

Comparison of Adaptive Boosting and Categorical Boosting in Heart Attack Diagnosis

Ali Amran^{1*}, Suryani², Nadiva Azro Fathinah³, Anita Desiani⁴, Indri Ramayanti⁵

^{1,2,3,4}Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Sriwijaya, Indralaya, 30662, Indonesia

⁵Fakultas Kedokteran, Universitas Muhammadiyah Palembang, 30263, Indonesia

Informasi Artikel

Diterima : 3 Januari 2026
Revisi : 23 Februari 2026
Publikasi : 31 Maret 2026

Kata Kunci:

AdaBoost
CatBoost
Klasifikasi
Serangan Jantung
Machine Learning

ABSTRAK

Serangan jantung merupakan salah satu penyebab utama kematian di seluruh dunia sehingga diperlukan metode deteksi dini yang akurat untuk membantu menurunkan angka mortalitas. Salah satu pendekatan yang dapat digunakan adalah machine learning dengan teknik klasifikasi berbasis algoritma ensemble boosting. Penelitian ini bertujuan untuk membandingkan kinerja dua algoritma ensemble, yaitu Adaptive Boosting (AdaBoost) dan Categorical Boosting (CatBoost), dalam mengklasifikasikan penyakit serangan jantung. Label yang digunakan pada penelitian ini adalah label positif dan negatif. Proses pengujian dilakukan menggunakan dua teknik evaluasi, yaitu percentage split dengan rasio 80% *training* dan 20% *testing*, serta 10-fold cross-validation. Kinerja model dievaluasi berdasarkan metrik akurasi, presisi, dan recall untuk mengukur kemampuan klasifikasi secara menyeluruh. Hasil penelitian menunjukkan bahwa pada metode percentage split, CatBoost memperoleh akurasi tertinggi sebesar 98,88%, sedangkan pada k-fold cross-validation mencapai 98,43%. Meskipun demikian, AdaBoost juga menunjukkan kinerja yang baik dengan seluruh metrik di atas 90%. Dengan demikian, model terbaik dalam penelitian ini adalah CatBoost dengan teknik k-fold cross-validation pada dataset serangan jantung.

ABSTRACT

Heart disease is one of the leading causes of death worldwide, and therefore, accurate early detection methods are needed to help reduce mortality rates. One approach that can be applied is machine learning using classification techniques based on ensemble boosting algorithms. This study aims to compare the performance of two ensemble algorithms, namely Adaptive Boosting (AdaBoost) and Categorical Boosting (CatBoost), in classifying heart attack disease. The labels used in this study are positive and negative. The evaluation process was conducted using two testing techniques: percentage split with a ratio of 80% training data and 20% testing data, and 10-fold cross-validation. Model performance was evaluated based on accuracy, precision, and recall to comprehensively measure classification capability. The results show that in the percentage split method, CatBoost achieved the highest accuracy of 98.88%, while in k-fold cross-validation it reached 98.43%. Nevertheless, AdaBoost also demonstrated good performance, with all evaluation metrics exceeding 90%. Therefore, the best-performing model in this study is CatBoost with the k-fold cross-validation technique on the heart attack dataset.

This is an open-access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license



*Penulis Koresponden

Email: ali_amran@mipa.unsri.ac.id

Cara sitasi IEEE::

- [1] A. Amran, S. Suryani, N. A. Fathinah, A. Desiani, dan I. Ramayanti, "Comparison of Adaptive Boosting and Categorical Boosting in Heart Attack Diagnosis," *Journal of Artificial Intelligence and Software Engineering (J-AISE)*, vol. 6, no. 1, p. 44-54, Maret 2026. doi: 10.30811/jaise.v6i1.9051

1. PENDAHULUAN

Serangan jantung adalah salah satu penyebab utama kematian di dunia. Menurut World Health Organization (WHO), lebih dari 17,9 juta orang meninggal akibat penyakit kardiovaskular setiap tahunnya, termasuk serangan jantung, penyakit jantung koroner, dan stroke [1]. Gejala serangan jantung sering kali samar dan tidak spesifik, sehingga banyak orang yang tidak menyadari kondisinya. Kurangnya kesadaran ini meningkatkan risiko keterlambatan pengobatan, yang dapat menyebabkan komplikasi serius atau bahkan kematian [2]. Deteksi dini serangan jantung diperlukan untuk mendeteksi dan mencegah potensi kasus kematian. Diagnosis dapat dilakukan dengan mengenali pola melalui analisis kumpulan data yang relevan. Deteksi awal risiko serangan jantung dapat ditingkatkan dengan memanfaatkan pendekatan matematis dalam data mining [3]. Salah satu pendekatan yang efektif untuk deteksi dini adalah dengan klasifikasi dengan algoritma dalam *machine learning* [4], [5]. Klasifikasi adalah teknik yang melibatkan penemuan pola atau fungsi untuk membedakan kelas data, dan dapat digunakan untuk memprediksi kasus yang belum teridentifikasi [6]. Algoritma yang dapat diterapkan dalam klasifikasi adalah Adaptive Boosting (AdaBoost) dan Categorical Boosting (CatBoost).

Adaptive Boosting adalah algoritma *machine learning* yang dikategorikan sebagai algoritma ensemble. Algoritma ensemble adalah algoritma yang menggabungkan beberapa *single classifier* dengan tujuan meningkatkan performa dengan mengambil keunggulan dari masing-masing *classifier* [7]. Algoritma AdaBoost dirancang untuk meningkatkan akurasi model klasifikasi karena bekerja dengan cara menggabungkan beberapa model lemah (*weak learners*) menjadi satu model kuat (*strong learner*) dengan menerapkan bobot [8]. Pada klasifikasi yang berbasis dua label, *weak learner* umumnya berupa *decision stump* [9]. *Decision stump* adalah sebuah pohon keputusan yang sangat sederhana dengan hanya satu aturan (split) berdasarkan satu fitur dan satu nilai ambang (threshold). Nilai ambang dipilih dengan mencoba berbagai nilai ambang pada setiap fitur dan memilih yang menghasilkan kesalahan (error) terkecil berdasarkan bobot data saat itu. Keunggulan utama AdaBoost adalah kemampuannya untuk meningkatkan akurasi prediksi dengan memberikan bobot yang lebih besar pada contoh data yang sulit diklasifikasikan [10]. El Hamdaoui et al. [11] menerapkan algoritma AdaBoost untuk mengklasifikasikan penyakit jantung dan mendapatkan akurasi sebesar 96,16%. Namun, penelitian tersebut memiliki keterbatasan dengan tidak menyertakan nilai sensitivitas model. Anwar et al. [12] mengimplementasikan algoritma AdaBoost untuk klasifikasi penyakit hati dan menghasilkan akurasi 85,04%. Penelitian ini dilakukan hanya dengan menggunakan satu metode pengujian yaitu k-fold cross validation. Meskipun mencapai akurasi tinggi, penelitian ini terbatas pada pengukuran akurasi tanpa mengukur jenis pengukuran kinerja lain. Dharmawan dkk. [13] juga menggunakan algoritma AdaBoost untuk mengklasifikasikan serangan jantung, mencapai akurasi 90,79%. Namun, evaluasi penelitian ini hanya dilakukan dengan akurasi tanpa membandingkan dengan algoritma lain ataupun metrik tambahan. Meskipun mencapai akurasi yang sangat baik, bahkan pada dataset yang tidak seimbang, algoritma AdaBoost memiliki kelemahan karena sensitif terhadap noise dan outlier dalam data. Hal ini dikarenakan bobot pada data yang sulit diklasifikasi dapat membuat model terlalu fokus pada data yang sebenarnya merupakan kesalahan, sehingga berpotensi menurunkan akurasi pada data yang lebih bersih [14].

Algoritma lain yang dapat digunakan untuk mengklasifikasikan dataset penyakit serangan jantung adalah Categorical Boosting (CatBoost). CatBoost dirancang untuk menangani data kategorik secara efisien dengan prapemrosesan minimal, CatBoost memiliki keunggulan yang berbeda dalam kasus yang melibatkan data terstruktur karena secara native mengelola fitur kategorikal tanpa memerlukan encoding manual seperti one-hot atau label encoding. [15]. Salah satu keunggulan utamanya adalah kemampuannya untuk memproses variabel kategorikal namun tetap tahan terhadap overfitting dan noise karena menggunakan pendekatan ordered boosting yang dapat menghindari kebocoran informasi saat pelatihan [16]. Samosir P. dan Salamah [15] menerapkan algoritma Catboost untuk klasifikasi penyakit kardiovaskular dan mendapatkan akurasi yang mencapai 75%. Penelitian ini terbatas hanya menggunakan satu metode pengujian dan akurasi yang masih dibawah 90%. Bhaskar dkk. [17] menerapkan CatBoost untuk mendeteksi penyakit ginjal kronis, dan menghasilkan akurasi sebesar 98,37%. Penelitian ini memiliki keterbatasan pada pengujian kinerjanya yang hanya mengukur akurasi. Haque dkk. [18] menggunakan CatBoost untuk mengklasifikasikan penyakit ginjal kronis (CKD), dengan akurasi sebesar 98,75%. Meskipun memiliki akurasi yang tinggi, penelitian ini hanya menggunakan satu metode klasifikasi yaitu metode percentage split. Selain itu, CatBoost memiliki keterbatasan dalam efisiensi waktu pelatihan pada dataset besar dan sensitif terhadap pemilihan parameter tertentu.

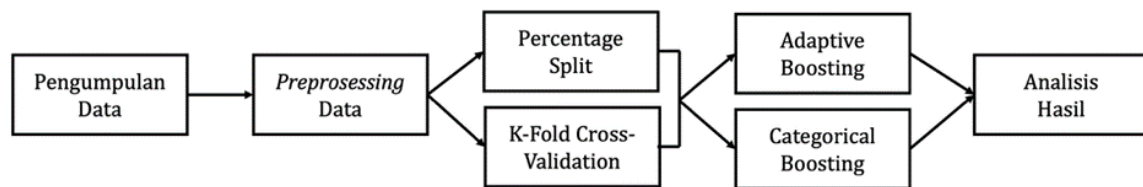
Dalam melakukan klasifikasi, terdapat dua teknik evaluasi model yang dapat digunakan, yaitu percentage split dan k-fold cross-validation. Metode percentage split memiliki kelebihan dalam hal kemudahan implementasi dan waktu pelatihan yang relatif singkat [19]. Metode ini juga memiliki kekurangan, terutama ketika data yang digunakan tidak seimbang [20]. Pembagian data yang tidak merata dapat mengakibatkan beberapa informasi penting tidak diikutsertakan dalam proses pelatihan. Untuk mengatasi kekurangan dari metode percentage split, teknik k-fold cross-validation dapat digunakan [21].

Teknik k-fold cross-validation adalah metode validasi model dalam machine learning yang membagi dataset menjadi k bagian [22]. Data dilatih sebanyak k kali, setiap kali menggunakan k - 1 bagian sebagai data latih dan 1 bagian sebagai data uji. Keuntungan utama dari teknik ini adalah memberikan hasil evaluasi yang lebih komprehensif dan mengurangi bias, terutama pada data yang tidak seimbang.

Penelitian ini membandingkan algoritma CatBoost dan AdaBoost dalam mengklasifikasikan penyakit serangan jantung. Adapun label yang digunakan ada dua, yaitu “positive” dan “negative”. Teknik pengujian digunakan adalah Percentage Split dengan rasion 80% training dan 20% testing, serta teknik K-Fold Cross-Validation dengan k = 10. Kinerja masing-masing algoritma dan teknik pengujian akan diukur menggunakan akurasi, presisi, dan recall agar dapat menentukan algoritma yang paling efektif untuk deteksi serangan jantung. Penelitian ini dapat dikembangkan dengan mengeksplorasi dataset yang lebih besar, mengklasifikasikan penyakit lainnya, atau perbandingan dengan algoritma klasifikasi tambahan.

2. METODE

Metode yang digunakan pada penelitian ini terdiri dari pengumpulan data, pre-processing data, data splitting, implementasi algoritma, dan analisis hasil. Diagram alir (flowchart) dari tahapan penelitian dapat dilihat pada gambar 1.



Gambar 1. Flowchart Tahapan Penelitian

2.1 Pengumpulan Data

Dataset yang digunakan dalam penelitian ini diperoleh dari platform Kaggle, yaitu Heart Attack Dataset yang dipublikasikan oleh Rashid dan Hassan [23]. Dataset tersebut terdiri dari 1319 data dengan 9 atribut. Dataset serangan jantung yang digunakan dalam penelitian ini memiliki ketidakseimbangan, dengan 61% data mewakili kasus dengan serangan jantung dan 39% tidak. Ketidakseimbangan ini dapat memengaruhi kinerja klasifikasi, karena model dapat menjadi bias terhadap kelas mayoritas [24]. Atribut yang ada pada dataset serangan jantung dapat dilihat pada tabel 1.

Tabel 1. Atribut, Jenis Data, dan Deskripsi yang Ada pada Dataset

Atribut	Tipe Data	Deskripsi
<i>Age</i>	Numerik	Umur Pasien
<i>Gender</i>	Nominal	Jenis kelamin pasien: 1 = Pria, 0 = Wanita
<i>Heart Rate</i>	Numerik	Detak jantung per menit (bpm)
<i>Systolic Blood Pressure</i>	Numerik	Tekanan pada arteri saat jantung berkontraksi (mmHg)
<i>Diastolic Blood Pressure</i>	Numerik	Tekanan pada arteri saat jantung beristirahat di antara denyut (mmHg)
<i>Blood Sugar</i>	Numerik	Kadar glukosa darah pasien
<i>CK-MB</i>	Numerik	Enzim jantung yang dilepaskan saat terjadi kerusakan otot jantung
<i>Troponin</i>	Numerik	Protein biomarker yang sangat spesifik untuk cedera otot jantung
<i>Result</i>	Kategorik	Label hasil yang menunjukkan apakah pasien mengalami serangan jantung (positif atau negatif)

Berdasarkan Tabel 1, dataset yang digunakan terdiri dari delapan atribut yang berhubungan dengan gejala dan satu atribut kelas. Atribut kelas mencakup dua kategori, yaitu positif dan negatif. Hasil positif menunjukkan pasien mengalami serangan jantung, dan hasil negatif menunjukkan bahwa pasien tidak mengalami serangan jantung.

2.2 Pre-Processing Data

Preprocessing dilakukan untuk memastikan bahwa data yang digunakan dalam pelatihan model bersih dan dapat diproses secara efektif. Data yang bersih berdampak pada akurasi hasil prediksi. Prapemrosesan data yang diterapkan dalam penelitian ini terdiri dari tiga tahap, yaitu sebagai berikut:

2.2.1 Menghapus Data Pencilan (Outlier)

Outlier adalah data yang menyimpang secara signifikan pada dataset dan dapat mengganggu proses training algoritma. Penghapusan outlier dilakukan untuk meningkatkan akurasi, mengurangi distorsi, dan mencegah overfitting, sehingga model dapat belajar dari data yang lebih representatif dan menghasilkan prediksi yang lebih baik. Pada dataset ini, outlier ditemukan pada kolom age, heart rate, systolic blood pressure, diastolic blood pressure, blood suhar, CK-MB, dan Troponin. Outlier yang dihapus ditentukan menggunakan metode IQR. Metode ini menggunakan kuartil pertama (Q1) dan kuartil ketiga (Q3), untuk menentukan IQR, yaitu adalah selisih dari Q3 dan Q1. Data yang berada di luar batas bawah ($Q1 - 1.5IQR$) dan batas atas ($Q3 + 1.5IQR$) dianggap outlier dan dapat dihapus

2.2.2 Feature Scaling

Feature scaling digunakan untuk menormalisasi atau mengubah rentang nilai pada kolom-kolom dalam dataset sehingga semua kolom berada dalam skala atau rentang yang sama. Hal ini dilakukan agar perbedaan skala antar kolom tidak dapat menyebabkan model menjadi bias terhadap kolom dengan nilai yang lebih besar dan hasil prediksi menjadi tidak optimal. Pada penelitian ini, feature scaling yang digunakan adalah min-max scaling. Min-Max Scaling mengubah skala data dengan cara mengubah nilai-nilai pada kolom agar berada dalam rentang tertentu, umumnya pada rentang 0 dan 1. Min-Max Scaling ini digunakan karena data tidak mengikuti distribusi normal untuk mempertahankan interpretasi asli dari nilai pada kolom. Transformasi pada setiap data untuk Min-Max scaling dapat direpresentasikan pada persamaan (1)

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (1)$$

di mana:

- X_{norm} = Letak data X pada min-max scaler
- X = Nilai asli data yang akan diubah skalanya
- X_{min} = Nilai terkecil yang ada pada kolom
- X_{max} = Nilai terbesar yang ada pada kolom

2.2.3 Keseimbangan Data

Pada dataset ini terdapat keseimbangan data yang relatif cukup baik dengan jumlah sampel kelas negatif sebanyak 446 dan kelas positif sebanyak 342. Meskipun perbedaan jumlah tidak terlalu besar, ketidakseimbangan tetap dapat memengaruhi performa model machine learning karena model cenderung lebih fokus pada kelas negatif. Untuk mengatasi masalah ini, dapat digunakan metode SMOTE (Synthetic Minority Over-sampling Technique). SMOTE bekerja dengan membuat sampel sintetis baru pada kelas minoritas positif melalui interpolasi antara sampel minoritas yang ada sehingga jumlah data menjadi lebih seimbang tanpa hanya menyalin data lama. Rumus utama yang digunakan dalam SMOTE untuk menghasilkan sampel sintetis baru dapat direpresnetasikan oleh persamaan (2)

$$X_{new} = X_i + \delta(X_{nn} - X_i) \quad (2)$$

di mana:

- X_{new} = sampel sintesis baru
- X_i = data dari sampel minoritas yang dipilih secara acak
- X_{nn} = tetangga terdekat (nearest neighbor) dari sampel minoritas X_i yang dipilih secara acak
- δ = bilangan acak antara 0 dan 1 yang menentukan posisi titik sintetis antara X_i dan X_{nn}

2.3 Data Split

Data yang digunakan dalam penelitian ini diolah dengan menggunakan dua metode pengujian, yaitu Percentage Split Method dan K-Fold Cross-Validation.

2.3.1 Percentage Split

Percentage Split adalah sebuah metode pengujian yang membagi dataset original menjadi beberapa bagian dengan rasio tertentu. Umumnya, pada percentage split, dataset dibagi menjadi data training dan testing. Seperti namanya, training data adalah data yang digunakan untuk melatih mesin agar mampu mengeluarkan hasil klasifikasi yang sesuai. Testing data, di sisi lain, merupakan data digunakan untuk

menguji seberapa besar kemampuan mesin mengklasifikasikan sebuah dataset setelah mempelajari data training. Dalam penelitian ini, perbandingan antar data training dan testing adalah 80:20. Artinya, 80% data pada dataset akan digunakan untuk melatih mesin, dan 20% sisanya digunakan untuk menguji kemampuan mesin setelah mempelajari 80% dari dataset tersebut.

2.3.2 K-Fold Cross-Validation

K-Fold Cross-Validation adalah sebuah metode pengujian yang membagi sebuah dataset sebanyak k lipatan. Kemudian akan dilakukan perulangan sebanyak k kali. Di setiap perulangan, akan digunakan sebanyak k-1 bagian untuk dijadikan data training dan satu bagian dijadikan data validasi. Perlu diingat bahwa data yang divalidasi selalu berbeda. Misalkan, apabila pada perulangan pertama menggunakan bagian pertama untuk data validasi, maka pada perulangan selanjutnya tidak menggunakan data itu lagi untuk dijadikan data validasi. Apabila setiap bagian telah dijadikan data validasi, maka diambil rata-rata dari hasil semua perulangan. Pada penelitian ini, digunakan nilai k sebanyak 10. Artinya, akan terjadi sebanyak 10 perulangan, yang di mana pada setiap perulangan akan menggunakan 9 bagian dari dataset sebagai data training dan 1 data validasi.

2.4 Implementasi Algoritma

2.4.1 Adaptive Boosting

Adaptive Boosting (AdaBoost) adalah algoritma pembelajaran mesin berbasis ensemble yang menggabungkan beberapa *weak learner* menjadi *strong learner*. Proses ini dilakukan secara iteratif dengan menerapkan bobot yang lebih tinggi pada data yang sulit diklasifikasikan dengan benar. Tahap klasifikasi menggunakan algoritma AdaBoost dilakukan dengan langkah-langkah sebagai berikut [14]:

1.) Menginisialisasikan bobot awal sampel

Pada langkah pertama, setiap data diberikan bobot yang sama untuk memberi pengaruh yang setara pada semua sampel. Bobot awal setiap data dapat dicari menggunakan persamaan (3).

$$w_i^{(1)} = \frac{1}{n} \quad (3)$$

di mana:

n = banyak sampel/data

$w_i^{(1)}$ = bobot awal untuk sampel ke-i

2.) Boosting Iteration

a. Pelatihan *weak learner*

Pada setiap iterasi m, model weak learner dilatih menggunakan bobot $w_i^{(m)}$. Model weak learner ini menghasilkan prediksi $k_m(x_i)$ pada setiap sampel. Model ini bertujuan untuk meminimalkan kesalahan klasifikasi dalam distribusi bobot.

b. Menghitung nilai kesalahan bobot

Tingkat kesalahan model berguna mengukur seberapa buruk model saat mengklasifikasikan data dengan bobot $w_i^{(m)}$. Persamaan untuk tingkat kesalahan model dapat dilihat pada persamaan (4).

$$\varepsilon_m = \frac{\sum_{i=1}^n y_i \neq k_m(x_i) w_i^{(m)}}{\sum_{i=1}^n y_i w_i^{(m)}} \quad (4)$$

Di mana:

m = nomor iterasi sekarang

ε_m = tingkat kesalahan model pada iterasi m (*error rate*)

y_i = tingkat kesalahan berbobot pada iterasi m

$k_m(x_i)$ = prediksi dari *weak learner* untuk sampel x_i pada iterasi ke-m

$w_i^{(m)}$ = bobot pada sampel ke-i pada iterasi ke-m

Semakin kecil ε_m , semakin baik weak learner di iterasi tersebut.

c. Menghitung nilai alpha

Nilai alpha α_m nantinya digunakan untuk memberikan bobot pada weak learner di iterasi ke-t. Semakin kecil kesalahan (misalnya mendekati 0), semakin besar nilai α_m . Perhitungan nilai α_m dapat dijumpai pada persamaan (5)

$$\alpha_m = \frac{1}{2} \ln \left(\frac{1-\varepsilon_m}{\varepsilon_m} \right) \quad (5)$$

Di mana:

α_m = bobot yang diberikan kepada weak learner pada iterasi ke-t.

- d. Memperbarui bobot
Pembaruan bobot dilakukan agar sehingga data yang salah diklasifikasikan oleh model akan mendapatkan bobot lebih tinggi. Bobot diperbarui dengan persamaan (6).

$$w_i^{(m+1)} = w_i^{(m)} \cdot \exp(-\alpha_m y_i k_m(x_i)) \quad (6)$$

- e. Menormalisasikan bobot
Penormalan bobot dilakukan agar total bobot seluruh sampel tetap proporsional (memiliki total 1). Normalisasi bobot dapat dilakukan dengan persamaan (7).

$$w_i^{(m+1)} = \frac{w_i^{(m+1)}}{\sum_{m=1}^n w_i^{(m+1)}} \quad (7)$$

- 3.) Menghitung prediksi akhir

Setelah m iterasi, model akhir dihitung dengan menjumlahkan prediksi dari *weak learners* yang telah dilatih. Prediksi akhir dihitung menggunakan persamaan (8).

$$K(x) = \text{sign}(\sum_{m=1}^n \alpha_m K_m(x)) \quad (8)$$

di mana:

$K(x)$ = prediksi akhir untuk sampel x

2.4.2 Categorical Boosting

CatBoost (Categorical Boosting) adalah algoritma yang dirancang khusus untuk menangani data kategori (categorical data) secara otomatis dan efisien, tanpa perlu melakukan one-hot encoding atau label encoding manual. Tahap pengklasifikasian menggunakan algoritma CatBoost dilakukan beberapa langkah berikut [16]:

- 1.) Iterasi

CatBoost bekerja berdasarkan prinsip fungsi kerugian (loss function) yang diminimalkan menggunakan pendekatan gradient descent. Pada kasus klasifikasi biner digunakan persamaan (9).

$$L(y, \hat{y}) = -\frac{1}{n} \sum_{i=1}^n [y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)] \quad (9)$$

Dengan:

n = banyak sampel/data

y_i = label kelas actual untuk sampel ke-i, dengan $y_i \in \{0,1\}$

\hat{y}_i = probabilitas prediksi untuk sampel ke-i, dengan $\hat{y}_i \in [0,1]$

Untuk iterasi sebanyak n dilakukan beberapa langkah:

- a. Menghitung Gradient Loss

Gradient dari fungsi kerugian dihitung terhadap prediksi saat ini $\hat{y}_i^{(t-1)}$ yaitu prediksi dari iterasi sebelumnya seperti pada persamaan (10).

$$g_i^t = \frac{\partial L(y_i, \hat{y}_i^{(t-1)})}{\partial \hat{y}_i^{(t-1)}} \quad (10)$$

Dengan:

T = Nomor iterasi sekarang

g_i^t = Nilai gradient (arah turunan kerugian) untuk data ke-i pada iterasi ke-t.

$\hat{y}_i^{(t-1)}$ = Prediksi pada iterasi sebelumnya.

Gradient ini mempresentasikan seberapa besar dan kearah mana model perlu dikoreksi.

- b. Melatih *Weak Learner* dengan *Oblivious Decision Tree*

Dengan memanfaatkan nilai gradient, algoritma melatih pohon keputusan simetris (oblivioius tree) sebagai weak learner $h^t(x)$ untuk memetakan data input x_i ke nilai koreksi.

- c. Menghitung prediksi sementara

Predikasi model diperbarui berdasarkan output dari weak learner dan learning rate η . Prediksi model dihitung menggunakan rumus pada persamaan (11).

$$\hat{y}_i^{(t)} = \hat{y}_i^{(t-1)} + \eta \cdot h^t(x_i) \quad (11)$$

dengan:

- $\hat{y}_i^{(t)}$ = Prediksi baru untuk data ke-i
 η = Learning rate
 $h^t(x_i)$ = Prediksi weak learner pada iterasi ke-

2.) Prediction

Setelah proses iterasi selesai, prediksi akhir dihitung sebagai jumlah dari seluruh *weak learner* yang terbentuk. Prediksi akhir dihitung menggunakan rumus persamaan (12).

$$F(x) = \sum_{t=1}^T \eta \cdot h^t(x) \quad (12)$$

Dengan:

- $F(x)$ = Prediksi gabungan untuk input x.
 T = Jumlah total iterasi.
 $h^t(x)$ = *weak learner* pada iterasi ke-t.

2.5 Analisis Hasil

Dalam proses ini, confusion matrix digunakan untuk merepresentasikan hasil diagnosis mesin dalam bentuk matriks tabel. Matriks ini terdiri dari dua kelas, positif dan negatif, yang berfungsi untuk menggambarkan jumlah contoh data yang diklasifikasikan dengan benar. Selanjutnya, confusion matrix digunakan untuk menghitung metrik evaluasi seperti akurasi, presisi, dan recall. Dalam konteks ini, confusion matrix klasifikasi biner diterapkan, seperti yang disajikan pada Tabel 2

Tabel 2. Confusion Matrix

Class		Prediction	
		Positive	Negative
Actual	Positive	True Positive	False Negative
	Negative	False Positive	True Negative

Di mana:

True Positive (TP): jumlah contoh positif aktual yang diprediksi dengan benar sebagai positif.

True Negative (TN): jumlah contoh negatif aktual yang diklasifikasikan dengan benar sebagai negatif.

False Positive (FP): jumlah contoh negatif yang salah diklasifikasikan sebagai positif,

False Negative (FN): jumlah contoh positif yang salah diklasifikasikan sebagai negatif.

Berdasarkan nilai-nilai yang didapatkan dalam *confussion matrix*, dapat dihitung nilai akurasi, presisi, dan recall. Akurasi mengukur proporsi dari total contoh yang diklasifikasikan dengan benar oleh model. Akurasi dapat dihitung dengan menggunakan Persamaan (13).

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (13)$$

Precision digunakan untuk mengukur seberapa tepat model dalam memprediksi kelas tertentu. Perhitungan untuk mendapatkan nilai presisi dapat ditemukan dalam Persamaan (14).

$$Precisson = \frac{TP}{TP+FP} \quad (14)$$

Recall adalah metrik yang menggambarkan seberapa banyak data yang memiliki nilai aktual tertentu dapat diklasifikasikan dengan benar oleh algoritma. Artinya, semakin tinggi nilai recall suatu algoritma, maka semakin banyak jumlah data dengan kelas tertentu yang dapat diklasifikasikan dengan benar oleh algoritma tersebut. Akurasi dapat dihitung dengan menggunakan persamaan (15).

$$Recall = \frac{TP}{TP+FN} \quad (15)$$

3. HASIL DAN PEMBAHASAN

Untuk mengidentifikasi algoritma mana yang memberikan hasil paling efektif dalam mendiagnosis serangan jantung, penelitian ini membandingkan presisi, recall, dan akurasi algoritma AdaBoost dan CatBoost. Perbandingan ini akan dilakukan dengan menggunakan metode uji *Percentage Split* dan *K-Fold Cross-Validation*.

3.1 Algoritma AdaBoost

Perbandingan antara hasil prediksi diagnosis menggunakan algoritma AdaBoost dengan metode Percentage Split dan K-Fold Cross-Validation dapat direpresentasikan dalam bentuk *confusion matrix*, seperti yang ditunjukkan pada Tabel 3.

Tabel 3. Confusion Matrix pada Algoritma AdaBoost Menggunakan Metode Percentage Split dan K-Fold Cross-Validation

Label		Prediction			
		Metode Percentage Split		Metode K-Fold Cross-Validation	
Aktual	Positif	Positive	Negative	Positive	Negative
	Negatif	87	2	432	14
3	87	7	439		

Tabel 3 menunjukkan bahwa AdaBoost dengan menggunakan teknik percentage split menghasilkan prediksi yang cukup merata, namun condong menghasilkan false positive. Akan tetapi, pada teknik K-Fold Cross-Validation, AdaBoost lebih banyak menghasilkan false negative dibandingkan false positive. Perbedaan ini mengindikasikan bahwa teknik validasi yang digunakan dapat memengaruhi sensitivitas dan spesifisitas model terhadap masing-masing kelas. Nilai-nilai pada tabel tersebut selanjutnya akan digunakan untuk menghitung akurasi, presisi, dan recall dari masing-masing teknik pengujian menggunakan persamaan (1), (2), dan (3). Hasil dari perhitungan tersebut dapat dilihat pada Tabel 4.

Tabel 4. Precision, Recall, dan Accuracy pada Algoritma AdaBoost Menggunakan Metode Percentage Split dan K-Fold Cross-Validation

Target Label	Metode Percentage Split			Metode K-Fold Cross-Validation		
	Presisi (%)	Recall (%)	Accuracy (%)	Presisi (%)	Recall (%)	Accuracy (%)
Positif	97	98	97.21	98	97	97.65
Negatif	98	97		97	98	

Tabel 4 menunjukkan bahwa algoritma AdaBoost mampu mempertahankan kinerja yang tinggi dalam presisi, recall, dan akurasi, baik pada metode pembagian persentase maupun K-Fold Cross-Validation. Nilai presisi, recall, dan akurasi yang berada di atas 97% menunjukkan bahwa model memiliki kemampuan yang sangat baik dalam membedakan antara kelas positif dan kelas negatif. Presisi yang tinggi mengindikasikan bahwa sebagian besar prediksi positif yang dihasilkan oleh model memang benar, sehingga jumlah kesalahan berupa false positive relatif kecil. Sementara itu, nilai recall yang tinggi menunjukkan bahwa model mampu mendeteksi hampir seluruh pasien yang benar-benar mengalami serangan jantung, sehingga kesalahan berupa false negative dapat diminimalkan. Meskipun perbedaan nilai presisi dan recall antara kedua metode pengujian tampak kecil, perbedaan tersebut menunjukkan bahwa kinerja model masih dipengaruhi oleh teknik evaluasi yang digunakan. Dalam kasus algoritma AdaBoost, Metode K-Fold Cross-Validation menunjukkan nilai akurasi yang lebih tinggi yaitu 97,56%. Hal ini menunjukkan bahwa model memiliki tingkat ketepatan prediksi keseluruhan yang lebih baik ketika diuji pada data yang lebih bervariasi.

3.2 CatBoost Algorithm

Hasil dari prediksi diagnosis menggunakan algoritma AdaBoost dengan menggunakan metode Percentage Split dan K-Fold Cross-Validation dapat dibandingkan dengan diagnosis yang sebenarnya. Perbandingan tersebut dapat direpresentasikan dalam bentuk *confusion matrix*, seperti yang ditunjukkan pada Tabel 5.

Tabel 5. Confusion Matrix pada Algoritma CatBoost Menggunakan Metode Percentage Split dan K-Fold Cross-Validation

Label		Prediction			
		Metode Percentage Split		Metode K-Fold Cross-Validation	
Aktual	Positif	Positive	Negative	Positive	Negative
	Negatif	89	0	436	10
2	88	4	442		

Tabel 5 memperlihatkan distribusi hasil prediksi algoritma CatBoost terhadap dua teknik pengujian yang digunakan. Pada teknik percentage split, model menunjukkan prediksi sempurna terhadap kelas negatif dan hampir sempurna terhadap kelas positif. Sementara itu, metode K-Fold Cross-Validation menunjukkan bahwa

meskipun akurasi tetap tinggi, terdapat peningkatan jumlah kesalahan klasifikasi, terutama terlihat pada banyak false negative. Hal ini mengindikasikan bahwa model mengalami sedikit penurunan sensitivitas ketika dievaluasi pada data yang lebih bervariasi. Perbedaan performa ini menunjukkan bahwa CatBoost memiliki kinerja yang sangat baik pada data pelatihan yang dibagi secara acak (percentage split), namun sedikit terpengaruh oleh variasi distribusi data pada validasi silang, yang menuntut kestabilan prediksi yang lebih kuat di seluruh subset data. Untuk evaluasi lebih lanjut, perhitungan akurasi, presisi, dan recall dari masing-masing metode pengujian perlu dilakukan menggunakan persamaan (1), (2), dan (3). Hasil dari perhitungan tersebut dapat dilihat pada Tabel 6.

Tabel 6. Precision, Recall, dan Accuracy pada Algoritma CatBoost Menggunakan Metode Percentage Split dan K-Fold Cross-Validation

Target Label	Metode Percentage Split			Metode K-Fold Cross-Validation		
	Presisi (%)	Recall (%)	Akurasi (%)	Presisi (%)	Recall (%)	Akurasi (%)
Positif	98	100	98.88	99	98	98.43
Negatif	100	98		98	99	

Tabel 6 menunjukkan bahwa algoritma CatBoost memiliki kinerja klasifikasi yang sangat baik. Hal ini ditunjukkan oleh nilai presisi, recall, dan akurasi yang sangat tinggi pada kedua teknik pengujian. Nilai presisi yang tinggi menunjukkan bahwa sebagian besar prediksi positif yang dihasilkan oleh model benar-benar merupakan kasus positif, sehingga jumlah kesalahan berupa false positive relatif kecil. Dalam metode pemisahan persentase, recall mencapai nilai maksimum (100%) untuk kelas positif, yang menunjukkan bahwa model mampu mengidentifikasi semua label positif tanpa kesalahan. Namun, dalam teknik K-Fold Cross-Validation, nilai recall adalah 98%, meskipun tidak sempurna, hasil ini masih berada dalam kategori sangat baik yaitu di atas 95%. Akurasi yang tinggi pada kedua metode evaluasi menunjukkan bahwa sebagian besar data, baik kelas positif maupun negatif, berhasil diklasifikasikan dengan benar oleh model. Konsistensi skor pada kedua teknik pengujian ini memperkuat bukti bahwa CatBoost adalah algoritma yang tangguh dalam menangani masalah klasifikasi, meskipun tetap dipengaruhi oleh variasi distribusi data pelatihan. Hal ini mempertegas pentingnya evaluasi menggunakan lebih dari satu metode validasi untuk memperoleh gambaran menyeluruh terhadap kemampuan generalisasi model.

3.3 Perbandingan AdaBoost and CatBoost

Pada penelitian ini akan dilakukan evaluasi lebih lanjut, yaitu membandingkan kedua algoritma beserta dua metode pengujian yang digunakan. Perbandingan tersebut dapat diamati dari nilai akurasi yang diperoleh oleh masing-masing algoritma. Hasil akurasi tersebut ditampilkan pada Tabel 7.

Tabel 7. Precision, Recall, dan Accuracy pada Algoritma AdaBoost dan CatBoost Menggunakan Metode Percentage Split dan K-Fold Cross-Validation

Metode	Algoritma	Target Label	Presisi(%)	Recall (%)	Akurasi (%)
Percentage Split	AdaBoost	Positif	97	98	97.21
		Negatif	98	97	
	CatBoost	Positif	98	100	98.88
		Negatif	100	98	
K-Fold Cross-Validation	AdaBoost	Positif	98	97	97.65
		Negatif	97	98	
	CatBoost	Positif	99	98	98.43
		Negatif	98	99	

Tabel 7 menunjukkan bahwa kedua algoritma AdaBoost dan CatBoost memiliki kinerja klasifikasi yang sangat baik dalam mendiagnosis serangan jantung, dengan nilai presisi, recall, dan akurasi semua di atas 95% pada kedua metode Percentage Split dan K-Fold Cross-Validation. Hal ini menunjukkan bahwa kedua algoritma mampu membedakan antara kelas positif dan negatif dengan tingkat akurasi yang tinggi.

Berdasarkan hasil evaluasi yang ditunjukkan dalam Tabel 7, dapat dilihat bahwa baik AdaBoost maupun CatBoost mampu memberikan kinerja klasifikasi yang tinggi, dengan nilai presisi, recall, dan akurasi di atas 95% pada kedua teknik pengujian. Namun, terdapat beberapa perbedaan yang perlu diperhatikan. Pada metode Pembagian Persentase, CatBoost menunjukkan hasil yang lebih stabil dengan nilai presisi 100% untuk kelas negatif dan recall 100% untuk kelas positif. Hal ini menunjukkan bahwa CatBoost mampu meminimalkan kesalahan klasifikasi, baik dalam bentuk false positive maupun false negative. Sebaliknya, meskipun kinerja AdaBoost juga sangat baik, masih terdapat sejumlah kecil kesalahan klasifikasi pada kedua jenis kesalahan

tersebut. Perbedaan ini menunjukkan bahwa CatBoost memiliki keunggulan dalam meminimalkan kesalahan klasifikasi pada data yang dibagi secara acak.

Ketika diuji menggunakan K-Fold Cross-Validation, kedua algoritma mengalami perbedaan performa yang relatif kecil. Akan tetapi, CatBoost menunjukkan konsistensi yang lebih baik, dengan nilai presisi dan recall yang seimbang di angka 98% untuk semua kelas. Sementara itu, AdaBoost mengalami sedikit ketidakseimbangan antara jumlah false negative dan false positive. Meskipun mengalami sedikit ketidakseimbangan AdaBoost tetap menunjukkan performa yang sangat baik yaitu diatas 95%. Hal ini menunjukkan bahwa CatBoost lebih robust terhadap variasi data yang diperkenalkan oleh proses cross-validasi.

Secara keseluruhan, CatBoost menunjukkan performa yang lebih unggul dalam hal stabilitas dan keakuratan klasifikasi, baik pada percentage split maupun K-Fold Cross-Validation. Keunggulan ini dapat dikaitkan dengan kemampuan CatBoost dalam menangani fitur kategorikal secara internal serta dapat mencegah overfitting dan menjaga generalisasi model. Dengan demikian, dalam konteks dataset pada penelitian ini, CatBoost dapat dianggap sesuai untuk digunakan dibandingkan AdaBoost.

4. KESIMPULAN

Berdasarkan penelitian yang telah dilakukan, hasil uji dengan menggunakan percentage split mendapatkan nilai terbaik pada algoritma CatBoost. Sedangkan Hasil pengujian teknik k-fold cross validation didapatkan nilai terbaik juga pada algoritma CatBoost. Hasil yang didapatkan dari teknik k-fold cross validation memiliki nilai yang lebih tinggi dibandingkan dengan nilai dari teknik percentage split. Meskipun CatBoost memberikan performa terbaik secara keseluruhan, algoritma AdaBoost dengan kedua teknik uji juga menunjukkan hasil yang sangat baik dan masih dapat digunakan dalam klasifikasi penyakit serangan jantung. Kesimpulan dari penelitian ini adalah algoritma terbaik yang diterapkan pada dataset penyakit serangan jantung adalah algoritma CatBoost dengan teknik percentage split. Penelitian ini dapat dikembangkan dengan membandingkan algoritma AdaBoost, CatBoost, dan algoritma lain dengan menggunakan metode lain untuk menemukan metode terbaik untuk klasifikasi penyakit serangan jantung.

REFERENSI

- [1] World Health Organization, "Cardiovascular Diseases (CVDs)," World Health Organization. Accessed: May 20, 2025. [Online]. Available: [https://www.who.int/news-room/fact-sheets/detail/cardiovascular-diseases-\(cvds\)](https://www.who.int/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds))
- [2] N. A. Baghdadi, S. M. Farghaly Abdelaliem, A. Malki, I. Gad, A. Ewis, and E. Atlam, "Advanced Machine Learning Techniques for Cardiovascular Disease Early Detection and Diagnosis," *J. Big Data*, vol. 10, no. 1, pp. 1–29, 2023, doi: 10.1186/s40537-023-00817-1.
- [3] S. K. Devi, S. Krishnapriya, and D. Kalita, "Prediction of heart disease using data mining techniques," *Indian J. Sci. Technol.*, vol. 9, no. 39, pp. 1–5, Oct. 2016, doi: 10.17485/ijst/2016/v9i39/102078.
- [4] A. J. A. Al-Khafaji and N. N. A. Sjarif, "A Comprehensive Review of Early Detection of COVID-19 Based on Machine Learning and Deep Learning Models," *International Journal of Electrical and Computer Engineering*, vol. 14, no. 4, pp. 4167–4174, Aug. 2024, doi: 10.11591/ijece.v14i4.pp4167-4174.
- [5] M. Jordan, J. Kleinberg, and B. Schölkopf, *Pattern Recognition and Machine Learning*. New York: Springer, 2006.
- [6] A. Mishra, B. B. Gupta, D. Peraković, and F. J. G. Peñalvo, "A Survey on Data mining classification approaches," *International Conference on Smart Systems and Advanced Computing*, 2021, [Online]. Available: <http://ceur-ws.org>
- [7] A. K. Putri and H. Suparwito, "Uji Algoritma Stacking Ensemble Classifier pada Kemampuan Adaptasi Mahasiswa Baru dalam Pembelajaran Online," *KONSTELASI: Konvergensi Teknologi dan Sistem Informasi*, vol. 3, no. 1, pp. 1–12, Jun. 2023.
- [8] J. Brownlee, "Boosting and AdaBoost for Machine Learning," *Machine Learning Mystery*. Accessed: May 26, 2025. [Online]. Available: <https://machinelearningmastery.com/boosting-and-adaboost-for-machine-learning/>
- [9] D. Singh and S. Agarwal, "XGBoost And AdaBoost," National Institute of Science Education and Research (NISER). Accessed: May 28, 2025. [Online]. Available: <https://www.niser.ac.in/~smishra/teach/cs460/23cs460/lectures/lec21.pdf>
- [10] C. Tu, H. Liu, and B. Xu, "AdaBoost typical Algorithm and its application research," in *MATEC Web of Conferences*, EDP Sciences, Dec. 2017, pp. 1–6. doi: 10.1051/mateconf/201713900222.
- [11] H. El Hamdaoui, S. Boujraf, N. E. H. Chaoui, B. Alami, and M. Maaroufi, "Improving Heart Disease Prediction Using Random Forest and AdaBoost Algorithms," *International Journal of*

- Online and Biomedical Engineering*, vol. 17, no. 11, pp. 60–75, 2021, doi: 10.3991/ijoe.v17i11.24781.
- [12] A. Sanusi, C. A. Putra, and F. A. Akbar, “Implementation of AdaBoost Algorithm on C50 for Improving the Performance of Liver Disease Classification,” *JEECS (Journal of Electrical Engineering and Computer Sciences)*, vol. 8, no. 2, pp. 93–102, Dec. 2023, doi: 10.54732/jeeecs.v8i2.1.
- [13] S. Dharmawan, V. Fernandes, and H. Halim, “Prediksi Serangan Jantung dengan Menggunakan Metode Logistic Regression Classifier dan Adaboost,” *Computatio: Journal of Computer Science and Information Systems*, vol. 8, no. 1, pp. 96–103, 2024.
- [14] A. William, “A Comprehensive Mathematical Approach to Understand AdaBoost,” Towards Data Science. Accessed: May 24, 2025. [Online]. Available: <https://towardsdatascience.com/a-comprehensive-mathematical-approach-to-understand-adaboost-f185104edced/>
- [15] P. Samosir P. and U. Salamah, “Perbandingan Performa Algoritma XGBoost, CatBoost Dan GBM Dalam Prediksi Penyakit Kardiovaskular,” *JSAI (Journal Scientific and Applied Informatics)*, vol. 8, no. 1, pp. 268–273, Jan. 2025.
- [16] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin, “CatBoost: Unbiased Boosting with Categorical Features,” Jun. 2017, [Online]. Available: <http://arxiv.org/abs/1706.09516>
- [17] N. Bhaskar, R. R. Borhade, S. Barekar, M. Bachute, and V. Bairagi, “CNN-CatBoost Ensemble Deep Learning Model for Enhanced Disease Detection and Classification of Kidney Disease,” *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 34, no. 1, pp. 144–151, Apr. 2024, doi: 10.11591/ijeeecs.v34.i1.pp144-151.
- [18] M. E. Haque, S. M. J. Islam, J. Maliha, Md. S. H. Sumon, R. Sharmin, and S. Rokoni, “Improving Chronic Kidney Disease Detection Efficiency: Fine Tuned CatBoost and Nature-Inspired Algorithms with Explainable AI,” 2025.
- [19] A. I. Nurhidayat, Asmunun, and D. Fatrianto, “Prediksi Kinerja Akademik Mahasiswa Menggunakan Machine Learning dengan Sequential Minimal Optimization untuk Pengelola Program Studi,” *JIEET: Journal Information Engineering and Educational Technology*, vol. 5, no. 2, pp. 84–91, 2021.
- [20] A. Widyanto, Kusrini, and Kusnawi, “Pengaruh Keseimbangan Data Terhadap Akurasi Model Support Vector Machine pada Data Set Donor Darah,” *Jurnal Teknologi Terpadu*, vol. 9, no. 2, pp. 79–88, 2023.
- [21] H. Hanum, S. Lamin, S. Yahdin, A. Desiani, D. Geovani, and R. An Fadhila Chaniago, “Percentage Split dan K-Fold Cross Validation pada Algoritma Support Vector Machine (SVM) pada Klasifikasi Penyakit Anemia,” *JSI : Jurnal Sistem Informasi (E-Journal)*, vol. 16, no. 2, pp. 462–470, Oct. 2024.
- [22] Wijiyanto, A. I. Pradana, Sopingi, and V. Atina, “Teknik K-Fold Cross Validation untuk Mengevaluasi Kinerja Mahasiswa,” *Jurnal Algoritma*, vol. 21, no. 1, pp. 239–248, May 2024, doi: 10.33364/algoritma/v.21-1.1618.
- [23] T. A. Rashid and B. Hassan, “Heart Attack Dataset.” Accessed: Jun. 08, 2024. [Online]. Available: <https://www.kaggle.com/datasets/fatemehmohammadinia/heart-attack-dataset-tarik-a-rashid>
- [24] M. Buda, A. Maki, and M. A. Mazurowski, “A Systematic Study of the Class Imbalance Problem in Convolutional Neural Networks,” *Neural Networks*, vol. 106, pp. 249–259, 2018, doi: 10.1016/j.neunet.2018.07.011.