

## Analisis Model Prediksi Penyakit Jantung Menggunakan *Adaptive Boosting*, *Gradient Boosting*, dan *Extreme Gradient Boosting*

Andrian Sah<sup>1\*</sup>, Chaeroen Niesa<sup>2</sup>, Rhaishudin Jafar Rumandan<sup>3</sup>, Muhammad Muharrom<sup>4</sup>

<sup>1</sup>Program Studi Sistem Informasi, Fakultas Ilmu Komputer, Universitas Yapis Papua

<sup>2</sup>Program Studi Informatika, Fakultas Komputer dan Multimedia, Universitas Islam Kebangsaan Indonesia

<sup>3</sup>Program Studi Manajemen Pendidikan Islam, Institut Agama Islam Negeri Ambon

<sup>4</sup>Program Studi Teknologi Informasi, Fakultas Teknik dan Informatika, Universitas Bina Sarana Informatika

Email: <sup>1</sup>cyberdefance23@gmail.com, <sup>2</sup>jeumalaniesa@gmail.com, <sup>3</sup>jafarrumadan@gmail.com,

<sup>4</sup>muhammad.muu@bsi.ac.id

Penulis Korespondensi\*

(received: 22-11-24, revised: 17-03-25, accepted: 07-05-25)

### Abstrak

Deteksi dini penyakit jantung merupakan langkah penting untuk meningkatkan kualitas diagnosis dan perawatan pasien. Namun, metode prediksi manual yang sering digunakan tenaga medis memiliki keterbatasan dalam efisiensi waktu, akurasi, dan kemampuan menangani volume data yang besar. Dalam bidang kecerdasan buatan, algoritma machine learning seperti *Adaptive Boosting* (AdaBoost), *Gradient Boosting*, dan *Extreme Gradient Boosting* (XGBoost) menawarkan potensi untuk meningkatkan akurasi prediksi, terutama dalam mengatasi tantangan pada dataset kecil yang sering mengalami ketidakseimbangan kelas dan risiko *overfitting*. Penelitian ini bertujuan untuk menganalisis kinerja ketiga algoritma *boosting* tersebut dalam memprediksi penyakit jantung. Hasil penelitian menunjukkan bahwa XGBoost memberikan performa terbaik dengan akurasi sebesar 84.78% dan ROC-AUC 0.9410, menjadikannya algoritma paling efektif dalam menangani pola data yang kompleks. *Gradient Boosting* menjadi model paling efisien dengan waktu pelatihan tercepat, yaitu 0.3655 detik, dengan akurasi dan ROC-AUC yang kompetitif. Sementara itu, AdaBoost menunjukkan kelemahan dalam menangani ketidakseimbangan kelas tetapi tetap memberikan hasil yang baik untuk kelas mayoritas. Berdasarkan evaluasi *precision*, *recall*, dan *F1-score*, XGBoost direkomendasikan untuk aplikasi prediksi penyakit jantung, terutama dalam situasi yang memerlukan akurasi tinggi, sedangkan *Gradient Boosting* cocok untuk kebutuhan *real-time*.

**Kata Kunci:** Prediksi Penyakit Jantung, *Machine Learning*, *Adaptive Boosting*, *Gradient Boosting*, *Extreme Gradient Boosting*.

### Abstract

*Early detection of heart disease is a crucial step in improving the quality of diagnosis and patient care. However, manual prediction methods commonly used by medical professionals face limitations in terms of time efficiency, accuracy, and the ability to handle large volumes of data. In the field of artificial intelligence, machine learning algorithms such as Adaptive Boosting (AdaBoost), Gradient Boosting, and Extreme Gradient Boosting (XGBoost) offer significant potential to improve prediction accuracy, especially in addressing challenges associated with small datasets that often suffer from class imbalance and the risk of overfitting. This study aims to analyze the performance of these three boosting algorithms in predicting heart disease. The results show that XGBoost delivers the best performance, achieving an accuracy of 84.78% and an ROC-AUC of 0.9410, making it the most effective algorithm for handling complex data patterns. Gradient Boosting emerges as the most efficient model with the fastest training time of 0.3655 seconds, while maintaining competitive accuracy and ROC-AUC. Meanwhile, AdaBoost exhibits weaknesses in handling class imbalance but still performs well for the majority class. Based on the evaluation of precision, recall, and F1-score, XGBoost is recommended for heart disease prediction applications, particularly in scenarios requiring high accuracy, while Gradient Boosting is more suitable for real-time needs.*

**Keywords:** Heart Disease Prediction, *Machine Learning*, *Adaptive Boosting*, *Gradient Boosting*, *Extreme Gradient Boosting*.

## 1. PENDAHULUAN

Penyakit jantung adalah salah satu penyebab utama kematian di seluruh dunia. Menurut laporan dari World Health Organization (WHO), penyakit ini bertanggung jawab atas lebih dari 17 juta kematian setiap tahun [1]. Deteksi dini penyakit jantung memiliki peran krusial dalam mengurangi angka kematian, karena memungkinkan pasien untuk mendapatkan perawatan yang lebih cepat dan lebih efektif [2]. Di sisi lain, metode prediksi manual yang dilakukan oleh tenaga medis sering kali memerlukan waktu yang cukup lama dan bergantung pada keahlian individu dalam menganalisis data pasien. Proses ini tidak hanya berisiko menghasilkan kesalahan akibat faktor subjektivitas, tetapi juga kurang efektif dalam menangani volume data yang besar, terutama dengan semakin bertambahnya jumlah pasien yang membutuhkan perhatian.

Dalam beberapa tahun terakhir, teknologi kecerdasan buatan, khususnya *machine learning*, telah digunakan secara luas untuk mendukung diagnosis penyakit jantung dengan tingkat akurasi yang semakin tinggi [3]–[5]. Di antara berbagai algoritma *machine learning*, metode *boosting* telah menjadi salah satu pendekatan yang paling populer dan efektif dalam meningkatkan akurasi model prediksi [6]. *Boosting* adalah teknik *ensemble* yang menggabungkan beberapa model sederhana (*weak learners*) untuk membangun model prediksi yang lebih kuat [7]. Algoritma seperti *Adaptive Boosting* (AdaBoost), *Gradient Boosting*, dan *Extreme Gradient Boosting* (XGBoost) memiliki karakteristik unik yang menjadikannya unggul dalam menangani data medis yang kompleks. AdaBoost menggunakan pendekatan iteratif untuk menyesuaikan bobot data [8], sementara *Gradient Boosting* mengoptimalkan kesalahan melalui gradien fungsi loss [9]. XGBoost, sebagai pengembangan lebih lanjut, menawarkan efisiensi waktu pelatihan dan kemampuan regularisasi yang lebih baik [10].

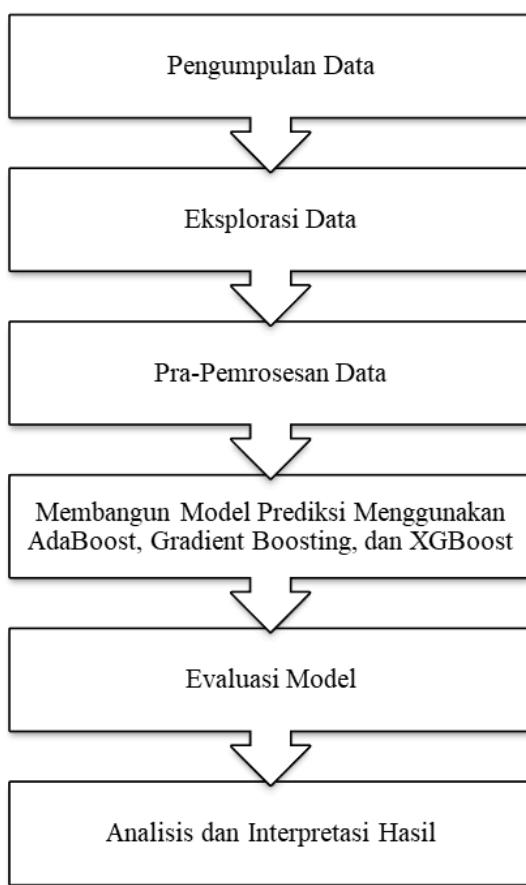
Namun, analisis model *machine learning* pada dataset medis, termasuk dataset penyakit jantung, menghadapi tantangan yang signifikan. Salah satu tantangan utama adalah jumlah dataset yang terbatas, yang sering kali hanya mencakup ratusan entri. Dataset kecil dapat menyebabkan model *machine learning* sulit untuk melakukan generalisasi, sehingga meningkatkan risiko *overfitting* [11]. Selain itu, dataset medis sering kali memiliki ketidakseimbangan kelas antara pasien dengan penyakit jantung dan yang tidak, yang dapat memengaruhi kemampuan model untuk memprediksi kelas minoritas dengan akurat. Karakteristik lain seperti keberadaan fitur yang redundan atau tidak relevan juga dapat mengurangi efisiensi model, terutama pada dataset kecil.

Metode *boosting* seperti AdaBoost, *Gradient Boosting*, dan XGBoost memiliki potensi untuk mengatasi tantangan ini. *Boosting* bekerja dengan membangun model secara bertahap, sehingga memungkinkan model untuk fokus pada kesalahan yang terjadi pada iterasi sebelumnya [12]. Selain itu, kemampuan *boosting* dalam mengoptimalkan fungsi loss dapat membantu dalam menangani ketidakseimbangan kelas pada dataset kecil. Namun, sejauh ini, studi yang membandingkan kinerja ketiga algoritma tersebut secara spesifik pada dataset kecil seperti dataset penyakit jantung masih sangat terbatas.

Penelitian ini bertujuan untuk menganalisis kinerja tiga algoritma boosting yaitu *Adaptive Boosting* (AdaBoost), *Gradient Boosting*, dan *Extreme Gradient Boosting* (XGBoost) dalam memprediksi penyakit jantung. Studi ini tidak hanya mengevaluasi akurasi, tetapi juga mempertimbangkan efisiensi waktu pelatihan, stabilitas model, dan kemampuannya untuk generalisasi pada dataset kecil. Dengan demikian, hasil dari penelitian ini memberikan kontribusi berupa rekomendasi algoritma yang optimal untuk prediksi penyakit jantung berdasarkan kinerja akurasi, efisiensi waktu, dan kemampuan generalisasi. Selain itu, penelitian dapat memberikan wawasan baru bagi pengembang sistem pendukung keputusan medis dalam memilih algoritma *machine learning* yang sesuai untuk dataset terbatas dan karakteristik medis yang kompleks.

## 2. METODOLOGI PENELITIAN

Metode penelitian merupakan kerangka kerja sistematis yang digunakan untuk mencapai tujuan penelitian secara terstruktur dan terorganisasi [13], [14]. Penelitian ini dilakukan dengan menggunakan pendekatan berbasis *machine learning* untuk menganalisis dan membandingkan kinerja algoritma *Adaptive Boosting* (AdaBoost), *Gradient Boosting*, dan *Extreme Gradient Boosting* (XGBoost) dalam memprediksi penyakit jantung. Metodologi penelitian mencakup beberapa tahapan utama yang ditampilkan pada Gambar 1.



Gambar 1. Prosedur Penelitian

Merujuk pada Gambar 1, setiap tahapan yang dilakukan dijelaskan secara rinci sebagai berikut.

## 2.1. Pengumpulan Data

Tahapan pertama dalam penelitian ini adalah pengumpulan dataset yang relevan untuk analisis. Pada penelitian ini, dataset yang digunakan adalah *Heart Disease Dataset* yang diperoleh dari *platform* Kaggle (<https://www.kaggle.com/datasets/yasserh/heart-disease-dataset>) [15]. Dataset ini berisi data klinis pasien yang digunakan untuk memprediksi kemungkinan penyakit jantung berdasarkan berbagai fitur yang relevan. Dataset ini memiliki 303 entri dengan 14 kolom, termasuk variabel target bernama *target*, yang menunjukkan ada atau tidaknya penyakit jantung pada pasien (1 untuk penyakit jantung, 0 untuk tidak).

*Heart Disease Dataset* yang digunakan dalam penelitian ini mencakup berbagai fitur klinis yang relevan untuk diagnosis penyakit jantung. Fitur-fitur tersebut meliputi *age*, yaitu usia pasien, dan *sex*, yang menunjukkan jenis kelamin pasien dengan nilai 1 untuk laki-laki dan 0 untuk perempuan. Selain itu, dataset ini mencatat *cp* (tipe nyeri dada) dengan kategori 1 hingga 4, serta *trestbps* (tekanan darah istirahat dalam mmHg) dan *chol* (kadar kolesterol serum dalam mg/dL). Dataset juga menyertakan *fbs* (gula darah puasa lebih dari 120 mg/dL, dengan nilai 1 menunjukkan "true" dan 0 "false") serta *restecg*, hasil dari pemeriksaan elektrokardiografi saat istirahat. Variabel *thalachh* mencatat detak jantung maksimum yang dicapai, sementara *exng* menunjukkan angina yang diinduksi oleh olahraga (1 untuk "ya" dan 0 untuk "tidak"). Selain itu, dataset memuat informasi *oldpeak* (depresi ST yang diinduksi oleh olahraga relatif terhadap kondisi istirahat), *slope* (kemiringan segmen ST saat puncak latihan), dan *caa* (jumlah pembuluh darah utama yang diwarnai oleh fluoroskopi, dengan rentang nilai 0–4). Variabel selanjutnya adalah *thall*, hasil tes thalium stres yang dikategorikan sebagai 1 hingga 3. Variabel terakhir yaitu *target*, yang menunjukkan ada atau tidaknya penyakit jantung, dengan nilai 1 menunjukkan adanya penyakit jantung dan 0 menunjukkan tidak adanya penyakit. Fitur-fitur ini memberikan dasar yang kuat untuk membangun model prediksi menggunakan algoritma machine learning.

## 2.2. Eksplorasi Data

Setelah dataset berhasil dikumpulkan, langkah berikutnya adalah melakukan eksplorasi data untuk memahami struktur, distribusi, serta pola yang terdapat dalam dataset. Tahap ini bertujuan untuk

mengidentifikasi pola awal yang berpotensi memengaruhi hasil prediksi. Berbagai visualisasi, seperti *boxplot*, *heatmap*, dan *pair plot*, digunakan untuk mengevaluasi distribusi variabel numerik, hubungan antar fitur, serta korelasi antar variabel. Hasil dari eksplorasi ini menjadi panduan penting dalam menentukan langkah-langkah pra-pemrosesan data yang tepat untuk memastikan kualitas data dan meningkatkan performa model prediksi.

### 2.3. Pra-pemrosesan Data

Tahapan ini bertujuan untuk mempersiapkan data agar siap digunakan dalam model machine learning dengan memastikan kualitas dan konsistensinya. Pada tahap pra-pemrosesan dilakukan penanganan nilai yang hilang (*missing values*), di mana nilai yang kosong diisi menggunakan metode imputasi, seperti median untuk data numerik atau modus untuk data kategori. Selanjutnya, dilakukan penanganan *outlier* menggunakan metode IQR (*Interquartile Range*) untuk mendeteksi dan menghapus nilai-nilai ekstrem yang dapat mengganggu stabilitas model. Data numerik kemudian dinormalisasi menggunakan teknik standarisasi untuk menyamakan skala fitur, sehingga mencegah dominasi fitur dengan skala besar terhadap model. Sebagai langkah terakhir, dataset dibagi menjadi data *training* dan data *testing* menggunakan rasio yang sesuai (misalnya 80:20), agar model dapat diuji dengan data yang belum pernah dilihat sebelumnya. Semua langkah ini dirancang untuk memastikan bahwa data bersih, relevan, dan siap untuk menghasilkan model prediksi yang akurat dan andal.

### 2.4. Membangun Model Prediksi Menggunakan AdaBoost, Gradient Boosting, dan XGBoost

Tahapan ini melibatkan pembangunan model prediksi dengan tiga algoritma *boosting*, yaitu *Adaptive Boosting* (AdaBoost), *Gradient Boosting*, dan *Extreme Gradient Boosting* (XGBoost). AdaBoost adalah algoritma *boosting* yang bekerja dengan menggabungkan beberapa model sederhana (*weak learners*), biasanya berupa *decision tree* dengan kedalaman satu (*stump*), untuk membentuk model yang lebih kuat (*strong learner*) [16]. Pada setiap iterasi, data yang salah diprediksi oleh model sebelumnya diberi bobot lebih besar sehingga model berikutnya dapat fokus pada kesalahan tersebut [17]. Proses ini berlanjut hingga jumlah iterasi yang telah ditentukan. Pembaruan bobot untuk setiap data digunakan persamaan (1).

$$w_i^{(t+1)} = w_i^{(t)} \cdot e^{\alpha_t \cdot I(y_i \neq h_t(x_i))} \quad (1)$$

di mana  $F_m(x)$  adalah bobot data,  $\alpha_t$  adalah kontribusi model  $t$  terhadap prediksi akhir, dan  $I(y_i \neq h_t(x_i))$  adalah fungsi indikator untuk kesalahan prediksi.

Prediksi akhir dilakukan dengan kombinasi tertimbang dari semua weak learners menggunakan persamaan (2). Dengan syarat bahwa semua titik data harus dipisahkan oleh *hyperplane* dengan margin yang benar, maka digunakan persamaan (2).

$$H(x) = \text{sign} \left( \sum_{t=1}^T \alpha_t h_t(x) \right) \quad (2)$$

di mana  $H(x)$  merupakan prediksi akhir,  $\alpha_t$  merupakan bobot model ke- $t$ , dan  $h_t(x)$  menunjukkan prediksi dari weak learner ke- $t$ .

*Gradient Boosting* adalah algoritma boosting yang mengoptimalkan fungsi *loss* dengan cara menambahkan model baru secara bertahap untuk memperbaiki kesalahan residu [18]. Berbeda dengan AdaBoost yang mengubah bobot data, *Gradient Boosting* mengurangi kesalahan residu dari model sebelumnya dengan mempelajari gradien dari fungsi *loss* [19]. Pada metode *Gradient Boosting* pembaruan model menggunakan persamaan (3).

$$F_m(x) = F_{m-1}(x) + \eta \cdot h_m(x) \quad (3)$$

di mana  $F_m(x)$  adalah bobot data ke- $m$ ,  $\eta$  merupakan learning rate yang mengontrol kontribusi model baru, dan  $h_m(x)$  menunjukkan weak learner baru yang mempelajari kesalahan residu.

Residunya dihitung menggunakan gradien dari fungsi loss melalui persamaan (4).

$$h_m(x) = -\nabla_F L(y, F_{m-1}(x)) \quad (4)$$

dimana  $-\nabla_F L(y, F_{m-1}(x))$  menunjukkan gradien dari fungsi loss, dan  $L(y, F(x))$  menunjukkan arah perbaikan XGBoost adalah pengembangan dari *Gradient Boosting* yang dirancang untuk lebih cepat dan efisien

dengan tambahan regularisasi untuk mencegah *overfitting* [19]. XGBoost menggunakan pendekatan berbasis *Taylor expansion* untuk mengoptimalkan fungsi loss, melalui persamaan (5).

$$L^{(t)} \approx \sum_{i=1}^n \left[ g_i \cdot h_t(x_i) + \frac{1}{2} h_i \cdot h_t^2(x_i) \right] + \Omega(h_t) \quad (5)$$

di mana  $\Omega(h_t)$  adalah fungsi regularisasi untuk mengontrol kompleksitas model.

Regularisasi pada XGBoost membuat lebih tahan terhadap *overfitting* dibanding *Gradient Boosting* [20]. Prediksi akhir XGBoost dilakukan dengan menjumlahkan output dari semua *tree* yang telah dibangun melalui persamaan (6).

$$F(x) = \sum_{k=1}^K T_k(x) \quad (6)$$

di mana  $T_k(x)$  adalah tree ke- $k$ .

## 2.5. Evaluasi Model

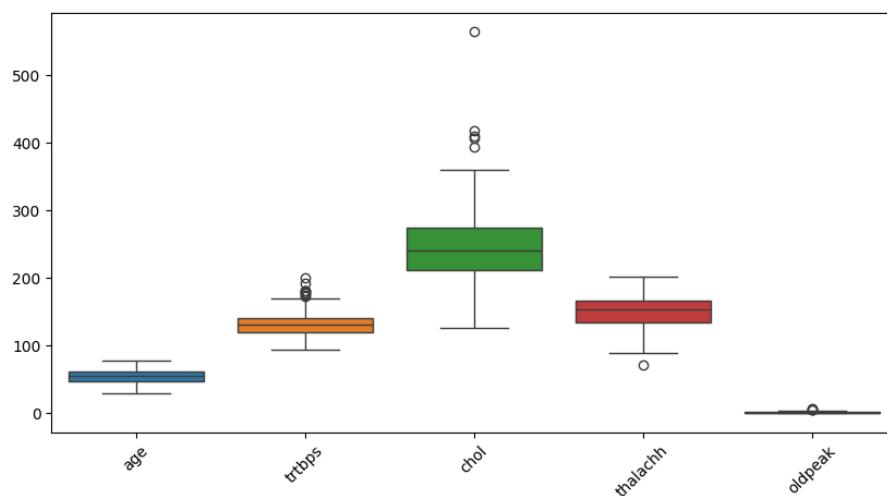
Evaluasi model dilakukan untuk menilai kinerja setiap algoritma berdasarkan metrik evaluasi yang relevan [21], [22]. Proses ini mencakup pengujian model menggunakan data uji yang tidak terlibat dalam tahap pelatihan [23]. Untuk mengukur kemampuan model evaluasi melibatkan penggunaan *confusion matrix* untuk memvisualisasikan distribusi prediksi model [24]. Berbagai metrik yang digunakan untuk mengukur kinerja, termasuk akurasi untuk mengukur persentase prediksi yang tepat, *precision* untuk menghitung proporsi prediksi positif yang benar, *recall* untuk menilai kemampuan model dalam mendeteksi kelas positif, *F1-score* sebagai rata-rata harmonis antara *precision* dan *recall*, serta ROC-AUC untuk mengevaluasi kemampuan model dalam membedakan antara kelas positif dan negatif. Di samping itu, waktu pelatihan juga dicatat untuk menilai efisiensi masing-masing algoritma.

## 2.6. Analisis dan Interpretasi Hasil

Hasil evaluasi dari setiap model dianalisis untuk menentukan algoritma yang memberikan kinerja terbaik dalam hal akurasi, efisiensi waktu pelatihan, dan kemampuan generalisasi. Perbandingan antara *Adaptive Boosting*, *Gradient Boosting*, dan XGBoost dilakukan berdasarkan metrik evaluasi dan waktu pelatihan. Hasil analisis dilakukan melalui diskusi mendalam untuk menjelaskan keunggulan dan kelemahan masing-masing algoritma, serta bagaimana algoritma tersebut menangani tantangan dataset yang kecil.

## 3. HASIL DAN PEMBAHASAN

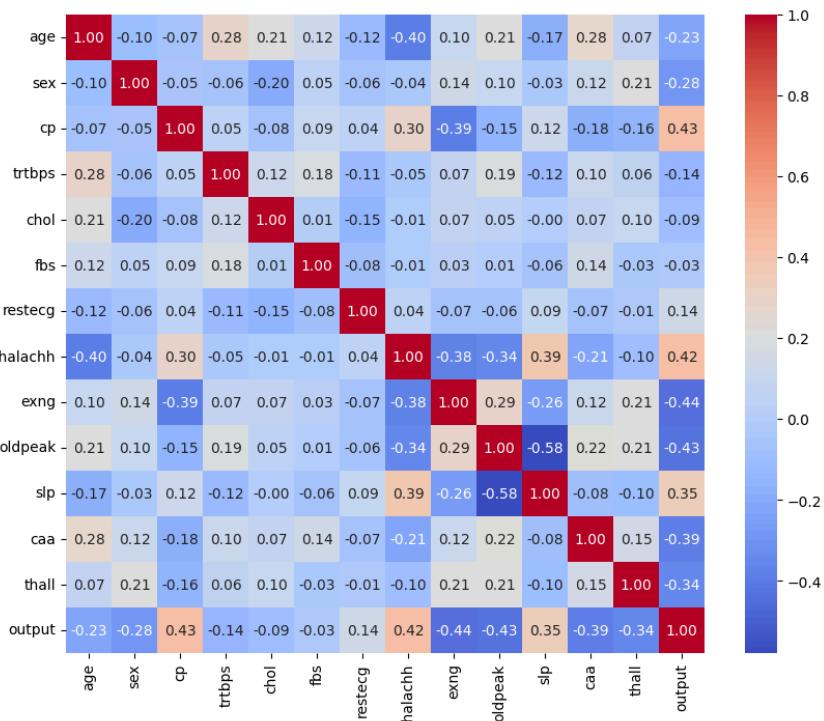
Untuk menganalisis prediksi penyakit jantung melalui pengembangan model *Adaptive Boosting* (AdaBoost), *Gradient Boosting*, dan *Extreme Gradient Boosting* (XGBoost), langkah pertama adalah menyiapkan dataset yang akan digunakan untuk pelatihan dan pengujian. Dalam penelitian ini, dataset yang digunakan adalah *Heart Disease Dataset* yang diambil dari *platform Kaggle* [15]. Dataset ini berisi data klinis pasien yang dirancang untuk memprediksi kemungkinan terjadinya penyakit jantung berdasarkan sejumlah fitur yang relevan. Dataset tersebut terdiri dari 303 entri dengan 14 kolom yang mewakili berbagai fitur. Setelah dataset diperoleh, langkah berikutnya adalah melakukan eksplorasi data untuk memahami secara mendalam karakteristik yang dimilikinya. Tahapan ini mencakup analisis distribusi data pada setiap fitur menggunakan visualisasi seperti distribusi fitur numerik, korelasi antar fitur, serta hubungan fitur dengan target. Langkah eksplorasi data dimulai dengan visualisasi distribusi fitur numerik yang terdapat dalam dataset. Visualisasi ini memiliki peran penting dalam mengidentifikasi karakteristik distribusi data numerik yang digunakan serta pola yang mungkin muncul. Distribusi variabel numerik pada dataset divisualisasikan menggunakan *boxplot*, seperti yang ditampilkan pada Gambar 2.



Gambar 2. Boxplot Fitur Numerik Pada Dataset

Visualisasi *boxplot* pada Gambar 2 menunjukkan distribusi data dari fitur numerik dalam dataset, seperti *age*, *trtbps*, *chol*, *thalachh*, dan *oldpeak*. *Boxplot* memberikan gambaran persebaran data, median, rentang interkuartil, dan mendeteksi *outlier*, yaitu data yang berada di luar nilai wajar. Dari visualisasi ini, terlihat beberapa fitur seperti *chol* (kolesterol) dan *trtbps* (tekanan darah istirahat) memiliki *outlier* signifikan yang mungkin relevan secara klinis untuk analisis lebih lanjut, sementara fitur seperti *age* memiliki distribusi yang wajar tanpa *outlier*.

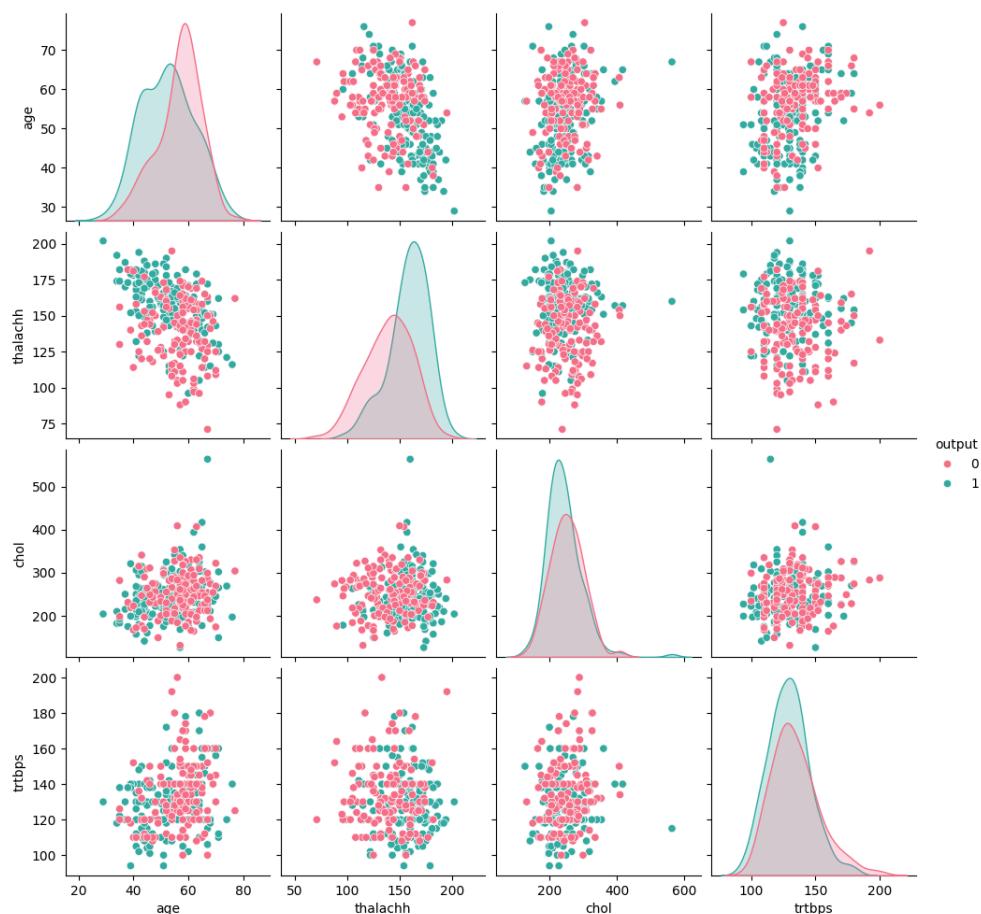
Visualisasi data selanjutnya adalah korelasi antar fitur, yang bertujuan untuk memahami hubungan linier antara setiap variabel dalam dataset. Informasi ini penting untuk diketahui agar dapat mengidentifikasi fitur-fitur yang memiliki hubungan signifikan dengan target, mengurangi risiko penggunaan fitur yang redundan, dan menentukan variabel mana yang paling relevan untuk analisis lebih lanjut. Visualisasi korelasi antar fitur melalui *heatmap* ditunjukkan pada Gambar 3.



Gambar 3. Heatmap Korelasi Antar Fitur

Visualisasi pada Gambar 3 adalah heatmap korelasi antar fitur yang menunjukkan hubungan linier antara variabel-variabel dalam dataset. Nilai korelasi berkisar dari -1 hingga 1, dengan warna mendekati merah menunjukkan korelasi positif yang kuat, sedangkan warna biru menunjukkan korelasi negatif. Misalnya, fitur *cp* (tipe nyeri dada) dan *output* memiliki korelasi positif (0.43), menunjukkan hubungan signifikan antara jenis nyeri dada dengan adanya penyakit jantung. Sebaliknya, *exng* (angina akibat olahraga) memiliki korelasi negatif (-0.44) terhadap *output*, menunjukkan bahwa angina yang diinduksi olahraga lebih sering terjadi pada pasien tanpa penyakit jantung.

Selanjutnya adalah visualisasi hubungan antar fitur dengan target, yang berfungsi untuk memahami bagaimana setiap fitur berinteraksi dengan variabel target. Visualisasi ini berguna untuk mengidentifikasi pola-pola spesifik yang membedakan kategori target, seperti pasien dengan atau tanpa penyakit jantung. Dengan mempelajari hubungan ini, dapat ditentukan fitur mana yang paling relevan dan berkontribusi signifikan terhadap prediksi. Visualisasi hubungan antar fitur dalam dataset di tampilkan pada Gambar 4.



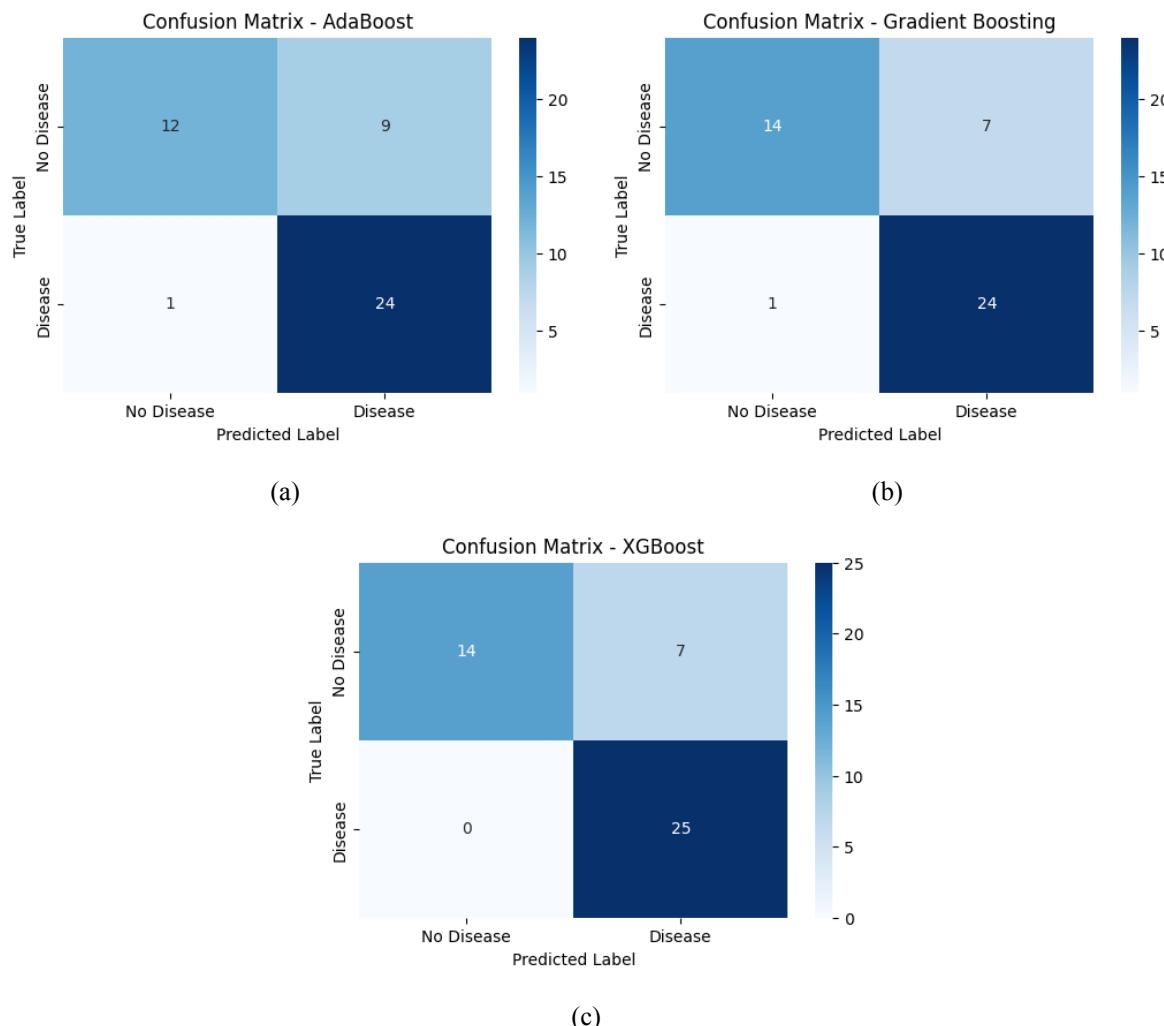
Gambar 4. Hubungan Antar Fitur Berdasarkan Target

Visualisasi pada Gambar 4 adalah *pair plot* yang menunjukkan hubungan antar fitur numerik dalam dataset, seperti *age*, *thalachh*, *chol*, dan *trtbps*, dengan memisahkan data berdasarkan kategori output (0 = tidak ada penyakit dan 1 = ada penyakit). Pair plot ini menggabungkan grafik distribusi (*diagonal*) dan *scatter plot* (*off-diagonal*), sehingga membantu dalam mengidentifikasi pola hubungan antar fitur dan perbedaan distribusi berdasarkan target. Misalnya, terlihat bahwa pasien dengan penyakit jantung cenderung memiliki detak jantung maksimum (*thalachh*) yang lebih rendah dibandingkan pasien tanpa penyakit jantung. Visualisasi ini penting untuk memahami pola interaksi antar fitur, mendeteksi keterkaitan signifikan dengan target, dan menentukan relevansi fitur untuk model prediksi.

Setelah karakteristik dataset yang digunakan dianalisis, langkah berikutnya adalah melakukan pra-pemrosesan untuk memastikan data siap digunakan oleh model. Berdasarkan hasil eksplorasi data, ditemukan adanya nilai yang hilang atau tidak lengkap, sehingga dilakukan imputasi dengan menggunakan nilai

median untuk menjaga distribusi data. Untuk menangani *outlier*, metode *Interquartile Range* (IQR) diterapkan guna menghapus data ekstrem yang berpotensi memengaruhi kinerja model. Selanjutnya, fitur-fitur dinormalisasi menggunakan *StandardScaler* agar skala data seragam (rata-rata 0 dan standar deviasi 1), sehingga algoritma dapat bekerja lebih efisien selama proses pelatihan. Setelah fitur dan target dipisahkan, dataset dibagi menjadi data latih (80%) dan data uji (20%). Rasio ini memberikan keseimbangan yang baik antara memastikan model dilatih dengan data yang cukup untuk generalisasi yang baik, serta memungkinkan evaluasi model yang objektif menggunakan data yang belum pernah dilihat sebelumnya.

Proses membangun model prediksi penyakit jantung dilakukan dengan tiga algoritma boosting, yaitu *Adaptive Boosting* (AdaBoost), *Gradient Boosting*, dan *Extreme Gradient Boosting* (XGBoost). Untuk AdaBoost, model dilatih menggunakan 100 *estimators* (*weak learners*), dengan parameter *default* untuk menjaga efisiensi. Pada *Gradient Boosting*, digunakan 100 *estimators*, *learning rate* 0.1, dan kedalaman maksimum pohon sebesar 3 untuk mengoptimalkan akurasi sambil mencegah *overfitting*. Sedangkan XGBoost dilatih dengan parameter serupa, namun dengan tambahan optimasi seperti *eval\_metric='logloss'* untuk menghitung fungsi *loss* selama pelatihan. Setiap model dilatih pada data training yang telah dinormalisasi, kemudian hasilnya diuji pada data *testing*. Evaluasi pertama yaitu melalui *confusion matrix*, di mana matriks ini digunakan untuk memvisualisasikan distribusi prediksi model dengan membandingkan jumlah prediksi yang benar dan salah untuk masing-masing kelas. Hasil *confusion matrix* dari tiga model yaitu AdaBoost, *Gradient Boosting*, dan XGBoost divisualisasikan pada Gambar 5.



Gambar 5. (a) *Confusion Matrix* Model AdaBoost; (b) *Confusion Matrix* Model Gradient Boosting; *Confusion Matrix* Model XGBoost

Hasil *confusion matrix* dari ketiga model menunjukkan distribusi prediksi terhadap label sebenarnya. Pada model AdaBoost (Gambar 5.a), terdapat 12 prediksi benar untuk kelas "No Disease" dan 24 prediksi benar untuk kelas "Disease," dengan 9 kesalahan klasifikasi untuk "No Disease" yang diprediksi sebagai "Disease" dan hanya 1 kesalahan untuk arah sebaliknya. Pada model Gradient Boosting (Gambar 5.b), performa sedikit lebih baik, dengan 14 prediksi benar untuk "No Disease" dan 24 untuk "Disease," serta kesalahan klasifikasi yang lebih sedikit pada kelas "No Disease" (7 kesalahan). Model XGBoost (Gambar 5.c) menunjukkan kinerja terbaik, dengan 14 prediksi benar untuk "No Disease" dan 25 untuk "Disease," serta tanpa kesalahan klasifikasi pada kelas "Disease," meskipun masih terdapat 7 kesalahan pada "No Disease." Secara keseluruhan, XGBoost menghasilkan prediksi paling stabil dengan kesalahan paling sedikit dibandingkan kedua model lainnya.

Selanjutnya, ketiga model dievaluasi menggunakan metrik seperti *precision*, *recall*, *F1-score*, *accuracy*, ROC-AUC, dan waktu pelatihan. Hasil *precision*, *recall*, dan *F1-score* untuk masing-masing model yang dikembangkan disajikan pada Tabel 1.

Tabel 1. *Precision*, *Recall*, dan *F1-Score* untuk Setiap Kelas

Model	<i>Precision</i>		<i>Recall</i>		<i>F1-Score</i>	
	No Disease	Disease	No Disease	Disease	No Disease	Disease
AdaBoost	0.9231	0.5714	0.7059	0.7273	0.9600	0.8286
Gradient Boosting	0.9333	0.6667	0.7778	0.7742	0.9600	0.8571
XGBoost	1.0000	0.6667	0.8000	0.7812	1.0000	0.8772

Berdasarkan Tabel 1, hasil *precision*, *recall*, dan *F1-score* menunjukkan variasi performa masing-masing model dalam memprediksi kelas "No Disease" dan "Disease." Model AdaBoost memiliki *precision* tinggi untuk kelas "No Disease" (0.9231) tetapi menunjukkan kelemahan pada *precision* kelas "Disease" (0.5714). Model Gradient Boosting lebih seimbang, dengan *precision* kelas "No Disease" sebesar 0.9333 dan *precision* kelas "Disease" meningkat menjadi 0.6667. Model XGBoost menampilkan *precision* tertinggi untuk kelas "No Disease" (1.0000), namun *precision* kelas "Disease" serupa dengan Gradient Boosting pada 0.6667. Dari sisi *recall*, XGBoost unggul untuk kelas "Disease" (1.0000), sementara *recall* untuk kelas "No Disease" sedikit lebih rendah dibanding Gradient Boosting. *F1-score*, yang menggabungkan *precision* dan *recall*, menunjukkan bahwa XGBoost memiliki performa keseluruhan terbaik pada kedua kelas.

Hasil evaluasi lainnya, yaitu *accuracy*, ROC-AUC, dan waktu pelatihan, disajikan dalam Tabel 2 untuk mempermudah analisis dan perbandingan kinerja model.

Tabel 2. *Accuracy*, ROC-AUC, dan Waktu Pelatihan

Model	<i>Accuracy</i>	ROC-AUC	<i>Training Time (seconds)</i>
AdaBoost	0.7826	0.9314	0.7742
Gradient Boosting	0.8261	0.9238	0.3655
XGBoost	0.8478	0.9410	0.6799

Pada Tabel 2, perbandingan akurasi, ROC-AUC, dan waktu pelatihan menunjukkan keunggulan XGBoost dalam hal akurasi (0.8478) dan ROC-AUC (0.9410). Namun, Gradient Boosting adalah model dengan waktu pelatihan tercepat (0.3655 detik) dibandingkan dengan AdaBoost (0.7742 detik) dan XGBoost (0.6799 detik). Meskipun AdaBoost memiliki ROC-AUC yang cukup tinggi (0.9314), akurasinya (0.7826) lebih rendah dibandingkan dua model lainnya. Hal ini menunjukkan bahwa Gradient Boosting dan XGBoost lebih efektif dalam menangkap pola data, meskipun membutuhkan waktu pelatihan yang lebih lama.

Dari sisi kelebihan, AdaBoost unggul dalam menangkap hubungan sederhana pada kelas mayoritas (*No Disease*), namun *precision* dan *recall* yang rendah untuk kelas minoritas (*Disease*) menjadi kekurangannya. Perbaikan dapat dilakukan dengan menyesuaikan bobot data untuk menangani ketidakseimbangan kelas. Gradient Boosting memiliki keseimbangan performa yang baik antara *precision*, *recall*, dan waktu pelatihan, menjadikannya pilihan efisien untuk dataset kecil. Perbaikan dapat dilakukan dengan menyesuaikan *hyperparameter* seperti kedalaman pohon dan *learning rate*. XGBoost adalah model dengan akurasi dan ROC-AUC terbaik, menunjukkan kemampuannya dalam memprediksi data yang kompleks. Namun, waktu pelatihan yang lebih lama dibanding Gradient Boosting dapat diperbaiki dengan *tuning hyperparameter*, seperti mengurangi jumlah *estimators* atau menyesuaikan *early stopping*.

Secara keseluruhan, XGBoost adalah pilihan terbaik untuk performa prediksi, namun Gradient Boosting lebih cocok jika efisiensi waktu menjadi prioritas. AdaBoost dapat digunakan pada dataset dengan distribusi kelas yang lebih seimbang untuk memaksimalkan *precision* dan *recall*. XGBoost cenderung memiliki performa

yang lebih baik karena menggunakan optimasi tambahan seperti regularisasi dan teknik *pruning* yang lebih canggih, yang membantu model menghindari *overfitting* dan menghasilkan prediksi yang lebih akurat. Namun, optimasi ini memerlukan waktu pelatihan yang lebih lama. Di sisi lain, *Gradient Boosting* lebih efisien dalam hal waktu karena tidak melibatkan proses kompleks seperti XGBoost, namun tetap memberikan hasil yang cukup baik dalam hal akurasi dan ROC-AUC.

#### 4. KESIMPULAN

Penelitian ini telah menganalisis kinerja tiga algoritma *boosting*, yaitu *Adaptive Boosting* (AdaBoost), *Gradient Boosting*, dan *Extreme Gradient Boosting* (XGBoost), dalam memprediksi penyakit jantung menggunakan dataset berukuran kecil. Berdasarkan hasil evaluasi, XGBoost menunjukkan performa terbaik dengan akurasi tertinggi sebesar 84.78% dan nilai ROC-AUC sebesar 0.9410, menjadikannya algoritma yang paling efektif dalam menangani pola data yang kompleks. Namun, *Gradient Boosting* menonjol dengan waktu pelatihan tercepat, yaitu 0.3655 detik, sambil tetap memberikan akurasi dan ROC-AUC yang kompetitif, sehingga menjadi pilihan efisien untuk kebutuhan *real-time*. Sementara itu, AdaBoost memberikan hasil yang baik pada kelas mayoritas, tetapi menunjukkan kelemahan dalam menangani ketidakseimbangan kelas pada dataset. Melalui analisis *precision*, *recall*, dan *F1-score*, ditemukan bahwa XGBoost memberikan prediksi yang lebih konsisten untuk kedua kelas dibandingkan dengan algoritma lainnya. Meskipun *Gradient Boosting* sedikit kalah dalam akurasi, algoritma ini menawarkan keseimbangan performa dan efisiensi waktu yang lebih baik. Sebaliknya, AdaBoost lebih sesuai digunakan untuk dataset dengan distribusi kelas yang lebih seimbang. Untuk penelitian selanjutnya, disarankan untuk mengeksplorasi *tuning hyperparameter* secara mendalam, seperti penyesuaian jumlah *estimators*, *learning rate*, dan kedalaman pohon, untuk lebih mengoptimalkan kinerja model. Selain itu, pengujian model pada dataset yang lebih besar atau dengan tingkat ketidakseimbangan kelas yang lebih signifikan dapat memberikan wawasan tambahan mengenai kemampuan generalisasi algoritma. Penelitian juga dapat menggabungkan metode *boosting* dengan pendekatan *ensemble* lainnya, seperti *bagging* atau *stacking*, guna meningkatkan kinerja prediksi lebih lanjut.

#### DAFTAR PUSTAKA

- [1] M. Marniati, S. Notoatmodjo, S. Kasiman, and R. K. Rochadi, *Lifestyle of Determinant: Penderita Penyakit Jantung Koroner*. Depok: PT. RajaGrafindo Persada - Rajawali Pers, 2022.
- [2] Solehudin Solehudin and Saiful Gunardi, "Deteksi Dini Serangan Jantung Saat Aktivitas Olahraga," *Faedah J. Has. Kegiat. Pengabdi. Masy. Indones.*, vol. 1, no. 3, pp. 257–265, 2023.
- [3] R. Hidayat, Y. S. Sy, T. Sujana, M. Husnrah, H. T. Saputra, and F. Okmayura, "Implementasi Machine Learning Untuk Prediksi Penyakit Jantung Menggunakan Algoritma Support Vector Machine," *BIOS J. Teknol. Inf. dan Rekayasa Komput.*, vol. 5, no. 2, pp. 161–168, 2024.
- [4] F. Handayani, "Komparasi Support Vector Machine, Logistic Regression Dan Artificial Neural Network Dalam Prediksi Penyakit Jantung," *J. Edukasi dan Penelit. Inform.*, vol. 7, no. 3, p. 329, 2021, doi: 10.26418/jp.v7i3.48053.
- [5] S. A. Putri, N. Selayanti, M. Kristanaya, M. P. Azzahra, M. G. Navsih, and K. M. Hindrayani, "Penerapan Machine Learning Algoritma Random Forest Untuk Prediksi Penyakit Jantung," in *Seminar Nasional Sains Data (SENADA)*, 2024, pp. 895–906.
- [6] D. F. Wicaksono, R. S. Basuki, and D. Setiawan, "Peningkatan Performa Model Machine Learning XGBoost Classifier melalui Teknik Oversampling dalam Prediksi Penyakit AIDS," *J. Media Inform. Budidarma*, vol. 8, no. 2, p. 736, 2024, doi: 10.30865/mib.v8i2.7501.
- [7] G. Airlangga, "Enhancing Medical Diagnostics with Ensemble Machine Learning: A Comparative Study of Gradient Boosting, XGBoost, LightGBM, and Blended Models," *J. Ris. Sist. Inf. Dan Tek. Inform.*, vol. 9, no. 2, pp. 1118–1132, 2024.
- [8] I. M. Latief, A. Subekti, and W. Gata, "Prediksi Tingkat Pelanggan Churn Pada Perusahaan Telekomunikasi Dengan Algoritma Adaboost," *J. Inform.*, vol. 21, no. 1, pp. 34–43, 2021.
- [9] D. T. Murdiansyah, "Prediksi Stroke Menggunakan Extreme Gradient Boosting," *JIKO (Jurnal Inform. dan Komputer)*, vol. 8, no. 2, p. 419, 2024, doi: 10.26798/jiko.v8i2.1295.
- [10] M. Alkaff, A. Baskara, and A. Ainiyyah, "Penerapan Metode XGBoost Untuk Memprediksi Jumlah Kejadian Kecelakaan Lalu Lintas di Kota Banjarmasin," *Gener. J.*, vol. 7, no. 1, pp. 61–69, 2023, doi: 10.29407/gj.v7i1.19807.
- [11] F. Y. Pamuji, A. R. Muslikh, R. M. Arief, and D. Muti, "Komparasi Metode Mean dan KNN Imputation dalam Mengatasi Missing Value pada Dataset Kecil," *J. Inform. Polinema*, vol. 10, no. 2, pp. 257–264, 2024, doi: 10.33795/jip.v10i2.5031.
- [12] Muhamad Fikri, "Klasifikasi Status Stunting Pada Anak Bawah Lima Tahun Menggunakan Extreme

- Gradient Boosting,” *Merkurius J. Ris. Sist. Inf. dan Tek. Inform.*, vol. 2, no. 4, pp. 173–184, 2024, doi: 10.61132/merkurius.v2i4.159.
- [13] A. Sah, S. Suhardi, and S. Nurhayati, “Geographic Information System of Patient Development in Jayapura Hospital During Pandemic,” *J. Teknol. Dan Open Source*, vol. 4, no. 2, pp. 149–154, 2021, doi: 10.36378/jtos.v4i2.1412.
- [14] A. Sah, J. Jusmawati, S. Nurhayati, M. Tonggiroh, and S. Bonay, “Sistem Informasi Manajemen Pada Puskesmas Kota Jayapura Berbasis Web,” *JTIM J. Teknol. Inf. dan Multimed.*, vol. 4, no. 3, pp. 212–220, 2022.
- [15] M. Yasser, “Heart Disease Dataset,” Kaggle. [Online]. Available: <https://www.kaggle.com/datasets/yasserh/heart-disease-dataset>
- [16] S. Ida Gultom, “Data Mining Implementation Determines a Healthy Lifestyle for KB Users Using the Adaboost Algorithm (Case Study :Dinas Serdang Bedagai),” *J. Inf. dan Teknol. Ilm.*, vol. 7, no. 3, pp. 298–304, 2020.
- [17] 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.
- [18] A. T. Pratiwi, A. Barizi, M. I. Maulana, and P. Rosyani, “Systematic Literature Review Penerapan Gradient Boosting Untuk Klasifikasi Penyakit Diabetes Tipe 2,” *BIIKMA Bul. Ilm. Ilmu Komput. dan Multimed.*, vol. 2, no. 3, pp. 454–458, 2024.
- [19] N. C. Sari and T. Linda Larasati, “Komparasi Algoritma Naïve Bayes dan Gradient Boosting untuk Prediksi Pasien Diabetes,” *J. Nas. Teknol. dan Sist. Inf.*, vol. 10, no. 2, pp. 118–125, 2024, doi: 10.25077/TEKNOSI.v10i2.2024.118-125.
- [20] A. Wibowo, “Prediksi Kekuatan Gempa Menggunakan Machine Learning Dengan Model XGBoost Sebagai Langkah Strategis Dalam Perencanaan Struktur Bangunan Tahan Gempa Di Indonesia,” *MESA (Teknik Mesin, Tek. Elektro, Tek. Sipil, Tek. Arsitektur)*, vol. 6, no. 1, pp. 18–29, 2023.
- [21] A. Candra, M. Erkamim, M. Muhamrom, and E. Prayitno, “Klasifikasi Stunting Pada Balita Berdasarkan Status Gizi Menggunakan Pendekatan Support Vector Machine (SVM),” *J. Ilm. FIFO*, vol. 16, no. 2, pp. 171–181, 2024.
- [22] 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.
- [23] R. I. Borman, R. Napianto, N. Nugroho, D. Pasha, Y. Rahmanto, and Y. E. P. Yudoutomo, “Implementation of PCA and KNN Algorithms in the Classification of Indonesian Medicinal Plants,” in *International Conference on Computer Science, Information Technology and Electrical Engineering (ICOMITEE)*, IEEE, 2021, pp. 46–50.
- [24] 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)*, IEEE, 2022.