

**LEMBAR PENGESAHAN**  
**PROPOSAL TUGAS AKHIR**  
**KLASIFIKASI ARITMIA PADA SINYAL**  
**ELEKTROKARDIOGRAM MENGGUNAKAN ENSEMBLE**  
**MACHINE LEARNING DENGAN KERANGKA KERJA**  
***BOOSTING***  
***(ARRHYTHMIA CLASSIFICATION IN***  
***ELECTROCARDIOGRAM SIGNALS USING***  
***ENSEMBLE MACHINE LEARNING WITH BOOSTING***  
***FRAMEWORK)***

**Telah disetujui dan disahkan sebagai Proposal Tugas Akhir**

**Program Studi S1 Teknik Biomedis**

**Fakultas Teknik Elektro**

**Universitas Telkom**

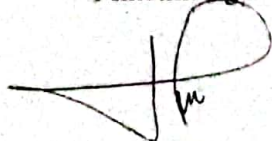
**Disusun oleh:**

**CATHERINE MICHELLE SUDARNO**

**1105213071**

**Bandung, 12 Desember 2024**

**Pembimbing I**



**Dr. Tito Waluyo Purboyo, S.Si., S.T., M.PMat.**

**NIP. 0407117301**

**Pembimbing II**



**Dr. Dziban Naufal, S.Si., M.T.**

**NIP. 24960005**

**KLASIFIKASI ARITMIA PADA SINYAL  
ELEKTROKARDIOGRAM MENGGUNAKAN ENSEMBLE  
MACHINE LEARNING DENGAN KERANGKA KERJA  
*BOOSTING*  
(*ARRHYTHMIA CLASSIFICATION IN  
ELECTROCARDIOGRAM SIGNALS USING  
ENSEMBLE MACHINE LEARNING WITH BOOSTING  
FRAMEWORK*)**

**PROPOSAL TUGAS AKHIR**

Disusun sebagai syarat mata kuliah Penyusunan Karya Ilmiah dan Proposal

Di Program Studi S1 Teknik Biomedis

Disusun oleh:

**CATHERINE MICHELLE SUDARNO**

**1105213071**



**Universitas  
Telkom**

**FAKULTAS TEKNIK ELEKTRO**

**UNIVERSITAS TELKOM**

**BANDUNG**

**2024**

## ABSTRAK

Penyakit jantung merupakan salah satu jenis Penyakit Tidak Menular (PTM) yang menjadi penyebab angka kematian tertinggi di dunia. Salah satu jenis penyakit jantung adalah aritmia, aritmia adalah variasi – variasi di luar irama normal jantung berupa kelainan pada kecepatan, keteraturan, tempat asal impuls, atau urutan aktivasi, dengan atau tanpa adanya penyakit jantung struktural yang mendasari. Pada tahun 2024, Kementerian Kesehatan Republik Indonesia menyatakan data epidemiologi global menunjukkan prevelansi aritmia jantung berkisar antara 1.5% hingga 5% dari populasi. Langkah pertama dalam diagnosis pasien yang dicurigai mengalami aritmia adalah melakukan pemeriksaan elektrokardiogram (EKG).

Salah satu metode untuk mendiagnosis aritmia secara akurat, efisien, dan otomatis adalah dengan pengklasifikasian elektrokardiogram menggunakan *ensemble machine learning*. Boosting adalah salah satu metode ensemble learning, yang melibatkan pembelajaran sekuensial adaptif dengan satu tipe model base-model yang digabungkan untuk hasil terbaik. Sinyal elektrokardiogram diklasifikasikan berdasarkan denyut sesuai standar AAMI menggunakan algoritma ensemble machine learning dengan kerangka kerja boosting seperti AdaBoost, GradientBoost, XGBoost, LightGBM, dan CatBoost.

Tujuan utama penelitian ini adalah untuk menemukan arsitektur terbaik dalam metode boosting untuk mengklasifikasikan berbagai jenis aritmia pada sinyal elektrokardiogram (EKG). Selain itu, penelitian ini juga bertujuan untuk membandingkan kinerja berbagai jenis arsitektur boosting dengan model tunggal, metode bagging, stacking, serta algoritma deep learning. Selanjutnya, penelitian ini akan mengoptimalkan nilai-nilai hyperparameter pada arsitektur boosting yang digunakan, guna meningkatkan performa sistem secara keseluruhan. Dengan pendekatan ini, diharapkan hasil klasifikasi aritmia pada sinyal elektrokardiogram dapat lebih akurat dan mampu mengidentifikasi pola-pola penting yang mungkin terlewatkan oleh metode lain.

**Kata Kunci:** *Aritmia, boosting, denyut jantung, ensemble machine learning, klasifikasi, sinyal EKG.*

## DAFTAR ISI

LEMBAR PENGESAHAN .....	i
ABSTRAK .....	iii
DAFTAR GAMBAR .....	vi
DAFTAR TABEL .....	vii
DAFTAR PERSAMAAN .....	viii
DAFTAR ISTILAH .....	ix
BAB I PENDAHULUAN .....	1
1.1. Latar Belakang Masalah .....	1
1.2. Rumusan Masalah .....	2
1.3. Tujuan dan Manfaat .....	2
1.4. Batasan Masalah .....	3
1.5. Metode Penelitian .....	3
1.6. Jadwal Pelaksanaan .....	4
BAB II TINJAUAN PUSTAKA .....	5
2.1. Jantung .....	5
2.1.1 Patofisiologi Aritmia .....	6
2.2. Elektrokardiogram .....	7
2.2.1 Karakteristik Sinyal Elektrokardiogram .....	7
2.3 <i>Artificial Intelligence</i> .....	8
2.3.1 <i>Machine Learning</i> .....	8
BAB III .....	18
PERANCANGAN SISTEM .....	18
3.1 Desain Sistem Klasifikasi .....	18
3.1.1 Diagram Blok Umum Sistem Klasifikasi .....	18
3.2 Desain Algoritma Sistem Klasifikasi .....	19
3.2.1 Akuisisi Data .....	20

3.2.2 <i>Exploratory Data Analysis</i> (EDA).....	24
3.2.3 <i>Pre-Processing</i> .....	24
3.2.4 <i>Time Series Feature Extraction Library</i> (TSFEL) .....	25
3.2.5 Klasifikasi .....	28
3.2.6 Evaluasi Metrik.....	38
3.3 Skenario pengujian algoritma klasifikasi .....	39
DAFTAR PUSTAKA .....	40

## DAFTAR GAMBAR

<b>Gambar 2.1</b> Anatomi jantung.....	5
<b>Gambar 2.2</b> Karakteristik sinyal elektrokardiogram.....	7
<b>Gambar 2.3</b> Kerangka kerja metode bagging.....	9
<b>Gambar 2.4</b> Kerangka kerja metode stacking.....	10
<b>Gambar 2.5</b> Kerangka kerja boosting.....	10
<b>Gambar 2.6</b> Proses algoritma AdaBoost.....	11
<b>Gambar 2.7</b> Proses algoritma GradientBoost.....	12
<b>Gambar 2.8</b> Struktur <i>decision tree</i> pada XGBoost, LightGBM, dan CatBoost...	14
<b>Gambar 2.9</b> Pemetaan <i>confusion matrix</i> .....	15
<b>Gambar 2.10</b> Grafik ROC.....	16
<b>Gambar 3.1</b> Diagram blok sistem secara umum.....	18
<b>Gambar 3.2</b> Diagram blok sistem secara detail.....	19
<b>Gambar 3.3</b> Ilustrasi akuisisi data.....	24
<b>Gambar 3.4</b> Proses ekstraksi ciri menggunakan TSFEL.....	25
<b>Gambar 3.5</b> Diagram alir <i>decision tree</i> .....	30
<b>Gambar 3.6</b> Diagram alir AdaBoost.....	32
<b>Gambar 3.7</b> Diagram alir GradientBoost.....	33
<b>Gambar 3.8</b> Diagram alir XGBoost.....	34
<b>Gambar 3.9</b> Diagram alir LightGBM.....	36
<b>Gambar 3.10</b> Diagram alir CatBoost.....	38
<b>Gambar 3.11</b> Alur kerja GridSearchCV.....	39

## DAFTAR TABEL

<b>Tabel 1.1</b> Contoh Jadwal dan <i>Milestone</i> .....	4
<b>Tabel 2.1</b> Standar klasifikasi aritmia AAMI.....	6
<b>Tabel 2.2</b> Penelitian sebelumnya.....	17
<b>Tabel 3.1</b> Pemetaan label MIT-BIH.....	23
<b>Tabel 3.2</b> Fitur spektral yang tersedia pada TSFEL.....	26
<b>Tabel 3.3</b> Fitur statistik yang tersedia pada TSFEL.....	27
<b>Tabel 3.4</b> Fitur temporal pada TSFEL.....	28

## DAFTAR PERSAMAAN

(1) $f(x) = \sum \alpha h(x)$ .....	10
(2) $Accuracy = \frac{tp+tn}{tp+fp+tn+fn} \times 100\%$ .....	15
(3) $Recall = \frac{tp}{tp+fn} \times 100\%$ .....	15
(4) $Precision = \frac{tp}{tp+fp} \times 100\%$ .....	16
(5) $F1\ score = 2 \times \frac{precision \times recall}{precision+recall} \times 100\%$ .....	16



## DAFTAR ISTILAH

AAMI: *Association for the Advancement of Medical Instrumentation*  
ADC: *Analog Digital Converter*  
AI: *Artificial Inteligence*  
AUC: *Area Under Curver*  
AV: *Atrioventricular*  
BEP: *Bayesian Error Pruning*  
CPU: *Central Processor Unit*  
CSV: *Comma Separated Values*  
DL: *Deep Learning*  
ML: *Machine Learning*  
DWT: *Discrete Wavelet Transform*  
ECDF: *Empirical Cumulative Distribution Function*  
EDA: *Explorary Data Analysis*  
EFB: *Exclusive Feature Bundling*  
EKG: *Elektrokardiogram*  
F: *Fusion Heartbeat*  
FN: *False Negative*  
FP: *False Positive*  
GOSS: *Gradient-based One-Side Sampling*  
LPCC: *Linear Prediction Cepstral Coefficient*  
MFCC: *Mel-Frequency Cepstral Coefficient*  
MIT: *Massachussetts Institute of Technology*  
MIT-BIH: *Massachussetts Institute of Technology – Beth Israel Hospital*  
N: *Normal heart rhythm*  
PERITMI: *Perhimpunan Aritmia Indonesia*  
PTM: *Penyakit Tidak Menular*  
Q: *Unknown ECG beat*  
ROC: *receiver operating characteristics*  
S: *Supraventricular abnormalities*  
TA: *Tugas Akhir*

TN: *True Negative*

TP: *True Positive*

TS: *Target Statistics*

TSFEL: *Time Series Feature Extraction Library*

V: *Ventricular escape*

# BAB I

## PENDAHULUAN

### 1.1. Latar Belakang Masalah

Salah satu organ tubuh yang berperan vital bagi manusia adalah jantung. Jantung memiliki peran penting dalam proses aliran darah dan merupakan penopang semua jaringan tubuh [1]. Penyakit jantung merupakan salah satu jenis Penyakit Tidak Menular (PTM) yang menjadi penyebab angka kematian tertinggi di dunia [2]. Salah satu jenis penyakit jantung adalah aritmia, aritmia adalah variasi – variasi di luar irama normal jantung berupa kelainan pada kecepatan, keteraturan, tempat asal impuls, atau urutan aktivasi, dengan atau tanpa adanya penyakit jantung struktural yang mendasari [3]. Empat jenis utama aritmia adalah detak prematur, aritmia ventrikular, bradikardia, dan aritmia supraventrikular [4].

Pada tahun 2024, Kementerian Kesehatan Republik Indonesia menyatakan data epidemiologi global menunjukkan prevalensi aritmia jantung berkisar antara 1.5% hingga 5% dari populasi, dengan fibrilasi atrium sebagai kasus yang paling tinggi [5]. Di Indonesia, prevalensi fibrilasi atrium berdasarkan survey rumah sakit tersier sekitar 3.2% [6]. Melalui organisasi seperti Perhimpunan Aritmia Indonesia (PERITMI)/*Indonesian Heart Rhythm Society* (InaHRS), pemerintah berupaya meningkatkan pengetahuan masyarakat dengan kampanye edukasi rutin serta menyediakan layanan kesehatan [7].

Langkah pertama dalam diagnosis pasien yang dicurigai mengalami aritmia adalah melakukan pemeriksaan elektrokardiogram (EKG) [8]. EKG adalah hasil pemeriksaan non-invasif yang melacak listrik yang dihasilkan jantung dari permukaan tubuh. Proses ini melibatkan alat elektrokardiograf untuk menghasilkan grafik yang menunjukkan aktivitas listrik jantung [9]. Salah satu metode untuk mendiagnosis aritmia dengan akurat, efisien, objektif dan otomatis adalah dengan teknik pengklasifikasian elektrokardiogram dengan *ensemble machine learning*. *Ensemble machine learning* menunjukkan kinerja yang lebih unggul dibandingkan *single machine learning* dalam menangani klasifikasi yang lebih dari dua kelas (*multiclass classification*) [10].

*Ensemble Learning* adalah paradigma *machine learning* dimana beberapa model (biasanya disebut "*weak learners*") dilatih untuk memecahkan masalah yang sama dan digabungkan untuk mendapatkan hasil yang lebih baik. Hipotesis utamanya adalah bahwa ketika model yang lemah digabungkan, didapat memperoleh model yang lebih akurat dibanding *single machine learning*. Boosting adalah salah satu metode *Ensemble Learning* yang menggunakan hanya satu tipe model base-model dengan melakukan pembelajaran secara sekuensial secara adaptif (hasil dari sebuah base-model tergantung dari hasil base-model sebelumnya), dan kemudian digabungkan untuk mendapatkan hasil yang terbaik. [11]. Boosting mempermudah pemahaman model dan membantu mengurangi kesalahan serta ketidaktepatan dalam *ensemble machine learning* [12].

Pada studi ini, sinyal elektrokardiogram diklasifikasikan berdasarkan denyutnya sesuai dengan standar AAMI. Sinyal diklasifikasi menggunakan algoritma *ensemble machine learning* dengan beberapa metode kerangka kerja boosting, yaitu AdaBoost, GradientBoost, XGBoost, LightGBM, dan CatBoost. Dengan pendekatan ini, diharapkan hasil klasifikasi aritmia pada sinyal elektrokardiogram dapat lebih akurat dan mampu mengidentifikasi pola-pola penting yang mungkin terlewatkan oleh metode lain.

### **1.2. Rumusan Masalah**

Berdasarkan latar belakang yang dibahas, berikut rumusan masalah yang akan diidentifikasi pada studi ini:

1. Apa jenis arsitektur boosting terbaik dalam mengklasifikasi jenis-jenis aritmia?
2. Bagaimana performa sistem dengan jenis-jenis arsitektur pada metode boosting dibandingkan dengan model tunggal, metode stacking, metode bagging, dan algoritma *deep learning* pada klasifikasi aritmia?
3. Bagaimana cara mengoptimalkan *hyperparameter* pada jenis-jenis arsitektur pada metode boosting?

### **1.3. Tujuan dan Manfaat**

Berdasarkan rumusan masalah yang telah dijabarkan, maka tujuan dari studi ini, yaitu:

1. Mengidentifikasi jenis arsitektur terbaik pada metode boosting dalam mengklasifikasi jenis-jenis aritmia pada sinyal elektrokardiogram.
2. Mengevaluasi performa sistem dengan jenis-jenis arsitektur pada metode boosting dibandingkan dengan model tunggal, metode bagging, metode stacking, dan algoritma *deep learning*.
3. Mengoptimalkan *hyperparameter* pada jenis-jenis arsitektur pada metode boosting.

#### **Manfaat Penelitian:**

Penggunaan *ensemble machine learning* dengan kerangka boosting dapat meningkatkan kemampuan sistem untuk mengklasifikasi jenis-jenis aritmia berdasarkan denyutnya secara *multiclass*. Studi ini dapat memberikan metode yang lebih baik dalam memanfaatkan sinyal EKG, sehingga dapat mendeteksi pola-pola yang kompleks dan sulit diidentifikasi oleh *single machine learning*.

#### **1.4. Batasan Masalah**

Adapun batasan dalam melakukan studi ini adalah sebagai berikut:

1. Data yang dipakai merupakan dataset yang diambil dari MIT-BIH.
2. Studi ini hanya berfokus pada klasifikasi aritmia menggunakan sinyal elektrokardiogram yang sudah tersedia dalam dataset terstandarisasi.
3. Analisis dilakukan menggunakan sinyal EKG dari pasien dewasa dengan kondisi aritmia tertentu, tanpa mempertimbangkan jenis kelamin, usia, atau kondisi komorbid lain.
4. Model yang dikembangkan dievaluasi berdasarkan metrik akurasi, *recall*, *precision*, f1 score, dan AUC, serta mempertimbangkan biaya komputasi atau waktu eksekusi.
5. Hanya 5 model boosting yang diuji pada studi ini, yaitu AdaBoost, XGBoost, GradientBoost, LightGBM, dan CatBoost.

#### **1.5. Metode Penelitian**

1. Studi literatur pada penelitian sebelumnya yang terkait dengan judul.
2. Akuisisi data sekunder dari MIT-BIH.

3. Analisis data secara statistik dengan *Exploratory Data Analysis* (EDA)
4. *Pre-processing* pada data.
5. Ekstraksi ciri dari data.
6. Perancangan algoritma sistem sebelum dioptimasi.
7. Klasifikasi sesuai standar AAMI.
8. Analisis performansi sistem.
9. Melakukan optimasi pada sistem.
10. Klasifikasi sesuai standar AAMI.
11. Analisis performansi sistem setelah dioptimasi.

### 1.6. Jadwal Pelaksanaan

Berisi jadwal pelaksanaan pengerjaan Tugas Akhir. Perlu ditetapkan beberapa *milestone* untuk menentukan pencapaian pekerjaan. Jadwal pelaksanaan akan menjadi acuan dalam mengevaluasi tahap-tahap pekerjaan seperti yang tertuang dalam *milestone* yang sudah ditetapkan seperti di Tabel 1.1.

**Tabel 1.1** Jadwal dan *Milestone*.

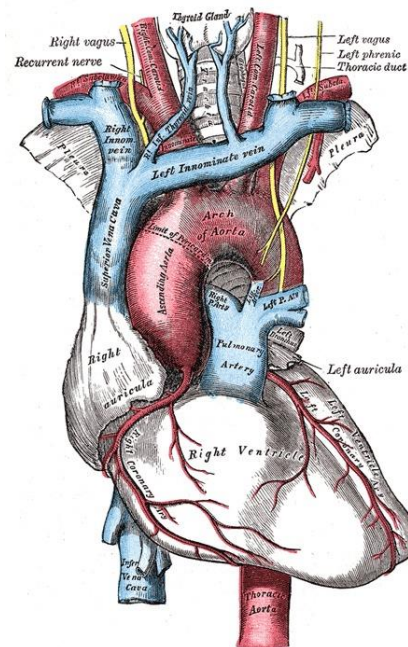
No.	Deskripsi Tahapan	Durasi	Tanggal Selesai	<i>Milestone</i>
1	Studi Literatur	6 bulan	30 Maret 2025	Memahami kajian penelitian sebelumnya
2	Desain Sistem	2 bulan	2 Desember 2024	Diagram blok dan spesifikasi arsitektur model
3	Pengumpulan dan Pemrosesan Data	6 minggu	6 Desember 2024	Dataset aritmia dapat digunakan
4	Implementasi Model	2 bulan	28 Februari 2025	Model boosting dan modifikasinya selesai
5	Pengujian dan Evaluasi Model	3 minggu	21 Maret 2025	Hasil akurasi dan evaluasi diperoleh
6	Penyusunan Laporan Buku TA	6 bulan	30 Maret 2025	Buku TA Selesai

## BAB II

### TINJAUAN PUSTAKA

#### 2.1. Jantung

Jantung adalah organ berotot di belakang tulang dada di antara paru-paru [13]. Bersama dengan pembuluh darah dan darah, jantung membentuk sistem peredaran darah tubuh [9]. Jantung terdiri dari empat ruang, yaitu atrium kanan, atrium kiri, ventrikel kanan, dan ventrikel kiri. Atrium dan ventrikel kanan sering disebut sebagai jantung kanan, dan ventrikel kanan dan kiri disebut sebagai ventrikel kanan dan kiri. Atrium kiri dan ventrikel kiri secara fungsional membentuk jantung kiri [13]. Atrium kanan menerima darah terdeoksigenasi dari tubuh dan mengalirkannya ke ventrikel kanan melalui katup trikuspid. Ventrikel kanan memompa darah ke paru-paru untuk oksigenasi melalui katup pulmonalis. Darah beroksigen dari paru-paru masuk ke atrium kiri melalui vena pulmonalis dan diteruskan ke ventrikel kiri melalui katup mitral. Ventrikel kiri kemudian memompa darah beroksigen ke seluruh tubuh melalui katup aorta, mengulangi siklus ini setiap detak jantung [14]. Gambar 2.1 menunjukkan anatomi jantung serta nama dari bagian-bagian jantung.



**Gambar 2.1** Anatomi jantung [14]

### 2.1.1 Patofisiologi Aritmia

Patofisiologi jantung adalah ilmu yang mempelajari bagaimana fungsi jantung terganggu karena berbagai faktor. Patofisiologi otot jantung bergantung pada kerusakan sel otot jantung, yang menyebabkan daya kontraksi yang tidak sesuai. Faktor umum pada patofisiologi adalah hipertensi, penyakit koroner, masalah bawaan, infark miokard, iskemia, syok, aritmia, penyakit genetik, kelainan struktural, efusi perikardial, emboli, tamponade, dan masih banyak lagi [15].

Aritmia adalah irama jantung yang tidak normal. Aritmia dapat mencakup berbagai gangguan denyut jantung dan kelainan irama. Pasien dengan aritmia dapat menunjukkan gejala yang beragam, mulai dari tidak ada sama sekali hingga henti jantung yang mendadak. Aritmia dapat bersifat paroksismal, yang membuatnya sulit untuk mengetahui seberapa sering terjadi. Berdasarkan denyut jantung, aritmia biasanya dibagi menjadi takikardia dan bradikardia. Takikardia didefinisikan sebagai irama abnormal dengan denyut jantung ventrikel 100 denyut per menit atau lebih sedangkan bradikardia didefinisikan sebagai denyut jantung di bawah 60 denyut per menit (bpm) [8]. Pada studi ini, aritmia akan diklasifikasikan berdasarkan standar AAMI (Association for the Advancement of Medical Instrumentation) seperti yang ditunjukkan pada Tabel 2.1.

**Tabel 2.1** Standar klasifikasi aritmia AAMI [16]

Heart rhythm label	Content description
N	Normal heart rhythm
S	Supraventricular abnormalities
V	Ventricular escape
F	Fusion Heartbeat
Q	Unknown ECG beat

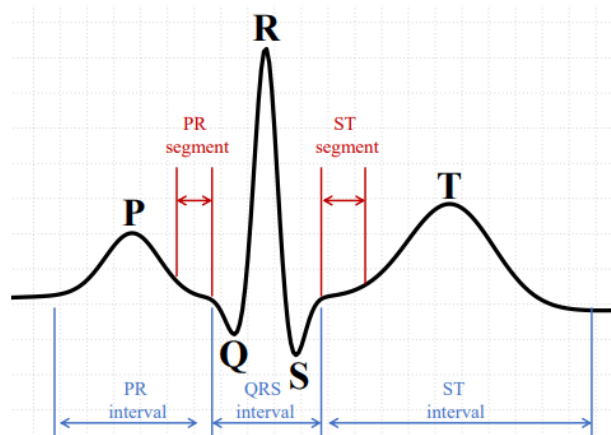


## 2.2. Elektrokardiogram

Langkah pertama dalam diagnosis pasien yang dicurigai mengalami aritmia adalah elektrokardiogram (EKG) [8]. Elektrokardiogram (EKG) adalah sebuah hasil rekaman aktivitas listrik pada jantung dan direkam secara non-invasif dari permukaan kulit di posisi jantung [9].

### 2.2.1 Karakteristik Sinyal Elektrokardiogram

Sinyal elektrokardiogram terdiri dari lima gelombang, yaitu gelombang P, Q, R, S, dan T. Setiap gelombang menunjukkan peristiwa tertentu yang terjadi selama siklus jantung. Gelombang QRS dianggap sebagai ciri yang paling menonjol dari sinyal EKG dari seluruh gelombang karena kehadirannya yang menonjol. Oleh karena itu, gelombang QRS digunakan sebagai titik referensi untuk seluruh siklus jantung [17]. Gambar 2.2 menunjukkan karakteristik sinyal elektrokardiogram pada siklus normal.



**Gambar 2.2** Karakteristik sinyal elektrokardiogram [18]

Gelombang P menunjukkan perubahan potensial pada depolarisasi atrium, sedangkan gelombang QRS menunjukkan perubahan potensial pada depolarisasi ventrikel. Interval PR menunjukkan waktu dimulainya depolarisasi atrium, yang dimulai dari awal gelombang P dan berakhir pada awal gelombang QRS. Segmen ST menunjukkan proses repolarisasi ventrikel yang lebih lambat, yang dimulai dari akhir gelombang QRS dan berakhir pada awal gelombang T. Gelombang T menunjukkan perubahan potensial pada depolarisasi ventrikel yang dimulai dengan gelombang QRS awal dan berakhir dengan akhir gelombang T. Interval ST merepresentasikan waktu saat ventrikel sedang mengalami depolarisasi penuh

sebelum kembali ke repolarisasi, yang dimulai dari akhir gelombang S dan awal gelombang T. Segmen PR merepresentasikan waktu saat impuls listrik bergerak melalui nodus AV (*Atrioventricular*) sebelum diteruskan ke ventrikel, dimulai dari akhir gelombang P dan awal gelombang QRS [19].

### **2.3 Artificial Intelligence**

Artificial Intelligence (AI) adalah istilah yang mengacu pada simulasi kecerdasan manusia oleh sistem atau mesin. Tujuan AI adalah untuk mengembangkan mesin yang memiliki kemampuan untuk berpikir seperti manusia dan meniru berbagai perilaku manusia, seperti memahami, menalar, belajar, merencanakan, memprediksi, dan sebagainya. Dengan revolusi industri yang terus berlanjut, semakin banyak mesin yang menggantikan tenaga kerja manusia di berbagai sektor masyarakat. Banyak ilmuwan berkonsentrasi pada bidang kecerdasan buatan sehingga menghasilkan bidang penelitian yang beragam jenisnya. Beberapa contoh bidang penelitian AI adalah algoritma pencarian, grafik pengetahuan, pemrosesan bahasa alami, sistem pakar, algoritma evolusi, machine learning (ML) dan deep learning (DL) [20].

#### **2.3.1 Machine Learning**

*Machine learning* adalah bagian dari AI yang digunakan untuk mengembangkan program komputer. Dengan menggunakan sejumlah algoritma dan statistik, program-program ini dipandu oleh sekumpulan data yang luas. *Machine learning* membantu melacak data dengan membuat model prediksi. Salah satu metode yang digunakan dalam *machine learning* adalah *supervised learning*, contoh dari *supervised learning* adalah klasifikasi. Klasifikasi adalah proses mengkategorikan sebuah objek ke dalam kategori yang telah ditentukan sebelumnya [21]. Meskipun *deep learning* mengguguli algoritma *machine learning* dasar dalam segi data berdimensi tinggi, *machine learning* dasar masih dapat mencapai hasil yang lebih baik daripada *deep learning* untuk input data berdimensi rendah, terutama dalam kasus ketersediaan data pelatihan yang terbatas [22].

##### **2.3.1.1 Ensemble Machine Learning**

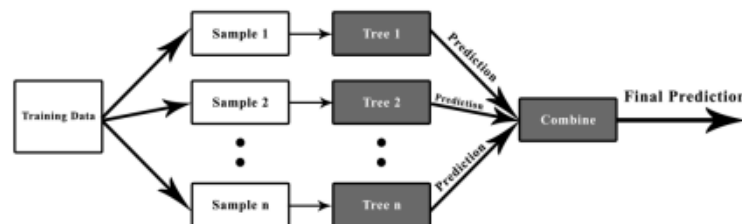
*Ensemble learning* adalah metode pembelajaran yang menggabungkan berbagai model dasar untuk membuat model tunggal yang lebih baik dibanding

model dasar [23]. Dengan kata lain *ensemble machine learning* adalah metode pembelajaran yang menggabungkan berbagai *machine learning* sederhana untuk membuat model baru yang lebih baik dibanding *machine learning* dasar.

*Ensemble learning* dapat mengurangi risiko *overfitting* dikarenakan keragaman model dasar. *Ensemble learning* juga lebih efektif dibandingkan model dasar di berbagai bidang penelitian. Metode *ensemble* yang paling umum digunakan adalah bagging, boosting, dan stacking [23].

#### a. Bagging

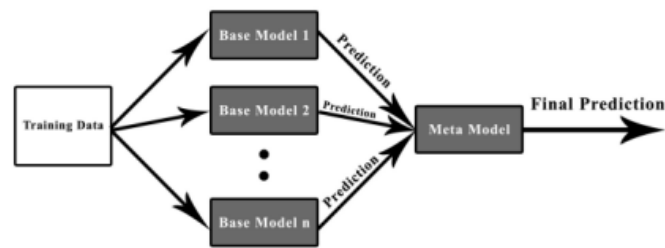
Bagging yang juga dikenal sebagai bootstrap aggregating, adalah algoritma yang membuat beberapa subset kecil data dari kumpulan data yang sebenarnya. Tujuan dari metode bagging adalah untuk membuat model prediksi yang lebih beragam dengan mengubah distribusi stokastik dari set data pelatihan, di mana perubahan kecil dalam kumpulan data pelatihan akan menghasilkan perubahan besar dalam prediksi model [23]. Kerangka kerja metode bagging dapat dilihat di Gambar 2.3.



**Gambar 2.3** Kerangka kerja metode bagging [11]

#### b. Stacking

Metode stacking atau dapat disebut sebagai *stacked generalization*, adalah teknik penggabungan model untuk menggabungkan informasi dari beberapa model dasar untuk menghasilkan model baru (meta-model). Arsitektur dari model stacking melibatkan dua atau lebih model dasar (model level 0) yang digabung menjadi meta-model (model level 1). Pada model level 1, model mempelajari cara menggabungkan prediksi model level 0 dengan efektif. Dalam klasifikasi, keluaran dari model level 0 berupa label kelas [23]. Kerangka kerja metode stacking dapat dilihat di Gambar 2.4.



**Gambar 2.4** Kerangka kerja metode stacking [11]

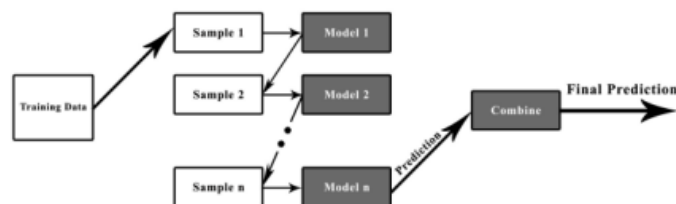
### c. Boosting

Metode boosting adalah proses berurutan dimana setiap model berikutnya mencoba memperbaiki kesalahan model sebelumnya. Metode ini terdiri dari beberapa model dasar yang lemah (*weak learner*) yang diurutkan secara adaptif yang menghasilkan satu *strong learner*. Setiap model meningkatkan perhatian (*weight*) pada kesalahan yang tidak ditangani dengan baik oleh model sebelumnya dalam urutan tersebut. Boosting, seperti bagging, dapat digunakan untuk masalah regresi dan klasifikasi. Boosting memberikan kemudahan interpretasi model dan membantu mengurangi varians dan bias dalam *ensemble machine learning* [23].

Fungsi dari metode boosting adalah sebagai berikut (1):

$$f(x) = \sum \alpha h(x) \quad (1)$$

Dimana  $f(x)$  dihasilkan dari beberapa weak learner  $h(x)$ . Hal ini dilakukan dengan membangun model dari data pelatihan, kemudian membuat model kedua yang mencoba memperbaiki kesalahan dari model pertama  $\alpha(x)$  [23]. Ada beberapa jenis boosting, yaitu AdaBoost, GradientBoost, XGBoost, LightGBM, dan CatBoost. Gambar 2.5 menunjukkan kerangka kerja boosting.



**Gambar 2.5** Kerangka kerja boosting [11]

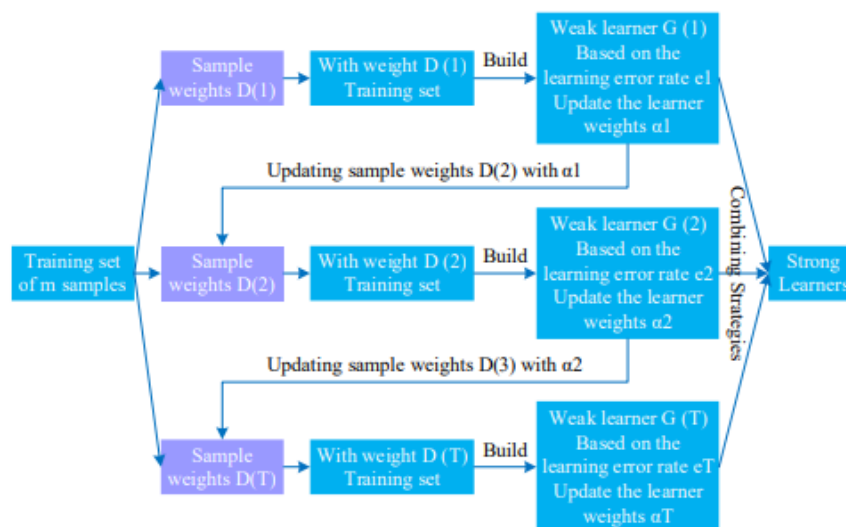
#### 1. AdaBoost (*Adaptive Boosting*)

Algoritma AdaBoost adalah metode peningkatan adaptif yang bekerja dengan cara meningkatkan performa pengklasifikasi secara bertahap selama proses

pelatihan hingga mencapai ambang batas atau tingkat kesalahan yang telah ditentukan sebelumnya [24]. Proses algoritma ini terdiri dari tiga langkah utama:

- a. **Pemberian Bobot Awal:** Pada tahap awal, semua sampel diberi bobot yang sama. Jika terdapat  $N$  sampel, maka setiap sampel memiliki bobot awal sebesar  $1/N$ .
- b. **Pelatihan Pengklasifikasi:** Pada langkah ini, pengklasifikasi dilatih untuk mengevaluasi data sampel. Pengklasifikasi pertama menentukan apakah suatu sampel diklasifikasikan dengan benar atau salah. Hasil klasifikasi ini memengaruhi bobot sampel pada iterasi berikutnya. Jika sampel diklasifikasikan dengan benar, bobotnya akan dikurangi, sedangkan bobot sampel yang salah diklasifikasikan akan meningkat. Penyesuaian bobot ini bertujuan agar pengklasifikasi selanjutnya lebih fokus pada sampel yang sulit diklasifikasikan, sehingga meningkatkan akurasi keseluruhan.
- c. **Kombinasi Pengklasifikasi:** Pada tahap akhir, beberapa pengklasifikasi digabungkan untuk membentuk pengklasifikasi yang lebih kuat. Setiap pengklasifikasi diberi bobot sesuai tingkat akurasi. Pengklasifikasi dengan akurasi tinggi diberi bobot lebih besar, sementara pengklasifikasi dengan akurasi rendah mendapatkan bobot lebih kecil.

Pendekatan ini memastikan pengklasifikasi lemah dapat bekerja bersama untuk menciptakan model yang lebih kuat dan akurat. Gambar 2.6 mengilustrasikan cara kerja AdaBoost.

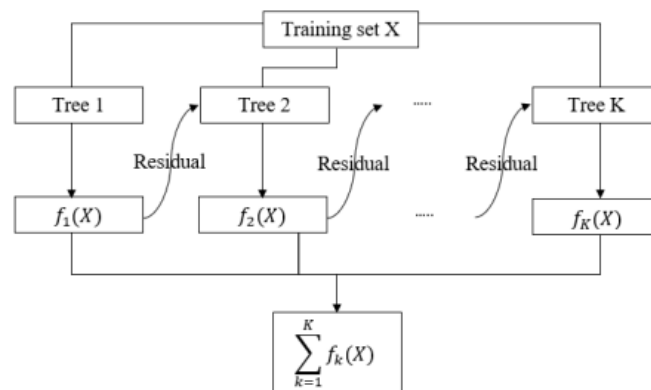


**Gambar 2.6** Proses algoritma AdaBoost [24]

## 2. GradientBoost (*Gradient Boosting*)

*Gradient boosting* adalah metode pembelajaran mesin yang digunakan untuk menyelesaikan masalah regresi dan klasifikasi. Seperti random forest, teknik ini menghasilkan prediksi berdasarkan output dari beberapa pohon keputusan. Namun, *gradient boosting* memanfaatkan pohon keputusan sebagai pembelajar lemah dan menggunakan konsep residual untuk memperbaiki prediksi. Residual dihitung dengan mengukur selisih antara prediksi saat ini dan nilai target yang diketahui [25].

Algoritma kemudian memetakan fitur lemah ke residual, secara bertahap memperbaiki model menuju nilai target melalui beberapa iterasi. Berbeda dengan random forest, gradient boosting membangun pohon keputusan secara berurutan, dengan setiap pohon baru dirancang untuk mengoreksi kesalahan dari pohon sebelumnya. Meskipun proses ini lebih memakan waktu secara komputasi, hasil yang diperoleh cenderung lebih akurat [25]. Gambar 2.7 mengilustrasikan cara kerja GradientBoost.



**Gambar 2.7** Proses algoritma GradientBoost [26]

## 3. XGBoost (*Extreme Gradient Boosting*)

XGBoost memperkenalkan tujuan pembelajaran yang teratur untuk meningkatkan kinerja model. Selain menghitung kesalahan berdasarkan perbedaan antara nilai yang diprediksi dan nilai yang sebenarnya, XGBoost juga menambahkan dua faktor regularisasi. Yang pertama berkaitan dengan jumlah cabang dalam setiap pohon keputusan, dan yang kedua berhubungan dengan bobot pada setiap cabang pohon [27].

Dengan menambahkan faktor-faktor ini, XGBoost dapat menghasilkan model yang lebih sederhana dan stabil, yang lebih baik dalam menggeneralisasi

data baru. Selain itu, XGBoost menggunakan metode optimasi yang lebih cepat, yaitu Newton descent, untuk mempercepat pelatihan pohon-pohon keputusan. Terakhir, XGBoost juga memperkenalkan algoritma baru untuk mempercepat proses pencarian pemisahan fitur saat melatih model [27].

#### 4. **LightGBM (*Light Gradient Boosting*)**

LightGBM memperkenalkan tiga strategi baru untuk membuat pelatihan menjadi lebih efisien, yaitu *a histogram-based split finding method*, *Exclusive Feature Bundling* (EFB) and *Gradient-based One-Side Sampling* (GOSS). EFB menggunakan metode untuk mengelompokkan fitur-fitur yang tidak saling berkaitan dan menggabungkannya, sehingga mengurangi jumlah fitur dalam dataset. Sementara itu, GOSS memilih sampel dataset yang paling penting berdasarkan gradien, tanpa mengubah distribusi keseluruhan data pelatihan. Kedua algoritma ini menyederhanakan proses pelatihan model dengan cara berbeda, mempercepat waktu pelatihan tanpa mengorbankan akurasi secara signifikan [28].

Selain itu, LightGBM menggunakan strategi pertumbuhan pohon yang berbeda dibandingkan XGBoost. Biasanya, pohon keputusan dibuat menggunakan pendekatan "*breadth-first*", di mana semua cabang pada tingkat yang sama dibangun secara bersamaan sebelum memperdalam cabang lebih lanjut, menghasilkan pohon dengan kedalaman yang seragam. Sebaliknya, LightGBM menggunakan pendekatan "*depth-first*", di mana algoritma membagi node berdasarkan peningkatan kinerja terbesar. Hal ini menghasilkan pohon yang tidak simetris, dengan beberapa cabang yang sangat dalam sementara yang lain tetap dangkal. Pendekatan ini memungkinkan model belajar lebih cepat tetapi berisiko terlalu menyesuaikan diri dengan dataset kecil [28].

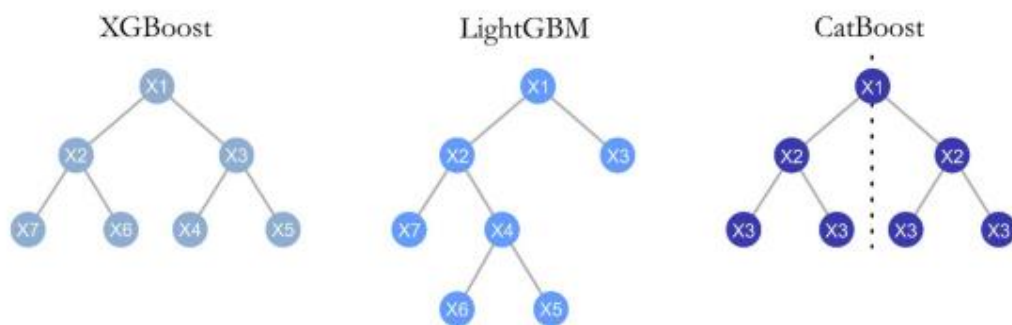
#### 5. **CatBoost (*Categorical Boosting*)**

CatBoost memiliki tiga fitur utama yang membedakannya dari LightGBM dan XGBoost [29].

- a. **Pengelolaan Variabel Kategorikal:** CatBoost menggunakan algoritma *Target Statistics* (TS) untuk mengolah variabel kategorikal. Pendekatan ini membantu meningkatkan kinerja pada data baru dengan mencegah kebocoran target selama pelatihan.

- b. *Ordered Boosting*: CatBoost memperkenalkan ordered boosting, variasi gradient boosting yang melatih setiap model pada bagian dataset pelatihan yang berbeda. Hal ini mengatasi masalah bias prediksi yang muncul saat pohon dilatih menggunakan gradien dari sampel yang sama. Pendekatan ini mengurangi risiko overfitting, terutama untuk dataset kecil.
- c. *Pohon Keputusan Simetris*: CatBoost menggunakan pohon keputusan simetris, di mana variabel dan ambang batas yang sama digunakan untuk pemisahan pada tingkat kedalaman tertentu. Simetri ini bertindak sebagai regularisasi, mencegah model menjadi terlalu kompleks. Selain itu, pendekatan ini memungkinkan estimasi ketidakpastian dalam prediksi, mirip dengan yang dilakukan model Proses Gaussian [29].

Gambar 2.8 menunjukkan perbedaan struktur *decision tree* yang diterapkan pada XGBoost, LightGBM, dan CatBoost.



**Gambar 2.8** Struktur *decision tree* pada XGBoost, LightGBM, dan CatBoost [30]

### 2.3.1.2 Evaluasi

Evaluasi sangat penting untuk menentukan keefektifan suatu model. Ada beberapa kriteria untuk mengevaluasi hal tersebut, yaitu

#### a. *Confusion Matrix*

*Confusion matrix* adalah tabel yang menunjukkan seberapa baik kinerja model klasifikasi dengan membandingkan nilai prediksi dengan nilai aktual untuk suatu dataset [31]. *Confusion matrix* memvisualisasikan kinerja model seperti di Gambar 2.9.



		Condition	
		Condition positive	Condition negative
Test Outcome	Test Positive	Ture Positive (TP)	False Negative (FN)
	Test negative	False Positive (FP)	Ture Negative (TN)

**Gambar 2.9** Pemetaan *confusion matrix* [32]

1. True Negative (TN): Model memprediksi data ada di kelas Negatif dan yang sebenarnya data memang ada di kelas Negatif.
  2. True Postive (TP): Model memprediksi data ada di kelas Positif dan yang sebenarnya data memang ada di kelas Positif.
  3. False Negative (FN): Model memprediksi data ada di kelas Negatif, namun yang sebenarnya data ada di kelas Positif.
  4. False Positive (FP): Model memprediksi data ada di kelas Positif, namun yang sebenarnya data ada di kelas Negatif.
- b. Akurasi

Akurasi menghitung ratio dari prediksi yang benar dari total jumlah sampel yang dievaluasi [33]. Akurasi memiliki rentang nilai 0 hingga 1, semakin tinggi nilainya maka kinerja model dalam mengklasifikasi semakin baik. Akurasi memiliki rumus (2):

$$Accuracy = \frac{tp+tn}{tp+fp+tn+fn} \times 100\% \quad (2)$$

c. Recall

*Recall* yang bisa disebut juga dengan *sensitivity*, digunakan untuk menghitung persentase kelas positif yang diklasifikasikan dengan benar [33]. *Recall* memiliki rentang nilai 0 hingga 1, semakin tinggi nilainya maka kinerja model dalam mengklasifikasi semakin baik. *Recall* memiliki rumus (3):

$$Recall = \frac{tp}{tp+fn} \times 100\% \quad (3)$$

d. *Precision*

*Precision* menghitung berapa banyak data yang diklasifikasikan sebagai kelas positif dengan benar dari total kelas positif yang diprediksi [31]. *Precision* memiliki rentang nilai 0 hingga 1, semakin tinggi nilainya maka kinerja model dalam mengklasifikasi semakin baik. *Precision* memiliki rumus (4):

$$Precision = \frac{tp}{tp+fp} \times 100\% \quad (4)$$

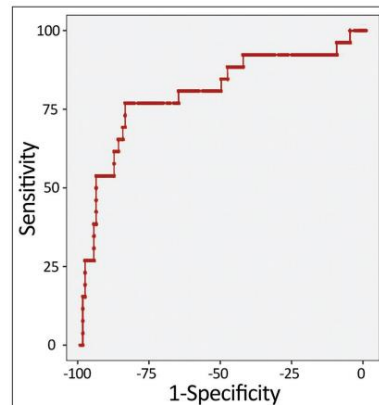
e. F1 Score

F1 Score merepresentasikan rata rata harmonik antara nilai recall dan precision [31]. *Precision* yang rendah disebabkan oleh banyaknya data yang diklasifikasikan ke label *false positive*, sedangkan *recall* yang rendah disebabkan oleh banyaknya data yang diklasifikasikan ke dalam label *false negative*. Oleh karena itu dibutuhkan F1 score untuk mempertimbangkan recall dan precision secara bersamaan. Berikut adalah rumus dari F1 score (5):

$$F1\ score = 2 \times \frac{precision \times recall}{precision+recall} \times 100\% \quad (5)$$

f. *Area Under Curve* (AUC)

AUC mengevaluasi kinerja model secara keseluruhan, yang dapat dihitung dari area di bawah kurva *receiver operating characteristics* (ROC), yang menunjukkan hubungan antara tingkat positif yang benar dan tingkat positif yang salah. AUC memiliki rentang nilai 0.5 hingga 1, semakin tinggi nilainya maka kinerja model dalam mengklasifikasi semakin baik [34]. Gambar 2.10 mengilustrasikan grafik ROC.



**Gambar 2.10** Grafik ROC

g. *Computational complexity*

*Computational complexity* yang bisa disebut juga dengan *computational cost*, mengacu pada tingkat kesulitan komputasi suatu fungsi matematika. *Computational cost* dibagi menjadi dua, yaitu kompleksitas waktu, yang menunjukkan berapa lama waktu yang dibutuhkan untuk menjalankan serangkaian operasi, dan kompleksitas ruang, yang menunjukkan seberapa banyak memori yang diperlukan untuk operasi tersebut [35].

## 2.4 Preliminary Research

Pada tabel 2.2 menunjukkan beberapa penelitian sebelumnya yang telah dilakukan dengan dataset yang sama. Hasil dari beberapa penelitian ini akan menjadi pembandingan dengan penelitian ini.

**Tabel 2.2** Penelitian sebelumnya

Metode	Metrik	N	S	V	F	Q
XGBoost [36]	Akurasi	98.87%	99.37%	99.44%	99.75%	99.99%
	Recall	99.73%	80.40%	96.06%	72.50%	0
Dual fully connected neural network [37]	Akurasi	93.4%	93.4%	93.4%	93.4%	-
	Recall	95.1%	90.3%	84.1%	0.25%	-
	Precision	98.3%	43.5%	89.5%	23.1%	-
	F1 score	0.97	0.59	0.87	-	-
AdaBoost [38]	Akurasi	94.15%	94.15%	94.15%	94.15%	-
	Recall	99.62%	1.91%	82.92%	0	-
	Precision	94.31%	40.22%	95.46%	0	-

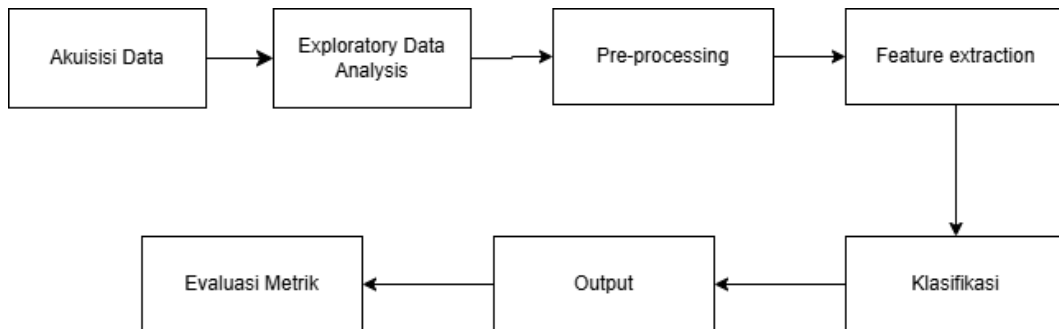
## BAB III

### PERANCANGAN SISTEM

#### 3.1 Desain Sistem Klasifikasi

Desain sistem merupakan langkah penting yang bertujuan untuk menentukan struktur, alur kerja, dan interaksi antara proses yang terlibat dalam algoritma pengklasifikasian data.

##### 3.1.1 Diagram Blok Umum Sistem Klasifikasi



**Gambar 3.1** Diagram blok sistem secara umum

Gambar 3.1 menyajikan representasi diagram blok dari sistem yang akan diterapkan dalam studi ini, memberikan gambaran struktur dan alur kerja yang mendasari implementasi algoritma. Sistem dimulai dari akuisisi data hingga evaluasi metrik untuk setiap keluarannya. Dataset ini berasal dari database aritmia MIT-BIH yang tersedia di Physionet yang kemudian diakses melalui Kaggle untuk mendapatkan dataset yang telah dilabel sesuai standar AAMI. Database ini mencakup sekitar 109.446 label denyut. Setelah data diakuisisi, data dianalisis menggunakan proses *Exploratory Data Analysis*. *Exploratory Data Analysis* (EDA) adalah proses awal untuk mengeksplorasi data dengan tujuan mengidentifikasi pola, menemukan anomali, menguji hipotesis, dan memeriksa asumsi yang ada. Dengan melakukan EDA, kesalahan dalam data seperti outlier, nilai yang hilang, duplikasi, kesalahan pengkodean, dan hubungan antar data dapat terdeteksi sejak awal, memungkinkan untuk melihat data secara menyeluruh sebelum membuat asumsi dan mengidentifikasi kesalahan.

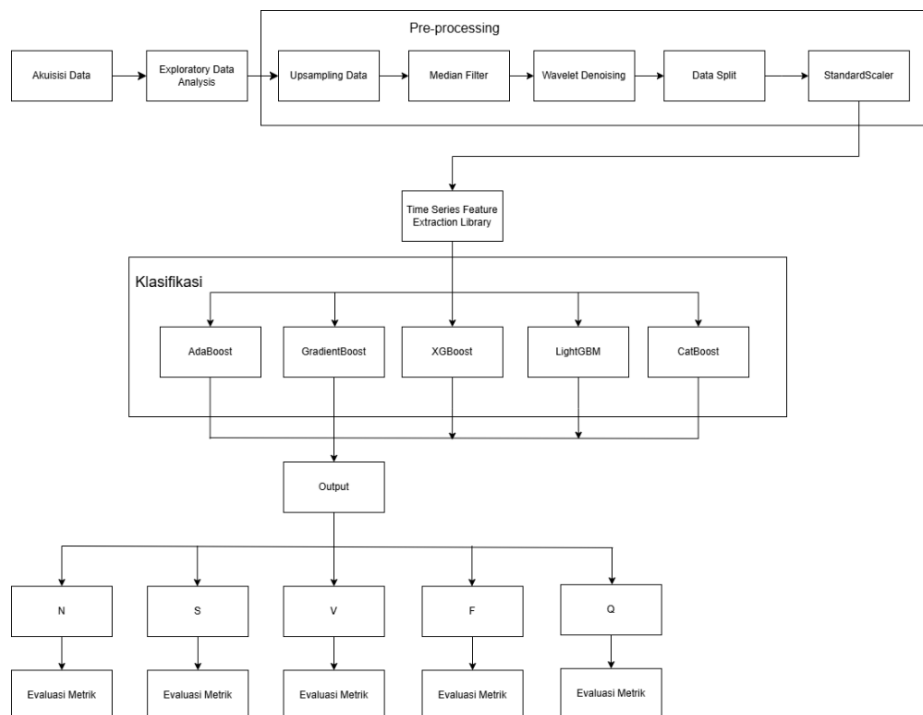
*Pre-processing* data perekaman EKG bertujuan untuk menyeimbangkan data lalu untuk menyaring dan mengurangi noise pada sinyal, sehingga menjadi optimal untuk mengenali kelas dalam pendekatan pembelajaran dalam proses selanjutnya.

Setelah itu, data dibagi menjadi data pelatihan dan data validasi. Data pelatihan distandarisasi dengan tujuan untuk mengubah nilai fitur menjadi distribusi dengan rata-rata 0 dan deviasi standar 1. Untuk memahami ciri dari suatu label maka dilakukan proses ekstraksi ciri. Kemudian data tersebut dimasukkan ke berbagai algoritma yang akan diujikan. Setiap keluaran dari algoritma tersebut akan dievaluasi untuk melihat kinerja model dalam mengklasifikasi setiap kelas.

Proses eksekusi algoritma dilakukan menggunakan platform Google Colab, yang mendukung bahasa pemrograman Python 3 dan memanfaatkan akselerator perangkat keras berupa GPU T4 untuk meningkatkan kinerja komputasi. Perangkat utama yang digunakan dalam pengujian algoritma adalah laptop Acer Aspire A514-54, dilengkapi prosesor Intel Core i3-1115G4 dengan kecepatan 3.00 GHz dan konfigurasi 4 CPU (*Central Processor Unit*) dilengkapi dengan RAM 8GB, yang menyediakan basis komputasi tambahan untuk pengolahan data.

### 3.2 Desain Algoritma Sistem Klasifikasi

Sebelum membuat algoritma sistem, diagram blok umum sistem diklasifikasi diperjelas menjadi lebih detail. Gambar 3.2 menyajikan sebuah diagram blok sistem yang dirancang dengan lebih detail, menggambarkan hubungan atau alur interaksi antar komponen dalam sistem klasifikasi.



**Gambar 3.2** Diagram blok sistem secara detail

### 3.2.1 Akuisisi Data

Dataset ini berasal dari database aritmia MIT-BIH yang tersedia di Physionet. Database ini menggunakan lebih dari 4000 rekaman Holter jangka panjang yang dikumpulkan antara tahun 1975 dan 1979 di Laboratorium Aritmia Rumah Sakit Beth Israel [39].

#### a. Kriteria pemilihan:

Sebanyak 60% rekaman diambil dari pasien rawat inap, sementara 40% sisanya dari pasien rawat jalan.

Database ini terdiri dari dua kelompok rekaman:

1. Kelompok pertama (rekaman 100-124): Dipilih secara acak untuk memberikan gambaran representatif tentang berbagai jenis gelombang dan artefak yang mungkin ditemukan oleh detektor aritmia dalam praktik klinis sehari-hari.
2. Kelompok kedua (rekaman 200-234): Dipilih untuk mencakup aritmia kompleks, seperti ventrikel, junctional, dan supraventrikular, serta gangguan konduksi. Beberapa rekaman di kelompok ini menonjol karena karakteristik ritme, variasi bentuk gelombang QRS, atau kualitas sinyalnya.

Secara keseluruhan, database ini mencakup 48 rekaman dengan durasi masing-masing lebih dari 30 menit. Subjek penelitian melibatkan 25 pria (usia 32-89 tahun) dan 22 wanita (usia 23-89 tahun). Rekaman 201 dan 202 diambil dari subjek pria yang sama [39].

#### b. Konfigurasi EKG

Sebagian besar rekaman menggunakan sinyal atas berupa lead tungkai yang dimodifikasi II (MLII), dihasilkan dengan menempatkan elektroda di dada. Sinyal bawah biasanya berupa lead V1 yang dimodifikasi, meskipun kadang menggunakan V2 atau V5, dan dalam satu kasus V4. Elektroda untuk sinyal bawah juga ditempatkan di dada. Konfigurasi ini merupakan metode standar yang digunakan di Laboratorium Aritmia BIH [40].

Sebaliknya, detak ektopik lebih mudah dikenali pada sinyal bawah, seperti pada rekaman 106. Salah satu pengecualian adalah rekaman 114, di mana sinyalnya terbalik. Hal ini mencerminkan situasi yang kadang terjadi dalam praktik klinis, sehingga detektor aritmia harus mampu mengenali kondisi ini. Pada rekaman 102

dan 104, lead II yang dimodifikasi tidak dapat digunakan karena pasien memiliki pembalutan bedah. Sebagai gantinya, lead V5 yang dimodifikasi digunakan untuk sinyal atas pada kedua rekaman tersebut [39].

c. Rekaman analog dan pemutaran ulang

Selama proses digitalisasi, rekaman analog diputar ulang menggunakan unit Del Mar Avionics model 660. Beberapa rekaman (112, 115–124, 205, 220, 223, dan 230–234) didigitalkan pada dua kali kecepatan normal, sedangkan sisanya diputar ulang pada kecepatan normal dengan perangkat tambahan khusus untuk unit tersebut. Pada beberapa perekam analog, ditemukan pergeseran waktu antar sinyal sekitar 40 milidetik. Pergeseran ini disebabkan oleh perbedaan pada posisi kepala pita dan goyangan vertikal pita selama proses perekaman dan pemutaran ulang. Masalah ini dapat menyebabkan kesulitan dalam analisis dua saluran yang dirancang untuk aplikasi *real-time* [39].

Kecepatan rekaman tidak berdampak signifikan pada detektor aritmia. Beberapa artefak frekuensi telah diidentifikasi, terutama akibat komponen mekanis dari perekam dan unit pemutaran [39].

Artefak frekuensi utama yang ditemukan meliputi:

- a. 0,167 Hz pada rekaman yang diputar ulang dengan kecepatan normal.
- b. 0,090 Hz, diikuti oleh artefak 0,083 Hz pada rekaman yang diputar ulang dengan dua kali kecepatan normal.
- c. Artefak lainnya (0,042 Hz, 0,42 Hz, 1,96 Hz, 9,1 Hz, dan 42 Hz) memiliki pengaruh lebih kecil. Artefak tambahan muncul pada rentang 0,10–0,18 Hz dan 0,30–0,54 Hz, terutama karena gulungan suplai.

Empat rekaman (102, 104, 107, dan 217) termasuk *paced beats*. Rekaman analog asli tidak merekam artefak alat pacu jantung dengan cukup detail untuk dikenali hanya dari amplitudo atau durasi denyut, tetapi data yang didigitalkan cukup akurat untuk menggunakan detektor khusus alat pacu jantung dalam analisis [39].

d. Digitalisasi

Selama digitalisasi, sinyal analog difilter untuk menghindari saturasi konverter analog-ke-digital (ADC) dan aliasing, dengan bandpass 0,1–100 Hz. Sinyal yang telah difilter didigitalkan pada 360 Hz per saluran menggunakan perangkat keras

yang dikembangkan oleh MIT dan Laboratorium Teknik Biomedis BIH. Frekuensi sampling 360 Hz dipilih untuk mempermudah implementasi filter digital 60 Hz (frekuensi utama) pada detektor aritmia. Sebagian besar noise 60 Hz muncul selama pemutaran ulang, dan pada rekaman dua kali kecepatan, noise ini bergeser menjadi 30 Hz (atau kelipatannya) [39].

Sampel dari masing-masing sinyal diambil hampir bersamaan, dengan perbedaan waktu antar sinyal hanya beberapa mikrodetik. Namun, pergeseran pita analog jauh lebih besar. ADC unipolar memiliki resolusi 11-bit dengan rentang  $\pm 5$  mV, menghasilkan nilai sampel dari 0 hingga 2047 (nilai 1024 mewakili nol volt). Untuk menghemat ruang penyimpanan, sampel 11-bit awalnya disimpan dalam format 8-bit dengan pengkodean perbedaan pertama. Dengan resolusi ADC dan frekuensi sampling ini, pengkodean memungkinkan perubahan maksimum yang dapat direkam sebesar  $\pm 225$  mV/s [39].

e. Anotasi

Label denyut awal dihasilkan menggunakan detektor QRS yang menandai setiap peristiwa sebagai denyut normal. Grafik sepanjang 150 kaki dicetak untuk setiap rekaman berdurasi 30 menit, lalu diperiksa secara independen oleh dua ahli jantung. Label tambahan ditambahkan untuk denyut yang terlewat, deteksi yang salah dihapus, dan label untuk denyut tidak normal disesuaikan. Anotasi tambahan seperti label ritme, kualitas sinyal, dan komentar juga disertakan [39].

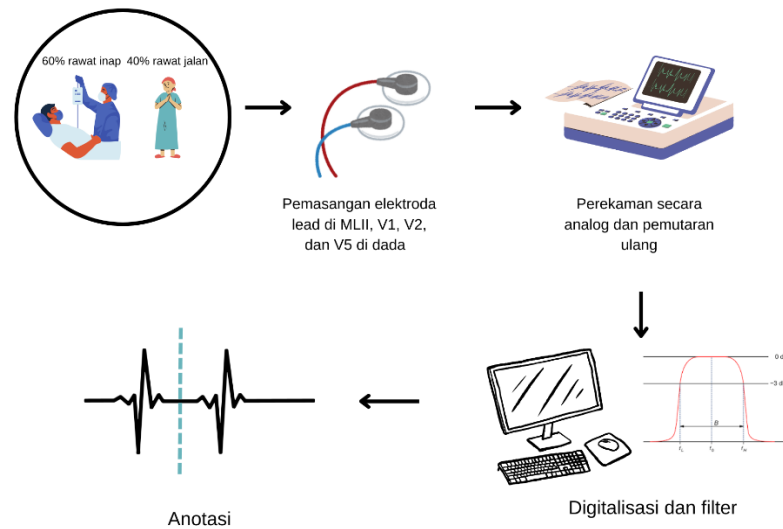
Setelah proses anotasi selesai, data dibandingkan secara otomatis untuk menemukan perbedaan yang kemudian diselesaikan melalui konsensus. Tahapan ini mencakup analisis konsistensi, pencarian interval R-R ekstrem, dan koreksi jika diperlukan. Semua label denyut ditempatkan sedekat mungkin pada puncak gelombang-R setelah melalui penyaringan digital. Ilustrasi akuisis data ditunjukkan pada Gambar 3.3. Database ini mencakup sekitar 109.446 label denyut, dengan berbagai koreksi dilakukan sejak dirilis pada 1980 [39]. Dataset yang diperoleh sudah dalam format CSV. Dalam format tersebut, 109.446 label dipetakan dan terbagi menjadi 5 kelas sesuai dengan standar AAMI seperti di Tabel 3.1.



**Tabel 3.1** Pemetaan label MIT-BIH [16]

Heart rhythm label	Content description	Include subcategories	Subclass label description
N (82.8%)	Normal heart rhythm	L	Left bundle branch block
		N	Normal ECG beat
		R	Right bundle branch block
		e	Atrial escape
		j	Borderline escape
S (7.3%)	Supraventricular abnormalities	A	Atrial premature beat
		J	Borderline premature beat
		S	Supraventricular premature beat
		a	Abnormal atrial premature beat
V (6.6%)	Ventricular escape	E	Ventricular escape beat
		V	Ventricular premature beats
F (2.5%)	Fusion heartbeat	F	Ventricular fusion heartbeat
Q (0.7%)	Unknown ECG Beat	P	Pacing heartbeat
		U	Uncategorized ECG beat
		f	Pacing and normal fusion heartbeat

Berikut adalah ringkasan akuisisi dataset MIT-BIH yang diilustrasikan dalam bentuk gambar. Ilustrasi tersebut ditunjukkan pada Gambar 3.3.



**Gambar 3.3** Ilustrasi akuisisi data

### 3.2.2 Exploratory Data Analysis (EDA)

Teknik yang digunakan dalam EDA adalah statistik deskriptif, yang merangkum data untuk memberikan gambaran umum tanpa menarik kesimpulan, seperti rata-rata, median, modus, standar deviasi, dan variansi. Hasil dari statistik deskriptif ini ditampilkan dalam bentuk tabel, diagram, grafik, dan lain-lain.

### 3.2.3 Pre-Processing

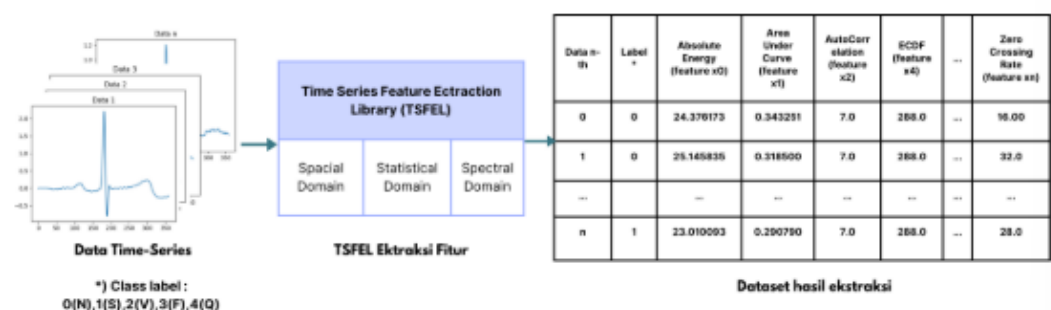
Sebelum melakukan *pre-processing*, jumlah data dengan label N diturunkan dan label lainnya dinaikkan dengan teknik *random upsampling* dan *random downsampling* agar data seimbang supaya proses *pre-processing* dapat diterapkan secara merata. Pemfilteran dilakukan dengan menerapkan filter median dengan lebar sinyal 200ms dan 600ms, kemudian mengurangi sinyal asli dan *baseline* untuk menghasilkan sinyal yang telah dikoreksi. Selanjutnya, proses denoising dilakukan pada sinyal ini dengan menggunakan db4 *Discrete Wavelet Transform* (DWT), dengan menerapkan high-pass filter pada koefisien DWT.

Setelah itu, data dibagi menjadi data train dan data validasi dengan rasio 80:20 untuk memastikan bahwa model yang dibangun dapat diterapkan dengan baik pada data baru. Data pelatihan membantu model memahami pola atau hubungan dalam data, dan data validasi menguji kinerja model selama proses pelatihan untuk

mengetahui sejauh mana model dapat menggeneralisasi dan mencegah overfitting (terlalu menyesuaikan diri dengan data pelatihan). Pembagian ini memungkinkan untuk mengetahui apakah model hanya mengingat data pelatihan atau benar-benar memahami pola saat ini untuk diterapkan pada data baru. Data pelatihan dan data validasi distandarisasi dengan StandardScaler untuk memastikan bahwa fitur-fitur dalam dataset memiliki skala yang sama.

### 3.2.4 Time Series Feature Extraction Library (TSFEL)

Setelah menerapkan StandardScaler, data *time series* akan distandarisasi, tetapi struktur yang mendasari data akan tetap sama. TSFEL masih dapat mengekstrak fitur-fitur yang berguna dari data pelatihan yang telah diskalakan. TSFEL dapat menangani data *time series* multidimensi, dengan fitur-fitur yang tersedia dibagi ke dalam tiga domain: statistik, temporal, dan spektral. Jumlah kategori mencakup lebih dari 60 fitur, dan dari setiap titik data dapat diekstrak 314 fitur berdasarkan parameter *default* yang digunakan untuk sinyal EKG dalam studi ini. Ilustrasi proses ekstraksi ciri menggunakan TSFEL ditunjukkan pada Gambar 3.4. Daftar fitur yang diperoleh setelah ekstraksi menggunakan TSFEL disajikan pada Tabel 3.2, Tabel 3.3, dan Tabel 3.4.



**Gambar 3.4** Proses ekstraksi ciri menggunakan TSFEL [33]

Tabel 3.2 Fitur spektral yang tersedia pada TSFEL [33]

Domain	Feature Types
Spectral	FFT Mean coefficient
	Fundamental Frequency
	Human range energy
	LPCC (Linear Prediction Cepstral Coefficient)
	MFCC (Mel-Frequency Cepstral Coefficient)
	Max power spectrum
	Maximum frequency
	Median frequency
	Power bandwidth
	Spectral centroid
	Spectral decrease
	Spectral distance
	Spectral entropy
	Spectral kurtosis
	Spectral positive turning points
	Spectral roll-off
	Spectral roll-on
	Spectral skewness

**Tabel 3.3** Fitur statistik yang tersedia pada TSFEL [33]

Domain	Feature Types
Statistik	Absolute energy
	Average power
	ECDF (Empirical Cumulative Distribution Function)
	ECDF Percentile
	ECDF Percentile Count
	Entropy
	Histogram
	Interquartile range
	Kurtosis
	Max
	Mean
	Mean absolute deviation
	Median
	Median absolute deviation
	Min
	Peak to peak distance
	Root mean square
	Skewness
	Standard deviation
	Variance

**Tabel 3.4** Fitur temporal pada TSFEL [33]

Domain	Feature
Temporal	Area under the curve
	Autocorrelation
	Centroid
	Mean absolute diff
	Mean diff
	Median absolute diff
	Median diff
	Negative turning points
	Neighbourhood peaks
	Positive turning points
	Signal distance
	Slope
	Sum absolute diff
	Zero crossing rate

### 3.2.5 Klasifikasi

Setelah data diekstrak, langkah selanjutnya adalah melakukan klasifikasi berdasarkan standar AAMI, yang membagi data ke dalam lima kelas, yaitu N (Normal), S (*Supraventricular*), V (*Ventricular*), F (*Fusion*), dan Q (*Unclassified*). Klasifikasi ini dilakukan dengan menggunakan salah satu metode *ensemble machine learning* yaitu boosting. Metode ini terdiri dari beberapa model dasar yang lemah (*weak learner*) yang diurutkan secara adaptif yang menghasilkan satu *strong learner*. Setiap model meningkatkan perhatian (*weight*) pada kesalahan yang tidak ditangani dengan baik oleh model sebelumnya dalam urutan tersebut. *Weak learner* yang digunakan dalam klasifikasi aritmia adalah *decision tree*.

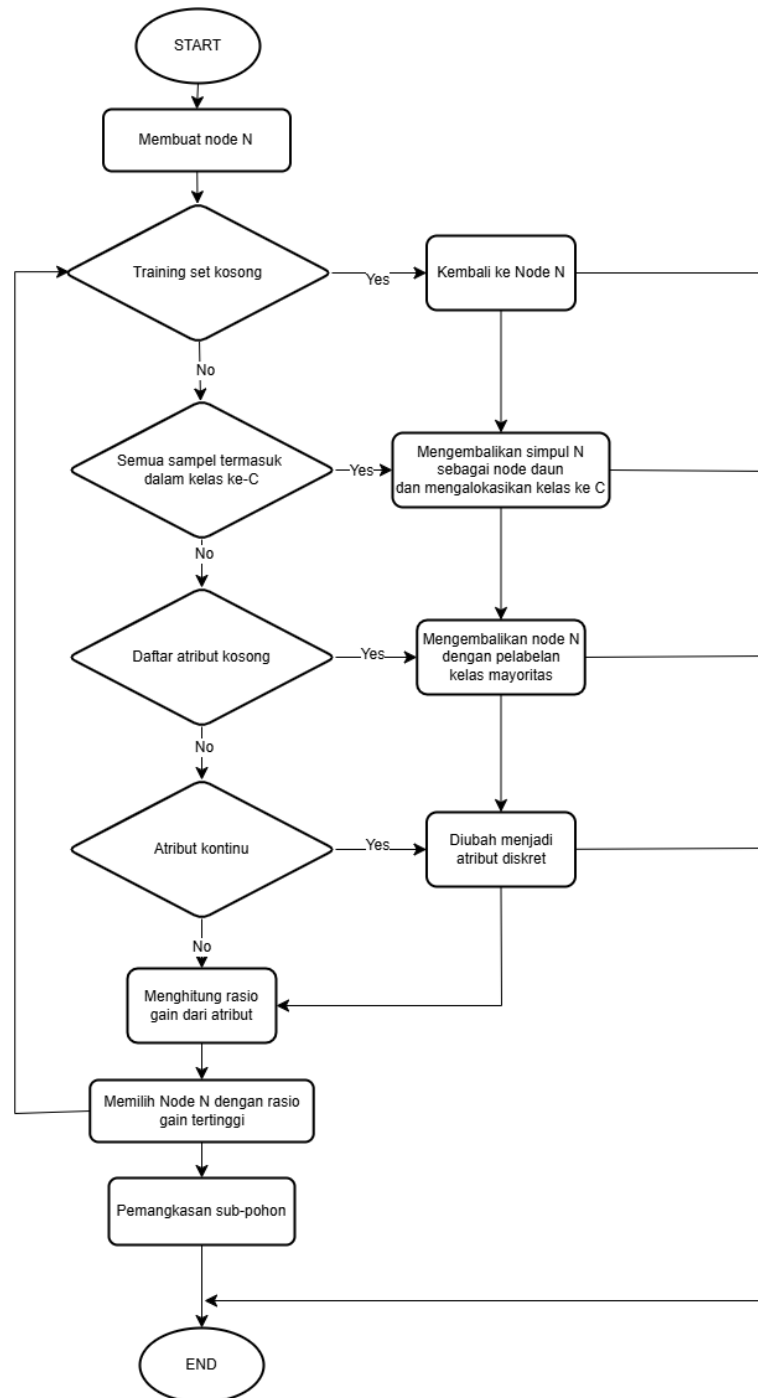
### 3.2.5.1 *Decision Tree*

*Decision tree* dirancang menggunakan algoritma C4.5 karena efektif dalam menangani data kategorikal dan kontinu. Berikut adalah cara kerja algoritma C4.5 dalam *decision tree*:

Input: Dataset (D), kedalaman maksimum (max\_depth), dan jumlah minimum sampel per node (min\_samples)

Output: Pohon keputusan akhir

- 1) Inisialisasi pohon keputusan sebagai struktur kosong:  $tree = \{ \}$
- 2) Untuk setiap node dalam pohon:
  - a) Pilih atribut untuk membagi berdasarkan rasio gain tertinggi.  
 Jika atribut bersifat kontinu:
    - i) Urutkan nilai atribut.
    - ii) Evaluasi semua kemungkinan titik pemisahan.
    - iii) Pilih titik pemisahan terbaik.
  - b) Bagi dataset berdasarkan atribut atau *threshold* yang dipilih.
    - i) Hitung subset data untuk setiap cabang.
    - ii) Simpan kriteria pembagian dalam pohon.
- 3) Bangun pohon secara rekursif:
  - a) Untuk setiap subset data:
    - i) Jika kriteria penghentian terpenuhi (misalnya, kedalaman maksimum atau jumlah sampel minimum per node):
      - Tetapkan label daun.
    - ii) Jika tidak:
      - Ulangi proses pembagian.
- 4) Pangkas pohon untuk mengoptimalkan performa:
  - a) Evaluasi pohon menggunakan *Bayesian Error Pruning* (BEP).
    - i) Untuk setiap subtree dalam pohon:
      - Hitung tingkat kesalahan sebelum dan sesudah pemangkasan.
      - Jika pemangkasan mengurangi tingkat kesalahan:
        - Hapus subtree.
- 5) Kembalikan pohon keputusan akhir.



**Gambar 3.5** Diagram alir *decision tree*

### 3.2.5.2 Boosting

Dalam studi ini, digunakan kerangka kerja boosting yang mencakup beberapa algoritma, yaitu AdaBoost, GradientBoost, XGBoost, LightGBM, dan CatBoost. Metode-metode ini bekerja dengan cara mengoptimalkan model secara iteratif, memperbaiki kesalahan yang dilakukan oleh model sebelumnya untuk meningkatkan kinerja klasifikasi secara keseluruhan.



## 1. AdaBoost

Model algoritma AdaBoost untuk mengklasifikasi aritmia berdasarkan denyutnya menggunakan 50 *decision tree* dengan kedalaman 1.

Proses pelatihan model algoritma AdaBoost ditunjukkan sebagai berikut:

Input: Dataset pelatihan MIT-BIH dengan  $n$  sampel:  $\{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$  di mana  $x_i$  adalah fitur dan  $y_i$  adalah label kelas (misal: -1 atau 1)

Output: Model klasifikasi akhir  $H(x)$

1) Inisialisasi bobot untuk setiap sampel:

Untuk  $i = 1$  hingga  $n$ :

$$W_1(i) = 1/n$$

2) Untuk  $m = 1$  hingga 50 (jumlah iterasi atau jumlah pohon keputusan):

a. Latih pohon keputusan  $G_m(x)$  dengan kedalaman maksimum 1 menggunakan bobot  $W_m$  pada dataset.

b. Hitung error  $G_m$ :

$$\text{error}_m = \sum (W_m(i) * I(y_i \neq G_m(x_i)))$$

di mana  $I(\text{condition})$  adalah fungsi indikator yang bernilai 1 jika condition benar, dan 0 jika salah.

c. Hitung koefisien  $\alpha_m$ :

$$\alpha_m = 0.5 * \ln((1 - \text{error}_m) / \text{error}_m)$$

d. Perbarui bobot  $W$  untuk iterasi berikutnya:

Untuk  $i = 1$  hingga  $n$ :

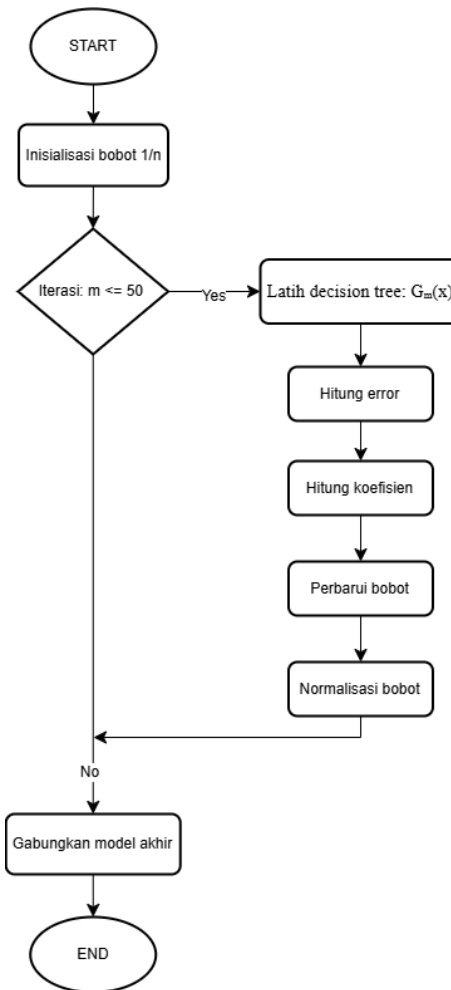
$$W_{m+1}(i) = W_m(i) * \exp(-\alpha_m * y_i * G_m(x_i))$$

Normalisasi  $W_{m+1}$  sehingga  $\sum W_{m+1}(i) = 1$

3) Gabungkan semua pohon keputusan yang telah dilatih menjadi model akhir:

$$H(x) = \text{sign}(\sum (\alpha_m * G_m(x)))$$

Diagram alir proses model pelatihan algoritma AdaBoost ditunjukkan pada Gambar 3.6.



**Gambar 3.6** Diagram alir AdaBoost

## 2. GradientBoost

Model GradientBoost untuk mengklasifikasi aritmia berdasarkan denyutnya menggunakan 50 *decision tree* dengan kedalaman 1. Proses pelatihan model GradientBoost ditunjukkan sebagai berikut:

Input: Dataset pelatihan MIT-BIH dengan  $n$  sampel:  $\{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$  di mana  $x_i$  adalah fitur dan  $y_i$  adalah label kelas (misal: -1 atau 1)

Output: Model klasifikasi akhir  $H(x)$

1) Inisialisasi model prediksi awal:

$$F_0(x) = (1/n) * \sum y_i \text{ untuk } i = 1 \text{ hingga } n$$

2) Untuk  $m = 1$  hingga 50 (jumlah iterasi atau jumlah pohon keputusan):

a. Hitung residu:

$$r_i = y_i - F_{m-1}(x_i) \text{ untuk } i = 1 \text{ hingga } n$$

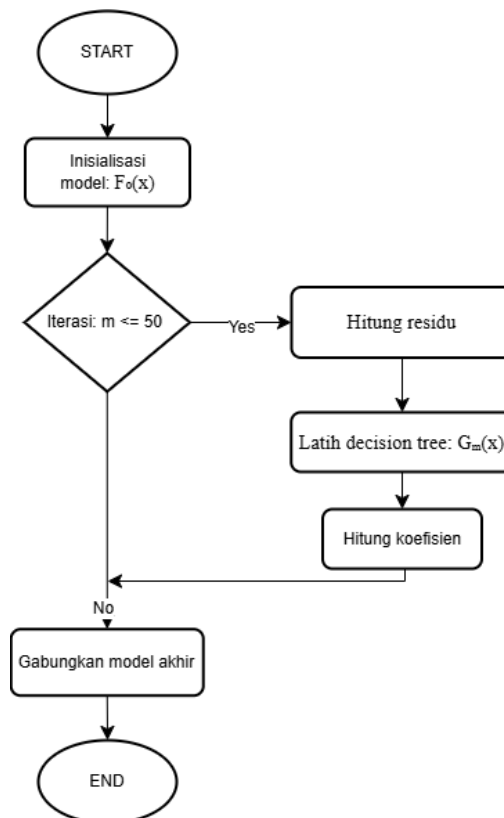
b. Latih pohon keputusan  $G_m(x)$  dengan kedalaman maksimum 1 pada dataset menggunakan  $(x_i, r_i)$  sebagai target.

c. Hitung koefisien pembaruan dengan learning rate  $\eta$  (contoh:  $\eta = 0.1$ ):

$$F_m(x) = F_{m-1}(x) + \eta * G_m(x)$$

3) Gabungkan semua pohon keputusan yang telah dilatih menjadi model akhir:  $H(x) = F_{50}(x)$

Diagram alir proses model pelatihan algoritma GradientBoost ditunjukkan pada Gambar 3.7.



**Gambar 3.7** Diagram alir GradientBoost

### 3. XGBoost

Model XGBoost untuk mengklasifikasi aritmia berdasarkan denyutnya menggunakan 50 *decision tree* dengan kedalaman 1. Proses pelatihan model XGBoost ditunjukkan sebagai berikut:

Input: Dataset pelatihan MIT-BIH dengan  $n$  sampel:  $\{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$  di mana  $x_i$  adalah fitur dan  $y_i$  adalah label kelas (misal: -1 atau 1)

Output: Model klasifikasi akhir  $H(x)$

1) Inisialisasi model prediksi awal:

$$F_0(x) = (1/n) * \sum(y_i) \text{ untuk } i = 1 \text{ hingga } n$$

2) Untuk  $m = 1$  hingga 50 (jumlah iterasi atau jumlah pohon keputusan):

a. Hitung residu:

$$r_i = y_i - F_{m-1}(x_i) \text{ untuk } i = 1 \text{ hingga } n$$

b. Latih pohon keputusan  $G_m(x)$  dengan kedalaman maksimum 1 pada dataset menggunakan  $(x_i, r_i)$  sebagai target.

c. Hitung *error* dan koefisien  $\alpha_m$  untuk pohon  $G_m(x)$ :

$$\text{error}_m = \sum(\text{loss}(r_i, G_m(x_i))) \text{ untuk } i = 1 \text{ hingga } n$$

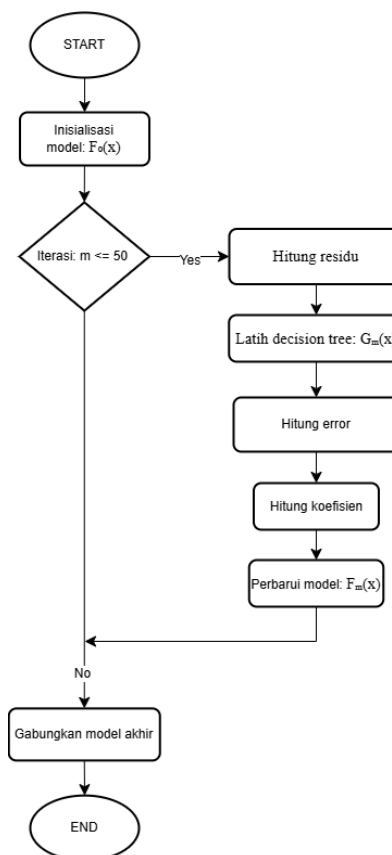
$$\alpha_m = \text{penalty term (dihitung dengan regularisasi atau optimasi)}$$

d. Perbarui model prediksi  $F_m(x)$ :

$$F_m(x) = F_{m-1}(x) + \eta * G_m(x)$$

3) Gabungkan semua pohon keputusan yang telah dilatih menjadi model akhir:  $H(x) = F_{50}(x)$

Diagram alir proses model pelatihan algoritma XGBoost ditunjukkan pada Gambar 3.8.



**Gambar 3.8** Diagram alir XGBoost

#### 4. LightGBM

Model LightGBM untuk mengklasifikasi aritmia berdasarkan denyutnya menggunakan 50 *decision tree* dengan kedalaman 1. Proses pelatihan model LightGBM ditunjukkan sebagai berikut:

Input: Dataset pelatihan MIT-BIH dengan  $n$  sampel:  $\{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$  di mana  $x_i$  adalah fitur dan  $y_i$  adalah label kelas (misal: -1 atau 1)

Output: Model klasifikasi akhir  $H(x)$

1) Inisialisasi model prediksi awal:

$F_0(x) = (1/n) * \sum(y_i)$  untuk  $i = 1$  hingga  $n$  (Model awal adalah rata-rata label kelas)

2) Untuk  $m = 1$  hingga 50 (jumlah iterasi atau jumlah pohon keputusan):

a. Hitung residu:

$$r_i = y_i - F_{m-1}(x_i) \text{ untuk } i = 1 \text{ hingga } n$$

b. Bangun histogram untuk setiap fitur:

- Bagi nilai fitur  $x_i$  ke dalam  $M$  bin (jumlah bin untuk histogram).

- Hitung distribusi residu pada setiap bin.

- Pilih bin yang memberikan pembagian terbaik untuk meminimalkan fungsi loss (misal, log loss atau squared error).

c. Latih pohon keputusan  $G_m(x)$  dengan kedalaman maksimum 1 pada dataset menggunakan histogram:

- Latih pohon untuk memprediksi residu  $r_i$ , dengan menggunakan pembagian terbaik yang dipilih dari histogram.

- Setiap pohon mempelajari split yang meminimalkan error berdasarkan histogram distribusi residual.

d. Hitung error dan koefisien  $\alpha_m$  untuk pohon  $G_m(x)$ :

$$\text{error}_m = \sum(\text{loss}(r_i, G_m(x_i))) \text{ untuk } i = 1 \text{ hingga } n$$

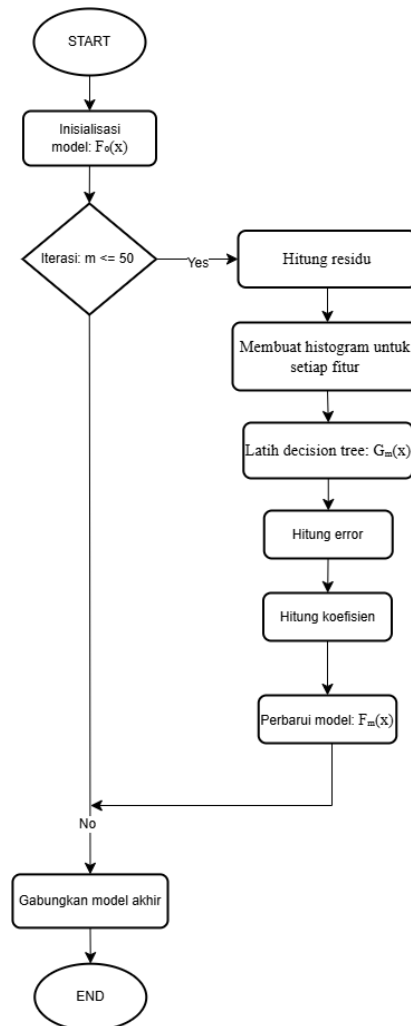
$\alpha_m = 0.5 * \ln((1 - \text{error}_m) / \text{error}_m)$  (koefisien untuk pembaruan model, sesuai dengan regulasi)

e. Perbarui model prediksi  $F_m(x)$ :

$F_m(x) = F_{m-1}(x) + \eta * G_m(x)$  (Model diperbarui dengan menambahkan hasil pohon keputusan dikalikan dengan learning rate)

3. Gabungkan semua pohon keputusan yang telah dilatih menjadi model akhir:  $H(x) = F_{50}(x)$

Diagram alir proses model pelatihan algoritma LightGBM ditunjukkan pada Gambar 3.9.



**Gambar 3.9** Diagram alir LightGBM

## 5. CatBoost

Model CatBoost untuk mengklasifikasi aritmia berdasarkan denyutnya menggunakan 50 *decision tree* dengan kedalaman 1. Proses pelatihan model CatBoost ditunjukkan sebagai berikut:

Input: Dataset pelatihan MIT-BIH dengan n sampel:  $\{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$  di mana  $x_i$  adalah fitur dan  $y_i$  adalah label kelas (misal: -1 atau 1)

Output: Model klasifikasi akhir  $H(x)$

1) Inisialisasi model prediksi awal:

$F_0(x) = (1/n) * \sum(y_i)$  untuk  $i = 1$  hingga  $n$  (Model awal adalah rata-rata dari label kelas)

2) Untuk  $m = 1$  hingga 50 (jumlah iterasi atau jumlah pohon keputusan):

a. Hitung residu:

$$r_i = y_i - F_{m-1}(x_i) \text{ untuk } i = 1 \text{ hingga } n$$

b. Proses fitur kategorikal:

- CatBoost menangani fitur kategorikal dengan teknik seperti *target encoding* atau *ordered boosting*.

- Fitur kategorikal dapat langsung dimasukkan dalam model tanpa perlu dilakukan *encoding* manual.

c. Latih pohon keputusan  $G_m(x)$  dengan kedalaman maksimum 1 pada dataset:

- Latih pohon keputusan untuk memprediksi residu  $(r_i)$ , dengan mempertimbangkan fitur kategorikal yang diproses dengan teknik yang sesuai.

- CatBoost menggunakan teknik *ordered boosting* yang menghindari overfitting pada dataset kecil dengan menggunakan metode pelatihan yang teratur.

d. Hitung error dan koefisien  $\alpha_m$  untuk pohon  $G_m(x)$ :

$$\text{error}_m = \sum(\text{loss}(r_i, G_m(x_i))) \text{ untuk } i = 1 \text{ hingga } n$$

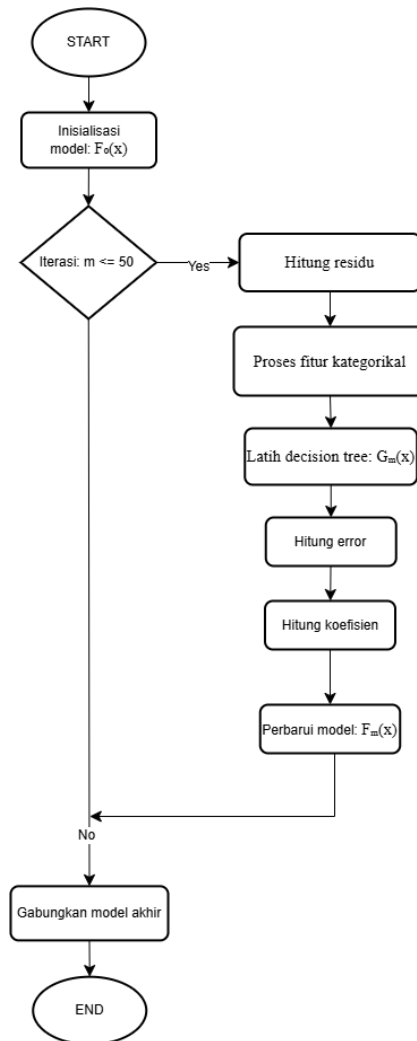
$\alpha_m$  = koefisien untuk pembaruan model, dihitung untuk mengoptimalkan loss function yang digunakan (misalnya, log loss atau squared error).

e. Perbarui model prediksi  $F_m(x)$ :

$F_m(x) = F_{m-1}(x) + \eta * G_m(x)$  (Model diperbarui dengan menambahkan hasil pohon keputusan dikalikan dengan learning rate)

3) Gabungkan semua pohon keputusan yang telah dilatih menjadi model akhir:  $H(x) = F_{50}(x)$

Diagram alir proses model pelatihan algoritma CatBoost ditunjukkan pada Gambar 3.10.



**Gambar 3.10** Diagram alir CatBoost

### 3.2.6 Evaluasi Metrik

Evaluasi metrik digunakan untuk menilai hasil keluaran model dengan membandingkannya terhadap data validasi yang telah disiapkan sebelumnya. Proses ini dilakukan untuk mengukur seberapa baik model dalam memprediksi atau mengklasifikasikan data yang belum terlihat sebelumnya. Setiap hasil keluaran model akan dievaluasi menggunakan beberapa metrik, termasuk *recall*, *precision*, *F1 score*, dan *confusion matrix*. Di sisi lain, untuk mengevaluasi kinerja algoritma yang digunakan, beberapa metrik lain seperti akurasi, AUC, dan *computational cost*.



### 3.3 Skenario pengujian algoritma klasifikasi

Studi ini menggunakan tiga skenario berbeda untuk mendapatkan pemahaman yang mendalam tentang metode boosting dalam mengklasifikasi aritmia.

#### 1. Skenario pertama

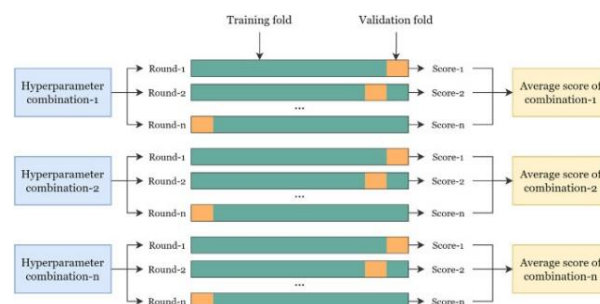
Pada skenario pertama, membandingkan performansi antar algoritma AdaBoost, GradientBoost, XGBoost, LightGBM, dan CatBoost sebelum dioptimasi. Tujuannya adalah untuk menentukan jenis algoritma dari metode boostin yang paling efektif dalam mengklasifikasi aritmia.

#### 2. Skenario kedua

Pada skenario kedua, membandingkan performansi model *decision tree* dengan AdaBoost, GradientBoost, XGBoost, LightGBM, dan CatBoost. Skenario ini menggunakan dataset dengan perlakuan yang sama serta mengklasifikasi aritmia berdasarkan standar AAMI.

#### 3. Skenario ketiga

Pada skenario ketiga, membandingkan performansi antar algoritma AdaBoost, GradientBoost, XGBoost, LightGBM, dan CatBoost setelah dioptimasi. Pada studi ini, optimasi dilakukan dengan menggunakan *tuning hyperparameter* dengan metode GridSearchCV. Metode ini dapat digunakan untuk menemukan nilai terbaik untuk parameter `n_estimator` dan `max_depth` untuk AdaBoost lalu `n_estimator`, `max_depth`, dan `learning_rate` pada GradientBoost, XGBoost, LightGBM, dan CatBoost. Gambar 3.11 menunjukkan proses GridSearchCV dan *cross validation* dalam menemukan nilai parameter yang baik.



**Gambar 3.11** Alur kerja GridSearchCV [41]

## DAFTAR PUSTAKA

- [1] T. Rusdiana, N. Aliza Putriana, I. Sopyan, D. Gozali, and P. Husni, “PEMBERIAN PEMAHAMAN MENGENAI SEDIAAN HERBAL YANG BERFUNGSI UNTUK PEMELIHARAAN KESEHATAN JANTUNG DAN GINJAL DI DESA CIBEUSI, SUMEDANG, JAWA BARAT,” *Jurnal Pengabdian kepada Masyarakat*, vol. 4, no. 6, pp. 139–141, 2019.
- [2] E. F. Elfi, E. Decroli, E. Nasrul, Y. Yanwirasti, and E. Darwin, “The Risk Factors of Coronary Heart Disease and its Relationship with Endothelial Nitric Oxide Synthase,” *Open Access Maced J Med Sci*, vol. 9, no. B, pp. 451–456, Jun. 2021, doi: 10.3889/oamjms.2021.6062.
- [3] D. Karen Widjaja and A. Arie Setiawan, “GAMBARAN GANGGUAN IRAMA JANTUNG YANG DISEBABKAN KARENA HIPERTIROID,” *Ariosta JKD*, vol. 6, no. 2, pp. 434–442, 2017.
- [4] G. Agus Andika, A. Sukohar, and A. Yonata, “Tatalaksana Aritmia: Fibrilasi Atrial.”
- [5] Kementerian Kesehatan Republik Indonesia, “JANTUNG SERING BERDEBAR (Bukan Jatuh Cinta, Waspada Aritmia Jantung).” Accessed: Oct. 13, 2024. [Online]. Available: <https://lms.kemkes.go.id/courses/9fa581e4-1239-446e-9e2d-5a7add775108>
- [6] Y. Yuniadi *et al.*, “Prevalence of atrial fibrillation based on tertiary hospital survey in Indonesia: A smartphone-based diagnosis,” *J Arrhythm*, Aug. 2024, doi: 10.1002/joa3.13137.
- [7] InaHRS, “Satu Dekade InaHRS: An Overview and Outlook.” Accessed: Oct. 01, 2024. [Online]. Available: <https://inahrs.or.id/news/satu-dekade-inahrs-an-overview-and-outlook>
- [8] D. S. Desai and S. Hajouli, “Arrhythmias,” Jan. 2024, Accessed: Oct. 14, 2024. [Online]. Available: <https://www.ncbi.nlm.nih.gov/books/NBK558923/>

- [9] Y. Sattar and L. Chhabra, “Electrocardiogram,” Jan. 2024, Accessed: Oct. 14, 2024. [Online]. Available: <https://www.ncbi.nlm.nih.gov/books/NBK549803/>
- [10] A. Hevathige and H. Asela, “Binary and Multi-Class Classification Using Supervised Machine Learning Algorithms and Ensemble Model,” 2021. [Online]. Available: <https://www.researchgate.net/publication/357117877>
- [11] L. Maretva Cendani and A. Wibowo, “Perbandingan Metode Ensemble Learning pada Klasifikasi Penyakit Diabetes,” 2022.
- [12] A. Mohammed and R. Kora, “A comprehensive review on ensemble deep learning: Opportunities and challenges,” Feb. 01, 2023, *King Saud bin Abdulaziz University*. doi: 10.1016/j.jksuci.2023.01.014.
- [13] A. Saxton, R. Chaudhry, and B. Manna, *Anatomy, Thorax, Heart Right Coronary Arteries*. 2023. Accessed: Oct. 27, 2024. [Online]. Available: <https://www.ncbi.nlm.nih.gov/books/NBK537357/>
- [14] I. Rehman and A. Rehman, *Anatomy, Thorax, Heart*. 2023. Accessed: Oct. 27, 2024. [Online]. Available: <https://www.ncbi.nlm.nih.gov/books/NBK470256/>
- [15] J. King and D. Lowery, *Physiology, Cardiac Output*. 2023. Accessed: Oct. 27, 2024. [Online]. Available: <https://www.ncbi.nlm.nih.gov/books/NBK470455/>
- [16] B. Zhang, J. Wen, and H. Ren, “A Classification Method of Arrhythmia Based on Adaboost Algorithm,” in *Journal of Physics: Conference Series*, IOP Publishing Ltd, 2020. doi: 10.1088/1742-6596/1682/1/012060.
- [17] C. Nayak, S. K. Saha, R. Kar, and D. Mandal, “An Efficient QRS Complex Detection Using Optimally Designed Digital Differentiator,” *Circuits Syst Signal Process*, vol. 38, no. 2, pp. 716–749, Feb. 2019, doi: 10.1007/s00034-018-0880-y.

- [18] Y. Li, F. Li, S. Lyu, M. Xu, and S. Wang, "Blind extraction of ECG signals based on similarity in the phase space," *Chaos Solitons Fractals*, vol. 147, Jun. 2021, doi: 10.1016/j.chaos.2021.110950.
- [19] J. Rahul, M. Sora, and L. DEV Sharma, "An overview on Biomedical Signal Analysis," 2019. [Online]. Available: <https://www.researchgate.net/publication/330779514>
- [20] Y. Xu *et al.*, "Artificial intelligence: A powerful paradigm for scientific research," Nov. 28, 2021, *Cell Press*. doi: 10.1016/j.xinn.2021.100179.
- [21] S. K. Singh, A. K. Tiwari, and H. K. Paliwal, "A state-of-the-art review on the utilization of machine learning in nanofluids, solar energy generation, and the prognosis of solar power," *Eng Anal Bound Elem*, vol. 155, pp. 62–86, 2023, doi: <https://doi.org/10.1016/j.enganabound.2023.06.003>.
- [22] Y. Zhang and C. Ling, "A strategy to apply machine learning to small datasets in materials science," *NPJ Comput Mater*, vol. 4, no. 1, Dec. 2018, doi: 10.1038/s41524-018-0081-z.
- [23] A. Mohammed and R. Kora, "A comprehensive review on ensemble deep learning: Opportunities and challenges," Feb. 01, 2023, *King Saud bin Abdulaziz University*. doi: 10.1016/j.jksuci.2023.01.014.
- [24] S. Wu and S. Meng, "Applied Mathematics and Nonlinear Sciences (aop) (aop) Applied Mathematics and Nonlinear Sciences A Modern Communication Path for Traditional Chinese Cultural Design Concepts Based on AdaBoost Model," 2023, doi: 10.2478/10.2478/amns.2023.2.00068.
- [25] A. Malik, Y. T. Javeri, M. Shah, and R. Mangrulkar, "Chapter 11 - Impact analysis of COVID-19 news headlines on global economy," in *Cyber-Physical Systems*, R. C. Poonia, B. Agarwal, S. Kumar, M. S. Khan, G. Marques, and J. Nayak, Eds., Academic Press, 2022, pp. 189–206. doi: <https://doi.org/10.1016/B978-0-12-824557-6.00001-7>.

- [26] M. Aladsani, H. Burton, S. A. Abdullah, and J. W. Wallace, "Machine Learning Driven Drift Capacity Model for Reinforced Concrete Walls," 2021. [Online]. Available: <https://www.researchgate.net/publication/356192786>
- [27] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Association for Computing Machinery, Aug. 2016, pp. 785–794. doi: 10.1145/2939672.2939785.
- [28] G. Ke *et al.*, "LightGBM: A Highly Efficient Gradient Boosting Decision Tree," 2017. [Online]. Available: <https://github.com/Microsoft/LightGBM>.
- [29] 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>
- [30] D. Boldini, F. Grisoni, D. Kuhn, L. Friedrich, and S. A. Sieber, "Practical guidelines for the use of gradient boosting for molecular property prediction," *J Cheminform*, vol. 15, no. 1, Dec. 2023, doi: 10.1186/s13321-023-00743-7.
- [31] P. Singh, N. Singh, K. K. Singh, and A. Singh, "Chapter 5 - Diagnosing of disease using machine learning," in *Machine Learning and the Internet of Medical Things in Healthcare*, K. K. Singh, M. Elhoseny, A. Singh, and A. A. Elngar, Eds., Academic Press, 2021, pp. 89–111. doi: <https://doi.org/10.1016/B978-0-12-821229-5.00003-3>.
- [32] Y. Liu and S. Yang, "Application of Decision Tree-Based Classification Algorithm on Content Marketing," *Journal of Mathematics*, vol. 2022, 2022, doi: 10.1155/2022/6469054.
- [33] D. Asmawati, L. Arif Sanjani, C. Dimas Renggana, C. Fatichah, and T. Mustaqim, "Arrhythmia Classification with ECG Signal using Extreme Gradient Boosting (XGBoost) Algorithm," *Journal of Technology and*

- Informatics (JoTI)*, vol. 6, no. 1, pp. 36–42, Oct. 2024, doi: 10.37802/joti.v6i1.792.
- [34] Ş. K. Çorbacioğlu and G. Aksel, “Receiver operating characteristic curve analysis in diagnostic accuracy studies: A guide to interpreting the area under the curve value,” Oct. 01, 2023, *Wolters Kluwer Medknow Publications*. doi: 10.4103/tjem.tjem\_182\_23.
- [35] S. Afzal, “Research on Computational Complexity of Machine Learning,” Feb. 2022. [Online]. Available: <https://www.researchgate.net/publication/359441560>
- [36] D. Asmawati, L. Arif Sanjani, C. Dimas Renggana, C. Fatichah, and T. Mustaqim, “Arrhythmia Classification with ECG Signal using Extreme Gradient Boosting (XGBoost) Algorithm,” *Journal of Technology and Informatics (JoTI)*, vol. 6, no. 1, pp. 36–42, Oct. 2024, doi: 10.37802/joti.v6i1.792.
- [37] H. Wang *et al.*, “A high-precision arrhythmia classification method based on dual fully connected neural network,” *Biomed Signal Process Control*, vol. 58, Apr. 2020, doi: 10.1016/j.bspc.2020.101874.
- [38] B. Zhang, J. Wen, and H. Ren, “A Classification Method of Arrhythmia Based on Adaboost Algorithm,” in *Journal of Physics: Conference Series*, IOP Publishing Ltd, 2020. doi: 10.1088/1742-6596/1682/1/012060.
- [39] G. Moody, “MIT-BIH Arrhythmia Database Directory.” Accessed: Dec. 10, 2024. [Online]. Available: <https://archive.physionet.org/physiobank/database/html/mitdbdir/mitdbdir.htm>
- [40] G. B. Moody and R. G. Mark, “The impact of the MIT-BIH arrhythmia database,” *IEEE Eng Med Biol Mag*, vol. 20, no. 3, pp. 45–50, 2001, doi: 10.1109/51.932724.

- [41] S. Mandala *et al.*, “An improved method to detect arrhythmia using ensemble learning-based model in multi lead electrocardiogram (ECG),” *PLoS One*, vol. 19, no. 4, p. e0297551, Apr. 2024, doi: 10.1371/journal.pone.0297551.

