gender female*, dan pada kelas *stunting* mengubah menjadi angka 1 untuk Yes dan angka 0 untuk No. Hasil setelah dilakukan normalisasi terdapat pada tabel 2.

Tabel 1. Data sebelum dinormalisasi

| <i>Gender</i> | <i>Age</i> | <i>Birth Weight</i> | <i>Birth Length</i> | <i>Body Weight</i> | <i>Body Length</i> | <i>Stunting</i> |
|---------------|------------|---------------------|---------------------|--------------------|--------------------|-----------------|
| <i>Male</i> | 17 | 3.0 | 49 | 10.0 | 72.2 | <i>No</i> |
| <i>Female</i> | 11 | 2.9 | 49 | 2.9 | 65.0 | <i>Yes</i> |

Tabel 2. Data setelah dinormalisasi

| <i>Gender</i> | <i>Age</i> | <i>Birth Weight</i> | <i>Birth Length</i> | <i>Body Weight</i> | <i>Body Length</i> | <i>Stunting</i> |
|---------------|------------|---------------------|---------------------|--------------------|--------------------|-----------------|
| 1 | 17 | 3.0 | 49 | 10.0 | 72.2 | 0 |
| 0 | 11 | 2.9 | 49 | 2.9 | 65.0 | 1 |

Tabel 3. Sebelum dan sesudah *balancing* data

| | <i>Stunting(1)</i> | <i>Tidak Stunting(0)</i> |
|--------------------------------|--------------------|--------------------------|
| Sebelum <i>class balancing</i> | 7955 | 2045 |
| Setelah <i>class balancing</i> | 7955 | 7955 |

Balancing data diperlukan jika jumlah data kelas tidak seimbang. Data dalam penelitian tidak seimbang, sehingga menggunakan (SMOTE) untuk menambahkan jumlah data pada data kategori tidak *stunting* (0), kemudian banyaknya data sama dengan data kategori *stunting* (1). SMOTE lebih efektif karena, bekerja dengan membuat data buatan pada kelas minoritas daripada hanya menduplikasi data asli, sehingga dapat membantu model belajar lebih baik terhadap pola-pola pada kelas minoritas. *Balancing* data dimanfaatkan untuk menangani masalah ketidakseimbangan dalam jumlah pengujian antara kelas yang berbeda, untuk meningkatkan metode pembelajaran mesin dengan memperlakukan semua kelas setara, serta menghindari *overfitting* pada prediksi yang muncul. Hal ini algoritma menjadi lebih tepat

dalam memprediksi setiap kelas dalam kumpulan data seperti hasil sebelum dan sesudah balancing data.

Selanjutnya, setelah tahap *balancing* data maka harus melakukan proses *standar scaler* agar tidak mempengaruhi perhitungan dan merusak kinerja mesin menjadi lebih berat. Tujuan dari *standard scaler* adalah untuk menstandarkan fitur-fitur dalam dataset sehingga memiliki skala yang seragam, yaitu rata-rata nol dan standar deviasi satu. Hasil yang sesudah dan sebelum dilakukan *standard scaler* terdapat pada tabel 4.

Tabel 4. Hasil *standard scaler*

| Variabel | Data Sebelum | Data Sesudah |
|---------------------|--------------|--------------|
| <i>Gender</i> | 0 | -1.24768523 |
| <i>Age</i> | 11 | -0.45304097 |
| <i>Birth Weight</i> | 2.9 | 0.46284728 |
| <i>Birth Length</i> | 49 | -0.1940576 |
| <i>Body Weight</i> | 2.9 | -2.6841891 |
| <i>Body Length</i> | 65.0 | -0.43828426 |

Tabel 5. Hasil akurasi

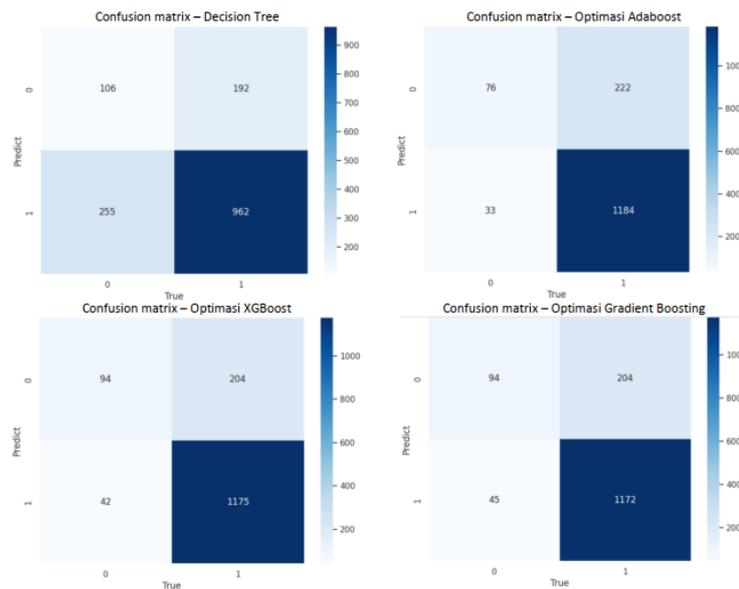
| Algoritma | Akurasi | Presisi | Recall | F1-score |
|-----------------------------------|---------|---------|--------|----------|
| <i>Decision Tree</i> | 70.5% | 72.7% | 70.5% | 71.5% |
| <i>Optimasi Adaboost</i> | 83.2% | 81.4% | 83.2% | 79.9% |
| <i>Optimasi XGBoost</i> | 83.8% | 82% | 83.8% | 81.2% |
| <i>Optimasi Gradient Boosting</i> | 83.6% | 81.7% | 83.6% | 81.1% |

Hasil pada tabel 5 menunjukkan hasil pengujian memiliki akurasi yang berbeda. Algoritma *decision tree* mempunyai hasil pengujian yang lebih rendah daripada yang dioptimasi menggunakan *teknik boosting*. Metode *XGBoost* memberikan hasil akurasi yang terbaik karena menerapkan berbagai optimasi yang meningkatkan performa, akurasi, dan efisiensi, selain itu mampu meminimalkan *overfitting* melalui regularisasi, yang jarang ada pada metode *boosting* lainnya.

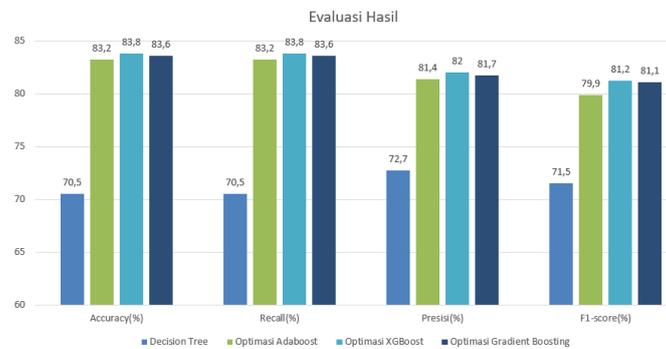
Evaluasi hasil pengujian dengan memanfaatkan *confusion matrix* dari algoritma *decision tree* serta *teknik boosting* untuk menghitung sejauh mana algoritma tersebut bisa memprediksi target kelas secara akurat. Dengan adanya *confusion matrix* dapat mengevaluasi hasil pengujian dengan memperkirakan seberapa model bisa memberikan hasil prediksi akurasi yang tepat dan akurat.

Selanjutnya hasil pada gambar 1 menampilkan evaluasi dari pemrosesan data dengan menggunakan *confusion matrix*. Pada *confusion matrix decision tree* kolom TP menunjukkan nilai 106, TN menunjukkan nilai 962, sedangkan FP memperoleh 192, serta FN memperoleh 255. Pada *confusion matrix optimasi Adaboost* kolom TP memperoleh nilai 76, TN memperoleh nilai 1184, FP memperoleh 222, dan FN memperoleh 33. Pada *confusion matrix optimasi XGBoost* kolom TP menunjukkan nilai 94, TN menunjukkan nilai 1175, FP memperoleh 204, dan FN memperoleh 42. Pada *confusion matrix optimasi Gradient Boosting* kolom TP menunjukkan nilai 94, TN menunjukkan nilai 1172, FP memperoleh 204, dan FN memperoleh 45.

Sementara itu, hasil yang disajikan pada gambar 2 merupakan diagram perbandingan akan mengevaluasi model yang berbeda yaitu *decision tree*, optimasi dengan *AdaBoost*, optimasi dengan *XGBoost*, dan optimasi dengan GB. Berdasarkan hasil perbandingan yang membuktikan bahwa optimasi *decision tree* menggunakan *XGBoost* dapat memberikan solusi dalam meningkatkan ketepatan klasifikasi secara efektif.



Gambar 1. Confusion matrix



Gambar 2. Perbandingan akurasi, presisi, recall, dan F1-score

Pembahasan

Berdasarkan hasil penelitian ini algoritma *decision tree* menggunakan optimasi *teknik boosting* mendapatkan akurasi yang berbeda-beda. Optimasi *decision tree* dengan *XGBoost* menghasilkan nilai akurasi tertinggi yaitu 83.8%. Dapat disimpulkan bahwa optimasi *decision tree* menggunakan *teknik boosting* sangat berpengaruh karena dapat meningkatkan nilai akurasi dibandingkan dengan algoritma *decision tree* tanpa optimasi. Berdasarkan hasil penelitian yang membuktikan bahwa optimasi *decision tree* menggunakan *XGBoost* dapat memberikan solusi dalam meningkatkan ketepatan klasifikasi secara efektif. Optimasi *decision tree* dengan *XGBoost* memberikan hasil yang efektif dibandingkan *teknik boosting* lainnya karena menerapkan berbagai optimasi yang meningkatkan performa, akurasi, dan efisiensi, selain itu mampu meminimalkan *overfitting* melalui regularisasi, yang jarang ada pada metode *boosting* yang lain.

Berdasarkan hasil yang diperoleh, dengan menerapkan optimasi menggunakan *teknik boosting* mampu meningkatkan hasil klasifikasi. Penelitian ini menggunakan *decision tree* dan optimasi menggunakan *Teknik Boosting* yang merupakan penelitian yang belum dilakukan oleh Amirudin & Wowor (2023). Penelitian sebelumnya oleh Amirudin & Wowor (2023) guna mengklasifikasi stunting memperoleh akurasi 78% yang sama-sama menggunakan algoritma *Decision Tree* tetapi tidak dioptimasi.

Meskipun pada penelitian sebelumnya sudah banyak mengklasifikasikan stunting balita dengan berbagai algoritma dan mendapatkan akurasi yang baik, tetapi belum ada penelitian

yang menerapkan algoritma *decision tree* yang dioptimasi menggunakan *teknik boosting*. Oleh karena itu, penelitian ini menggunakan algoritma *decision tree* yang dioptimasi menggunakan *teknik boosting* guna meningkatkan nilai akurasi. Optimasi dengan *teknik boosting* dapat meningkatkan hasil klasifikasi, maka dengan mendapatkan hasil lebih tinggi dapat mengklasifikasi data stunting secara efektif dan akurat. Dalam pencegahan stunting hal ini berdampak positif, karena metode yang tepat dapat memberikan bantuan untuk mengidentifikasi stunting balita secara lebih pasti. Dengan cara ini, upaya pencegahan yang tepat bisa dilakukan dengan lebih berhasil, serta memberikan pengaruh yang jauh lebih baik dalam mengatasi *stunting* pada balita.

SIMPULAN

Berdasarkan hasil pengujian yang diperoleh dari penelitian ini terlihat bahwa penerapan *teknik boosting* memiliki peluang dalam mengatasi masalah klasifikasi dalam dataset ini. Metode *decision tree* yang dioptimasi menggunakan *XGBoost* memberikan akurasi tertinggi sebesar 83,8%, presisi 82%, *recall* 83,8%, dan *F1-score* 81,2% yang menunjukkan potensi besar dalam meningkatkan klasifikasi stunting balita. Hal ini dapat menunjukkan bahwa dengan menggunakan *teknik boosting* dapat secara signifikan meningkatkan ketepatan hasil prediksi dalam klasifikasi *stunting* balita dengan hasil yang lebih tepat. Pada penelitian selanjutnya diharapkan untuk menguji pendekatan lain seperti *random forest* atau *teknik neural networks* untuk menilai kinerja yang lebih baik.

REFERENSI

- Amirudin, M., & Wowor, A. D. (2023). Analisis Perbandingan Klasifikasi Balita Beresiko Stunting Menggunakan Metode Support Vector Machine dan Decision Tree. *Centive*, 3(1), 581–591.
- Anku, E. K., & Duah, H. O. (2024). Predicting and identifying factors associated with undernutrition among children under five years in Ghana using machine learning algorithms. *PLoS ONE*, 19(2), 1–16. <https://doi.org/10.1371/journal.pone.0296625>
- Anugrah, M. I., Zeniarja, J., & Setiawan, D. S. (2024). Peningkatan Performa Model Hard Voting Classifier dengan Teknik Oversampling ADASYN pada Penyakit Diabetes. *Edumatic: Jurnal Pendidikan Informatika*, 8(1), 290–299. <https://doi.org/10.29408/edumatic.v8i1.25838>
- Bitew, F. H., Sparks, C. S., & Nyarko, S. H. (2022). Machine learning algorithms for predicting undernutrition among under-five children in Ethiopia. *Public Health Nutrition*, 25(2), 269–280. <https://doi.org/10.1017/S1368980021004262>
- Damayanti, D. K. D., & Jakfar, M. (2023). Klasifikasi Status Stunting Balita Menggunakan Algoritma Fuzzy C-Means (Studi Kasus Posyandu Rw 01 Kelurahan Jepara Surabaya). *MATHunesa: Jurnal Ilmiah Matematika*, 11(3), 533–542. <https://doi.org/10.26740/mathunesa.v11n03.p524-533>
- Finda, S. M., & Utomo, D. W. (2024). Klasifikasi Stunting Balita menggunakan Metode Ensemble Learning dan Random Forest. *Infotekmesin*, 15(2), 287–295.
- Hanif, K. H., & Muntiari, N. R. (2024). Penerapan Algoritma Decision Tree, Svm, Naive Bayes Dalam Deteksi Stunting Pada Balita. *METHOMIKA Jurnal Manajemen Informatika Dan Komputerisasi Akuntansi*, 8(1), 105–109. <https://doi.org/10.46880/jmika.vol8no1.pp105-109>
- 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>
- Harelvi, D. A., Salma, A., Kurniawati, Y., & Fitri, F. (2024). *The Comparison of C4 . 5 and C5 . 0 Algorithms in Classifying the Nutritional Status of Stunted Toddlers*. 2, 213–218.

- Herliansyah, V., Latuconsina, R., & Dinimaharawati, A. (2021). Prediksi Stunting Pada Balita Dengan Menggunakan Algoritma Klasifikasi Naïve-bayes. *eProceedings of Engineering*, 8(5), 6642–6649.
- Masacgi, G. N., & Rohman, M. S. (2023). Optimasi Model Algoritma Klasifikasi menggunakan Metode Bagging pada Stunting Balita. *Edumatic: Jurnal Pendidikan Informatika*, 7(2), 455–464. <https://doi.org/10.29408/edumatic.v7i2.23812>
- Matdoan, M. Y., Matdoan, U. A., & Saleh Far-Far, M. (2022). Algoritma K-Means Untuk Klasifikasi Provinsi di Indonesia Berdasarkan Paket Pelayanan Stunting. *PANRITA Journal of Science, Technology, and Arts*, 1(2), 41–46.
- Muche, A., Gezie, L. D., Baraki, A. G. egzabher, & Amsalu, E. T. (2021). Predictors of stunting among children age 6–59 months in Ethiopia using Bayesian multi-level analysis. *Scientific Reports*, 11(1), 1–12. <https://doi.org/10.1038/s41598-021-82755-7>
- Mulyanto, Y., Idifitriani, F., & Wati, A. (2024). Klasifikasi Data Mining Untuk Penentuan Stunting Pada Balita Menggunakan Metoden Naïve Bayes. *Jurnal Mnemonic*, 7(2), 129–135. <https://doi.org/10.36040/mnemonic.v7i2.8849>
- Nabila, Z., Rahman Isnain, A., & Abidin, Z. (2021). Analisis Data Mining Untuk Clustering Kasus Covid-19 Di Provinsi Lampung Dengan Algoritma K-Means. *Jurnal Teknologi Dan Sistem Informasi (JTSI)*, 2(2), 100.
- Pramadhana, D. (2021). Klasifikasi Penyakit Diabetes Menggunakan Metode CFS dan ROS dengan Algoritma J48 Berbasis Adaboost. *Edumatic: Jurnal Pendidikan Informatika*, 5(1), 89–98. <https://doi.org/10.29408/edumatic.v5i1.3336>
- Pratama, Y. A., Budiman, F., Winarno, S., & Kurniawan, D. (2023). Analisis Optimasi Algoritma Decision Tree, Logistic Regression dan SVM Menggunakan Soft Voting. *Jurnal Media Informatika Budidarma*, 7(4), 1908-1919. <https://doi.org/10.36040/mnemonic.v7i2.8849>
- Rahayu, S., Yumarlin, M. Z., Bororing, J. E., & Hadiyat, R. (2022). Implementasi Metode K-Nearest Neighbor (K-NN) untuk Analisis Sentimen Kepuasan Pengguna Aplikasi Teknologi Finansial FLIP. *Edumatic: Jurnal Pendidikan Informatika*, 6(1), 98-106. <https://doi.org/10.29408/edumatic.v6i1.5433>
- Setiawan, A., & Suryono, R. R. (2024). Analisis Sentimen Ibu Kota Nusantara menggunakan Algoritma Support Vector Machine dan Naïve Bayes. *Edumatic: Jurnal Pendidikan Informatika*, 8(1), 183-192.
- Setiyani, L., Wahidin, M., Awaludin, D., & Purwani, S. (2020). Analisis Prediksi Kelulusan Mahasiswa Tepat Waktu Menggunakan Metode Data Mining Naïve Bayes: Systematic Review. *Faktor Exacta*, 13(1), 35-43. <https://doi.org/10.30998/faktorexacta.v13i1.5548>
- Sholihah, N. N., & Hermawan, A. (2023). Implementation of Random Forest and Smote Methods for Economic Status Classification in Cirebon City. *Jurnal Teknik Informatika (Jutif)*, 4(6), 1387–1397. <https://doi.org/10.52436/1.jutif.2023.4.6.1135>
- Sihombing, L. O., Hannie, H., & Dermawan, B. A. (2021). Sentimen Analisis Customer Review Produk Shopee Indonesia Menggunakan Algoritma Naïve Bayes Classifier. *Edumatic: Jurnal Pendidikan Informatika*, 5(2), 233-242. <https://doi.org/10.29408/edumatic.v5i2.4089>
- Takdirillah, R. (2020). Penerapan Data Mining Menggunakan Algoritma Apriori Terhadap Data Transaksi Penjualan Bisnis Ritel. *Edumatic: Jurnal Pendidikan Informatika*, 4(1), 37-46. <https://doi.org/10.29408/edumatic.v4i1.2081>
- Titimeidara, M. Y., & Hadikurniawati, W. (2021). Implementasi Metode Naïve Bayes Classifier Untuk Klasifikasi Status Gizi Stunting Pada Balita. *Jurnal Ilmiah Informatika*, 9(01), 54–59. <https://doi.org/10.33884/jif.v9i01.3741>
- Uska, M. Z., Wirasasmita, R. H., Usuluddin, U., & Arianti, B. D. D. (2020). Evaluation of rapidminer-application in data mining learning using persiva model. *Edumatic: Jurnal*

Pendidikan Informatika, 4(2), 164-171. <https://doi.org/10.29408/edumatic.v4i2.2688>
Wahyuni, S. D., & Kusumodestoni, R. H. (2024). Optimalisasi Algoritma Support Vector Machine (SVM) Dalam Klasifikasi Kejadian Data Stunting. *Bulletin of Information Technology (BIT)*, 5(2), 56–64.