JOISIE (Journal Of Information Systems And Informatics Engineering)
Vol. 9, No.2, Desember 2025, Hlm 368-379
Received: 8 November 2025 Revised: 4 Desember 2025, Accepted: 8 Desember 2025
p- ISSN: 2503-5304
e- ISSN: 2527-3116

OPTIMASI ALGORITMA XGBOOST UNTUK KLASIFIKASI STATUS GIZI BALITA BB/TB MENGGUNAKAN GRIDSEARCHCV

Avi Oktaviani[1)], Basuki Rahmat[2)], Kartini[3)]
[1,2,3]Fakultas Ilmu Komputer, UPN "Veteran" Jawa Timur
Jl. Rungkut Madya, Gn. Anyar, Kec. Gn. Anyar, Surabaya, Jawa Timur 60294
Email : email: penulis avioktaviani1002@gmail.com, basukirahmat.if@upnjatim.ac.id, kartini.if@upnjatim.ac.id

Abstract

Nutritional problems remain a major public health concern in many countries, including Indonesia. This study designs and evaluates an automatic classification system for toddler nutritional status based on weight-for-height indicators using the XGBoost algorithm. The dataset comprises 1,763 Indonesian children aged 24–60 months, with nutritional status categorized into six classes (gizi baik, risiko gizi lebih, gizi kurang, gizi buruk, gizi lebih, obesitas). Predictor variables include sex, age in months, body weight, body height, and weight-for-height Z-score. Preprocessing includes data cleaning, removal of duplicates, label encoding, standardization, and an 80:20 train–test split. A baseline XGBoost model is compared with models tuned using GridSearch with 3-fold and 10-fold cross-validation. Performance is evaluated using accuracy, macro precision, macro recall, macro F1-score, and log loss. The best model, XGBoost with GridSearch 10-fold, achieves an accuracy of 0.8689, F1-score of 0.8149, and log loss of 0.3395, improving over the baseline log loss of 0.4190. These findings indicate that XGBoost with GridSearch hyperparameter tuning yields well-calibrated probabilistic predictions and is a promising decision-support tool for early detection of malnutrition and obesity in toddlers in primary health care settings.

Keywords: *XGBoost, Grid Search, Toddler nutrition, Classification*

Abstrak

Masalah gizi tetap menjadi perhatian utama kesehatan masyarakat di banyak negara, termasuk Indonesia. Studi ini merancang dan mengevaluasi sistem klasifikasi otomatis untuk status gizi balita berdasarkan indikator berat badan terhadap tinggi badan menggunakan algoritma XGBoost. Dataset terdiri dari 1.763 anak Indonesia berusia 24–60 bulan, dengan status gizi dikategorikan menjadi enam kelas (gizi baik, risiko gizi lebih, gizi kurang, gizi buruk, gizi lebih, obesitas). Variabel prediktor meliputi jenis kelamin, usia dalam bulan, berat badan, tinggi badan, dan skor Z berat badan terhadap tinggi badan. Praproses meliputi pembersihan data, penghapusan duplikat, pengkodean label, standardisasi, dan pembagian data latih-uji 80:20. Model XGBoost dasar dibandingkan dengan model yang disetel menggunakan GridSearch dengan validasi silang 3-fold dan 10-fold. Kinerja dievaluasi menggunakan akurasi, presisi makro, recall makro, skor F1 makro, dan log loss. Model terbaik, XGBoost dengan GridSearch 10-fold, mencapai akurasi 0,8689, skor F1 0,8149, dan log loss 0,3395, lebih baik daripada log loss dasar sebesar 0,4190. Temuan ini menunjukkan bahwa penyetelan hyperparameter XGBoost dengan GridSearch menghasilkan prediksi probabilistik yang terkalibrasi dengan baik dan merupakan alat pendukung keputusan yang menjanjikan untuk deteksi dini kekurangan gizi dan obesitas pada balita di fasilitas perawatan kesehatan primer.

Kata Kunci: XGBoost, Pencarian Grid, Nutrisi Balita, Klasifikasi

1. PENDAHULUAN

Status gizi merupakan salah satu indikator utama untuk menentukan kecukupan pertumbuhan dan perkembangan balita (Muhamad Fikri, 2024). Permasalahan gizi seperti gizi buruk dan obesitas masih menjadi tantangan utama dalam bidang kesehatan di banyak negara, termasuk Indonesia (Amalia Alhamid et al., 2021). Kelompok usia balita merupakan salah satu kelompok yang paling rentan mengalami masalah gizi karena berada pada fase pertumbuhan yang sangat cepat dan membutuhkan asupan nutrisi yang optimal (Fibrinika Tuta Setiani et al., 2024). Kesalahan dalam

https://doi.org/10.35145/joisie.v9i2.5601

menegakkan diagnosis status gizi balita dapat berakibat fatal, seperti gangguan perkembangan kognitif, memperberat penyakit kronis, hingga meningkatkan risiko kematian (Paramita et al., 2024). Berdasarkan Survei Status Gizi Indonesia (SSGI) tahun 2022, prevalensi wasting pada balita di Indonesia mencapai 7,7%, meningkat dari 7,1% pada tahun 2021 (Kemenkes RI, 2022). Kondisi gizi yang buruk pada balita tidak hanya berdampak pada kesehatan jangka pendek, tetapi juga memengaruhi kualitas sumber daya manusia di masa mendatang (Lidya et al., 2021). Dalam praktik di lapangan, kategori seperti gizi buruk dan obesitas masih sering terabaikan ketika penilaian status gizi dilakukan secara manual hanya berdasarkan interpretasi grafik pertumbuhan (Masturina et al., 2023). Balita dengan berat badan lebih sering kali memerlukan konfirmasi kembali menggunakan indikator berat badan menurut tinggi badan (BB/TB) agar penilaian status gizinya lebih akurat(Bahar et al., 2024).

Perkembangan metode komputasi dan pembelajaran mesin menawarkan peluang untuk meningkatkan akurasi dan konsistensi klasifikasi status gizi (Puerwandono & Maulana, 2023). Salah satu pendekatan yang banyak digunakan adalah ensemble learning, yang menggabungkan beberapa model sederhana (*weak learner*) untuk membentuk model yang lebih kuat (*better learner*) (Zhang & Haghani, 20 klasifikasi15). Gradient boosting menjadi salah satu teknik yang populer karena mampu meningkatkan akurasi model klasifikasi secara signifikan (Handayani et al., n.d.-a). XGBoost (Extreme Gradient Boosting) merupakan pengembangan dari metode gradient boosting yang dirancang agar lebih efisien dan memiliki performa prediksi yang tinggi untuk tugas klasifikasi maupun regresi (Liew et al., 2021). Pada XGBoost, penambahan weak learner memanfaatkan pemrosesan multi-threaded sehingga penggunaan inti CPU menjadi lebih efisien dan waktu komputasi lebih cepat dibandingkan implementasi gradient boosting konvensional (Anggoro & Mukti, 2021). Algoritma ini diketahui mampu menangani dataset berdimensi tinggi, toleran terhadap missing value, dan cukup efisien secara komputasi (Handayani et al., n.d.-b). Namun, keefektifan XGBoost sangat bergantung pada pemilihan kombinasi hyperparameter, seperti learning_rate, max_depth, dan parameter lainnya, sehingga dibutuhkan strategi penalaan (tuning) yang sistematis (Syahputra & Saputro, 2024a).

Salah satu metode yang sering digunakan untuk tuning hyperparameter adalah GridSearch. Metode ini bekerja dengan membentuk kisi (grid) kombinasi hyperparameter yang ditentukan sebelumnya, kemudian mengevaluasi setiap kombinasi secara sistematis untuk mencari konfigurasi terbaik bagi model (Tran et al., 2023). GridSearch umumnya direkomendasikan untuk ruang parameter yang relatif terbatas namun ingin dieksplorasi secara menyeluruh. Dalam konteks pembelajaran mesin, GridSearch telah dimanfaatkan untuk berbagai kasus, termasuk optimasi model pada proses klasifikasi di bidang kesehatan dan manufaktur (Ogunsanya et al., 2023). Penggunaan GridSearch untuk tuning hyperparameter XGBoost berpotensi menjadi pendekatan yang kritis dalam klasifikasi status gizi balita karena mampu mengeksplorasi kombinasi parameter secara menyeluruh dan meningkatkan kinerja model secara terukur (Uzir et al., 2016).

Beberapa penelitian terdahulu menunjukkan bahwa XGBoost memiliki kinerja yang baik untuk klasifikasi status gizi balita. Anggoro dan Mukti melaporkan bahwa XGBoost memberikan performa terbaik dibandingkan dua metode lain dengan skema pembagian data 80:20, dengan nilai presisi 0,9849, recall 0,9848, akurasi 0,9848, F1-score 0,9848, dan ROC-AUC 0,9994 (Anggoro & Mukti, 2021). Keunggulan ini dikaitkan dengan teknik regularisasi yang efektif, kemampuan menangani missing value, serta algoritma boosting yang efisien melalui paralelisasi. Namun, penelitian tersebut masih menggunakan kombinasi hyperparameter default sehingga potensi peningkatan kinerja model XGBoost melalui tuning hyperparameter belum dieksplorasi secara mendalam. Penelitian lain yang dilakukan oleh Abdurrahman dan rekan menunjukkan bahwa optimasi XGBoost menggunakan GridSearch dan Random Search pada klasifikasi penyakit diabetes mampu meningkatkan akurasi model secara signifikan. Mereka melaporkan bahwa penalaan terstruktur melalui GridSearch dapat meningkatkan performa, meskipun lebih mahal secara komputasi, sementara Random Search lebih efisien namun tetap mampu menemukan kombinasi hyperparameter yang kompetitif (Abdurrahman et al., 2022). Meskipun demikian, studi tersebut berfokus pada penyakit diabetes dan tidak membahas secara spesifik klasifikasi status gizi balita maupun indikator BB/TB.

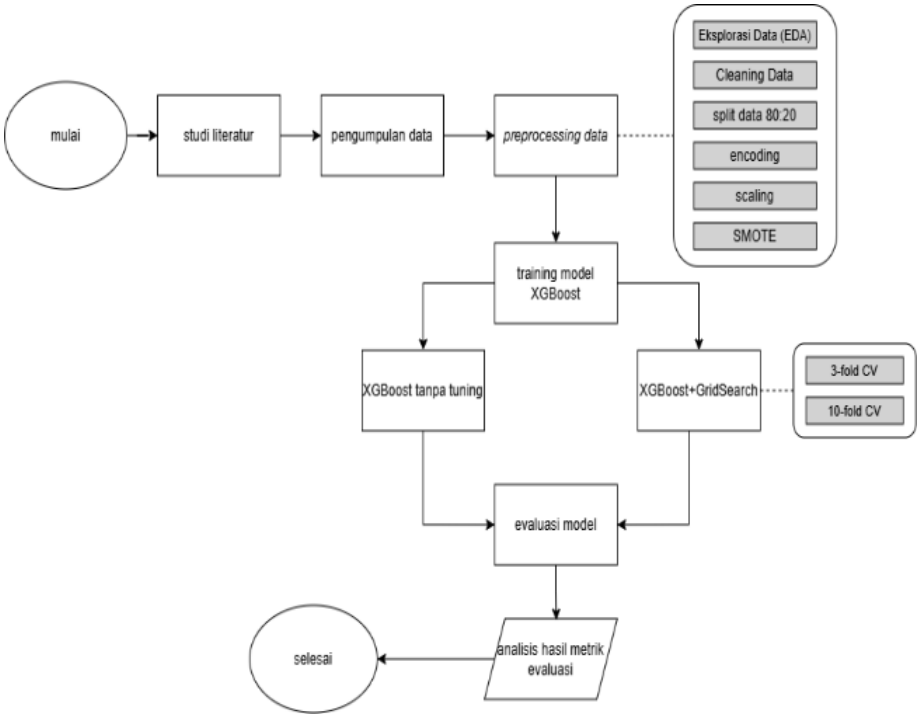
Berdasarkan uraian tersebut, dapat diidentifikasi adanya celah penelitian (research gap), yaitu belum banyak kajian yang secara khusus menerapkan XGBoost dengan tuning hyperparameter GridSearch pada klasifikasi status gizi balita berbasis indikator BB/TB dengan skema multi-kelas.

https://doi.org/10.35145/joisie.v9i2.5601

Penelitian terdahulu yang menggunakan XGBoost untuk klasifikasi status gizi umumnya masih mengandalkan parameter default, sedangkan penelitian yang mengkaji optimasi XGBoost dengan GridSearch lebih berfokus pada domain penyakit lain dan belum menilai secara rinci kalibrasi probabilitas melalui log loss dalam konteks status gizi balita. Oleh karena itu, penelitian ini bertujuan untuk mengoptimasi algoritma XGBoost menggunakan GridSearch sebagai metode tuning hyperparameter pada klasifikasi status gizi balita berdasarkan indikator BB/TB. Penelitian ini mengevaluasi kinerja model menggunakan *confusion matrix*, classification report (akurasi, presisi, recall, dan F1-score), serta log loss untuk menilai baik akurasi klasifikasi maupun kualitas kalibrasi probabilistik. Diharapkan, model yang dihasilkan dapat memberikan kontribusi sebagai alat bantu pengambilan keputusan yang lebih andal bagi tenaga kesehatan dalam deteksi dini masalah gizi pada balita.

2. METODE PENELITIAN

Penelitian ini melakukan pengumpulan data dari Puskesmas Sukorame, Kota Kediri. Dataset yang digunakan merupakan data balita usia 24-60 bulan dengan antropometri BB/TB. Dataset ini berupa data mentah yang berjumlah 1764 sampel balita dengan fitur jenis kelamin, Usia, BB, TB, ZScore BB/TB, dan status gizi. Data yang digunakan merupakan data pertumbuhan balita pada bulan Februari 2025. Data tersebut dapat membantu untuk memastikan keandalan dan validitas hasil sesuai dengan tujuan penelitian.



Gambar 1. Diagram alur penelitian

Tahapan selanjutnya dilakukan preprocessing data dengan menganalisa penyebaran data, pembersihan data, encoding dan standarisasi.

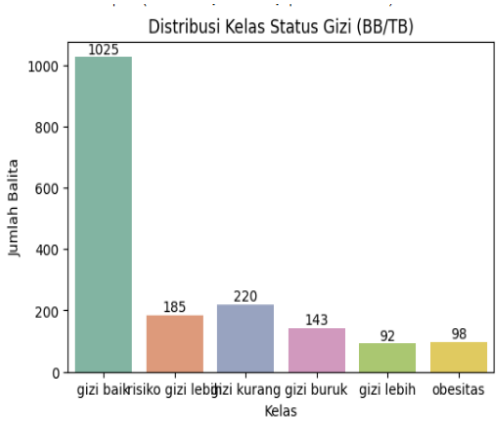
2.1. Pengolahan Data

Pada tahap ini, dilakukan pengolahan data sebelum digunakan dalam model klasifikasi. Pengolahan data merupakan langkah penting dalam penelitian ini untuk memastikan dataset yang digunakan dalam pelatihan dan pengujian model sesuai dengan format (Çetin & Yıldız, 2022).

1. *Exploration Data Analysis* (EDA)

   Tahap ini merupakan langkah awal penting sebelum data diolah. Status gizi balita berdasarkan antropometri BB/TB terdiri dari 6 kategori, yaitu gizi buruk, gizi kurang, gizi baik, risiko gizi lebih, gizi lebih, dan obesitas. Pada tahap ini, EDA membantu memahami sebaran data gizi balita (Goswami & Bhatia, 2023).

https://doi.org/10.35145/joisie.v9i2.5601



Gambar 2. Diagram distribusi gizi balita

Berdasarkan hasil eksplorasi, diketahui bahwa distribusi kategori tidak seimbang. Gizi baik merupakan kategori dengan distribusi tertinggi, sedangkan kategori lainnya kurang dari 250. Karena itu perlu dilakukan balancing data agar lebih merata.

2. *Cleaning data*

Pada tahap ini dilakukan pembersihan data pada missing value dan data yang duplikat (Fan et al., 2021). Dalam pembersihan missing value dilakukan identifikasi terlebih dahulu pada dataset untuk membantu proses analisis dan pengambilan keputusan (Chiang et al., 2021) .

=== Sampel Data dengan Missing Values ===

	Kelamin	BB	TB	ZS BB/TB	Usia	BB/TB
1757	P	15.2	105.0	NaN	60	NaN
1759	L	15.2	101.0	NaN	60	NaN

Gambar 3. Sample *missing value*

Pada Gambar 3. Ditemukan 2 sampel dengan kondisi *missing valued* fitur ZS BB/TB dan BB//TB. Sehingga perlu dilakukan penghapusan kedua sampel tersebut.

Selanjutnya dilakukan identifikasi data yang duplikat pada dataset. Data duplikat berpotensi membuat distribusi kelas nampak lebih besar dan membuat nilai metrik seolah lebih tinggi dari seharusnya.

Jumlah baris duplikat (termasuk semua kemunculan duplikat): 18

Contoh baris duplikat:

	Kelamin	BB	TB	ZS BB/TB	Usia	BB/TB
54	P	10.7	84.0	-0.46	30	gizi baik
57	L	10.3	82.5	-0.90	26	gizi baik
333	P	15.0	87.0	2.10	30	gizi lebih
569	L	15.7	90.0	2.50	35	gizi lebih
582	P	10.5	82.5	-0.30	27	gizi baik

Gambar 4. Setelah dilakukan *cleaning duplicate*

Gambar 4. Ditemukan 18 data yang duplikat pada 2 kelas, yaitu kelas gizi baik dan gizi lebih. Pada kelas gizi baik terdapat 12 data duplikat, sedangkan pada kelas gizi lebih terdapat 6 data duplikat sehingga dilakukan pembersihan data duplikat. Setelah dilakukan *cleaning data,* data yang tersisa sebanyak 1753 dataset

3. Split Data

Split data dilakukan untuk memisahkan data uji sejak awal untuk mencegah terjadinya *leakage(Erlin et al., 2022)*. Pada penelitian ini data dengan rasio 80:20 untuk *train:test*. Rasio ini memberikan data pelatihan lebih banyak, sehingga fit cenderung lebih kuat.

```
=== Data Split  ===
X_train_80: (1403, 4)
X_test_20: (351, 4)
y_train_80: (1403,)
y_test_20: (351,)
```

Gambar 5. Pembagian data latih dan uji

Setelah proses pembagian, diperoleh data latih sebanyak 1.403 sampel dan data uji sebanyak 351 sampel. Pembagian ini dilakukan secara stratified berdasarkan kelas status gizi, sehingga proporsi tiap kategori pada data latih dan data uji tetap sebanding dengan distribusi awal. Selain itu, data uji tidak dilibatkan dalam proses pelatihan maupun tuning hyperparameter dan hanya digunakan pada tahap akhir untuk mengukur kemampuan generalisasi model.

https://doi.org/10.35145/joisie.v9i2.5601

4. Encoding

Encoding merupakan tahap mengubah fitur menjadi numerik agar dapat diproses model. Fitur yang di-encode adalah fitur kelamin dan BB/TB yang masih bertipe object ke represintasi numerik. Pada fitur BB/TB dilakukan encoding mengubah status gizi balita menjadi numerik berdasarkan urutan kondisi balita.

```
RangeIndex: 1754 entries, 0 to 1753
Data columns (total 6 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   Kelamin    1754 non-null   object
 1   BB         1754 non-null   float64
 2   TB         1754 non-null   float64
 3   ZS BB/TB   1754 non-null   float64
 4   Usia       1754 non-null   int64
 5   BB/TB      1754 non-null   object
dtypes: float64(3), int64(1), object(2)
memory usage: 82.3+ KB
```

```
RangeIndex: 1754 entries, 0 to 1753
Data columns (total 6 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   Kelamin    1754 non-null   int64
 1   BB         1754 non-null   float64
 2   TB         1754 non-null   float64
 3   ZS BB/TB   1754 non-null   float64
 4   Usia       1754 non-null   int64
 5   BB/TB      1754 non-null   int64
dtypes: float64(3), int64(3)
memory usage: 82.3 KB
```

Gambar 6. Tipe data sebelum dan setelah *encoding*

Setelah proses encoding dilakukan, tipe data pada fitur kategorik mengalami perubahan sebagaimana ditunjukkan pada Gambar 6. Sebelum encoding, fitur *Kelamin* dan *BB/TB* masih bertipe object, sedangkan fitur numerik lainnya, yaitu *BB*, *TB*, *ZS BB/TB*, dan *Usia* sudah bertipe numerik (float64 atau int64). Pada tahap encoding, fitur *Kelamin* diubah menjadi representasi bilangan bulat (int64) dengan pemberian kode numerik untuk tiap kategori jenis kelamin. Fitur *BB/TB* juga di-encode menjadi int64 dengan pemetaan kategori status gizi balita ke dalam nilai numerik berurutan sesuai tingkat kondisinya

5. Scaling

Tahap ini bertujuan untuk meratakan skala nilai fitur numerik agar lebih stabil dan tiap fitur berkontribusi secara proporsional. Semua fitur kecuali kelamin, BB/TB, dan ZS BB/TB dikeluarkan dari scaling. Untuk memastikan hanya prediktor numerik yang distandarisasi.

```
=== Data setelah Standarisasi ===
   Kelamin        BB        TB  ZS BB/TB      Usia  BB/TB
0        0 -0.065071 -0.439152      0.09 -0.797525      2
1        0 -0.065071  0.455230     -1.38 -0.212555      2
2        1 -0.246855 -0.176099     -0.71 -0.017565      2
3        0  1.473101  1.244390      1.39  0.177425      3
4        0 -0.065071  1.336459     -2.86  0.762395      1
```

Gambar 7. Data setelah distandarisasi

Gambar 7 menunjukkan hasil setelah dilakukan standarisasi. Nilai pada fitur numerik *BB*, *TB*, dan *Usia* tidak lagi dinyatakan dalam satuan aslinya (kilogram, sentimeter, dan bulan), tetapi diubah menjadi nilai tanpa satuan dengan skala rata-rata 0 dan simpangan baku 1. Perubahan ini terlihat pada Gambar 7, di mana angka pada kolom *BB*, *TB*, dan *Usia* sudah berada pada kisaran sekitar −3 hingga +3. Nilai negatif menunjukkan bahwa pengukuran pada sampel tersebut berada di bawah rata-rata populasi, sedangkan nilai positif menunjukkan bahwa pengukuran berada di atas rata-rata.

6. Oversampling dengan SMOTE

Tujuan dari tahap ini untuk meningkatkan representasi kelas minor tanpa adanya duplikasi data sehingga menjadikan recall per kelas lebih seimbang. SMOTE hanya dilakukan pada data train sehingga data test tetap menggunakan data asli.

```
Distribusi Kelas Sebelum SMOTE:
BB/TB
2    815
1    176
3    148
0    114
5     79
4     71
```

```
Distribusi Kelas Setelah SMOTE:
BB/TB
2    815
4    815
3    815
1    815
0    815
5    815
```

Gambar 6. Tipe data sebelum dan setelah *encoding*

Gambar 6 menunjukkan perubahan distribusi kelas pada data latih sebelum dan sesudah dilakukan oversampling dengan SMOTE. Sebelum SMOTE, distribusi label *BB/TB* pada data latih sangat tidak seimbang, dengan rincian: kelas 2 (gizi baik) sebanyak 815 sampel, kelas 1 (gizi kurang) 176 sampel, kelas 3 (risiko gizi lebih) 148 sampel, kelas 0 (gizi buruk) 114 sampel, kelas 5 (obesitas) 79 sampel, dan kelas 4 (gizi lebih) 71 sampel. Kondisi ini menunjukkan dominasi kelas gizi baik dan minimnya representasi kelas minoritas (gizi buruk, gizi lebih, dan obesitas), sehingga berpotensi menyebabkan model lebih "berpihak" pada kelas mayoritas dan mengabaikan kelas minor.

2.2. XGBoost

Proses XGBoost dimulai dengan membuat prediksi sederhana sebagai base model. Selesai melakukan prediksi awal, selanjutnya menghitung residu atau nilai error yang merupakan perbedaan antara nilai prediksi dengan dengan nilai asli dari data latih. Pohon keputusan pertama dibangun

https://doi.org/10.35145/joisie.v9i2.5601

XGBoost dalam ensemble yang berfokus mempelajari residu untuk mengurangi eror secara keseluruhan. Di tahap inilah algoritma menemukan titik pemisah terbaik dalam fitur dengan mengurangi kesalahan paling banyak. Kesalahan pada pohon sebelumnya dimanfaatkan XGBoost dengan cara menargetkan kesalahan yang tersisa untuk meningkatkan akurasi.

XGBoost mengoptimalkan *loss function* secara sistematis dengan mengukur seberapa baik prediksi model. XGBoost meminimalisir *loss function* untuk memastikan bahwa *ensemble* berada di jalur yang sesuai untuk membuat prediksi yang akurat.

1. Menghitung *Gradien* dan *Hesse Matrix*

   Gradien (turunan pertama) dan *hesse matrix* (turunan kedua) digunakan XGBoost untuk menentukan arah dan ukuran perubahan dalam model. Fungsi gradien dan hesse matrix diformulasikan sebagai.

$$g_i = \frac{\partial L}{\partial p_i} = p_i - y_i \qquad (1)$$

$$h_i = \frac{\partial^2 L}{\partial p_i^2} = p_i(1 - p_i) \qquad (2)$$

- Gradien (g_i) : besaran error yang masih tersisa untuk setiap data
- Hesse (h_i) : mengukur curam loss function untuk membantuk optimasi
- p_i: Probabilitas prediksi
- L : fungsi loss yang digunakan untuk optimasi

2. Parameter XGBoost

   Kinerja XGBoost sangat dipengaruhi oleh pemilihan parameter yang tepat. Pada awal pengujian, XGBoost dikonfigurasi sebagai baseline dengan menetapkan beberapa parameter inti agar sesuai dengan tugas klasifikasi *multiclass*, sementara parameter lain menggunakan nilai default dari *library* XGBoost. Hal ini bertujuan untuk memperoleh acuan netral sebelum dilakukan tuning *hyperparameter* (Syahputra & Saputro, 2024b).

Tabel 1. Parameter baseline XGBoost

Parameter	Nilai
Learning_rate	0.2
N_estimator	100
Max_depth	4
Subsample	1.0
Colsample_bytree	1.0

   Beberapa parameter yang penting pada XGBoost yaitu *learning Rate* (η) berfungsi untuk mengontrol ukuran langkah dalam proses pembaruan parameter, *max depth* untuk menentukan kedalaman maksimum pohon keputusan yang akan dibangun, *n estimators* digunakan untuk menentukan jumlah pohon keputusan yang digunakan dalam proses pelatihan model, *subsample* berfungsi menentukan proporsi sampel data yang akan digunakan untuk melatih setiap pohon dalam ensemble model, colsample_bytree untuk menentukan proporsi fitur yang digunakan dalam setiap pohon untuk mengurangi overfitting dan meningkatkan generalisasi model

3. *Pseudocode* XGBoost

   Untuk memperjelas alur kerja model yang diusulkan, tahapan pemodelan dirangkum dalam bentuk pseudocode. Representasi ini bertujuan memberikan gambaran terstruktur mengenai urutan proses yang dilakukan, mulai dari prapemrosesan data, pembagian data latih dan uji, penanganan ketidakseimbangan kelas dengan SMOTE, hingga tuning hyperparameter XGBoost menggunakan GridSearchCV dan evaluasi akhir pada data uji.

   A. *Preprocessing data*

      Membersihkan data dari *missing value* dan data duplikat. Lalu melakukan encoding pada fitur kategorik (kelamin, BB/TB). Pisahkan data menjadi data latih dan uji, X dan y dengan stratified split. SMOTE pada kelas minoritas hanya pada data latih.

   B. *Tuning hyperparameter*

      Definiskan ruang *hyperparameter n_estimator, learning_rate, max_depth, subsample,* dan *colsample.*

https://doi.org/10.35145/joisie.v9i2.5601

C. Pelatihan model
Latih data dari kombinasi *hyperparameter*.
D. Evaluasi pada data uji
Prediksi probabilitas kelas pada data uji. Tentukan label prediksi dari probabilitas maksimum. Hitung metrik Evaluasi.

2.3. Tuning Hyperparameter Gridsearch

Gridsearch adalah salah satu metode pencarian *hyperparameter* yang dilakukan secara sistematis dengan mencoba semua kombinasi nilai *hyperparameter* dalam ruang pencarian (*search space*) yang telah ditentukan (Nugraha & Sasongko, n.d.-a). Teknik optimasi ini cukup mudah untuk diimplementasikan dalam eksplorasi parameter yang telah ditentukan. Berikut adalah proses Gridsearch:

1. Penentuan ruang *hyperparameter*
Ruang pencarian meliputi nilai-nilai yang mungkin untuk setiap hyperparameter.

Tabel 2. Ruang hyparameter Gridsearch

Parameter	Nilai
Learning_rate	[0.05, 0.1, 0.2]
N_estimator	[100, 200, 300]
Max_depth	[4, 6, 8]
Subsample	[0.7, 0.9, 1.0]
Colsample_bytree	[0.7, 0.9, 1.0]

Penentuan ruang pencarian sangat penting karena untuk peningkatan performa model dengan mencari kombinasi parameter terbaik.

2. CV (*Cross-Validation)*
Setiap kombinasi *hyperparameter* diuji menggunakan validasi silang (*cross-validation*) untuk memastikan performa model tidak bias terhadap data tertentu (Nugraha & Sasongko, n.d.-b). *K-fold cross validation* merupakan teknik yang kuat untuk melakukan evaluasi model prediksi (Pathak et al., n.d.). Proses ini membagi dataset menjadi k subset atau fold, dimana setiap fold digunakan sebagai set validasi secara bergilir di saat k-fold yang tersisa digunakan sebagai pelatihan. Pada pengujian Gridsearch nilai k yang akan digunakan 3-fold dan 10-fold.

2.4. Library

Pada penelitian ini, seluruh proses pengolahan data, pemodelan, dan evaluasi dilakukan menggunakan bahasa pemrograman *Python* dengan bantuan sejumlah *library* pendukung. Setiap library memiliki peran spesifik dalam alur penelitian, mulai dari prapemrosesan data, penanganan ketidakseimbangan kelas, pembangunan model klasifikasi, hingga visualisasi hasil. Berikut *library* yang digunakan dalam penelitian ini.

1. *XGBoost :* untuk membangun model klasifikasi status gizi balita berbasis gradient boosting.
2. *GridSearchCV :* untuk melakukan pencarian dan pemilihan kombinasi hyperparameter XGBoost terbaik melalui validasi silang (cross-validation).
3. *Sklearn* : untuk prapemrosesan data (encoding, standardisasi, split data), perhitungan metrik evaluasi (accuracy, F1, log loss), serta skema validasi dan pemodelan pendukung lainnya.
4. *SMOTE* : untuk melakukan oversampling kelas minoritas pada data latih sehingga distribusi kelas menjadi lebih seimbang.
5. *Pandas* : untuk memuat, mengelola, dan memanipulasi data dalam bentuk tabel (*DataFrame*).
6. *Matplotlib* : untuk membuat visualisasi dasar seperti grafik log loss, kurva, dan plot sederhana lainnya.
7. *Seaborn* : membuat visualisasi statistik yang lebih informatif, misalnya heatmap *confusion matrix* dan distribusi data.

2.5. Confusion matrix

Confusion matrix digunakan sebagai alat untuk mengevaluasi kinerja dari algoritma klasifikasi (Heydarian et al., 2022). Matriks ini menggambarkan perbandingan antara hasil prediksi model dengan label kelas sebenarnya, sehingga memungkinkan analisis performa secara komprehensif.

https://doi.org/10.35145/joisie.v9i2.5601

Setiap entrik dalam matrik menunjukkan frekuensi klasifikasi dari kombinasi prediksi dan kondisi aktual. Elemen-elemen penting dari *Confusion matrix* meliputi *True Positive* (TP), *False Positive* (FP), *True Negative* (TN), *False Negative* (FN).

1. Akurasi

   Akurasi merupakan metrik dasar evaluasi model yang sering digunakan dalam pelatihan model. Akurasi dihitung berdasarkan rasio antara jumlah prediksi yang benar dengan jumlah total sampel yang telah diuji. Berikut persamaannnya.

$$Accuracy = \frac{TP+TN}{TP\ +FP+FN+TN} \quad (3)$$

   Keterangan :

- TP (*True Positive*) = Jumlah sampel positif yang diprediksi benar.
- TN (*True Negative*) = Jumlah sampel negatif yang diprediksi benar.
- FP (*False Positive*) = Jumlah sampel negatif yang diprediksi sebagai positif.
- FN (*False Negative*) = Jumlah sampel positif yang diprediksi sebagai negatif.

2. Presisi

   Presisi berfungsi untuk mengukur seberapa banyak sampel yang diprediksi positif benar dibandingkan dengan seluruh sampel yang diprediksi positif. Berikut persamaanya.

$$Precision\ = \frac{TP}{TP + FP} \quad (4)$$

3. *Recall*

   Recall merupakan pengukur sensitivitas seberapa banyak sampel positif yang telah terdeteksi oleh model yang telah dilatih. Berikut persamaan *recall*.

$$Recall\ = \frac{TP}{TP + FN} \quad (5)$$

4. *F1-score*

   F1-score merupakan metrik gabungan dari presisi dan *recall* yang dihitung sebagai rata-rata dari keduanya. Berikut persamaan *f1-score.*

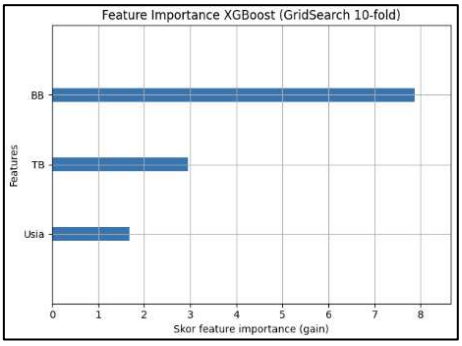
$$F1 - score\ = 2\ x\ \frac{Presisi\ x\ Recall}{Presisi + Recall} \quad (6)$$

3. HASIL DAN PEMBAHASAN

Pembahasan difokuskan pada interpretasi teknis dari setiap metrik, perbandingan antar skenario pengujian, serta implikasinya terhadap kemampuan model dalam mengklasifikasikan seluruh kelas status gizi secara lebih seimbang dan andal.

3.1 Feature Importance

Gambar 8. dapat dilihat bahwa fitur BB memiliki kontribusi paling besar terhadap pembentukan keputusan model, diikuti oleh TB, sedangkan Usia memberikan pengaruh yang relatif lebih kecil. Dominasi BB menunjukkan bahwa variasi berat badan antar balita merupakan informasi yang paling informatif untuk membedakan kelas status gizi, dengan TB berperan sebagai pendukung pola tersebut.



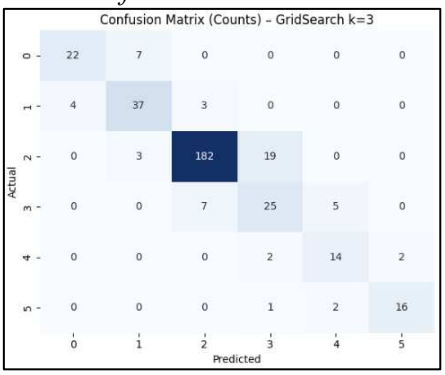
Gambar 8. Grafik *feature importance*

https://doi.org/10.35145/joisie.v9i2.5601

Dengan demikian, model memanfaatkan terutama kombinasi berat dan tinggi badan untuk memisahkan kelas gizi, sedangkan usia lebih berfungsi sebagai variabel penjelas tambahan.

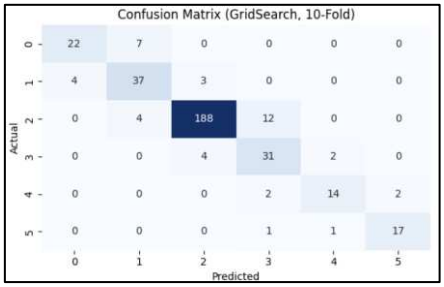
3.2 Evaluasi model

Evaluasi difokuskan pada kemampuan model dalam mengklasifikasikan enam kelas status gizi secara seimbang, yang dinilai melalui *confusion matrix* dan classification report.



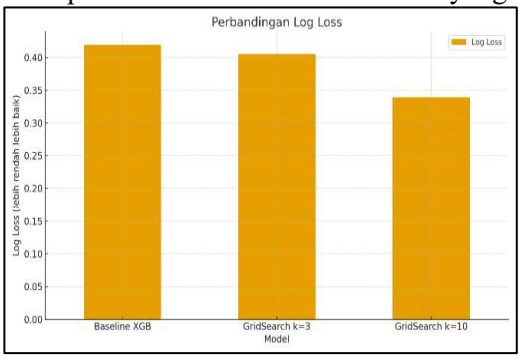
Gambar 9. *Confusion matrix* uji *GS CV 3-fold*

Gambar 9. menunjukkan bahwa mayoritas sampel berada pada diagonal utama sehingga sebagian besar kelas terprediksi dengan benar. Kelas 0, 1, 2, 3, 4, dan 5 masing-masing didominasi oleh prediksi yang tepat, dengan hanya sebagian kecil sampel yang bergeser ke kelas lain. Kesalahan yang muncul terutama terjadi antar kelas yang berurutan seperti kelas 0 ke 1, kelas 2 ke 3 yang mengindikasikan bahwa model masih sulit membedakan batas di antara kelas yang memiliki karakteristik fitur sangat mirip. Meskipun demikian, sebaran nilai diagonal yang relatif besar menunjukkan bahwa performa klasifikasi model secara keseluruhan sudah cukup baik.



Gambar 10. *Confusion matrix* uji *GS CV 10-fold*

Gambar 10. menunjukkan bahwa sebagian besar sampel berada pada diagonal utama sehingga setiap kelas didominasi oleh prediksi yang benar. Kelas 0, 1, 2, 3, 4, dan 5 masing-masing memiliki 22, 37, 188, 31, 14, dan 17 prediksi tepat, dengan kesalahan relatif sedikit dan umumnya hanya bergeser ke kelas yang berdekatan. Pola ini mengindikasikan bahwa model semakin mampu membedakan antar kelas dibandingkan konfigurasi 3-fold, sementara sisa kesalahan terutama disebabkan oleh kemiripan karakteristik fitur di batas antar kelas yang berurutan. Secara keseluruhan, sebaran nilai diagonal yang sangat dominan pada *confusion matrix* ini menegaskan bahwa konfigurasi GridSearch 10-fold memberikan performa klasifikasi multi-kelas yang lebih stabil dan akurat.



Gambar 11. Perbandingan Log loss

https://doi.org/10.35145/joisie.v9i2.5601

Gambar 11. menunjukkan bahwa pada baseline logloss berada di 0.4190. Sedangkan dengan tuning GridSearch 3-fold CV, log loss mengalami penurunan menjadi 0.4050 yang berarti penurunan dari XGBoost baseline ke XGBoost GridSearch 3-fold CV terjadi sebanyak 3,3%. Pada pengujian XGBoost GridSearch 10-fold CV, log loss berada di 0.3395. Penurunan dari baseline ke GridSearch 10-fold CV terjadi sebanyak 19%. Hal ini menandakan bahwa probabilitas prediksi lebih terkalibrasi setelah tuning, terutama pada 10-fold CV.

3.3 Hasil pengujian

Pada pengujian, model XGBoost tuning GridSearch 10-fold CV juga memberikan hasil terbaik jika dibandingkan dengan model *baseline* XGBoost. Pada nilai akurasi, jarak antara baseline XGBoost ke XGBoost GridSearch 10-fold CV mengalami peningkatan sebanyak 0.0398, pada presisi juga mengalami peningkatan sebanyak 0.0441, recall meningkat sebanyak 0.0362, f1-score meningkat sebanyak 0.0449. Hal ini membuat model 10-fold CV lebih unggul karena CV yang lebih rapat mengakibatkan nilai tuning lebih stabil, sehingga pemilihan hyperparameter lebih tepat.

Tabel 2. Hasil perbandingan *classification report*

Model	acc	*precision*	*recall*	*f1-score*
XGB	0.8291	0.7595	0.7939	0.7709
XGB+GS 3-fold CV	0.8433	0.7781	0.7979	0.7851
XGB + RS 3-fold CV	0.8376	0.7762	0.7497	0.7804
XGB+GS 10-fold CV	**0.8746**	**0.8134**	**0.8317**	**0.8208**
XGB + RS 10-fold CV	0.8462	0.7936	0.8014	0.7936

Peningkatan ini menunjukkan bahwa GridSearch meningkatkan kinerja XGBoost dibandingkan dengan baseline pada klasifikasi gizi balita antropometeri BB/TB, terutama pada 10-fold CV paling konsisten. Namun dari sisi biaya komputasi, GridSearch 10-fold CV memerlukan waktu tuning jauh lebih besar. Hal ini menunjukkan bahwa GridSearchCV memang memberikan kinerja terbaik, namun harus mempertimbangkan biaya komputasinya.

4. SIMPULAN

GridSearch terbukti mampu meningkatkan kinerja algoritma XGBoost pada klasifikasi status gizi balita berbasis indikator BB/TB. Model dengan kinerja paling konsisten dan unggul adalah XGBoost yang dituning menggunakan GridSearchCV dengan validasi silang 10-fold. Pada konfigurasi ini, akurasi akhir model mencapai 0,8689 dengan nilai log loss sebesar 0,3395, lebih rendah dibandingkan model baseline. Cross-validation yang lebih rapat membuat pemilihan hyperparameter menjadi lebih andal, meskipun harus dibayar dengan waktu komputasi tuning yang jauh lebih besar, sementara waktu pelatihan akhir (final fit) relatif tetap singkat.

Penelitian ini masih terbatas pada data balita dari satu puskesmas dan satu periode pengukuran, sehingga generalisasi model ke wilayah lain perlu diuji lebih lanjut. Selain itu, model yang diuji berfokus pada XGBoost sehingga perbandingan langsung dengan algoritma klasifikasi lain belum dilakukan. Model yang dihasilkan berpotensi dimanfaatkan sebagai alat bantu keputusan bagi tenaga gizi di puskesmas untuk menilai status gizi balita berbasis indikator BB/TB secara lebih konsisten, cepat, dan terukur. Untuk penelitian selanjutnya, disarankan mengombinasikan GridSearchCV dengan metode tuning hyperparameter lain, seperti Random Search atau Bayesian Optimization, guna mengurangi waktu komputasi tanpa mengorbankan kinerja model.

5. DAFTAR PUSTAKA

Abdurrahman, G., Oktavianto, H., & Sintawati, M. (2022). Optimasi Algoritma XGBoost Classifier Menggunakan Hyperparameter Gridesearch dan Random Search Pada Klasifikasi Penyakit Diabetes. In *Informatics Journal* (Vol. 7, Issue 3).

Amalia Alhamid, S., Tiara Carolin, B., Lubis, R., Studi Kebidanan, P., Ilmu Kesehatan, F., & Nasional Jakarta, U. (2021). *Studi Mengenai Status Gizi Balita* (Vol. 7, Issue 1).

https://doi.org/10.35145/joisie.v9i2.5601

Anggoro, D. A., & Mukti, S. S. (2021). Performance Comparison of Grid Search and Random Search Methods for Hyperparameter Tuning in Extreme Gradient Boosting Algorithm to Predict Chronic Kidney Failure. *International Journal of Intelligent Engineering and Systems*, *14*(6), 198–207. https://doi.org/10.22266/ijies2021.1231.19

Bahar, Muh. A., Galistiani, G. F., Eliyanti, U., & Mohi, A. R. (2024). Gambaran Nilai Utilitas Kesehatan Anak dengan Malnutrisi : Studi pada Kasus Stunting, Wasting, dan Underweight di Indonesia. *Jurnal Mandala Pharmacon Indonesia*, *10*(2), 610–617. https://doi.org/10.35311/jmpi.v10i2.656

Çetin, V., & Yıldız, O. (2022). A comprehensive review on data preprocessing techniques in data analysis. *Pamukkale University Journal of Engineering Sciences*, *28*(2), 299–312. https://doi.org/10.5505/pajes.2021.62687

Chiang, H. Y., Liang, L. Y., Lin, C. C., Chen, Y. J., Wu, M. Y., Chen, S. H., Wu, P. H., Kuo, C. C., & Chi, C. Y. (2021). Electronic medical record-based deep data cleaning and phenotyping improve the diagnostic validity and mortality assessment of infective endocarditis: Medical big data initiative of CMUH. *BioMedicine (Taiwan)*, *11*(3), 59–67. https://doi.org/10.37796/2211-8039.1267

Erlin, E., Desnelita, Y., Nasution, N., Suryati, L., & Zoromi, F. (2022). Dampak SMOTE terhadap Kinerja Random Forest Classifier berdasarkan Data Tidak seimbang. *MATRIK : Jurnal Manajemen, Teknik Informatika Dan Rekayasa Komputer*, *21*(3), 677–690. https://doi.org/10.30812/matrik.v21i3.1726

Fan, C., Chen, M., Wang, X., Wang, J., & Huang, B. (2021). A Review on Data Preprocessing Techniques Toward Efficient and Reliable Knowledge Discovery From Building Operational Data. In *Frontiers in Energy Research* (Vol. 9). Frontiers Media S.A. https://doi.org/10.3389/fenrg.2021.652801

Fibrinika Tuta Setiani, Heny Lestari, & Abdullah Azam Mustajab. (2024). Nutritional Status of Toddlers Aged 0-59 Months: a Descriptive Study. *International Journal of Health and Medicine*, *1*(4), 156–164. https://doi.org/10.62951/ijhm.v1i4.101

Goswami, P., & Bhatia, D. (2023). Application of Machine Learning in FPGA EDA Tool Development. *IEEE Access*, *11*, 109564–109580. https://doi.org/10.1109/ACCESS.2023.3322358

Handayani, S., Toresa, D., Informatika, T., Ilmu Komputer, F., & Lancang Kuning, U. (n.d.-a). *Peningkatan Performa Model Gradient Boosting dalam Klasifikasi Stroke Melalui Optimasi Grid Search*.

Handayani, S., Toresa, D., Informatika, T., Ilmu Komputer, F., & Lancang Kuning, U. (n.d.-b). *Peningkatan Performa Model Gradient Boosting dalam Klasifikasi Stroke Melalui Optimasi Grid Search*.

Heydarian, M., Doyle, T. E., & Samavi, R. (2022). MLCM: Multi-Label Confusion Matrix. *IEEE Access*, *10*, 19083–19095. https://doi.org/10.1109/ACCESS.2022.3151048

Lidya, N., Fakultas, S., Klabat, K. U., & Mononutu, J. A. (2021). *Hubungan Antara Status Sosial Ekonomi Dengan Status Gizi Balita Di Kelurahan Buha Kecamatan Mapanget Kota Manado* (Vol. 3, Issue 1). http://ejournal.unklab.ac.id/index.php/kjn

Liew, X. Y., Hameed, N., & Clos, J. (2021). An investigation of XGBoost-based algorithm for breast cancer classification. *Machine Learning with Applications*, *6*, 100154. https://doi.org/10.1016/j.mlwa.2021.100154

Masturina, M. L., Salam, A., Indriasari, R., Razak Thaha, A., Jafar, N., Studi Ilmu Gizi, P., & Kesehatan Masyarakat, F. (2023). Description of family characteristics and nutritional status in toddlers Gambaran karakteristik keluarga dan status gizi balita. *Community Research of Epidemiology Journal*, *3*(2). https://doi.org/10.24252/corejournal.v%vi%i.37731

Muhamad Fikri. (2024). Klasifikasi Status Stunting Pada Anak Bawah Lima Tahun Menggunakan Extreme Gradient Boosting. *Merkurius : Jurnal Riset Sistem Informasi Dan Teknik Informatika*, *2*(4), 173–184. https://doi.org/10.61132/merkurius.v2i4.159

Nugraha, W., & Sasongko, A. (n.d.-a). *SISTEMASI: Jurnal Sistem Informasi Hyperparameter Tuning pada Algoritma Klasifikasi dengan Grid Search Hyperparameter Tuning on Classification Algorithm with Grid Search* (Vol. 11, Issue 2). http://sistemasi.ftik.unisi.ac.id

Nugraha, W., & Sasongko, A. (n.d.-b). *SISTEMASI: Jurnal Sistem Informasi Hyperparameter Tuning pada Algoritma Klasifikasi dengan Grid Search Hyperparameter Tuning on Classification Algorithm with Grid Search* (Vol. 11, Issue 2). http://sistemasi.ftik.unisi.ac.id

Ogunsanya, M., Isichei, J., & Desai, S. (2023). *Manufacturing Letters Grid Search Hyperparameter Tuning in Additive Manufacturing Processes-NC-ND license (https://creativecommons.org/licenses/by-nc-nd/4.0) Peer-review under responsibility of the Scientific Committee of the NAMRI/SME*. www.sciencedirect.com

Paramita, I. S., Atasasih, H., & Rahayu, D. (2024). *Penilaian Status Gizi Antropometri Pada Balita Penerbit Salnesia (CV. Sarana Ilmu Indonesia)*.

Pathak, A. K., Chaubey, M., & Gupta, M. (n.d.). *Randomized-Grid Search for Hyperparameter Tuning in Decision Tree Model to Improve Performance of Cardiovascular Disease Classification*.

Puerwandono, E., & Maulana, I. (2023). Penerapan Algoritma Svm Untuk Klasifikasi Citra Daun Sirih Application Of The Svm Algorithm To Image Classification Of Betel Leaf. *Journal of Information Technology and Computer Science (INTECOMS)*, *6*(2).

Syahputra, A. A., & Saputro, R. E. (2024a). Application of the XGBoost Model with Hyperparameter Tuning for Industry Classification for Job Applicants. *Sinkron*, *8*(3), 1920–1931. https://doi.org/10.33395/sinkron.v8i3.13840

Syahputra, A. A., & Saputro, R. E. (2024b). Application of the XGBoost Model with Hyperparameter Tuning for Industry Classification for Job Applicants. *Sinkron*, *8*(3), 1920–1931. https://doi.org/10.33395/sinkron.v8i3.13840

Tran, N. T., Tran, T. T. G., Nguyen, T. A., & Lam, M. B. (2023). A new grid search algorithm based on XGBoost model for load forecasting. *Bulletin of Electrical Engineering and Informatics*, *12*(4), 1857–1866. https://doi.org/10.11591/eei.v12i4.5016

Uzir, N., Raman, S., Banerjee, S., & Nishant Uzir Sunil R, R. S. (2016). Experimenting XGBoost Algorithm for Prediction and Classification of Different Datasets Experimenting XGBoost Algorithm for Prediction and Classifi cation of Different Datasets. *International Journal of Control Theory and Applications*, *9*. https://www.researchgate.net/publication/318132203

Zhang, Y., & Haghani, A. (2015). A gradient boosting method to improve travel time prediction. *Transportation Research Part C: Emerging Technologies*, *58*, 308–324. https://doi.org/10.1016/j.trc.2015.02.019

https://doi.org/10.35145/joisie.v9i2.5601