UNIVERSITAS LOGISTIK & BISNIS INTERNASIONAL

KOMPARASI PERFORMA MODEL TERHADAP KLASIFIKASI SINYAL MIT-BIH ARRHYTHMIA DATABASE

M. RIZKY
RONI ANDARSYAH

KOMPARASI PERFORMA MODEL TERHADAP KLASIFIKASI SINYAL MIT-BIH ARRHYTHMIA DATABASE

KOMPARASI PERFORMA MODEL TERHADAP KLASIFIKASI SINYAL MIT-BIH ARRHYTHMIA DATABASE

M. RIZKY
RONI ANDARSYAH

bukupedia

KOMPARASI PERFORMA MODEL TERHADAP KLASIFIKASI SINYAL MIT-BIH ARRHYTHMIA DATABASE

Penulis:
M. RIZKY
Roni Andarsyah

Editor:

M. Yusril Helmi Setyawan

Cetakan Pertama: Isi dengan Bulan saat upload buku

Cover: M. RIZKY

Tata Letak: Tim Kreatif Penerbit

Penerbit Buku Pedia
Athena Residence Blok. E No. 1, Desa Ciwaruga,
Kec. Parongpong, Kab. Bandung Barat 40559

Website: https://www.bukupedia.co.id
E-mail: penerbit@bukupedia.co.id

- Cet. I –: BukuPedia, 2023
Dimensi : 14,8 x 21 cm
ISBN: KOSONGKAN DULU

Undang-undang No.19 Tahun 2002 Tentang Hak Cipta Pasal 72

Barang siapa dengan sengaja melanggar dan tanpa hak melakukan perbuatan sebagaimana dimaksud dalam pasal ayat (1) atau pasal 49 ayat (1) dan ayat (2) dipidana dengan pidana penjara masing-masing paling sedikit 1 (satu) bulan dan/atau denda paling sedikit Rp.1.000.000,00 (satu juta rupiah), atau pidana penjara paling lama 7 (tujuh) tahun dan/atau denda paling banyak Rp.5.000.000.000,00 (lima miliar rupiah).

Barang siapa dengan sengaja menyiarkan, memamerkan, mengedarkan, atau menjual kepada umum suatu ciptaan atau barang hasil pelanggaran hak cipta terkait sebagai dimaksud pada ayat (1) dipidana dengan pidana penjara paling lama 5 (lima) tahun dan/atau denda paling banyak Rp.500.000.000,00 (lima ratus juta rupiah).

KATA PENGANTAR

Machine Learning merupakan teknologi yang berkembang sangat pesat saat ini. Banyak sekali produk-produk digital yang memanfaatkan teknologi Machine Learning mulai dari bidang transportasi, kedokteran, dan lainnya. Seiring banyaknya yang menggunakan Machine Learning, hampir semua produk yang di develop sekarang sudah mengimplementasikan Machine Learning.

Pemanfaatan Machine Learning di bidang kesehatan memang sangat perlu dilakukan terlebih lagi pentingnya penanganan dini kepada pasien yang mengidap penyakit khususnya pada penyakit jantung. Dengan melakukan pendeteksian dini pada suatu penyakit dapat meminimalisir berkembangnya penyakit tersebut.

Buku ini berisi bertujuan sebagai pembelajaran tentang bagaimana melakukan ekstraksi terhadap sinyal EKG dan melakukan klasifikasi pada sinyal tersebut berdasarkan *class*-nya. Buku ini diterbitkan bukan hanya karena penulis yang berperan, tetapi ada banyak pihak yang membantu.

Penulis yakin bahwa kesempurnaan hanya milik Allah SWT. Mohon maaf atas kekurangan buku ini. Oleh karena itu, penulis mohon maaf atas segala kekurangan dan kesalahan, baik yang disengaja maupun yang tidak disengaja. Saya berharap para pembaca dapat mengambil manfaat dari buku ini, terima kasih, dan selamat membaca.

Bandung, 9 Januari 2023

Penulis

DAFTAR ISI

DAFTAR GAMBAR

BAB 1
PENDAHULUAN

Dalam bab ini berisi tentang latar belakang penelitian yang dilakukan, permasalahan yang ingin dicapai, tujuan dan manfaat, ruang lingkup.

A. LATAR BELAKANG

Tidak bisa dipungkiri lagi bahwa perkembangan Artificial Intelligence begitu sangat cepat. Seiring berkembangnya teknologi yang sangat cepat, banyak sekali jenis - jenis teknologi yang bermunculan untuk membantu bahkan menggantikan pekerjaan manusia salah satunya di bidang kesehatan.Salah satu cabang Artificial Intelligence yang sekarang banyak sekali diminati adalah Machine Learning dan sampai saat ini masih terus berkembang pesat di kalangan programmer atau khususnya di dunia IT. Machine Learning sendiri terdiri atas 2 bagian yaitu Machine Learning Supervised dan Machine Learning Unsupervised.

Machine Learning Supervised adalah struktur dari suatu data yang hendak dianalisis telah ditentukan dahulu dan Machine Learning mencari data di struktur tersebut, sedangkan Machine Learning Unsupervised struktur dari suatu data dicari oleh Machine Learning itu sendiri [2]. Salah satu algoritma Supervised Learning yang sering digunakan pada proses klasifikasi adalah algoritma Random Forest (RF).

RF adalah teknik bagging yang memiliki karakteristik signifikan yang berjalan efisien pada dataset besar. Random forest dapat menangani ribuan variabel masukan tanpa penghapusan variabel dan memperkirakan fitur penting untuk klasifikasi [1].

Di dalam dunia medis, teknologi - teknologi banyak sekali diterapkan untuk memenuhi kebutuhan medis itu sendiri seperti AI pendeteksi pasien positif Covid atau tidak dengan memanfaatkan hembusan nafas dari pasien tersebut. Dan masih banyak lagi hal - hal yang bisa kita manfaatkan untuk mengembangkan teknologi di bidang kesehatan salah satunya adalah dengan memanfaatkan sinyal ECG atau Electrocardiography untuk mengklasifikasi penyakit gagal jantung pada pasien. ECG merupakan sebuah informasi sinyal yang digambarkan dalam bentuk diagram yang menampilkan informasi penting mengenai keadaan jantung manusia. Electrocardiography atau ECG adalah rekaman aktivitas listrik yang dihasilkan melalui siklus detak jantung [2]. Pada kegiatan internship yang penulis lakukan akan berfokus pada Ekstraksi dan klasifikasi sinyal MIT-BIH Arrhythmia menggunakan model Random Forest.

B. Rumusan Masalah

Dari latar belakang yang telah dijelaskan, bisa diambil rumusan masalahnya sebagai berikut:

1. Bagaimana proses ekstraksi fitur pada sinyal ECG?
2. Algoritma yang cocok untuk proses klasifikasi pada sinyal ECG?
3. Bagaimana proses implementasi model yang akan digunakan untuk mengklasifikasi data pada sinyal ECG?

C. Tujuan

Pada penelitian ini memiliki tujuan sebagai berikut:

1. Dapat memahami bagaimana proses ekstraksi fitur pada sinyal ECG
2. Dapat menentukan model yang terbaik untuk proses klasifikasi sinyal ECG
3. Dapat memahami bagaimana implementasi model yang dipilih dalam proses klasifikasi

D. MANFAAT

Manfaat dari penelitian ini adalah sebagai berikut:

1. Memahami bagaimana proses ekstraksi fitur pada sinyal ECG
2. Mengetahui model yang cocok dan menghasilkan akurasi yang tinggi untuk proses klasifikasi.

E. RUANG LINGKUP

Untuk menghindari pembahasan lebih luas, maka dalam penelitian ini peneliti mengambil objek penelitian sebagai berikut:

1. Data yang digunakan adalah data ECG atau Sinyal *Electrocardiogram* yang didapatkan pada situs *PhysioNet Database.*
2. Dataset yang digunakan merupakan data yang dipublish pada tanggal 24 Mei 1997.
3. Pendekatan yang digunakan adalah beberapa model Machine Learning Tradisional.
4. Bahasa pemrograman menggunakan bahasa *Python*.
5. Aplikasi yang digunakan adalah *Google Colab*.

F. SISTEMATIKA PENULISAN

Berdasarkan latar belakang dan perumusan masalah diatas, maka penyusunan buku ini dibuat dalam suatu sistematika yang terdiri dalam lima BAB, yaitu:

BAB 1 PENDAHULUAN

Bab ini berisi penjelasan terkait dengan pemaparan teori umum dengan topik yang dibahas secara global dan mengaitkan dengan referensi yang ada. Tujuan menjelaskan tentang solusi dari masalah yang ada. Ruang lingkup menjelaskan mengenai batasan dalam pemodelan dan aplikasi tersebut. Serta sistematika penulisan menjelaskan tentang isi dari aplikasi tersebut.

BAB 2 TINJAUAN PUSTAKA

Bab ini berisi konsep dasar dan pendukung dari sistem yang akan dibangun dengan menggunakan metode tertentu dan teori-teori dasar pengetahuan.

BAB 3 METODOLOGI PENELITIAN

Bab ini berisi penjelasan ruang lingkup, diagram alur metodologi penelitian beserta tahapan – tahapan diagram alur penelitian untuk menyelesaikan penelitian yang sedang dilakukan sehingga bisa mencapai tujuan yang diharapkan.

BAB 4 HASIL PENELITIAN

Bab ini berisi implementasi hasil penelitian berdasarkan capaian yang ingin dicapai pada metodologi penelitian.

BAB 5 PENUTUP

Bab ini berisi tentang kesimpulan mengenai capaian tujuan penelitian yang telah dilakukan.

BAB 2
LANDASAN TEORI

Bab ini berisi konsep dasar dan pendukung dari sistem yang akan dibangun dengan menggunakan metode tertentu dan teori-teorlasifikasi yang luar biasa, namun output yang dihasilkan jarang benar dalam kenyataan [17]. Terlepas dari hal itu, pada kenyataanya Naive Bayes bekerja dengan baik diimplementasikan di dunia ii dasar pengetahuan yang menjadi landasan kita dalam melakukan penelitian.

2.1. *MIT-BIH Arrhythmia Database*

MIT-BIH Arrhythmia database adalah rangkaian uji standar yang umumnya tersedia untuk mengevaluasi aritmia deteksi. Sejak 1980, *database* ini telah digunakan untuk dasar penelitian untuk dinamika jantung di sekitar 500 lokasi di seluruh dunia. Basis data ini sebagian besar digunakan untuk tujuan medis dan penelitian dari deteksi dan analisis aritmia jantung yang berbeda. Basis data ini mencoba menyediakan informasi yang tepat untuk mendeteksi aritmia ventrikel [3].

Aritmia adalah perubahan detak jantung yang tidak normal karena detak jantung yang tidak tepat yang menyebabkan kegagalan dalam pemompaan darah. Aritmia dapat menyebabkan kematian jantung mendadak. Gejala aritmia yang umum adalah denyut prematur,

jantung berdebar, pusing, kelelahan, dan pingsan. Aritmia lebih sering terjadi pada orang yang menderita tekanan darah tinggi, diabetes dan arteri koroner penyakit [4]. Sinyal *Electrocardiogram* yang akan digunakan pada kegiatan internship ini diambil dari *MIT-BIH Arrhythmia database*.

2.2. *Electrocardiogram (ECG)*

Electrocardiogram (ECG) adalah tes medis yang mengukur aktivitas listrik jantung. ECG digunakan untuk mendiagnosis dan memantau berbagai kondisi jantung, seperti serangan jantung, aritmia, dan gagal jantung. ECG akan merekam aktivitas listrik kecil yang dihasilkan oleh jantung selama periode waktu tertentu dengan menempatkan elektroda pada tubuh pasien [5]. Rekaman ECG berisi *noise* dan amplitudo yang bervariasi dari setiap orang sehingga sulit dalam proses mendiagnosis [6].

Electrocardiogram (ECG) memberikan informasi penting tentang berbagai kondisi manusia [7]. Untuk melakukan ECG, petugas kesehatan akan menempelkan tambalan kecil dan lengket yang disebut elektroda ke dada, lengan, dan kaki pasien. Elektroda terhubung ke mesin ECG, yang merekam sinyal listrik yang dihasilkan oleh jantung saat bergerak ke seluruh tubuh. Mesin tersebut menghasilkan jejak aktivitas listrik jantung, yang disebut strip ECG, yang kemudian diinterpretasikan oleh petugas kesehatan.

Machine Learning dapat diterapkan pada analisis data ECG untuk meningkatkan akurasi dan efisiensi diagnosis dan pengobatan.

Misalnya, algoritma pembelajaran mesin dapat dilatih untuk mengenali pola dalam data ECG yang menunjukkan kondisi jantung tertentu. Algoritma ini kemudian dapat digunakan untuk menganalisis data ECG secara otomatis dan memberikan rekomendasi diagnosis atau pengobatan. Algoritma pembelajaran mesin dapat dilatih untuk mengklasifikasikan data ECG ke dalam kategori yang berbeda, seperti normal atau abnormal, atau untuk mengidentifikasi kondisi jantung tertentu.

Biasanya, klasifikasi sinyal ECG memiliki empat fase: preprocessing, segmentasi, ekstraksi fitur, dan klasifikasi. Fase preprocessing terutama ditujukan untuk mendeteksi dan melemahkan frekuensi sinyal ECG yang terkait dengan artefak, yang juga biasanya melakukan normalisasi dan peningkatan sinyal. Setelah preprocessing, segmentasi akan membagi sinyal menjadi segmen yang lebih kecil, yang dapat mengekspresikan aktivitas listrik jantung dengan lebih baik [6].

2.3. Machine Learning

Pembelajaran Mesin atau Machine Learning merupakan kemajuan teknologi yang penting karena dapat membantu dalam mengambil keputusan dengan mekanisme prediksi dan klasifikasi berdasarkan data yang ada [8]. Berfokus pada performance yang tinggi, teknik pembelajaran mesin atau machine learning diterapkan pada bisnis dengan data yang berkembang pesat. Karena pendekatan desain cocok untuk komunikasi komputasi paralel dan terdistribusi yang berevolusi atau data bisnis yang dinamis dan berkembang kedalam model Machine Learning [9].

Teknologi berbasis komputer modern banyak yang telah menggunakan Pembelajaran Mesin atau Machine Learning. Machine Learning merupakan cabang dari Kecerdasan Buatan atau Artificial Intelligence yang luas dan sudah berkembang pesat saat ini yang memungkinkan komputer untuk belajar dan berkembang secara otomatis tanpa harus diprogram secara eksplisit. Teknologi ini berasal dari mempelajari pengenalan pola dan teori pembelajaran komputasi. Secara umum, metode pembelajaran yang umum digunakan oleh Machine Learning dapat diklasifikasikan menjadi Supervised, Unsupervised, dan Reinforcement Learning [10].

2.3.1. Supervised Learning

Supervised Learning merupakan metode Machine Learning untuk menyimpulkan fungsi dari data *train* ada. Algoritma Supervised Learning biasanya berisi kumpulan sampel input (*feature*) dan label yang berkaitan dengan kumpulan data tersebut. Tujuan dari pengklasifikasian adalah untuk menemukan batas yang sesuai yang dapat memprediksi label yang benar pada data *test*. Secara singkat, dalam Supervised Learning memiliki setiap contoh data yang berpasangan yang terdiri dari objek masukan (*input*) dan objek keluaran (*output*) yang diinginkan. Algoritma Supervised Learning menganalisis data *train* dan menghasilkan fungsi (*model*) [11]. Beberapa contoh metode algoritma pada Supervised Learning:

2.3.1.1. Random Forest

Random Forest adalah metode Machine Learning yang diperkenalkan pada tahun 2001 oleh Leo Breiman. Metode ini menggunakan serangkaian besar dari Decision Tree dengan korelasi timbal balik yang rendah dan fitur yang dipilih secara acak menggunakan metode bagging (Bootstrap AGGregatING) [12].

Random Forest merupakan salah satu metode pengklasifikasian terbaik dan banyak digunakan untuk regresi dan juga klasifikasi. Random Forest memiliki algoritma yang sederhana sehingga menjadi salah satu pilihan yang menarik untuk mengklasifikasi teks. Selain itu, Random Forest juga memiliki kemampuan untuk mengolah data berdimensi tinggi dan memiliki performa yang tinggi walaupun menggunakan data yang banyak sehingga menjadi salah satu keuntungan menggunakan model ini dibandingkan dengan model Machine Learning lainnya [13].

Pemilihan model ini didasarkan karena pada faktanya bahwa Random Forest secara luas dianggap sebagai salah satu metode Machine Learning yang paling sukses dan banyak digunakan hingga saat ini [14].

2.3.1.2. Naive Bayes

Naive Bayes adalah salah satu pengklasifikasi terkemuka yang telah banyak dikutip oleh banyak peneliti

dan digunakan di banyak aspek karena kesederhanaannya dan kinerja dari klasifikasi yang nyata [15]. Diantara bermacam-macam teknik atau metode klasifikasi saat ini, pengklasifikasi Naive Bayes (*NB*) berperan penting karena kesederhanaan, traktabilitas dan efisiensinya [16].

Naive Bayes juga merupakan pengklasifikasi yang sangat kompeten dalam banyak aplikasi di dunia nyata. Meskipun Naive Bayes telah menunjukan akurasi klasifikasi yang luar biasa, namun output yang dihasilkan jarang benar dalam kenyataan [17]. Terlepas dari hal itu, pada kenyataanya Naive Bayes bekerja dengan baik diimplementasikan di dunia ini seperti memprediksi waktu, pemfilteran spam, prakiraan cuaca, dan diagnosis medis [18].

Pengklasifikasian Naive Bayes didasarkan pada kombinasi Teorema Bayes dan asumsi independensi atribut. Pengklasifikasi Naive Bayes didasarkan pada asumsi yang disederhanakan bahwa nilai atribut bersifat independen secara kondisional, berdasarkan asumsi nilai target yang diberikan. Pendekatan Bayes untuk klasifikasi kasus baru terdiri dari penetapan nilai target yang paling mungkin, dengan asumsi bahwa ada [19].

2.3.1.3. K-Nearest Neighbor

K-Nearest Neighbor (KNN) merupakan salah satu algoritma klasifikasi yang paling stabil dalam kelompok algoritma klasifikasi supervised. Dikarenakan kesederhanaan dan implementasi algoritma yang mudah.

K-Nearest Neighbor (KNN) merupakan salah satu metode klasifikasi nonparametric. KNN menjadi terkenal karena algoritmanya yang luas dan yang paling mudah. KNN dapat menyimpan semua masalah atau studi kasus yang ada dan mengklasifikasikan berdasarkan kesamaannya. Secara umum, KNN menggunakan jarak Euclidean untuk menemukan data yang paling mirip dengan kelompoknya [20].

Pada metode ini, nilai yang hilang dari variabel tertentu diganti dengan nilai rata-rata atau nilai mean dari KNN terdekat dari pengamatan variabel yang sama. Fungsi jarak yang berbeda dapat digunakan untuk memilih tetangga yang memungkinkan metode untuk menyertakan variabel numerik dan kategori. Keuntungan utama KNN adalah tidak memerlukan spesifikasi model prediktif apapun [21].

2.3.1.4. Support Vector Machine (SVM)

Support Vector Machine membuktikan bahwa salah satu algoritma yang memiliki performance yang powerful

selama beberapa dekade terakhir dan mengandalkan prinsip SRM. Support Vector Learning (SVM) umumnya digunakan untuk masalah klasifikasi dan regresi.

Support Vector Machine (SVM) bekerja dengan membangun hyperplane yang memisahkan sampel berdasarkan pendekatan margin yang maksimum. Berbeda dengan Artificial Neural Network (ANN) yang memiliki kelemahan local minimal. Support Vector Machine memberikan solusi dengan menyelesaikan masalah optimasi dengan konveks [22].

Metode Support Vector Machine juga disebut model klasifikasi biner. Dalam ruang dua dimensi, garis lurus menjadikan garis segmentasi yang paling cocok di tengah 2 kelas data, dan untuk kumpulan data berdimensi tinggi, ini untuk menetapkan bidang keputusan yang optimal sebagai tolak ukur klasifikasi. Prinsip dasar Support Vector Machine (SVM) mensyaratkan bahwa ketika masalah klasifikasi diselesaikan, jarak dari titik sampel terdekat ke permukaan keputusan adalah yang terbesar, yaitu jarak minimum memaksimalkan dua kelas titik sampel untuk memisahkan tepi [23].

2.3.2. Unsupervised Learning

Unsupervised Learning hanya dapat digunakan untuk tugas pengelompokan (clustering). Banyak pendekatan

menggunakan Unsupervised Learning untuk mendukung tugas klasifikasi. Misalnya, algoritma pengelompokan (clustering) dapat meningkatkan kinerja tugas klasifikasi dengan mengelompokan objek data ke dalam kelompok yang lebih homogen.

Unsupervised Learning banyak digunakan untuk preprocessing data seperti ekstraksi fitur, pemilihan fitur, dan resampling. Namun, ada banyak juga kasus penggunaan pembelajaran tanpa pengawasan sebagai algoritma pilihan untuk klasifikasi dengan kinerja yang sebanding dan mungkin lebih baik dibandingkan Supervised Learning [24].

Pada algoritma Unsupervised Learning yang mampu memisahkan data tanpa sebuah pengetahuan yang dalam tentang berbagai jenis peristiwa meningkatkan efisiensi analisis secara luar biasa, dan memungkinkan analisis hilir untuk berkonsentrasi pada upaya penyesuaian hanya pada peristiwa yang menarik. Selain itu, algoritma pengelompokan memungkinkan lebih banyak eksplorasi data, berpotensi memungkinkan jenis reaksi baru dan tak terduga [25].

2.3.2.1. K-Means

Di era big data, sejumlah besar sumber daya data dikumpulkan dari kehidupan orang sehari-hari, ditransfer ke dalam internet, dan disimpan pada pusat data [26]. Pengelompokan data (Clustering), sebagai bagian penting

dari data mining, dan sudah dianggap sebagai tugas penting dalam Unsupervised Learning.

Untuk kumpulan data tertentu, clustering akan membaginya menjadi beberapa kelompok atau cluster yang sedemikian rupa sehingga objek data dalam kelompok atau cluster yang sama berupa satu sama lain [27]. K-Means adalah pengelompokan masalah yang dipelajari dengan baik yang menghasilkan aplikasi di banyak bidang dan merupakan bagian dari Unsupervised Learning. K-Means merupakan salah satu masalah paling mendasar dalam ilmu komputer [28].

BAB 3
METODOLOGI PENELITIAN

Bab ini berisi penjelasan ruang lingkup, diagram alur metodologi penelitian beserta tahapan – tahapan diagram alur penelitian untuk menyelesaikan penelitian yang sedang dilakukan sehingga bisa mencapai tujuan yang diharapkan.

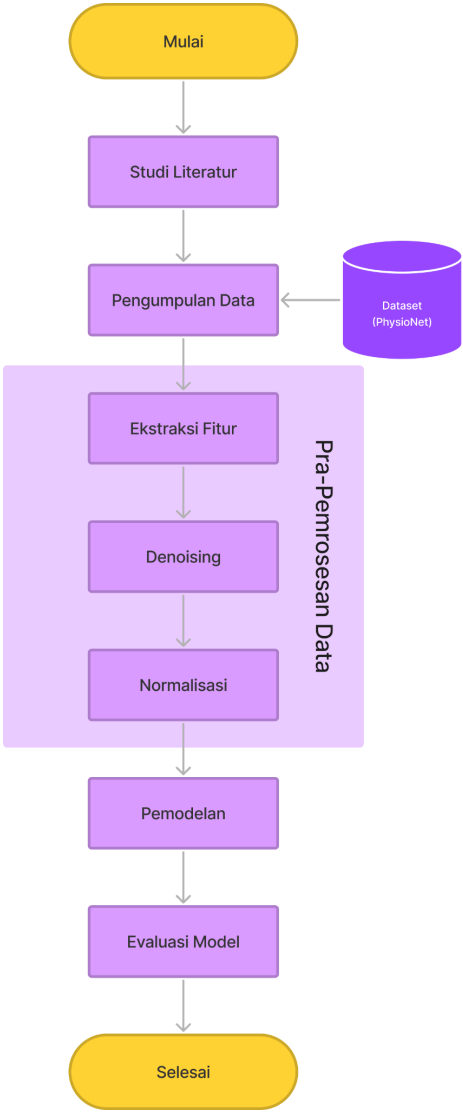
3.1. Ruang Lingkup

Penelitian ini akan membahas tentang ekstraksi dan klasifikasi MIT-BIH Arrhythmia Database dengan menggunakan salah satu model Machine Learning yaitu Random Forest. Sumber data yang akan digunakan berasal dari situs PhysioNet yang merupakan Database Complex Physiologic Signals. Data yang akan diambil pada penelitian ini adalah MIT-BIH Arrhythmia Database dengan 48 record dan masing-masing durasi yang tersedia. Dari data-data tersebut akan digabungkan lalu dilakukan ekstraksi dan klasifikasi sinyal dengan Random Forest.

3.2. Alur Metodologi Penelitian

Pada penelitian ini, metode akan menjadi hal penting dalam melakukan ekstraksi hingga klasifikasi sinyal karena akan mempengaruhi hasil klasifikasi atau output dari model tersebut. Oleh

karena itu, alur metodologi yang tepat untuk mendapatkan hasil yang terbaik. Alur penelitian akan ditunjukan pada gambar 3.1.



Gambar 3.1 Flow Diagram Metodologi Penelitian

3.3. Indikator Capaian Penelitian

Berdasarkan gambar diatas, terdapat beberapa indikator capaian yaitu sebagai berikut:

No	Tahapan	Indikator Capaian
1	Studi Literatur	Uraian Teori - teori
2	Pengumpulan Data	Dataset sinyal MIT-BIH Arrhythmia
3	Pra-Pemrosesan Data	Ekstraksi Fitur, Denoising, Normalisasi
4	Pemodelan	Model Random Forest untuk Klasifikasi Sinyal
5	Evaluasi Model	Melakukan perbandingan dengan beberapa model

Tabel 3.1 Indikator Capaian Penelitian

3.3.1. Studi Literatur

Pada studi literatur ini berisikan tentang uraian teori-teori bahan penelitian orang lain yang didapatkan pada jurnal - jurnal nasional maupun internasional. Pada bagian ini akan dijadikan acuan dari kegiatan penelitian ini untuk mengimplementasikan dan juga mengembangkan sesuai dengan teori-teori yang dijelaskan.

3.3.2. Pengumpulan Data

Pada tahap ini, data yang akan digunakan adalah data yang diambil langsung dari website PhysioNet yang merupakan *Research Resource for Complex Physiological Signals* untuk melakukan penelitian dan pendidikan biomedis dan menawarkan akses gratis pada database yang disediakan. PhysioNet didirikan pada tahun 1999 dibawah naungan *National Institutes of Health* (*NIH*).

Data yang dikumpulkan merupakan data hasil rekaman detak jantung dari beberapa orang. Data yang digunakan berbentuk CSV dengan nilai hasil dari indikator grafik yang digambarkan. Pada data ini juga diberikan anotasi di setiap detak jantungnya untuk dijadikan label dari dataset tersebut.

3.3.3. Pra-Pemrosesan Data

Pra-Pemrosesan data merupakan proses pembersihan data dengan melakukan ekstraksi fitur dengan menggabungkan data menjadi satu data frame dengan masing-masing notasinya yang nantinya data tersebut akan dilakukan *denoising* agar data yang digunakan menjadi optimal dan tidak terdapat data-data yang *outlier* atau jauh dari nilai aslinya. Setelah itu, data yang sudah dilakukan denoising akan di normalisasi agar skala yang dipakai memiliki nilai yang sama.

3.3.4. Pemodelan

Pada tahap ini akan dilakukan pemodelan dengan menggunakan data yang sebelumnya. Data yang tersedia perlu dilakukan split untuk keperluan data *train* dan data *test* untuk nanti dilakukan proses pemodelan menggunakan data *train* tersebut.

3.3.5. Evaluasi Model

Tahap evaluasi model merupakan tahapan untuk mengukur kinerja dari model yang dihasilkan apakah sudah memiliki akurasi yang baik atau tidak sehingga pada proses ini kita bisa melakukan perbandingan kinerja antara model yang tersedia dan model mana yang memiliki akurasi yang baik sesuai dengan hasil pra-pemrosesan data tersebut.

BAB 4
HASIL PENELITIAN

4.1. Data

Data yang akan digunakan diambil langsung dari website PhysioNet yang merupakan *Research Resource for Complex Physiological Signals* untuk melakukan penelitian dan pendidikan biomedis. Data yang dikumpulkan merupakan data hasil rekaman detak jantung dari beberapa orang. Sejak tahun 1975, laboratorium Beth Israel Deaconess Medical Center dan MIT menyediakan wadah untuk menganalisis aritmia dan beberapa penelitian terkait. Dan MIT-BIH merupakan bagian utama dari penelitian tersebut yang diselesaikan serta didistribusikan pada tahun 1980. Sehingga dataset ini merupakan perangkat uji standar pertama yang tersedia secara umum untuk evaluasi detektor aritmia.

Bentuk data yang digunakan merupakan data CSV dengan nilai yang membentuk grafik dari sebuah detak jantung atau *beat*. Data tersebut memiliki *annotation* nya masing-masing disetiap *beat* yang bisa dijadikan label untuk proses klasifikasi. Berikut record data yang akan digunakan:

Record	Channel	Gender	Age	Medications
100	MLII, V5	Male	69	Aldomet, Inderal
101	MLII, V1	Female	75	Diapres
102	V5, V2	Female	84	Digoxin
103	MLII, V2	Male	-	Diapres, Xyloprim
104	V5, V2	Female	66	Digoxin, Pronestyl
105	MLII, V1	Female	73	Digoxin, Nitropaste, Pronestyl
106	MLII, V1	Female	24	Inderal
107	MLII, V1	Male	63	Digoxin
108	MLII, V1	Female	87	Digoxin, Quinaglute
109	MLII, V1	Male	64	Quinidine
111	MLII, V1	Female	47	Digoxin, Lasix
112	MLII, V1	Male	54	Digoxin, Pronestyl
113	MLII, V1	Female	24	-
114	V5, MLII	Female	72	Digoxin
115	MLII, V1	Female	39	-
116	MLII, V1	Male	68	-
117	MLII, V2	Male	69	-
118	MLII, V1	Male	69	Digoxin, Norpace
119	MLII, V1	Female	51	Pronestyl
121	MLII, V1	Female	83	Digoxin, Isordil, Nitropaste

Record	Channel	Gender	Age	Medications
122	MLII, V1	Male	51	Digoxin, Lasix, Pronestyl
123	MLII, V5	Female	63	Digoxin, Inderal
124	MLII, V4	Male	77	Digoxin, Isordil, Quinidine
200	MLII, V1	Male	64	Digoxin, Quinidine
201	MLII, V1	Male	68	Digoxin, Hydrochlorothiazide, Inderal, KCl
202	MLII, V1	Male	68	Digoxin, Hydrochlorothiazide, Inderal, KCl
203	MLII, V1	Male	43	Coumadin, Digoxin, Heparin, Hygroton, Lasix
205	MLII, V1	Male	59	Digoxin, Quinaglute
207	MLII, V1	Female	89	Digoxin, Quinaglute
208	MLII, V1	Female	23	-
209	MLII, V1	Male	62	Aldomet, Hydrodiuril, Inderal
210	MLII, V1	Male	89	-
212	MLII, V1	Female	32	-
213	MLII, V1	Male	61	Digoxin
214	MLII, V1	Male	53	Digoxin, Dilantin
215	MLII, V1	Male	81	-
217	MLII, V1	Male	65	Digoxin, Lasix, Quinidine
219	MLII, V1	Male	-	Digoxin

Record	Channel	Gender	Age	Medications
220	MLII, V1	Female	87	Digoxin
221	MLII, V1	Male	83	Hydrochlorthiazide, Lasix
222	MLII, V1	Female	84	Digoxin, Quinidine
223	MLII, V1	Male	73	-
228	MLII, V1	Female	80	Digoxin, Norpace
230	MLII, V1	Male	32	Dilantin
231	MLII, V1	Female	72	-
232	MLII, V1	Female	76	Aldomet, Inderal
233	MLII, V1	Male	57	Dilantin
234	MLII, V1	Female	56	-

Tabel 4.1 Record Dataset

Dari tabel 4.1 menunjukan record dataset yang dihimpun dari website resmi PhysioNet akan digunakan dengan masing-masing kolom seperti *Record, Channel, Gender, Age,* dan *Medications*. Kolom *Channel* merupakan kolom yang nilainya akan digunakan sebagai proses klasifikasi detak jantung yang nantinya akan digabungkan pada annotations nya masing-masing tiap *beat*-nya. Berikut contoh dataset pada sebuah record:

'sample #'	'MLII'	'V1'
0	955	992

1	955	992
2	955	992
3	955	992
4	955	992
5	955	992
6	955	992
7	955	992
8	958	994
9	960	995
10	960	990
…	…	…
649999	1024	1024

Tabel 4.2 Contoh Record Dataset MLII dan V1

Pada tabel 4.2 memiliki 3 kolom yang masing-masing nilai yang berbeda-beda tiap barisnya. Pada kolom MLII memiliki nilai yang akan dijadikan nilai plotting pada sebuah detak jantung dengan range plotting 360 baris sehingga 360 baris pada kolom MLII merupakan 1 kali plotting detak jantung atau *beat*.

Di Setiap kolom tidak memiliki nilai yang kosong atau NULL, sehingga bisa langsung digunakan tanpa harus mengisi lagi nilai-nilai yang kosong. Setelah data-data disiapkan, proses selanjutnya adalah dengan melakukan Pra-Pemrosesan data.

4.2. Pra-Pemrosesan Data

Pada tahap ini, kita akan mempersiapkan data dengan mengolahnya agar data tersebut dapat digunakan untuk proses akhir atau pemodelan. Beberapa langkah Pra-Pemrosesan data yang akan dilakukan seperti Split Extension File, Record Signal Dataset, Denoising, Normalisasi, Merge Annotation, Rebalancing Dataset, Split Dataset.

4.2.1. Split Extension File

Dataset yang digunakan merupakan kumpulan beberapa record dan annotation, untuk itu perlu dilakukan split untuk untuk memisahkan file record dan annotation masing-masing record.

```
def split_file(filenames):
 records = list()
 annotations = list()

 for f in filenames:
    filename, file_extension = os.path.splitext(f)

    if(file_extension == '.csv'):
       records.append(path + filename + file_extension)
    else:
       annotations.append(path + filename + file_extension)

 return records, annotations
```

Pada fungsi diatas terdapat kondisional yang memisahkan jenis extension file “.csv” dan memasukkan ke dalam sebuah array. Terdapat 2 extension yang di pisah yaitu “.csv” untuk dataset attribute dan “.txt” untuk label.

4.2.2. Record Signal Dataset

Tahap ini dilakukan untuk proses menyiapkan bentuk nilai dari record dataset untuk dilakukan plotting apakah hasil plotting sudah sesuai dengan bentuk sinyal ECG atau tidak.

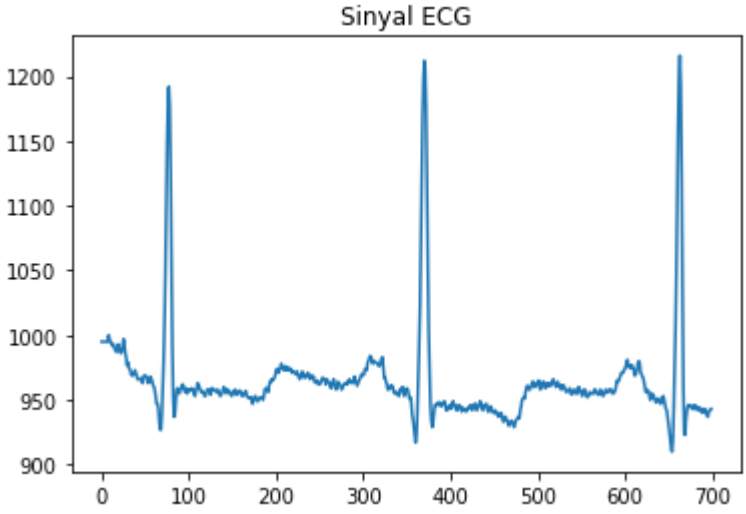
```
def get_record(record):
 signals = []

 with open(record, 'rt') as csvfile:
    spamreader = csv.reader(csvfile, delimiter = ',', quotechar =
'|')
    row_index = -1
    for row in spamreader:
       if(row_index >= 0):
          signals.insert(row_index, int(row[1]))
       row_index += 1

 return signals
```

Fungsi diatas akan membaca file CSV pada parameter dengan mengambil nilai MLII pada CSV untuk dijadikan attribute pada proses plot sinyal maupun proses klasifikasi. Nilai tersebut akan disimpan pada sebuah array yang nantinya akan diplot

menggunakan array tersebut. Untuk melakukan plot sinyal menggunakan library matplotlib dari python.

```
plt.title("Sinyal ECG")
plt.plot(signals[0:700])
plt.show()
```

Dari potongan kode diatas akan menghasilkan plot dari contoh sinyal ECG dari dataset yang disiapkan sebelumnya. Berikut contoh gambar hasil plot sinyal ECG:



Gambar 4.1 Contoh Sinyal ECG

Pada gambar diatas terlihat bahwa hasil plotting membentuk sinyal ECG atau Detak Jantung. Terdapat 3 *beat* yang dihasilkan dari hasil plotting dengan mengambil nilai mulai dari baris ke 1 sampai baris ke 700. Tetapi pada sinyal ECG tersebut

masih memiliki *noise* yang merupakan sinyal gangguan. Untuk itu, perlu dilakukan Denoising pada dataset untuk menghilangkan *noise* tersebut.

4.2.3. Denoising

Denoising merupakan tahapan yang sangat penting dilakukan untuk menghilangkan *noise* pada sinyal. Jika *noise* tersebut tidak dihilangkan akan mengganggu dan menurunkan performa dari hasil modeling nanti karena banyak bentuk sinyal yang hanya mengganggu dari hasil sinyal yang sebenarnya.

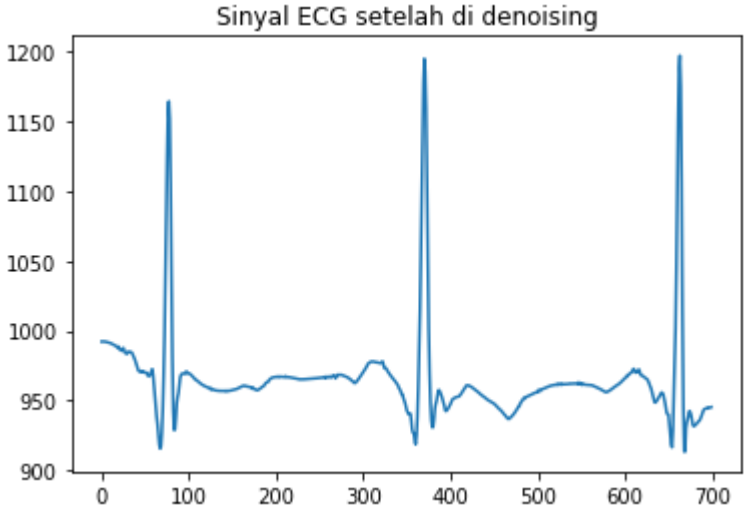
```
def denoise(df):
  w = pywt.Wavelet('sym4')
  maxlev = pywt.dwt_max_level(len(df), w.dec_len)
  threshold = 0.04

  coeffs = pywt.wavedec(df, 'sym4', level=maxlev)
  for i in range(1, len(coeffs)):
    coeffs[i] = pywt.threshold(coeffs[i],
threshold*max(coeffs[i]))

  datarec = pywt.waverec(coeffs, 'sym4')

  return datarec
```

Untuk melakukan Denoising menggunakan library "pywt" dengan mengkalkulasikan Wavelet untuk mengelompokan nilai menjadi sebuah Wavelet tunggal. Untuk nilai *Threshold* yang digunakan adalah "0.04". Setelah sinyal

dilakukan Denoising, maka akan membentuk sinyal yang sempurna dan tidak terdapat banyak *noise.*

```
signals = denoise(signals)
plt.title("Sinyal ECG setelah di denoising")
plt.plot(signals[0:700])
plt.show()
```

Potongan kode diatas akan melakukan Denoising menggunakan fungsi yang telah dibuat sebelumnya yaitu fungsi denoise(). Setelah dilakukan Denoising maka akan melakukan plot sinyal sebanyak 700 baris nilai pada dataset.



Gambar 4.2 Contoh Sinyal ECG setelah Denoising

Terlihat pada gambar diatas bahwa sinyal yang di plotting sudah bersih dari *noise* dan memiliki bentuk sinyal yang lebih

sempurna daripada sinyal yang sebelumnya belum dilakukan Denoising. Karena sinyal ECG memiliki nilai yang berbeda-beda tiap recordnya, maka perlu dilakukan *Normalisasi* pada dataset tersebut untuk menyamakan skalanya.

4.2.4. Normalisasi

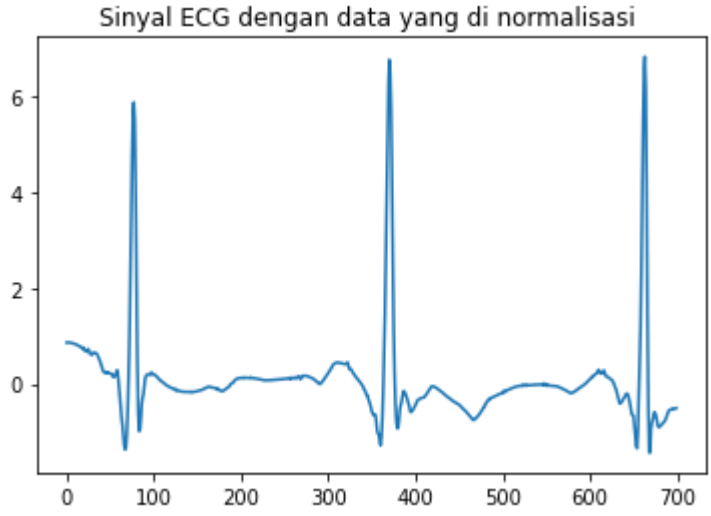
Normalisasi merupakan tahapan yang perlu dilakukan untuk mengubah nilai dengan tipe data Numerik pada himpunan data agar skala dari himpunan data tersebut sama. Akan Tetapi, tidak semua dataset bisa dilakukan normalisasi tergantung dari algoritma atau model yang akan dipakai untuk proses klasifikasi.

```
signals = stats.zscore(signals)
```

Fungsi diatas akan melakukan normalisasi nilai dari data atau sinyal yang telah dilakukan Denoising sebelumnya. Setelah dilakukan normalisasi maka data sinyal tersebut bisa langsung digunakan untuk klasifikasi. Untuk proses normalisasi ini akan mengubah nilai dari MLII pada data sebelumnya menjadi nilai yang sudah dinormalisasikan tentunya.

```
plt.title("Sinyal ECG dengan data yang di normalisasi ")
plt.plot(signals[0:700])
plt.show()
```

Dari fungsi diatas menggunakan data atau nilai sinyal yang sudah dinormalisasikan dari data yang sudah di denoising sebelumnya. Karena normalisasi hanya mengubah nilai skalanya menjadi kecil, maka harusnya hasil plotingan yang sudah di denoising dengan yang dinormalisasikan tidak akan berubah.



Gambar 4.3 Contoh Sinyal ECG setelah Normalisasi

Terlihat pada nilai Y dari hasil plotingan diatas berubah yang awalnya bernilai 900an menjadi nilai dengan range -1 sampai 7 tetapi tetap membentuk sinyal atau detak jantung yang sama seperti sebelumnya. Sebelum ke proses selanjutnya harus dilakukan penggabungan annotation pada setiap detak jantung atau *beat*.

4.2.5. Merge Annotation

Pada tahapan ini, menggabungkan annotation atau label pada data yang akan kita lakukan klasifikasi sangat penting, karena annotation inilah yang akan menentukan bahwa model yang dibuat berjalan dengan baik atau tidak nantinya. Pada proses ini juga akan dilakukan plotting dari masing-masing *class* atau label pada dataset tersebut.

```
example_beat_N_printed = False
example_beat_L_printed = False
example_beat_R_printed = False
example_beat_A_printed = False
example_beat_V_printed = False
```

Diatas merupakan variabel yang akan digunakan untuk pengkondisian untuk melakukan plotting pada masing-masing *class* atau label. Karena untuk mendapatkan nilai yang sesuai dengan *class* yang diinginkan, perlu pengkondisian pada suatu *looping*.

```
for r in range(0, len(records)):
  signals = []

  with open(records[r], 'rt') as csvfile:
     spamreader = csv.reader(csvfile, delimiter = ',', quotechar =
'|')
     row_index = -1
     for row in spamreader:
```

```
        if(row_index >= 0):
            signals.insert(row_index, int(row[1]))
        row_index += 1
```

Pada kodingan diatas akan melakukan *looping* sesuai dengan banyaknya file records yang telah di split sebelumnya lalu akan membuka isi file CSV tersebut dan membaca nilainya. Nilai tersebut akan dimasukan kedalam array untuk nantinya digabungkan dengan record yang lainnya.

```
  with open(annotations[r], 'r') as fileID:
    data = fileID.readlines()
    beat = list()

    for d in range(1, len(data)):
      splitted = data[d].split(' ')
      splitted = filter(None, splitted)
      next(splitted)

      pos = int(next(splitted))

      arrhythmia_type = next(splitted)
```

Setelah membuka dan membaca nilai dari CSV, maka dilanjutkan dengan membuka file annotation untuk mengambil sampel dari masing-masing beat dan juga mengambil *class* atau label dari data CSV sebelumnya. Proses selanjutnya ada dengan

mengecek apakah *class* atau label tersebut sudah sesuai dengan yang diperlukan untuk proses klasifikasi.

```
    if(arrhythmia_type in classes):
      arrhythmia_index = classes.index(arrhythmia_type)
      count_classes[arrhythmia_index] += 1
      if(window_size <= pos and pos < (len(signals) -
window_size)):
        beat = signals[pos - window_size : pos + window_size]

        X.append(beat)
        y.append(arrhythmia_index)
```

Kodingan diatas menunjukan bahwa apabila *class* atau label yang diambil dari file annotation ada pada variabel *classes*, maka *beat* tersebut bisa di pakai untuk dataset untuk proses pengklasifikasian nanti. Setelah proses pengkondisian, maka beat tersebut dimasukan kedalam array untuk dijadikan dataframe nantinya. Pada variabel X merupakan kolom beat sedangkan untuk variabel y untuk *class* atau label dari beat tersebut.

Selanjutnya adalah dengan melakukan plot pada masing-masing *class* berdasarkan nilai pada variabel *classes* yang terdiri dari 5 *class* seperti berikut:

```
classes = ['N', 'L', 'R', 'A', 'V']
```

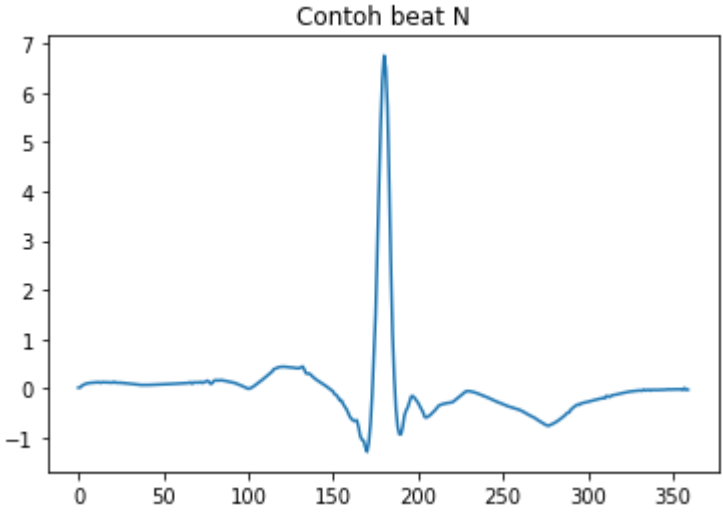
Keterangan:

- N: Normal beat (displayed as "•" by the PhysioBank ATM, LightWAVE, pschart, and psfd)
- L: Left bundle branch block beat
- R: Right bundle branch block beat
- A: Atrial premature beat
- V: Premature ventricular contraction

Dari keterangan diatas merupakan arti dari masing-masing label yang akan dipakai untuk proses klasifikasi. N merupakan beat normal manusia dan diikuti dengan A dan V untuk beat yang prematur.

```
        if arrhythmia_type == 'N' and not
example_beat_N_printed:
            print("Contoh Beat")
            example_beat_N_printed = True
            plt.title("Contoh beat " + arrhythmia_type)
            plt.plot(beat)
            plt.show()
```

Pada kodingan diatas apabila *class* bernilai ‘N’ dan *class* tersebut belum dilakukan plot, maka kodingan tersebut akan melakukan plot detak jantung atau *beat* dengan label ‘N’.

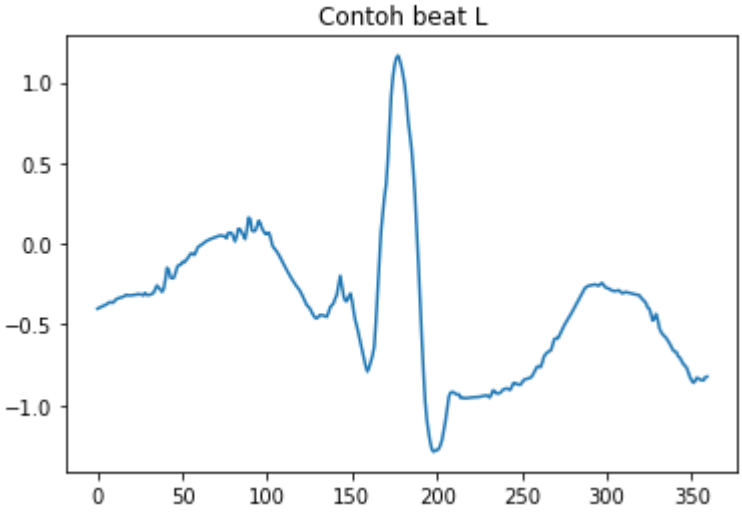


Gambar 4.4 Contoh Sinyal ECG dengan *class* N

Hasil plot diatas merupakan merupakan detak jantung Normalnya manusia yang digambarkan dengan satu gelombang tengah.

```
        if arrhythmia_type == 'L' and not
example_beat_L_printed:
            print("Contoh Beat")
            example_beat_L_printed = True
            plt.title("Contoh beat " + arrhythmia_type)
            plt.plot(beat)
            plt.show()
```

Kodingan diatas akan menampilkan plotting *beat* dengan *class* L apabila variabel *arrhythmia_type* bernilai ‘L’.



Gambar 4.5 Contoh Sinyal ECG dengan *class* L

Terlihat bahwa bentuk plot dari detak jantung dengan *class* L sangat berbeda dengan bentuk *beat* dengan *class* N.

```
        if arrhythmia_type == 'R' and not
example_beat_R_printed:
            print("Contoh Beat")
            example_beat_R_printed = True
            plt.title("Contoh beat " + arrhythmia_type)
            plt.plot(beat)
            plt.show()
```

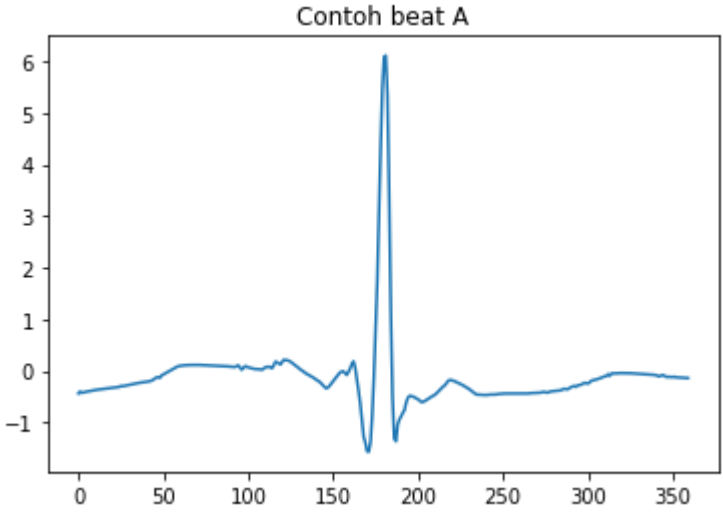
Kodingan diatas akan menampilkan plotting *beat* dengan *class* R apabila variabel *arrhythmia_type* bernilai ‘R’.

Gambar 4.6 Contoh Sinyal ECG dengan *class* R

Terlihat bahwa bentuk plot dari detak jantung dengan *class* L sangat tidar beraturan dan masih memiliki banyak *noise*.

```
        if arrhythmia_type == 'A' and not
example_beat_A_printed:
            print("Contoh Beat")
            example_beat_A_printed = True
            plt.title("Contoh beat " + arrhythmia_type)
            plt.plot(beat)
            plt.show()
```

Kodingan diatas akan menampilkan plotting *beat* dengan *class* A apabila variabel *arrhythmia_type* bernilai ‘A’.

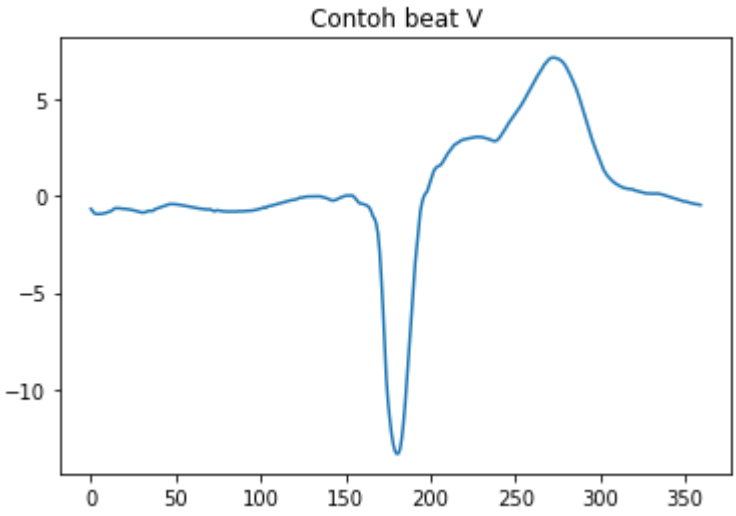


Gambar 4.7 Contoh Sinyal ECG dengan *class* A

Terdapat sedikit perbedaan bentuk plot dari detak jantung dari *class* N dan *class* A. Perbedaan terletak pada sisi kiri gelombang dengan adanya lekukan kebawah.

```
        if arrhythmia_type == 'V' and not
example_beat_V_printed:
            print("Contoh Beat")
            example_beat_V_printed = True
            plt.title("Contoh beat " + arrhythmia_type)
            plt.plot(beat)
            plt.show()
```

Kodingan diatas akan menampilkan plotting *beat* dengan *class* V apabila variabel *arrhythmia_type* bernilai ‘V’.



Gambar 4.8 Contoh Sinyal ECG dengan *class* V

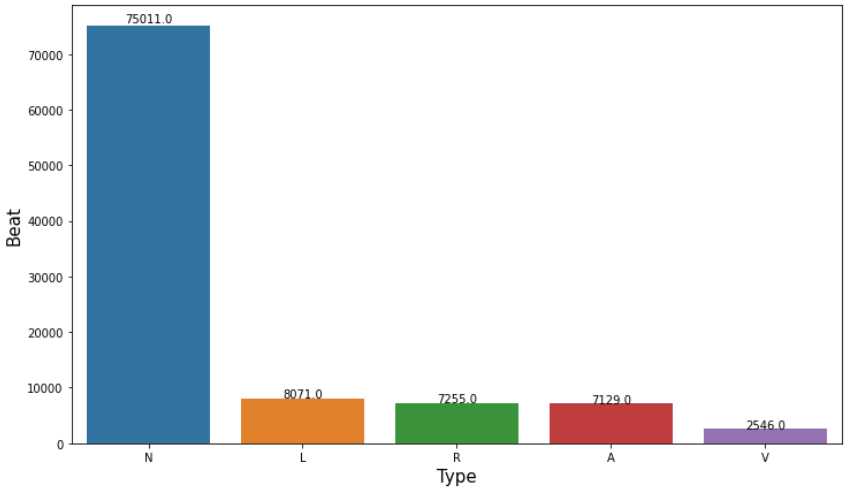
Perbedaan yang sangat menonjol adalah berbentuk gelombang terbalik dari bentuk beat dengan *class* N.

Kemudian, langkah selanjutnya adalah melakukan plotting diagram semua beat pada dataset dengan mengelompokan berdasarkan jumlah dari *class* yang akan digunakan.

```
per_class = X_train_df[X_train_df.shape[1] - 1].value_counts()
plt.figure(figsize=(12,7))
ax = sns.barplot(x=['N', 'L', 'R', 'A', 'V'], y=per_class)
plt.xlabel("Type", fontsize=15)
plt.ylabel("Beat", fontsize=15)
for p in ax.patches:
  width = p.get_width()
  height = p.get_height()
```

```
  x = p.get_x()
  y = p.get_y()
  ax.annotate(f"{height}", (x + width/2, y+ height*1.01),
ha="center")
plt.show()
```

Kemudian, langkah selanjutnya adalah melakukan plotting diagram semua beat pada sesuai dengan *class* nya masing-masing.



Gambar 4.9 Diagram jumlah *beat* masing-masing *class*

Pada diagram diatas terlihat bahwa *beat* dengan *class* N mencapai 75.011 *beat* sedangkan *class* lainnya memiliki jumlah *beat* kurang dari 10.000 *beat*. Oleh karena itu, data tersebut perlu dilakukan penyetaraan data sesuai dengan *class* nya atau Rebalancing Dataset.

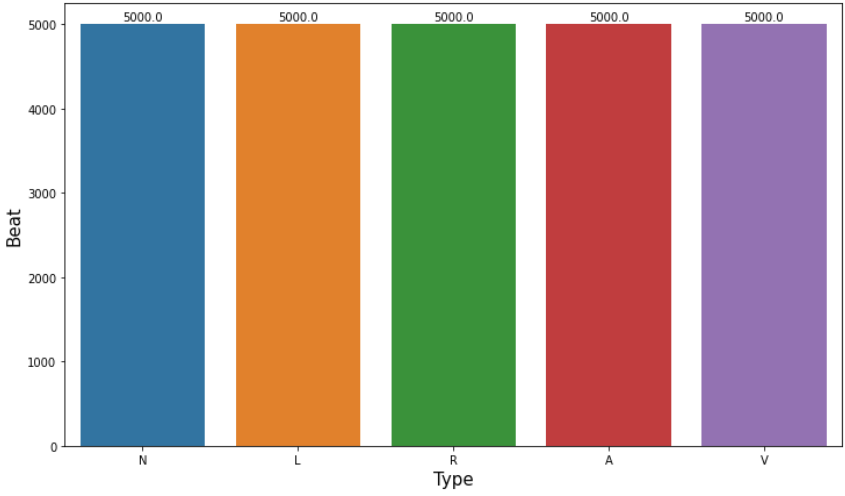
4.2.6. Rebalancing Dataset

Tahap Rebalancing Dataset merupakan proses penyeimbangan data sesuai dengan *class* masing-masing. Rasio data yang tidak seimbang sesuai dengan *class* perlu dilakukan rebalancing data agar data yang dilakukan *train* nanti menjadi proporsional.

```
def rebalancing_dataframe(X_train_df):
 df_0 = (X_train_df[X_train_df[X_train_df.shape[1]-1] == 0])
 .sample(n = 5000, random_state = 42)

 df_1 = X_train_df[X_train_df[X_train_df.shape[1]-1] == 1]
 df_2 = X_train_df[X_train_df[X_train_df.shape[1]-1] == 2]
 df_3 = X_train_df[X_train_df[X_train_df.shape[1]-1] == 3]
 df_4 = X_train_df[X_train_df[X_train_df.shape[1]-1] == 4]

 df_1_upsample = resample(df_1, replace = True, n_samples =
5000,
 random_state = 122)
 df_2_upsample = resample(df_2, replace = True, n_samples =
5000,
 random_state = 123)
 df_3_upsample = resample(df_3, replace = True, n_samples =
5000,
 random_state = 124)
 df_4_upsample = resample(df_4, replace = True, n_samples =
5000,
 random_state = 125)

 X_train_df = pd.concat([df_0, df_1_upsample,
df_2_upsample,
```

```
df_3_upsample, df_4_upsample])

return X_train_df
```

Dari kodingan di atas, tahapan rebalancing ini mengambil 5.000 sampel atau *beat* masing-masing *class* nya. Sehingga model mempelajari data yang seimbang dengan masing-masing *class* tersebut. Setelah dilakukan rebalancing, maka bentuk diagram dari dataset tersebut akan seimbang. Untuk mengetahui apakah data sudah seimbang atau tidak, perlu dilakukan plotting untuk mengetahui bentuk diagramnya.



Gambar 4.10 Diagram dataset setelah di Rebalancing

Terlihat dari diagram tersebut sudah memiliki masing-masing *beat* sebanyak 5.000 *beat* dengan jumlah *class*

yang sama. Sehingga dataset tersebut sudah bisa dilakukan split untuk dilakukan *train* dan *test* pada tahap pemodelan.

4.2.7. Split Dataset

Pada tahap ini akan membagi 2 dataset untuk dilakukan *train* dan *test* untuk dilakukan pemodelan.

```
X = X_train_df.loc[:, :359]
Y = X_train_df[360]
x_train, x_test, y_train, y_test = train_test_split(X, Y,
test_size=0.3, random_state=1)

print("X_train : ", np.shape(x_train))
print("X_test  : ", np.shape(x_test))
```

Output:

```
X_train :  (17500, 360)
X_test  :  (7500, 360)
```

Setelah dilakukan split dataset, maka akan ada data *train* dan data *test.* Data *train* berfungsi untuk model melakukan pembelajaran terhadap data yang telah disiapkan. Setelah model melakukan pembelajaran terhadap data tersebut, maka model tersebut akan melakukan proses pengecekan terhadap data *test* yang sudah mempunyai hasil atau outputnya untuk divalidasi apakah hasil klasifikasi atau output dari model tersebut tepat atau tidak.

4.3. Pemodelan

Tahap modeling merupakan tahap yang paling penting dari sebuah Machine Learning dikarenakan tahap modeling akan menghasilkan sebuah model yang memiliki output yang dapat mengklasifikasi sebuah masalah atau objek serta memiliki akurasi dalam pembuatannya.

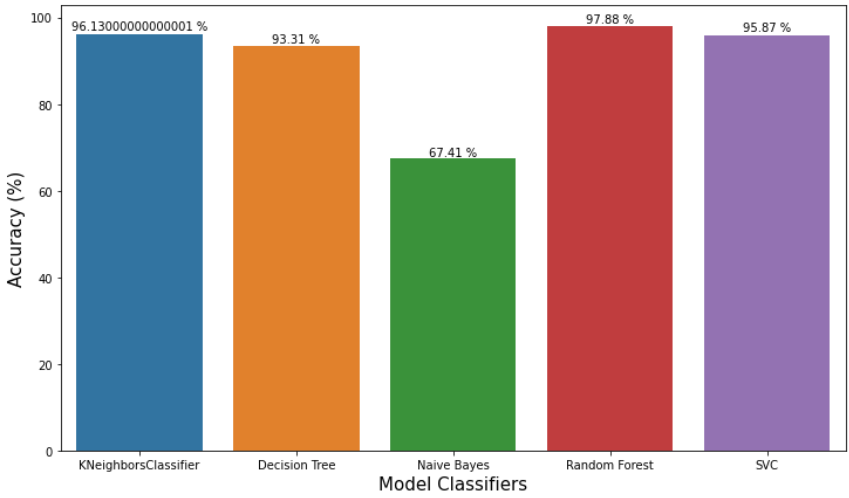
```
rf_model = RandomForestClassifier()
rf_model.fit(x_train, y_train)
rf_prediction = rf_model.predict(x_test)
rf_accuracy = (round(accuracy_score(rf_prediction, y_test), 4) * 100)
accuracy_list.append(rf_accuracy)
print("Accuracy (%) : ", rf_accuracy)
```

Output:
Accuracy (%) : 97.88%

Dari hasil modeling tersebut mendapatkan akurasi yang tinggi yaitu sebesar 97.88%. Akurasi tersebut sudah termasuk akurasi yang tinggi dikarenakan tidak termasuk dalam model yang *overfitting* maupun *underfitting*.

4.4. Evaluasi Model

Untuk memastikan model yang dibuat adalah model yang memiliki akurasi yang paling tinggi, maka perlu dilakukan evaluasi model. Pada evaluasi ini akan membandingkan hasil klasifikasi dari beberapa model seperti K-Neighbors Classifier, Decision Tree, Naive Bayes, dan Support Vector Machine.



Gambar 4.11 Perbandingan Akurasi Model

Setelah dilakukan plot terlihat pada gambar diatas bahwa Random Forest memiliki akurasi yang paling tinggi dengan akurasi 97.88% diantara model yang ada dan diikuti dengan model Naive Bayes yang memiliki akurasi yang paling rendah dengan akurasi 67.41%. Berikut penjelasan dari karakteristik dari masing-masing model yang dilakukan komparasi:

4.4.1. KNeighborsClassifier

KNN identik dengan menghitung jarak masing-masing kelas yang akan dilakukan klasifikasi. Nilai jarak yang dihitung merupakan hasil kemiripan dari data lama (K) yang terdekat. Nilai K pada algoritma ini dijadikan parameter pada pemodelan untuk mencari nilai K terbaik. Sehingga algoritma ini akan langsung

mengklasifikasikan berdasarkan data (K) yang terdekat dengan hasil nilai perhitungan tersebut.

4.4.2. Decision Tree

Karakteristik dari Decision Tree adalah dengan memprediksi suatu masalah dengan membentuk suatu pohon keputusan dengan memecah kedalam himpunan yang lebih kecil dan secara bertahap akan terus dilakukan pengembangan dalam pengambilan keputusannya. Node keputusan dan node daun merupakan hasil akhir dari algoritma ini. Parameter *max_deph* sangat mempengaruhi hasil klasifikasi karena model tersebut akan mengambil keputusan berdasarkan nilai dari *max_deph* tersebut.

4.4.3. Naive Bayes

Pada model ini akan menghitung nilai probabilitas dari suatu data atau objek yang akan dilakukan klasifikasi dan sangat cocok untuk diterapkan pada data yang bernilai biner.

4.4.4. Random Forest

Random Forest merupakan salah satu metode dari Decision Tree sehingga tidak heran kalau alur dari algoritma ini memiliki karakteristik yang hampir sama. Tetapi yang membedakannya adalah Random Forest itu sendiri merupakan kombinasi dari beberapa Tree atau pohon yang dijadikan sebuah model.

4.4.5. SVC

Model Support Vector Classification memiliki karakteristik dengan mencari hyperplane terbaik dengan cara mencari jarak diantara kedua kelas tersebut. Fungsi dari hyperplane itu sendiri adalah dengan membagi 2 kelas.

BAB 5
KESIMPULAN

5.1. Kesimpulan

Dari model yang telah dilakukan komparasi, dapat diambil kesimpulan bahwa setiap model memiliki karakteristik dan algoritmanya masing-masing sehingga hal tersebut dapat mempengaruhi hasil akurasi dari masing-masing model tersebut. Berdasarkan hasil pemodelan yang dilakukan, didapatkan model dengan akurasi yang paling tinggi yaitu model Random Forest dengan nilai akurasi 97.88% dan di posisi paling rendah yaitu model Naive Bayes dengan nilai akurasi 67.41%.

5.2. Saran

Saran dari penelitian ini adalah dengan melakukan pengembangan terhadap model dan juga metode penelitian yang akan digunakan dimasa yang akan datang menambahkan proses ekstraksi nilai RR dan jarak antar sinyal atau *beat.* Adapun beberapa tujuannya adalah sebagai berikut:

A. Mendapatkan model dan metode penelitian yang lebih baik dan cocok.

B. Mendapatkan nilai akurasi yang lebih tinggi dari sebelumnya.

DAFTAR PUSTAKA

[1] Wasimuddin, M., Elleithy, K., Abuzneid, A. S., Faezipour, M., & Abuzaghleh, O. (2020). Stages-based ECG signal analysis from traditional signal processing to machine learning approaches: A survey. *IEEE Access*, *8*, 177782-177803.

[2] Pojon, M. (2017). *Using machine learning to predict student performance* (Master's thesis).

[3] Kuila, S., Dhanda, N., & Joardar, S. (2020). Feature Extraction and Classification of MIT-BIH Arrhythmia Database. In Proceedings of the 2nd International Conference on Communication, Devices and Computing (pp. 417-427). Springer, Singapore.

[4] Apandi, Z. F. M., Ikeura, R., & Hayakawa, S. (2018, August). Arrhythmia detection using MIT-BIH dataset: A review. In *2018 International Conference on Computational Approach in Smart Systems Design and Applications (ICASSDA)* (pp. 1-5). IEEE.

[5] Li, T., & Zhou, M. (2016). ECG classification using wavelet packet entropy and random forests. *Entropy*, *18*(8), 285.

[6] Kumar, R. G., & Kumaraswamy, Y. S. (2012). Investigating cardiac arrhythmia in ECG using random forest classification. *International Journal of Computer Applications*, *37*(4), 31-34.

[7] Wasimuddin, M., Elleithy, K., Abuzneid, A. S., Faezipour, M., &

Abuzaghleh, O. (2020). Stages-based ECG signal analysis from traditional signal processing to machine learning approaches: A survey. *IEEE Access*, *8*, 177782-177803.

[8] Kamila, N. K., Frnda, J., Pani, S. K., Das, R., Islam, S. M., Bharti, P. K., & Muduli, K. (2022). Machine learning model design for high performance cloud computing & load balancing resiliency: An innovative approach. *Journal of King Saud University-Computer and Information Sciences*.

[9] Renggli, C., Ashkboos, S., Aghagolzadeh, M., Alistarh, D., & Hoefler, T. (2019, November). Sparcml: High-performance sparse communication for machine learning. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis* (pp. 1-15).

[10] Bhatt, H., Shah, V., Shah, K., Shah, R., & Shah, M. (2022). State-of-the-art machine learning techniques for melanoma skin cancer detection and classification: a comprehensive review. *Intelligent Medicine*.

[11] Khanzadeh, M., Chowdhury, S., Marufuzzaman, M., Tschopp, M. A., & Bian, L. (2018). Porosity prediction: Supervised-learning of thermal history for direct laser deposition. *Journal of manufacturing systems*, *47*, 69-82.

[12] Khanzadeh, M., Chowdhury, S., Marufuzzaman, M., Tschopp, M. A., & Bian, L. (2018). Porosity prediction: Supervised-learning of thermal history for direct laser deposition. *Journal of manufacturing*

systems, *47*, 69-82.

[13] Jalal, N., Mehmood, A., Choi, G. S., & Ashraf, I. (2022). A novel improved random forest for text classification using feature ranking and optimal number of trees. *Journal of King Saud University-Computer and Information Sciences*.

[14] Makariou, D., Barrieu, P., & Chen, Y. (2021). A random forest based approach for predicting spreads in the primary catastrophe bond market. *Insurance: Mathematics and Economics*, *101*, 140-162.

[15] Alizadeh, S. H., Hediehloo, A., & Harzevili, N. S. (2021). Multi independent latent component extension of naive bayes classifier. *Knowledge-Based Systems*, *213*, 106646.

[16] Blanquero, R., Carrizosa, E., Ramírez-Cobo, P., & Sillero-Denamiel, M. R. (2021). Variable selection for Naïve Bayes classification. *Computers & Operations Research*, *135*, 105456.

[17] Chen, S., Webb, G. I., Liu, L., & Ma, X. (2020). A novel selective naïve Bayes algorithm. *Knowledge-Based Systems*, *192*, 105361.

[18] Shaban, W. M., Rabie, A. H., Saleh, A. I., & Abo-Elsoud, M. A. (2021). Accurate detection of COVID-19 patients based on distance biased Naïve Bayes (DBNB) classification strategy. *Pattern Recognition*, *119*, 108110.

[19] Andrejiova, M., & Grincova, A. (2018). Classification of impact damage on a rubber-textile conveyor belt using Naïve-Bayes methodology. *Wear*, *414*, 59-67.

[20] Zamri, N., Pairan, M. A., Azman, W. N. A. W., Abas, S. S., Abdullah, L., Naim, S., ... & Gao, M. (2022). River quality classification using different distances in k-nearest neighbors algorithm. *Procedia Computer Science*, *204*, 180-186.

[21] Cubillos, M., Wøhlk, S., & Wulff, J. N. (2022). A bi-objective k-nearest-neighbors-based imputation method for multilevel data. *Expert Systems with Applications*, 117298.

[22] Kumar, B., & Gupta, D. (2021). Universum based Lagrangian twin bounded support vector machine to classify EEG signals. *Computer Methods and Programs in Biomedicine*, *208*, 106244.

[23] Zhang, H., Shi, Y., Yang, X., & Zhou, R. (2021). A firefly algorithm modified support vector machine for the credit risk assessment of supply chain finance. *Research in International Business and Finance*, *58*, 101482.

[24] Jafari-Marandi, R. (2021). Supervised or unsupervised learning? Investigating the role of pattern recognition assumptions in the success of binary predictive prescriptions. *Neurocomputing*, *434*, 165-193.

[25] Solli, R., Bazin, D., Hjorth-Jensen, M., Kuchera, M. P., & Strauss, R. R. (2021). Unsupervised learning for identifying events in active target experiments. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, *1010*, 165461.

[26] Zhao, D., Hu, X., Xiong, S., Tian, J., Xiang, J., Zhou, J., & Li, H.

(2021). K-means clustering and kNN classification based on negative databases. *Applied Soft Computing*, *110*, 107732.

[27] Wang, X., Wang, Z., Sheng, M., Li, Q., & Sheng, W. (2021). An adaptive and opposite K-means operation based memetic algorithm for data clustering. *Neurocomputing*, *437*, 131-142.

[28] Zhang, Z., Feng, Q., Huang, J., Guo, Y., Xu, J., & Wang, J. (2021). A local search algorithm for k-means with outliers. *Neurocomputing*, *450*, 230-241.

TENTANG PENULIS

M. RIZKY, lahir di Kabupaten Dompu pada tanggal 17 April 2000. Pendidikan tingkat dasar hingga menengah dan atas ditempuh di Kabupaten Dompu. Lulus D4 di Program Studi Teknik Informatika Politeknik Pos Indonesia (sekarang Universitas Logistik dan Bisnis Internasional).

Roni Andarsyah, S.T., M.Kom., SFPC. Lulus D3 di Program Studi Teknik Informatika Politeknik Pos Indonesia (sekarang Universitas Logistik dan Bisnis Internasional), lulus S1 di Program Studi Teknik Informatika ST Inten Bandung, dan lulus S2 di Teknik Informatika STMIK LIKMI. Saat ini telah menjadi dosen D4 Teknik Informatika di ULBI dan menjabat sebagai Ketua Program Studi Sarjana Terapan Informatika.

DI ERA MODERN INI, BANYAK SEKALI KEGIATAN – KEGIATAN OPERASIONAL MAUPUN KEGIATAN LAINNYA YANG MELIBATKAN ATAU MENGGUNAKAN ARTIFICIAL INTELLIGENCE (AI). TIDAK BISA DIPUNGKIRI LAGI BAHWA TEKNOLOGI YANG BERKEMBANG PESAT SEPERTI SEKARANG INI TENTU DIBUAT UNTUK MEMPERMUDAH PEKERJAAN MANUSIA BAHKAN MENGGANTIKAN PERAN MANUSIA. SALAH SATU BAGIAN DARI ARTIFICIAL INTELLIGENCE (AI) ADALAH MACHINE LEARNING. MACHINE LEARNING SEBAGAI METODE DALAM SISTEM KECERDASAN BUATAN YANG MAMPU MENGKLASIFIKASIKAN DATA YANG DIMASUKKAN UNTUK KEPERLUAN DAN KEBUTUHAN MASING - MASING. BANYAK APLIKASI YANG MENERAPKAN MACHINE LEARNING UNTUK KEPERLUAN KLASIFIKASI DATA, MEMPREDIKSI HUBUNGAN ANTAR DATA, MEMBACA POLA DATA, DAN BANYAK IMPLEMENTASI LAINNYA. DALAM LAPORAN INI AKAN DIBAHAS PENGGUNAAN KELUARAN STRUKTUR DARI MACHINE LEARNING DIGUNAKAN UNTUK MENDETEKSI SINYAL ECG APAKAH SUDAH SESUAI DENGAN DATA YANG SUDAH DI TETAPKAN PADA ANOTASI. BERDASARKAN HASIL PENELITIAN MENUNJUKKAN PEMODELAN STRUKTUR DARI MACHINE LEARNING DAPAT DIGUNAKAN UNTUK MENDETEKSI KESESUAIAN DARI SINYAL BAIK PEMODELAN SECARA TERPISAH ATAUPUN PEMODELAN SECARA GABUNGAN.