

Analisis Perbandingan Metode *AdaBoost*, *Gradient Boosting*, dan *XGBoost* Untuk Kalsifikasi Status Gizi Pada Balita

Moh. Erkamim^{1*}, Adam M Tanniewa², Irfan AP², Nurhayati³

¹ Program Studi Sistem Informasi Kota Cerdas, Universitas Tunas Pembangunan, Surakarta, Indonesia

² Program Studi Teknik Informatika, Fakultas Teknik, Universitas Sulawesi Barat, Majene, Indonesia

³ Program Studi Teknik Informatika, Fakultas Teknik, Universitas Muhammadiyah Tangerang, Tangerang, Indonesia

Email: ^{1*}erkamim@lecture.utp.ac.id, ²adamtanniewa76@gmail.com, ³ipalaloi@gmail.com, ⁴nurhayati09011@ft-umt.ac.id

Email Penulis Korespondensi: erkamim@lecture.utp.ac.id

Submitted: 31/07/2024; Accepted: 24/12/2024; Published: 25/12/2024

Abstrak—Masalah gizi pada balita merupakan isu krusial yang berdampak signifikan terhadap kesehatan dan perkembangan anak di Indonesia. Kekurangan gizi dapat mengakibatkan berbagai masalah kesehatan jangka panjang. Oleh karena itu, deteksi dan klasifikasi status gizi pada balita sangat penting. Penelitian ini bertujuan untuk melakukan analisis dan membandingkan teknik *boosting* untuk mengklasifikasikan status gizi balita, dengan fokus pada tiga teknik *boosting*: *AdaBoost*, *Gradient Boosting*, dan *XGBoost*. Hal ini dilakukan karena teknik *boosting* bekerja dengan cara membangun model secara berurutan, di mana setiap model baru berusaha memperbaiki kesalahan prediksi dari model sebelumnya. Hasil penelitian menunjukkan bahwa model *XGBoost* memberikan kinerja terbaik dengan *precision* 0.9849, *recall* 0.9848, *accuracy* 0.9848, *F1 score* 0.9848, dan ROC-AUC 0.9994 pada rasio pembagian data 80:20. Sebaliknya, model *AdaBoost* menunjukkan hasil terendah dengan *precision* 0.6294, *recall* 0.6292, *accuracy* 0.6292, *F1 score* 0.6291, dan ROC-AUC 0.7581 pada rasio pembagian data 90:10, yang disebabkan oleh sensitivitasnya terhadap *outliers* dan *noise* dalam data. Temuan ini menunjukkan bahwa *XGBoost* adalah model *boosting* terbaik untuk klasifikasi status gizi balita, diikuti oleh *Gradient Boosting*, dengan *AdaBoost* berada di posisi terakhir. Kinerja luar biasa dari *XGBoost* disebabkan oleh penggunaan teknik regularisasi, kemampuan menangani *missing values* secara efektif, serta algoritma *boosting* yang efisien dan cepat melalui teknik paralelisasi.

Kata Kunci: Metode *Boosting*; *AdaBoost*; *Gradient Boosting*; *XGBoost*; Klasifikasi Status Gizi

Abstract—Nutritional issues in toddlers are a crucial issue that significantly impacts the health and development of children in Indonesia. Malnutrition can lead to various long-term health problems. Therefore, detecting and classifying the nutritional status of toddlers is very important. This study aims to analyze and compare boosting techniques to classify the nutritional status of toddlers, focusing on three boosting techniques: *AdaBoost*, *Gradient Boosting*, and *XGBoost*. This is done because boosting techniques work by sequentially building models, where each new model attempts to correct the prediction errors of the previous model. The results show that the *XGBoost* model provides the best performance with a precision of 0.9849, recall of 0.9848, accuracy of 0.9848, F1 score of 0.9848, and ROC-AUC of 0.9994 at an 80:20 data split ratio. Conversely, the *AdaBoost* model shows the lowest results with a precision of 0.6294, recall of 0.6292, accuracy of 0.6292, F1 score of 0.6291, and ROC-AUC of 0.7581 at a 90:10 data split ratio, caused by its sensitivity to outliers and noise in the data. These findings indicate that *XGBoost* is the best boosting model for classifying the nutritional status of toddlers, followed by *Gradient Boosting*, with *AdaBoost* in the last position. The outstanding performance of *XGBoost* is due to the use of regularization techniques, effective handling of missing values, and efficient and fast boosting algorithms through parallel processing techniques.

Keywords: Boosting Method; *AdaBoost*; *Gradient Boosting*; *XGBoost*; Nutritional Status Classification

1. PENDAHULUAN

Masalah gizi pada balita merupakan isu krusial yang berdampak signifikan terhadap kesehatan dan perkembangan anak. Di Indonesia, masalah kekurangan gizi pada balita masih menjadi tantangan besar. Menurut data dari Kementerian Kesehatan Indonesia, prevalensi balita dengan gizi buruk dan kurang pada tahun 2021 mencapai 17,7%, sedangkan prevalensi stunting berada di angka 24,4% [1]. Kekurangan gizi pada masa balita dapat mengakibatkan berbagai masalah kesehatan jangka panjang, termasuk stunting, penurunan kemampuan kognitif, dan meningkatnya risiko penyakit kronis di masa dewasa [2][3]. Oleh karena itu, upaya untuk mendeteksi dan mengklasifikasikan status gizi pada balita menjadi sangat penting sebagai langkah awal dalam penanganan masalah ini [4]. Seiring dengan kemajuan teknologi informasi, penggunaan teknik data mining dengan pembelajaran mesin (*machine learning*) dapat menjadi solusi permasalahan ini. Data mining berperan penting dalam ekstraksi informasi berharga dari data data-data yang telah dikumpulkan. Setelah proses data mining, *machine learning* diterapkan untuk membangun model prediktif yang mampu mengklasifikasikan status gizi balita berdasarkan berbagai fitur-fitur yang berkaitan. Namun pemilihan algoritma *machine learning* yang tepat menjadi tantangan besar untuk dapat diselesaikan, karena pemilihan metode yang tepat dapat menghemat sumber daya dan meningkatkan efisiensi program kesehatan yang berfokus pada penanganan masalah gizi balita.

Penelitian terdahulu mengenai klasifikasi status gizi telah dilakukan oleh beberapa peneliti sebelumnya. Terdapat penelitian yang menggunakan algoritma *K-Nearest Neighbor* (KNN) untuk mengklasifikasikan status gizi balita di Puskesmas Sa'dan dengan data yang diambil dari internet mulai tahun 2017-2021 [5]. Hasil penelitian menunjukkan bahwa KNN dapat mengklasifikasikan status gizi balita dengan akurasi sebesar 91,8%. Selain penelitian tersebut terdapat penelitian yang menggunakan pendekatan KNN juga yang digunakan untuk mengklasifikasikan status gizi balita dengan data dari Posyandu Nusa Indah 4 [6]. Dengan menggunakan 516 data *training* dan 100 data *testing*, penelitian ini menghasilkan akurasi sebesar 94,64%, *precision* sebesar 94,64%, dan *recall* sebesar 94,64%. Penelitian

lainnya, menggunakan algoritma yang berbeda yakni *Decision Tree* yang digunakan untuk memprediksi status gizi balita [7]. Model yang dihasilkan mampu memberikan prediksi akurat dengan nilai akurasi sebesar 92,73%. Berikutnya, penelitian yang menggunakan pengembangan dari *Decision Tree* yaitu algoritma C4.5 untuk klasifikasi status gizi balita dengan data dari Puskesmas Janji dan Puskesmas Suka Makmur [8]. Namun, pada penelitian ini tingkat akurasi tidak lebih tinggi dari penelitian sebelumnya yaitu sebesar 80%. Penelitian selanjutnya mengenai penggunaan algoritma *Random Forest* untuk mengklasifikasikan status gizi anak di Kecamatan Nglegok [9]. Menggunakan data antropometri dari 2.542 balita, penelitian ini menghasilkan akurasi sebesar 88,6%, *precision* 88,1%, *recall* 88,6%, dan *F1 score* 88,2%. Dari penelitian yang ada, belum terdapat penelitian yang menggunakan pendekatan yang dapat meningkatkan kinerja model melalui penggabungan model sederhana menjadi satu model yang lebih kuat atau yang disebut dengan teknik *boosting*.

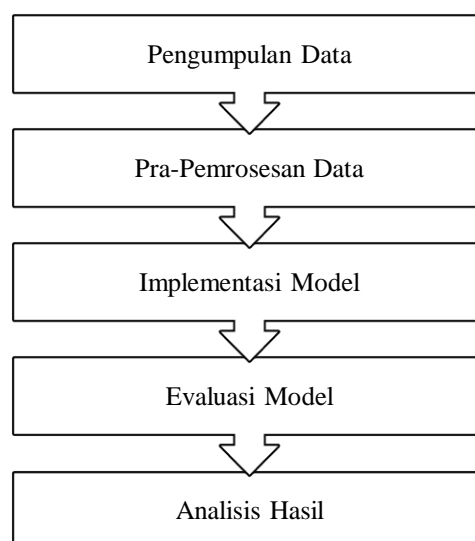
Perbedaan penelitian yang dilakukan dengan penelitian terdahulu yaitu pada penelitian ini mencoba melakukan analisis kinerja teknik *boosting* diantaranya *AdaBoost (Adaptive Boosting)*, *Gradient Boosting*, dan *XGBoost (Extreme Gradient Boosting)* untuk mengklasifikasikan status gizi balita. Hal ini dilakukan karena *boosting* bekerja dengan cara membangun model secara berurutan, di mana setiap model baru berusaha memperbaiki kesalahan prediksi dari model sebelumnya [10]. Algoritma seperti *AdaBoost (Adaptive Boosting)* menyesuaikan bobot dari sampel yang salah klasifikasi sehingga model berikutnya lebih fokus pada kesalahan tersebut [11]. *Gradient Boosting*, di sisi lain, mengoptimalkan fungsi *loss* dengan pendekatan *gradient descent* pada setiap iterasi [12]. *XGBoost (Extreme Gradient Boosting)* adalah versi yang lebih efisien dari *Gradient Boosting*, menggunakan teknik regulasi dan pemrosesan paralel untuk meningkatkan akurasi dan efisiensi [13]. Keunggulan utama *boosting* terletak pada kemampuannya untuk mengurangi bias dan *variance*, serta meningkatkan akurasi model secara keseluruhan, menjadikannya sangat efektif untuk berbagai jenis tugas klasifikasi dan regresi.

Sehingga, yang menjadi tujuan pada penelitian ini yaitu untuk melakukan analisis dan membandingkan teknik *data mining* dan *machine learning* untuk mengklasifikasikan status gizi balita, dengan fokus pada tiga teknik *boosting*: *AdaBoost*, *Gradient Boosting*, dan *XGBoost*. Dengan melakukan perbandingan ini, diharapkan dapat ditemukan metode yang paling efektif dan efisien untuk diterapkan dalam klasifikasi status gizi balita, sehingga dapat mendukung upaya penanganan masalah gizi dengan lebih baik. Hasil dari penelitian berkontribusi dalam bidang kesehatan masyarakat dan menjadi referensi bagi peneliti lain yang ingin mengembangkan metode klasifikasi yang lebih baik di masa depan terutama dalam penggunaan teknik *boosting*.

2. METODOLOGI PENELITIAN

2.1 Tahapan Penelitian

Tahapan penelitian berkaitan dengan langkah-langkah sistematis yang diambil untuk mencapai tujuan penelitian [14]. Penelitian ini bertujuan untuk membandingkan kinerja ketiga metode *boosting* yaitu *AdaBoost*, *Gradient Boosting*, dan *XGBoost* dalam mengklasifikasikan status gizi balita. Sehingga, susunan tahapan penelitian yang digunakan divisualkan pada Gambar 1.



Gambar 1. Langkah-Langkah Penelitian

Gambar 1 menampilkan tahap-tahap pelaksanaan penelitian. Di bawah ini diuraikan penjelasan yang lebih terperinci mengenai langkah-langkah penelitian yang dijalankan:

a. Pengumpulan Data

Langkah pertama dalam penelitian ini adalah pengumpulan data, yang melibatkan pengumpulan dataset yang relevan mengenai status gizi balita. Dataset yang digunakan diambil dari website Kaggle dengan nama "*Stunting Toddler Detection*" [15]. Dataset ini menggunakan rumus *z-score* untuk menentukan status stunting sesuai dengan pedoman dari *World Health Organization* (WHO), dengan fokus pada deteksi stunting pada anak-anak balita. Dataset ini terdiri dari 121.000 baris data yang berisi informasi tentang usia, jenis kelamin, tinggi badan, dan status gizi balita. Status gizi dikategorikan menjadi empat: Normal, *Severely Stunted*, *Stunted*, dan Tinggi. "*Severely Stunted*" menunjukkan kondisi yang sangat serius dengan *z-score* kurang dari -3 *Standard Deviation*, "*Stunted*" menunjukkan kondisi kerdil dengan *z-score* antara -3 dan kurang dari -2 *Standard Deviation*, "Normal" menunjukkan status gizi yang sehat dengan *z-score* antara -2 dan +3 *Standard Deviation*, dan "Tinggi" menunjukkan pertumbuhan di atas rata-rata dengan *z-score* lebih dari +3 *Standard Deviation*.

b. Pra-Pemrosesan Data

Setelah data dikumpulkan, tahap berikutnya adalah pra-pemrosesan data. Tahap ini melibatkan pembersihan data untuk menghilangkan kesalahan dan data yang hilang, serta transformasi data ke dalam format yang sesuai untuk analisis lebih lanjut [16]. Pra-pemrosesan juga termasuk normalisasi atau standarisasi data untuk memastikan semua fitur berada dalam skala yang sama, yang penting untuk meningkatkan kinerja algoritma *boosting*. Proses ini juga dapat melibatkan pengkodean variabel kategorikal dan pembagian data menjadi set pelatihan dan set pengujian.

c. Implementasi Model

Pada tahap pembangunan model, algoritma *AdaBoost*, *Gradient Boosting*, dan *XGBoost* diterapkan pada data yang telah di pra-pemrosesan. Setiap algoritma akan dilatih menggunakan data pelatihan untuk membangun model yang mampu mengklasifikasikan status gizi balita. Tahap ini melibatkan pengaturan parameter model dan pelatihan model dengan teknik pembelajaran yang sesuai. Pembangunan model ini bertujuan untuk menghasilkan model prediksi yang optimal untuk setiap algoritma.

d. Evaluasi Model

Setelah model dibangun, tahap selanjutnya adalah evaluasi model. Pada tahap evaluasi model yang telah dilatih diuji menggunakan set pengujian untuk menilai kinerja dari model yang dibangun [17]. Evaluasi ini melibatkan penggunaan metrik evaluasi seperti akurasi, presisi, *recall*, dan *F1 score* untuk mengukur performa masing-masing model. Selain itu, analisis kurva ROC dan AUC juga dapat digunakan untuk mengevaluasi kemampuan prediksi model secara lebih mendalam. Tujuan dari tahap ini adalah untuk menentukan algoritma mana yang memberikan hasil terbaik dalam klasifikasi status gizi balita.

e. Analisis Hasil

Tahap terakhir adalah analisis hasil, di mana hasil evaluasi model dibandingkan dan dianalisis untuk menarik kesimpulan. Analisis ini mencakup perbandingan kinerja masing-masing algoritma dan diskusi mengenai kelebihan dan kelemahan setiap metode. Hasil analisis ini akan digunakan untuk menentukan metode yang paling efektif untuk klasifikasi status gizi balita dan memberikan rekomendasi untuk implementasi di lapangan. Kesimpulan yang diperoleh dari analisis ini juga akan membantu dalam memahami lebih baik bagaimana teknik *boosting* dapat digunakan dalam konteks klasifikasi status gizi balita.

2.2 Metode *AdaBoost*

AdaBoost, yang merupakan singkatan dari *Adaptive Boosting*, adalah salah satu algoritma *boosting* pertama yang diperkenalkan oleh Yoav Freund dan Robert Schapire pada tahun 1996 [18]. *AdaBoost* adalah metode ansambel yang bertujuan untuk mengubah sejumlah model prediksi lemah (*weak learners*) menjadi model prediksi yang kuat (*strong learner*) [19]. Dalam konteks klasifikasi, *weak learner* biasanya adalah model yang sedikit lebih baik dari tebakan acak. Algoritma ini bekerja secara iteratif, di mana setiap iterasi meningkatkan performa model dengan fokus pada kesalahan yang dibuat oleh model sebelumnya [20].

Prinsip kerja dari algoritma *AdaBoost* dijelaskan secara rinci sebagai berikut:

a. Inisialisasi Bobot

Setiap pengamatan dalam dataset diberikan bobot yang sama di awal. Misalnya, jika terdapat N pengamatan, setiap bobot $w_i = \frac{1}{N}$.

b. Iterasi

Pada setiap iterasi t :

1. Pelatihan *Weak Learner*: Pelatihan model lemah pada data yang dibobotkan.
2. Kesalahan Model: Hitung kesalahan model $h_t(x)$ terhadap bobot data melalui persamaan (1).

$$\epsilon_t = \frac{\sum_{i=1}^N w_i I(y_i \neq h_t(x_i))}{\sum_{i=1}^N w_i} \quad (1)$$

dimana $I(y_i \neq h_t(x_i))$ adalah fungsi indikator yang bernilai 1 jika prediksi salah dan 0 jika benar.

3. Bobot Model: Beri bobot pada model lemah berdasarkan kesalahannya menggunakan persamaan (2).

$$\alpha_t = \ln \left(\frac{1 - \epsilon_t}{\epsilon_t} \right) \quad (2)$$

4. Pembaruan Bobot: Perbarui bobot pengamatan sehingga pengamatan yang salah klasifikasi mendapatkan bobot lebih tinggi melalui persamaan (3).

$$w_i \leftarrow w_i \exp\left(\alpha_t I(y_i \neq h_t(x_i))\right) \quad (3)$$

Setelah pembaruan, bobot di-normalisasi sehingga total bobot tetap 1.

- c. Kombinasi Model

Model akhir adalah kombinasi dari semua model lemah yang telah diberi bobot menggunakan persamaan (4).

$$H(x) = \text{sign}\left(\sum_{t=1}^T \alpha_t h_t(x)\right) \quad (4)$$

2.3 Metode Gradient Boosting

Gradient Boosting adalah teknik *boosting* yang dikembangkan oleh Jerome Friedman pada akhir 1990-an dan awal 2000-an [21]. Algoritma ini menggabungkan konsep *boosting* dengan pendekatan *gradient descent* untuk mengoptimalkan fungsi *loss* [22]. Dalam *Gradient Boosting*, model dibangun secara bertahap, dengan setiap model baru mengoreksi kesalahan dari model sebelumnya [12]. *Gradient Boosting* secara efektif meminimalkan fungsi *loss* dengan mengikuti gradien dari fungsi *loss*. Pendekatan ini memungkinkan model untuk memperbaiki kesalahan yang dibuat oleh model sebelumnya secara iteratif dan konsisten.

Prinsip kerja dari algoritma *Gradient Boosting* dijelaskan secara rinci sebagai berikut:

- a. Inisialisasi Bobot

Mulai dengan model awal yang sederhana $F_0(x)$, biasanya berupa rata-rata target jika melakukan regresi atau probabilitas dasar untuk klasifikasi.

- b. Iterasi

Pada setiap iterasi t :

1. Hitung Residu: Hitung residu (kesalahan) dari model saat ini menggunakan persamaan (5).

$$r_{i,t} = - \left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)} \right]_{F=D=F_{t-1}} \quad (5)$$

dimana L adalah fungsi *loss* dan F adalah model prediksi.

2. Pelatihan *Weak Learner*: Latih model lemah $h_t(x)$ pada residu r_t .

3. Pembaruan Model: Perbarui model prediksi dengan menambahkan model lemah yang baru menggunakan persamaan (6).

$$F_t(x) = F_{t-1}(x) + v h_t(x) \quad (7)$$

- c. Kombinasi Model

Kombinasi Model: Model akhir adalah kombinasi dari semua model lemah melalui persamaan (8).

$$F(x) = F_0(x) + \sum_{t=1}^T v h_t(x) \quad (8)$$

2.4 Metode XGBoost

XGBoost (Extreme Gradient Boosting) diperkenalkan oleh Tianqi Chen pada tahun 2014 [23]. Algoritma ini dikembangkan sebagai bagian dari proyek *open-source* dan telah menjadi sangat populer dalam komunitas *data science* dan *machine learning* karena kinerja dan efisiensinya yang tinggi. *XGBoost* mengintegrasikan beberapa teknik optimasi untuk meningkatkan kinerja dan efisiensi komputasi, termasuk regulasi, pemrosesan paralel, dan penanganan *sparsity* [24]. *XGBoost* menggunakan teknik tambahan seperti *shrinkage* (pengurangan kontribusi setiap pohon) dan *subsampling* (pengambilan subset data secara acak untuk setiap pohon) untuk lebih meningkatkan kinerja [25]. Teknik-teknik ini membantu mengurangi *overfitting* dan mempercepat proses pelatihan.

Prinsip kerja dari algoritma *XGBoost (Extreme Gradient Boosting)* dijelaskan secara rinci sebagai berikut:

- a. Fungsi Tujuan

Fungsi tujuan *XGBoost* terdiri dari dua komponen utama: fungsi *loss* dan istilah regulasi untuk mengontrol kompleksitas model. Fungsi tujuan dapat dilihat pada persamaan (9).

$$\text{Obj}(t) = \sum_{i=1}^n L(y_i, \hat{y}_i^{(t)}) + \sum_{k=1}^t \Omega(f_k) \quad (9)$$

dimana Ω adalah istilah regulasi yang membantu mencegah *overfitting*.

- b. Istilah Regulasi

Istilah "Regulasi" pada *XGBoost* adalah "*Regularization*." *Regularization* dalam konteks *XGBoost* mengacu pada teknik yang digunakan untuk mencegah *overfitting* dengan menambahkan penalti terhadap kompleksitas model. *XGBoost* menggunakan dua jenis regularisasi: L1 (*Lasso*) dan L2 (*Ridge*). Regularisasi ini membantu model menjadi lebih generalis dan meningkatkan performanya pada data yang belum pernah dilihat. Regularisasi untuk kompleksitas model dihitung menggunakan persamaan (10).

$$\Omega(f) = \gamma T + \frac{1}{2} \lambda \sum_{j=1}^T w_j^2 \quad (10)$$

di mana T adalah jumlah daun dalam pohon dan w_j adalah bobot daun j .

c. Fungsi *Loss*

Fungsi *loss* dihitung pada setiap iterasi melalui persamaan (11).

$$Obj(t) \approx \sum_{i=1}^n \left[g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) \right] + \Omega(f_t) \tag{11}$$

dimana g_i dan h_i adalah gradien dan hessian dari fungsi *loss*, masing-masing.

d. Pembaruan Model

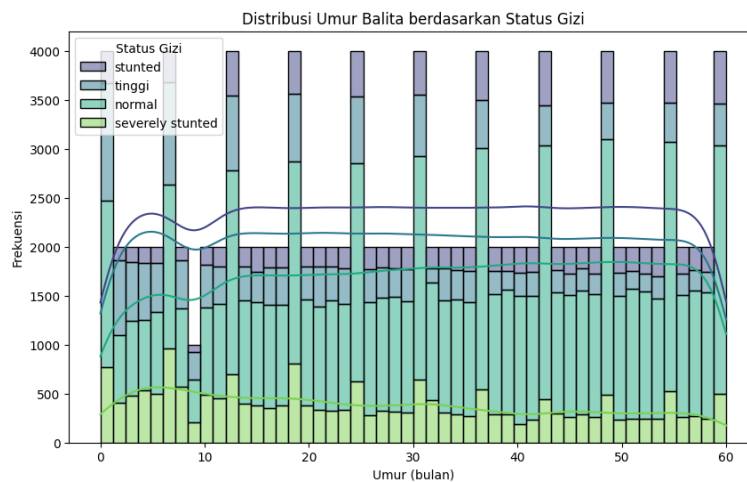
Pembaruan model dilakukan dengan persamaan (12).

$$\hat{y}_i^{(t)} = \hat{y}_i^{(t-1)} + \eta f_t(x_i) \tag{12}$$

di mana η adalah laju pembelajaran.

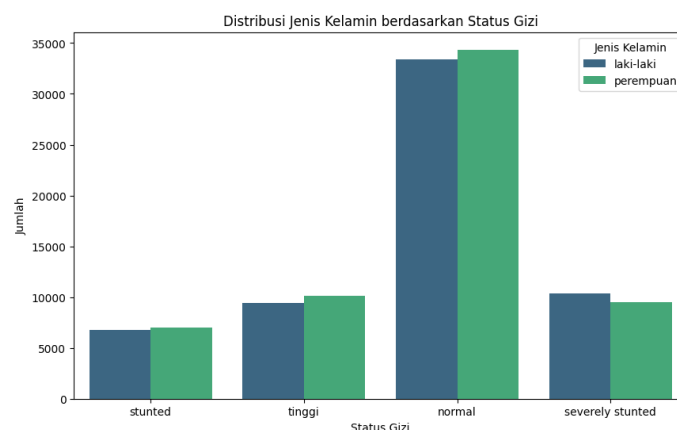
3. HASIL DAN PEMBAHASAN

Pada penelitian ini, dilakukan klasifikasi status gizi balita dengan menganalisis tiga teknik *boosting* yaitu *AdaBoost*, *Gradient Boosting*, dan *XGBoost*. Dataset yang digunakan untuk klasifikasi status gizi mencakup berbagai fitur seperti umur, jenis kelamin, dan tinggi badan. Tahap pertama diawali dengan pra-pemrosesan data, dimana proses ini dilakukan dengan menghapus duplikasi dan menerapkan label *encoding* pada fitur kategorikal, sehingga siap untuk digunakan dalam pemodelan. Untuk memberikan gambaran mengenai distribusi data, maka dibuat visualisasi data untuk beberapa fitur. Untuk fitur umur berdasarkan status gizi divisualisasikan pada Gambar 2.



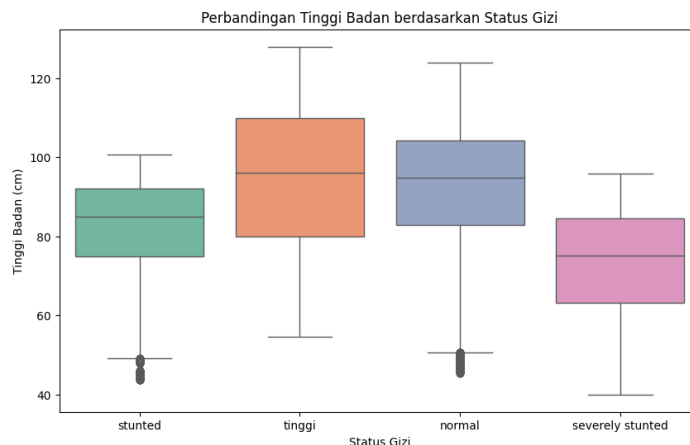
Gambar 2. Distribusi Data Umur Berdasarkan Status Gizi

Grafik histogram yang ditampilkan pada Gambar 2 menunjukkan distribusi umur balita berdasarkan status gizi, dimana sebagian besar balita dalam dataset memiliki status gizi normal, diikuti oleh balita dengan status gizi tinggi. Balita dengan status gizi *stunted* dan *severely stunted* jumlahnya lebih sedikit dan merata di setiap kelompok umur. Namun, dari data ini dapat menjadi informasi pola status gizi berdasarkan umurnya. Selanjutnya, untuk fitur jenis kelamin terhadap status gizi distribusi dataset yang digunakan tervisualisasi pada Gambar 3.



Gambar 3. Distribusi Data Jenis Kelamin Berdasarkan Status Gizi

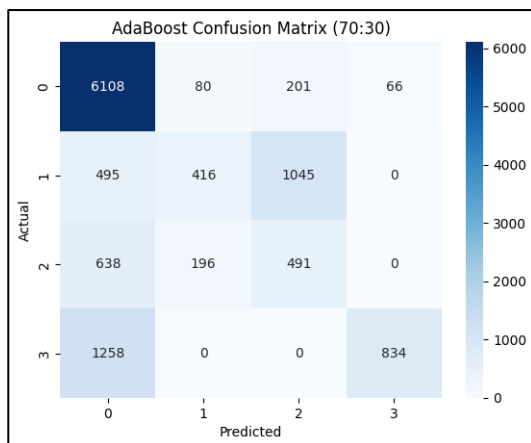
Grafik bar yang ditampilkan pada Gambar 3 menunjukkan distribusi jenis kelamin balita berdasarkan status gizi, dimana distribusi status gizi berdasarkan jenis kelamin relatif merata di antara laki-laki dan perempuan, dengan sedikit lebih banyak balita perempuan yang memiliki status gizi normal. Kategori normal mendominasi populasi balita dalam data ini, sementara kategori *stunted* dan *severely stunted* memiliki jumlah yang jauh lebih rendah. Pola-pola data yang ada dari distribusi data jenis kelamin terhadap status gizi ini menjadi informasi untuk model yang dibangun dalam melakukan klasifikasi. Berikutnya, yaitu untuk distribusi data tinggi badan berdasarkan status gizi. Distribusi data ini divisualisasikan pada Gambar 4.



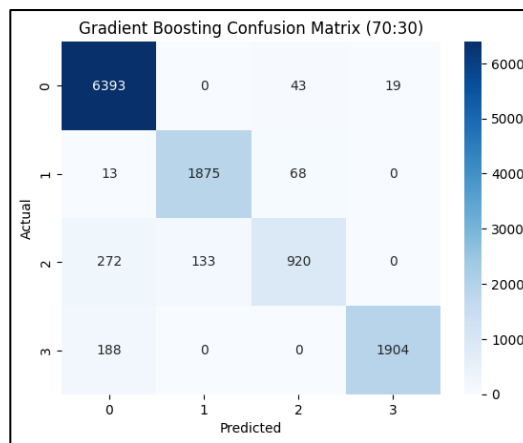
Gambar 4. Distribusi Data Tinggi Badan Berdasarkan Status Gizi

Grafik *boxplot* yang ditampilkan pada Gambar 4 menunjukkan distribusi tinggi badan balita berdasarkan status gizi, dimana status gizi balita memiliki korelasi yang kuat dengan tinggi badan. Balita dengan status gizi tinggi memiliki rentang tinggi badan yang lebih besar dan median yang lebih tinggi dibandingkan dengan kategori lainnya. Sebaliknya, balita dengan status gizi *stunted* dan *severely stunted* memiliki tinggi badan yang lebih rendah secara signifikan, dengan beberapa balita yang sangat pendek (*outlier*). Pola yang terdapat pada distribusi data tinggi badan terhadap status gizi ini menjadi informasi untuk model dalam melakukan klasifikasi status gizi.

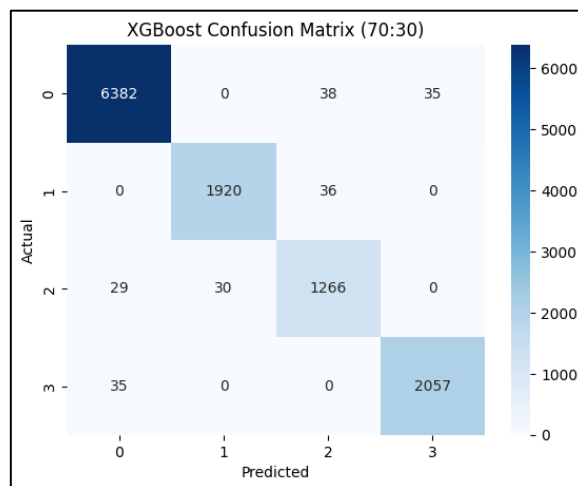
Proses selanjutnya yaitu membangun model klasifikasi menggunakan tiga teknik *boosting* yaitu *AdaBoost*, *Gradient Boosting*, dan *XGBoost*. Proses membangun model dalam menggunakan Bahasa pemrograman Python dan menggunakan *code editor* yaitu Google Collab. Langkah membangun model diawali dengan melakukan *encoding* pada kolom 'Jenis Kelamin' dan 'Status Gizi' menggunakan '*LabelEncoder*' untuk mengubah nilai kategorikal menjadi numerik. Setelah itu, fitur dan target dipisahkan, dan fitur-fitur tersebut distandarisasi menggunakan '*StandardScaler*' untuk memastikan bahwa semua fitur berada dalam skala yang sama. Dataset kemudian dibagi menjadi data pelatihan dan data uji dengan rasio yang berbeda-beda. Untuk menganalisis kinerja masing-masing metode digunakan tiga pembagian dataset yaitu: 70:30, 80:20, dan 90:10. Setelah pembagian data, tiga model berbeda dilatih: *AdaBoost*, *Gradient Boosting*, dan *XGBoost*. Model *AdaBoost* dilatih dengan menggunakan '*AdaBoostClassifier*', model *Gradient Boosting* dilatih dengan '*GradientBoostingClassifier*', dan model *XGBoost* dilatih dengan '*XGBClassifier*' dari pustaka '*xgboost*'. Setiap model dilatih menggunakan data pelatihan (*x_train* dan *y_train*). Tiga model pembelajaran mesin: *AdaBoost*, *Gradient Boosting*, dan *XGBoost*, dilatih menggunakan data latih dan dievaluasi pada data uji. Evaluasi dilakukan dengan menghitung metrik seperti *precision*, *recall*, *F1 score*, akurasi, dan ROC-AUC. Untuk mendapatkan nilai-nilai tersebut menggunakan *confusion matrix* dan kurva ROC-AUC. Hasil *confusion matrix* untuk masing-masing model *boosting* yang diuji ditampilkan pada Gambar 5.



(a)



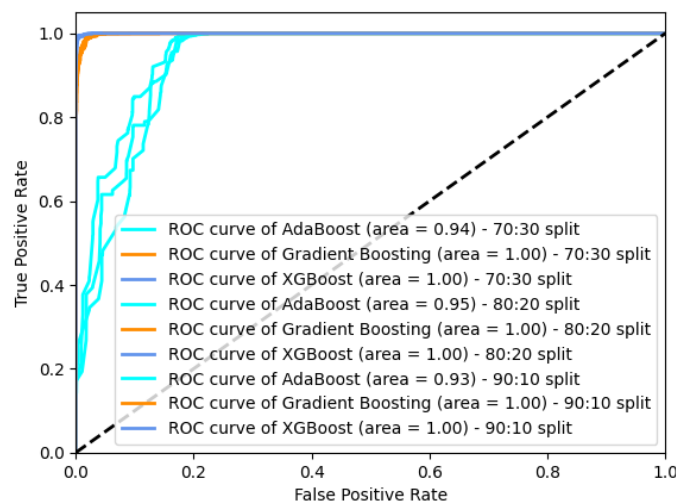
(b)



(c)

Gambar 5. Confusion Matrix Untuk Pembagian Dataset 70:30 Pada Metode: (a) AdaBoost, (b) Gradient Boosting, dan (c) XGBoost

Pada Gambar 5 menunjukkan hasil *confusion matrix* dengan pembagian dataset 70:30 pada masing-masing metode *boosting* yang diuji. *Confusion matrix* disusun dengan pembagian dataset yang lain seperti 80:20 dan 90:10. *Confusion matrix* digunakan untuk menilai kinerja model klasifikasi dengan menggambarkan bagaimana hasil prediksi model dibandingkan dengan label kelas yang sebenarnya. Setelah, *confusion matrix* telah disusun selanjutnya melakukan evaluasi menggunakan metrik lainnya yaitu ROC-AUC (*Receiver Operating Characteristic - Area Under the Curve*). ROC-AUC merupakan metrik evaluasi yang digunakan untuk menilai performa model klasifikasi, khususnya dalam kasus di mana data tidak seimbang. ROC-AUC mengukur kemampuan model dalam membedakan antara kelas positif dan negatif. Hasil ROC-AUC untuk rasio dataset 70:30, 80:20, dan 90:10 pada model *AdaBoost*, *Gradient Boosting*, dan *XGBoost* divisualisasikan pada curve pada Gambar 6.



Gambar 6. ROC-AUC Curve Pada Metode *AdaBoost*, *Gradient Boosting*, dan *XGBoost*

Gambar 6 menampilkan kurva ROC untuk tiga model pembelajaran mesin dengan teknik *boosting*, yaitu *AdaBoost*, *Gradient Boosting*, dan *XGBoost*, dengan berbagai rasio pembagian data (70:30, 80:20, dan 90:10). Kurva ROC digunakan untuk mengevaluasi kinerja klasifikasi model dengan mengukur *trade-off* antara *True Positive Rate* (TPR) dan *False Positive Rate* (FPR). Langkah selanjutnya dari metrik-metrik yang telah dihasilkan kemudian dihitung parameter-parameter uji seperti *precision*, *recall*, *F1 score*, akurasi dan ROC-AUC. Hasil uji akurasi untuk masing-masing model *boosting* dengan pembagian dataset 70:30, 80:20, 90:10 disusun dalam Tabel 1.

Tabel 1. Perbandingan Hasil Evaluasi

Rasio Dataset	Model Boosting	Precision	Recall	F1 Score	Accuracy	ROC-AUC
70:30	<i>AdaBoost</i>	0.6646	0.6636	0.6635	0.6636	0.7544
	<i>Gradient Boosting</i>	0.9380	0.9378	0.9378	0.9378	0.9933
	<i>XGBoost</i>	0.9829	0.9828	0.9828	0.9828	0.9993

Rasio Dataset	Model Boosting	Precision	Recall	F1 Score	Accuracy	ROC-AUC
80:20	AdaBoost	0.6448	0.6443	0.6442	0.6443	0.7718
	Gradient Boosting	0.9405	0.9401	0.9401	0.9401	0.9935
	XGBoost	0.9849	0.9848	0.9848	0.9848	0.9994
90:10	AdaBoost	0.6294	0.6292	0.6291	0.6292	0.7581
	Gradient Boosting	0.9406	0.9404	0.9403	0.9404	0.9936
	XGBoost	0.9843	0.9843	0.9842	0.9843	0.9994

Padat Tabel 1 yang menunjukkan hasil evaluasi perbandingan model *boosting* untuk klasifikasi status gizi balita menunjukkan bahwa *XGBoost*, *Gradient Boosting*, dan *AdaBoost* memiliki perbedaan kinerja yang signifikan. Dalam semua rasio pembagian data (70:30, 80:20, dan 90:10), *XGBoost* secara konsisten menunjukkan kinerja terbaik dengan nilai *precision*, *recall*, *accuracy*, *F1 score*, dan ROC-AUC yang mendekati sempurna. Misalnya, pada rasio 70:30, *XGBoost* mencapai *precision* 0.9829, *recall* 0.9828, *accuracy* 0.9828, *F1 score* 0.9828, dan ROC-AUC 0.9993. Hasil ini menunjukkan bahwa *XGBoost* memiliki kemampuan yang luar biasa dalam mengklasifikasikan status gizi balita dengan akurasi tinggi dan kemampuan diskriminatif yang sangat baik. Kinerja *XGBoost* tetap stabil dan optimal meskipun ukuran set pelatihan berubah, menandakan kemampuannya dalam menangani kompleksitas data dan generalisasi yang kuat. *Gradient Boosting* juga menunjukkan kinerja yang sangat baik dan konsisten di semua rasio pembagian data. Dengan *precision*, *recall*, *accuracy*, *F1 score*, dan ROC-AUC yang hampir mendekati nilai maksimal, *Gradient Boosting* memberikan hasil yang sangat memuaskan. Pada rasio 70:30, model ini mencapai *precision* 0.9380, *recall* 0.9378, *accuracy* 0.9378, *F1 score* 0.9378, dan ROC-AUC 0.9933. Hasil yang serupa juga terlihat pada rasio 80:20 dan 90:10, menunjukkan bahwa *Gradient Boosting* adalah model yang sangat andal untuk tugas klasifikasi ini. Meskipun sedikit di bawah *XGBoost*, *Gradient Boosting* tetap menjadi pilihan yang sangat baik karena konsistensinya dalam memberikan hasil yang akurat dan stabil. Sebaliknya, *AdaBoost* menunjukkan kinerja yang lebih rendah dibandingkan dengan dua model lainnya. Pada rasio 70:30, *AdaBoost* hanya mencapai *precision* 0.6646, *recall* 0.6636, *accuracy* 0.6636, *F1 score* 0.6635, dan ROC-AUC 0.7544. Kinerja ini menurun lebih lanjut pada rasio 80:20 dan 90:10, dengan nilai *precision*, *recall*, *accuracy*, *F1 score*, dan ROC-AUC yang terus menurun. Ini menunjukkan bahwa *AdaBoost* kurang mampu menangani variasi dan kompleksitas data dibandingkan dengan *XGBoost* dan *Gradient Boosting*, dan model ini lebih sensitif terhadap perubahan ukuran set pelatihan.

Secara keseluruhan, analisis ini mengindikasikan bahwa *XGBoost* adalah model terbaik untuk klasifikasi status gizi balita, diikuti oleh *Gradient Boosting*. Kedua model ini menunjukkan kinerja yang superior dalam semua metrik evaluasi dan tetap konsisten di berbagai rasio pembagian data, menjadikannya pilihan yang paling andal untuk tugas ini. *AdaBoost*, meskipun memberikan hasil yang cukup baik, tidak dapat bersaing dengan kinerja dari *XGBoost* dan *Gradient Boosting*. Oleh karena itu, dalam konteks klasifikasi status gizi balita, *XGBoost* adalah model yang paling direkomendasikan. *XGBoost* (*Extreme Gradient Boosting*) mendapatkan hasil yang lebih baik karena beberapa alasan utama. Pertama, *XGBoost* menggunakan teknik regularisasi, yang membantu mencegah *overfitting* dengan mengurangi kompleksitas model. Kedua, *XGBoost* memiliki kemampuan untuk menangani *missing values* secara efektif dan membuat keputusan yang optimal tentang bagaimana menangani data yang hilang selama proses pelatihan. Selain itu, *XGBoost* menggunakan algoritma *boosting* yang lebih efisien dan cepat dengan memanfaatkan teknik paralelisasi, sehingga dapat memproses dataset besar dengan lebih cepat dan akurat. Di sisi lain, *AdaBoost* tidak mendapatkan hasil yang optimal karena beberapa keterbatasan inherennya. *AdaBoost* bekerja dengan memberikan bobot lebih pada sampel yang salah diklasifikasikan dalam iterasi sebelumnya dan membangun model yang lebih kuat dari model lemah. Meskipun pendekatan ini efektif dalam beberapa kasus, *AdaBoost* sangat sensitif terhadap *outliers* dan *noise* dalam data. Hal ini disebabkan oleh peningkatan bobot yang diberikan pada misclassifications, yang dapat memperkuat kesalahan jika data mengandung banyak *noise* atau *outliers*.

4. KESIMPULAN

Penelitian telah mengklasifikasikan status gizi balita dengan menganalisis tiga teknik *boosting* yaitu *AdaBoost*, *Gradient Boosting*, dan *XGBoost*. Hasil tertinggi pada evaluasi klasifikasi status gizi balita diperoleh dari model *XGBoost* dengan rasio pembagian data 80:20, di mana model ini mencapai *precision* 0.9849, *recall* 0.9848, *accuracy* 0.9848, *F1 score* 0.9848, dan ROC-AUC 0.9994. Kinerja luar biasa dari *XGBoost* disebabkan oleh penggunaan teknik regularisasi, kemampuan menangani *missing values* secara efektif, serta algoritma *boosting* yang efisien dan cepat melalui teknik paralelisasi. Semua faktor ini memungkinkan *XGBoost* untuk memproses dataset besar dengan lebih akurat dan konsisten, sehingga menghasilkan performa yang hampir sempurna dalam berbagai metrik evaluasi. Sebaliknya, hasil terendah diperoleh dari model *AdaBoost* dengan rasio pembagian data 90:10, di mana *precision* hanya mencapai 0.6294, *recall* 0.6292, *accuracy* 0.6292, *F1 score* 0.6291, dan ROC-AUC 0.7581. Kinerja rendah dari *AdaBoost* disebabkan oleh sensitivitasnya terhadap *outliers* dan *noise* dalam data, serta kurangnya mekanisme regularisasi yang kuat. Algoritma *AdaBoost* memberikan bobot lebih pada sampel yang salah diklasifikasikan dalam iterasi sebelumnya, yang dapat memperkuat kesalahan jika data mengandung banyak *noise* atau *outliers*. Temuan ini menunjukkan bahwa *XGBoost* adalah model terbaik untuk klasifikasi status gizi balita, diikuti oleh *Gradient Boosting*, dengan *AdaBoost* berada di posisi terakhir. *XGBoost* dan *Gradient Boosting* menunjukkan kinerja yang superior dalam

semua metrik evaluasi dan tetap konsisten di berbagai rasio pembagian data, menjadikannya pilihan yang paling andal untuk tugas ini. Untuk penelitian selanjutnya, disarankan untuk mengeksplorasi beberapa aspek tambahan guna meningkatkan hasil klasifikasi dan memahami lebih dalam tentang data dan model yang digunakan. Penggunaan kombinasi model (*ensemble methods*) seperti stacking atau blending bisa diuji untuk melihat apakah kombinasi dari beberapa model dapat memberikan hasil yang lebih baik daripada menggunakan satu model saja. Selain itu, dibutuhkan *tuning hyperparameter* yang lebih mendalam untuk masing-masing model dapat dilakukan untuk memastikan bahwa model bekerja pada konfigurasi optimalnya.

REFERENCES

- [1] Z. Prihatini and S. W. Wibawa, "Angka Stunting Indonesia 24,4 Persen, 7 Provinsi Catat Kasus Tertinggi," *kompas.com*, 2022. <https://www.kompas.com/sains/read/2022/07/01/140000123/angka-stunting-indonesia-24-4-persen-7-provinsi-catat-kasus-tertinggi>
- [2] S. Sukmawati, *Faktor-faktor yang Berhubungan dengan Kejadian Stunting pada Balita*. Pekalongan: Penerbit NEM, 2023.
- [3] A. Wahyu, L. Ginting, and N. D. Sinaga, *Faktor Penyebab Terjadinya Stunting*. Sukabumi: CV Jejak (Jejak Publisher), 2022.
- [4] B. Analysis and C. Analysis, "Upaya Penanganan Stunting di Indonesia: Analisis Bibliometrik dan Analisis Konten," *J. Ilmu Pemerintah. Suara Khatulistiwa*, vol. VIII, no. 01, pp. 44–59, 2023.
- [5] M. L. Dambe, S. Y. Padang, and M. S. Adha, "Evaluasi K-Nearest Neighbour Untuk Klasifikasi Status Gizi Balita," *INFINITY*, vol. 3, no. 1, pp. 33–40, 2023, doi: 10.34148/infinity.v9i1.xxx.
- [6] 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," *J. Tek.*, vol. 18, no. 2, pp. 489–500, 1978.
- [7] D. N. A. Kurniawan and M. Maryam, "Implementasi Metode Decision Tree pada Sistem Prediksi Status Gizi Balita," *J. Sains Komput. Inform.*, vol. 7, no. 2, pp. 731–739, 2023.
- [8] Y. R. Nasution, A. Armansyah, M. Furqan, and T. R. Matondang, "Penerapan Algoritma C4.5 Pada Klasifikasi Status Gizi Balita," *J. FASILKOM*, vol. 14, no. 1, pp. 216–225, 2024.
- [9] P. Handayani, A. C. Fauzan, and H. Harliana, "Machine Learning Klasifikasi Status Gizi Balita Menggunakan Algoritma Random Forest," *KLIK Kaji. Ilm. Inform. dan Komput.*, vol. 4, no. 6, pp. 3064–3072, 2024, doi: 10.30865/klik.v4i6.1909.
- [10] N. Nurhayati, *Teknik Ensemble Learning Untuk Peningkatan Performa Akurasi Model Prediksi (Seleksi Mahasiswa Penerima Beasiswa)*. Tangerang: Pascal Books, 2022.
- [11] G. Abdurrahman, "Klasifikasi Penyakit Diabetes Melitus Menggunakan Adaboost Classifier," *JUSTINDO (Jurnal Sist. Teknol. Inf. Indones.)*, vol. 7, no. 1, pp. 59–66, 2022.
- [12] V. Atlantic, E. Sulistianingsih, and H. Perdana, "Gradient Boosting Machine Pada Klasifikasi Kelulusan Mahasiswa," *Bul. Ilm. Math. Stat. dan Ter.*, vol. 13, no. 2, pp. 165–174, 2024.
- [13] A. H. Permana, F. R. Umbara, and F. Kasyidi, "Klasifikasi Penyakit Jantung Tipe Kardiovaskular Menggunakan Adaptive Synthetic Sampling dan Algoritma Extreme Gradient Boosting," *Build. Informatics, Technol. Sci.*, vol. 6, no. 1, pp. 499–508, 2024, doi: 10.47065/bits.v6i1.5421.
- [14] Y. Fernando, R. Napianto, and R. I. Borman, "Implementasi Algoritma Dempster-Shafer Theory Pada Sistem Pakar Diagnosa Penyakit Psikologis Gangguan Kontrol Impuls," *Insearch Inf. Syst. Res. J.*, vol. 2, no. 2, pp. 46–54, 2022.
- [15] R. P. Pradana, "Stunting Toddler Detection," *Kaggle*, 2024. <https://www.kaggle.com/datasets/rendiputra/stunting-balita-detection-121k-rows/>
- [16] R. I. Borman and M. Wati, "Penerapan Data Mining Dalam Klasifikasi Data Anggota Kopdit Sejahtera Bandarlampung Dengan Algoritma Naïve Bayes," *J. Ilm. Fak. Ilmu Komput.*, vol. 9, no. 1, pp. 25–34, 2020.
- [17] R. I. Borman, F. Rossi, D. Alamsyah, R. Nuraini, and Y. Jusman, "Classification of Medicinal Wild Plants Using Radial Basis Function Neural Network with Least Mean Square," in *International Conference on Electronic and Electrical Engineering and Intelligent System (ICE3IS)*, 2022.
- [18] T. H. Saragih, M. Muliadi, M. R. Faisal, and M. A. I. N. R. Said, "AdaBoost Classifier untuk Klasifikasi Tanaman Jarak Pagar," *J. Komputasi*, vol. 9, no. 2, pp. 60–66, 2021.
- [19] R. T. Febianto, D. Suranti, and R. T. Alinse, "Penerapan Algoritma Adaboost Dalam Mengetahui Pola Pengguna KB di Puskesmas Tanjung Harapan," *J. Sci. Soc. Res.*, vol. VII, no. 1, pp. 145–155, 2024.
- [20] N. Novianti, M. Zarlis, and P. Sihombing, "Penerapan Algoritma Adaboost Untuk Peningkatan Kinerja Klasifikasi Data Mining Pada Imbalance Dataset Diabetes," *J. Media Inform. Budidarma*, vol. 6, no. 2, pp. 1200–1206, 2022, doi: 10.30865/mib.v6i2.4017.
- [21] M. Ridwansyah and H. Zakaria, "Implementasi Algoritma Gradient Boosting Pada Aplikasi Hutang Piutang Perorangan Secara Berbasis Web Untuk Meningkatkan Akurasi Prediksi Pelunasan Hutang (Studi Kasus : PT Naila Kreasi Mandiri)," *JURIHUM J. Inov. dan Hum.*, vol. 1, no. 4, pp. 440–451, 2023.
- [22] S. E. Suryana, B. Warsito, and S. Suparti, "Penerapan Gradient Boosting dengan Hyperopt Untuk Memprediksi Keberhasilan Telemarketing Bank," *J. Gaussian*, vol. 10, no. 4, pp. 617–623, 2021.
- [23] J. Melvin and A. Soraya, "Analisis Perbandingan Algoritma XGBoost dan Algoritma Random Forest Ensemble Learning pada Klasifikasi Keputusan Kredit," *J. Ris. Rumpun Mat. dan Ilmu Pengetah. Alam*, vol. 2, no. 2, pp. 87–103, 2023.
- [24] S. E. H. Yulianti, O. Soesanto, and Y. Sukmawaty, "Penerapan Metode Extreme Gradient Boosting (XGBOOST) pada Klasifikasi Nasabah Kartu Kredit," *J. Math. Theory Appl.*, vol. 4, no. 1, pp. 21–26, 2022.
- [25] I. M. K. Karo, "Implementasi Metode XGBoost dan Feature Importance untuk Klasifikasi pada Kebakaran Hutan dan Lahan," *J. Softw. Eng. Inf. Commun. Technol.*, vol. 1, no. 1, pp. 11–18, 2020.