

Studi Algoritma Deep Learning Untuk Mendeteksi Aritmia Berdasarkan Sinyal Elektrokardiogram

Tugas Akhir

Kelompok Keahlian: Telematics

Nariswari Yoga Nandita
NIM: 1301194202



Program Studi Sarjana Informatika
Fakultas Informatika
Universitas Telkom
Bandung
2023

Lembar Pengesahan

Studi Algoritma Deep Learning Untuk Mendeteksi Aritmia
Berdasarkan Sinyal Elektrokardiogram

*Study of Deep Learning Algorithm for Arrhythmia Detection based
on Electrocardiogram Signals*

Nariswari Yoga Nandita
NIM: 1301194202

Tugas Akhir ini diterima dan disahkan untuk memenuhi sebagian dari syarat
untuk memperoleh gelar sarjana Sarjana Komputer
Program Studi Sarjana Informatika
Fakultas Informatika Universitas Telkom

Bandung, 9 September 2023
Menyetujui,

Pembimbing 1



Satria Mandala, S. T., M.Sc., Ph.D.

NIP: 16730040

Mengesahkan,
Kepala Program Studi Informatika



Dr. Erwin Budi Setiawan, S.Si., M.T
NIP: 00760045

Abstrak

Aritmia jantung adalah salah satu penyakit kardiovaskular yang dikarenakan ketidakaturan dalam ritme jantung, yang menyebabkan jantung berdetak terlalu cepat atau terlalu lambat. Aritmia sering dianggap penyakit ringan tetapi dapat mempengaruhi kinerja fungsi jantung secara keseluruhan dan menyebabkan penyakit yang lebih parah, bahkan dapat menyebabkan kematian mendadak. Pada umumnya deteksi aritmia terdiri dari tiga tahapan, yaitu *pre-processing*, ekstraksi ciri, dan klasifikasi. Metode deteksi aritmia saat ini belum berhasil menunjukkan peningkatan yang signifikan meskipun telah dilakukan berbagai percobaan dengan berbagai teknik, oleh karena itu, perlu untuk meningkatkan identifikasi aritmia, dari sekian banyak literatur yang mengusulkan metode deteksi aritmia, algoritma *deep learning* masih memberikan akurasi deteksi rendah. Oleh karena itu, dibutuhkan metode yang optimal untuk meningkatkan nilai akurasi deteksi aritmia. Disisi lain, pengembangan model algoritma *deep learning* jarang dilakukan. Untuk menyelesaikan masalah-masalah di atas, tugas akhir ini mengusulkan pengembangan algoritma *deep learning* yang mendukung peningkatan akurasi deteksi aritmia. *Deep learning* adalah metode yang paling menjanjikan untuk deteksi aritmia dengan menggunakan fitur yang telah diekstraksi. Dan tugas akhir ini juga melakukan analisis terhadap model algoritma yang diusulkan. Metode yang digunakan dalam penelitian tugas akhir ini adalah 1. Studi literatur tentang klasifikasi aritmia, 2. Perancangan dan pengembangan algoritma klasifikasi, 3. Pengujian performansi dan analisis algoritma *deep learning*. Hasilnya menunjukkan, metode yang diusulkan mencapai akurasi 98,49% pada *Convolutional Neural Network*, akurasi 95,36% pada *Long Short Term Memory*, dan 96,07% pada *Gated Recurrent Unit*.

Kata Kunci: Aritmia, *deep learning*, elektrokardiogram, ekstraksi fitur

Abstract

Cardiac arrhythmia is one of the cardiovascular diseases caused by irregularities in the heart rhythm, leading to the heart beating too fast or too slow. Arrhythmia is often considered a mild condition, but it can affect the overall performance of the heart and lead to more severe illnesses, including sudden death. Generally, arrhythmia detection involves three stages: pre-processing, feature extraction, and classification. Current arrhythmia detection methods have failed to demonstrate significant improvement despite multiple attempts using various techniques. Therefore, there is a need to enhance arrhythmia identification. Among the many proposed methods for arrhythmia detection, deep learning algorithms still provide low detection accuracy. Hence, an optimal method is required to improve the accuracy of arrhythmia detection. Furthermore, the development of deep learning algorithm models is rarely conducted. To address the aforementioned issues, this final project proposes the development of a deep learning algorithm to support improved arrhythmia detection accuracy. Deep learning is the most promising method for arrhythmia detection using extracted features. This final project also includes an analysis of the proposed algorithm models. The methods employed in this research are as follows: 1. Literature review on arrhythmia classification. 2. Design and development of classification algorithms. 3. Performance testing and analysis of deep learning algorithms. The results demonstrate the proposed method achieving 98.49% accuracy on Convolutional Neural Network, 95.36% accuracy on Long Short Term Memory, and 96.07% on Gated Recurrent Unit.

Keywords: Arrhythmia, deep learning, electrocardiogram, feature extraction

Lembar Persembahan

Puji dan syukur penulis panjatkan kepada Allah SWT, yang telah memberikan rahmat, hidayah, dan karunia-Nya sehingga penulis dapat menyelesaikan penulisan skripsi ini dengan judul "Studi Algoritma Deep Learning Untuk Mendeteksi Aritmia Berdasarkan Sinyal Elektrokardiogram". Segala puji hanya milik Allah, Tuhan semesta alam.

Tugas Akhir ini tidak akan selesai tanpa dukungan dan bantuan dari berbagai pihak. Oleh karena itu, dengan tulus dan ikhlas penulis sampaikan rasa terima kasih kepada:

1. Abah, Ibu dan Adik saya Adinda, terima kasih atas doa, dukungan moral, dan cinta kasih yang selalu diberikan. Semua jerih payah dan pengorbanan mereka adalah sumber motivasi penulis dalam menyelesaikan Tugas Akhir ini.
2. Terima kasih kepada Bapak Satria Mandala, S. T., M.Sc., Ph.D. sebagai pembimbing utama, yang telah memberikan bimbingan, arahan, serta masukan berharga selama penulisan Tugas Akhir ini.
3. Terima kasih kepada teman-teman Bu Garma yang telah memberikan dukungan, semangat, serta berbagi pengalaman selama studi di perguruan tinggi ini.
4. Untuk keluarga besar dan semua pihak yang tidak dapat disebutkan satu per satu, penulis mengucapkan rasa terima kasih yang dalam atas segala doa, dukungan, dan semangat yang telah diberikan selama perjalanan pengerjaan Tugas Akhir ini. Kebersamaan dalam menghadapi berbagai tantangan adalah pendorong utama dalam mencapai titik ini. Semua pengorbanan dan dorongan batin yang diberikan tidak akan pernah terlupakan. Terima kasih atas semua yang telah dilakukan.

Akhir kata, semoga hasil penelitian ini bermanfaat dan dapat memberikan sumbangsih dalam pengembangan ilmu pengetahuan. Penulis menyadari bahwa masih banyak keterbatasan dalam penelitian ini, oleh karena itu masukan dan kritik yang membangun sangat penulis harapkan untuk perbaikan di masa yang akan datang.

Kata Pengantar

Puji syukur penulis panjatkan kepada Allah SWT atas segala rahmat, nikmat, dan hidayah-Nya, yang telah melimpahkan berkah-Nya sehingga penulis dapat menyelesaikan penyusunan tugas akhir ini dengan judul "Sudi Algoritma Deep Learning Untuk Mendeteksi Aritmia Berdasarkan Sinyal Elektrokardiogram". Penulis menyadari bahwa perjalanan penulisan tugas akhir ini tidaklah mudah, dan atas berkat-Nya segala hal menjadi mungkin. Tugas akhir ini merupakan hasil perjalanan panjang dalam mengeksplorasi dan memahami topik yang telah penulis pilih. Dalam proses ini, penulis mendapatkan banyak pelajaran dan pengalaman berharga yang akan membawa dampak positif dalam perjalanan akademis dan profesional penulis di masa depan. Penulis ingin mengucapkan terima kasih kepada Bapak Satria Mandala, S. T., M.Sc., Ph.D. selaku pembimbing pendamping atas bimbingan, arahan, dan masukan yang berharga dalam menyusun tugas akhir ini. Dukungan dan pengarahan dari kedua pembimbing sangatlah berarti bagi perbaikan dan pengembangan tugas akhir ini. Tidak lupa, penulis juga ingin menyampaikan apresiasi kepada rekan-rekan seangkatan yang telah memberikan dukungan, inspirasi, dan kolaborasi selama proses perkuliahan. Pengalaman bersama kalian telah memberikan warna tersendiri dalam perjalanan akademis ini. Penulis juga ingin mengucapkan terima kasih kepada teman-teman yang telah membantu dalam berbagai tahap penelitian, baik dalam pengumpulan data, analisis, maupun diskusi konstruktif yang mendorong perbaikan tugas akhir ini.

Akhir kata, penulis menyadari bahwa tugas akhir ini jauh dari kesempurnaan. Oleh karena itu, masukan, kritik, dan saran yang membangun sangat penulis harapkan untuk perbaikan dan pengembangan penelitian di masa yang akan datang. Semoga tugas akhir ini dapat memberikan manfaat dan kontribusi yang bermanfaat dalam pengembangan ilmu pengetahuan dan aplikasinya di dunia nyata.

Bandung, 28 Agustus 2023



Nariswari Yoga Nandita

Daftar Isi

Lembar-Persetujuan	i
Abstrak	ii
Abstract	iii
Lembar Persembahan	iv
Kata Pengantar	v
Daftar Isi	vi
Daftar Gambar	viii
Daftar Tabel	x
I Pendahuluan	1
1.1 Latar Belakang	1
1.2 Perumusan Masalah	2
1.3 Pernyataan Masalah	3
1.4 Tujuan	3
1.5 Batasan Masalah	3
1.6 Hipotesis	3
1.7 Sistematika Penulisan	4
II Kajian Pustaka	5
2.1 Penelitian Terkait	5
2.2 Deep Learning	16
2.2.1 Convolutional Neural Network	17
2.2.2 Long Short-Term Memory	17
2.2.3 Gated Recurent Unit	18
2.3 Aritmia	19
2.4 Elektrokardiogram (EKG)	20
2.5 Fitur Sinyal Elektrokardiogram	21

III Metodologi	23
3.1 Metode Penelitian	23
3.1.1 Framework Penelitian	23
3.1.2 Metodologi untuk Mencapai Tujuan Penelitian	26
3.1.3 Analisis Kebutuhan Sistem	29
3.1.4 Dataset	29
3.1.5 Default Parameter	30
3.1.6 Hyperparameter Tuning	30
3.1.7 Metrik Uji	31
3.1.8 Metode Pengujian	32
3.1.9 Perbandingan Hasil Penelitian	34
3.2 Ringkasan	34
IV Hasil dan Pembahasan	35
4.1 Hasil Objektif Pertama	35
4.2 Hasil Objektif Kedua	35
4.2.1 Preprocessing	35
4.2.2 Ekstraksi Fitur	37
4.2.3 Hasil Algoritma Klasifikasi	37
4.3 Hasil Objektif Ketiga	61
4.4 Pembahasan	65
4.4.1 Perbandingan Hasil Model Default dan Hasil Model Hyperparameter Tuning	65
4.4.2 Penjelasan Hasil Hyperparameter Tuning	67
4.4.3 Perbandingan Hasil Akurasi dengan Penelitian Sebelumnya	68
4.5 Ringkasan	69
V Kesimpulan dan Saran	70
5.1 Kesimpulan	70
5.2 Saran	71
Daftar Pustaka	72

Daftar Gambar

2.1	Taksonomi <i>Deep Learning</i>	16
2.2	Flowchart model CNN	17
2.3	Flowchart model LSTM	18
2.4	Flowchart model GRU	19
2.5	Gelombang Elektrokardiogram (EKG)	20
2.6	Contoh perbedaan fitur dinamis dan morfologi	22
2.7	<i>previous RR, post RR, local RR, dan average RR.</i>	22
3.1	Diagram Alir penelitian <i>Framework</i>	24
3.2	Diagram Alir Metodologi Objektif Pertama	26
3.3	Diagram Alir Metodologi Objektif Kedua	27
3.4	Diagram Alir Metodologi Objektif Ketiga	28
4.1	Contoh sinyal EKG sebelum pra-pemrosesan yang menampilkan sinyal ECG mentah.	36
4.2	Contoh sinyal EKG setelah pra-pemrosesan yang menggambarkan sinyal ECG setelah noise dihilangkan.	36
4.3	Sinyal EKG setelah normalisasi Z-Score.	37
4.4	Grafik <i>accuracy</i> dan <i>loss</i> model <i>convolutional neural network</i> selama <i>training</i> dan <i>validation</i>	38
4.5	Fold 5 dari <i>confusion matrix</i> model <i>convolutional neural network</i>	39
4.6	Grafik metrik performansi model <i>convolutional neural network</i> . .	40
4.7	Confusion matrix dari model <i>convolutional neural network</i> . . .	41
4.8	Grafik <i>accuracy</i> dan <i>loss</i> model selama <i>training</i> dan <i>validation</i> setelah <i>hyperparameter tuning</i>	42
4.9	Fold 5 dari <i>confusion matrix</i> model CNN setelah <i>hyperparameter</i> <i>tuning</i>	43
4.10	Grafik metrik performansi model <i>convolutional neural network</i> setelah <i>hyperparameter tuning</i>	44
4.11	<i>Confusion Matrix</i> model CNN setelah <i>hyperparameter tuning</i> . .	45
4.12	Grafik <i>accuracy</i> dan <i>loss</i> model <i>long short term memory</i> selama <i>training</i> dan <i>validation</i>	46
4.13	Fold 5 <i>confusion matrix</i> model <i>long short term memory</i>	47
4.14	Grafik metrik performansi model <i>long short term memory</i>	48

4.15	<i>Confusion matrix</i> model <i>long short term memory</i>	49
4.16	Grafik <i>accuracy</i> dan <i>loss</i> model <i>long short term memory</i> selama <i>training</i> dan <i>validation</i> setelah <i>hyperparameter tuning</i>	50
4.17	Fold 5 dari <i>Confusion Matrix</i> LSTM setelah <i>hyperparameter tuning</i>	51
4.18	Grafik metrik performansi model <i>long short term memory</i> setelah <i>hyperparameter tuning</i>	52
4.19	<i>Confusion matrix</i> LSTM setelah <i>hyperparameter tuning</i>	53
4.20	Grafik <i>accuracy</i> dan <i>loss</i> model <i>gated recurrent unit</i> selama <i>training</i> dan <i>validation</i>	54
4.21	Fold 5 dari <i>confusion matrix</i> model <i>gated recurrent unit</i>	55
4.22	Grafik metrik performansi model <i>gated recurrent unit</i>	56
4.23	<i>Confusion matrix</i> model <i>gated recurrent unit</i>	57
4.24	Grafik <i>accuracy</i> dan <i>loss</i> model <i>gated recurrent unit</i> selama <i>training</i> dan <i>validation</i> setelah <i>hyperparameter tuning</i>	58
4.25	Fold 2 dari <i>confusion matrix</i> model GRU setelah <i>hyperparameter tuning</i>	59
4.26	Grafik metrik performansi model <i>gated recurrent unit</i> setelah <i>hyperparameter tuning</i>	60
4.27	<i>Confusion matrix</i> GRU setelah <i>hyperparameter tuning</i>	61
4.28	Grafik perbandingan metrik performansi model <i>convolutional neural network</i> sebelum dan setelah <i>hyperparameter tuning</i>	62
4.29	Grafik perbandingan metrik performansi model <i>long short term memory</i> sebelum dan setelah <i>hyperparameter tuning</i>	63
4.30	Grafik perbandingan metrik performansi model <i>gated recurrent unit</i> sebelum dan setelah <i>hyperparameter tuning</i>	64

Daftar Tabel

2.1	Ringkasan riset terkait	9
2.1	Ringkasan riset terkait	10
2.1	Ringkasan riset terkait	11
2.1	Ringkasan riset terkait	12
2.1	Ringkasan riset terkait	13
2.1	Ringkasan riset terkait	14
2.1	Ringkasan riset terkait	15
3.1	<i>Nilai Parameter</i>	30
3.2	<i>Hyperparameter Tuning</i>	31
3.3	<i>Algoritma Deep Learning</i>	32
4.1	Metrik performansi <i>Convolutional Neural Network</i> (%).	39
4.2	Metrik performansi <i>Convolutional Neural Network</i> untuk klasifikasi aritmia berdasarkan kelas (%).	40
4.5	Metrik performansi <i>convolutional neural network</i> setelah <i>hyperparameter tuning</i> (%).	43
4.6	Metrik performansi <i>convolutional neural network</i> untuk klasifikasi aritmia berdasarkan kelas setelah <i>hyperparameter tuning</i> (%).	44
4.9	Metrik performansi <i>long short term memory</i> (%).	47
4.10	Metrik performansi <i>long short term memory</i> untuk klasifikasi aritmia berdasarkan kelas (%).	48
4.13	Metrik performansi <i>long short term memory</i> setelah <i>hyperparameter tuning</i> (%).	52
4.14	Metrik performansi <i>long short term memory</i> untuk klasifikasi aritmia berdasarkan kelas setelah <i>hyperparameter tuning</i> (%).	52
4.17	Metrik performansi <i>gated recurrent unit</i> (%).	55
4.18	Metrik performansi <i>gated recurrent unit</i> untuk klasifikasi aritmia berdasarkan kelas (%).	56
4.21	Metrik performansi <i>gated recurrent unit</i> setelah <i>hyperparameter tuning</i> (%).	59
4.22	Metrik performansi <i>gated recurrent unit</i> untuk klasifikasi aritmia berdasarkan kelas setelah <i>hyperparameter tuning</i> (%).	60

4.23	Perbandingan metrik performansi <i>convolutional neural network</i> untuk klasifikasi aritmia sebelum dan setelah <i>hyperparameter tuning</i> (%).	62
4.24	Perbandingan metrik performansi <i>long short term memory</i> untuk klasifikasi aritmia sebelum dan setelah <i>hyperparameter tuning</i> (%).	63
4.25	Perbandingan metrik performansi <i>gated recurrent unit</i> untuk klasifikasi aritmia sebelum dan setelah <i>hyperparameter tuning</i> (%).	64
4.31	Perbandingan state-of-the-art menggunakan database yang sama (MIT-BIH Database)	68

Bab I

Pendahuluan

1.1 Latar Belakang

Menurut World Health Organization, penyakit kardiovaskular (CVD) seperti miokardia dan aritmia adalah penyebab utama kematian di seluruh dunia, terhitung lebih dari 17.9 juta kematian setiap tahunnya, mewakili 32% kematian global [43]. Aritmia jantung adalah salah satu penyakit kardiovaskular.

Aritmia adalah kelainan pada jantung karena ketidakaturan dalam detak jantung, yang menyebabkannya berdetak terlalu cepat atau terlalu lambat, hal ini disebabkan oleh konduksi intrakardiak yang buruk atau ritme jantung yang salah. Aritmia sering dianggap penyakit ringan tetapi dapat mempengaruhi kinerja fungsi jantung secara keseluruhan dan menyebabkan penyakit yang lebih parah, bahkan dapat menyebabkan kematian mendadak. Aritmia dapat terjadi secara intermiten, terutama pada tahap awal masalah, dengan demikian sulit untuk mendeteksi aritmia dalam waktu singkat, membutuhkan pemantauan pasien secara rutin [7]. Aritmia jantung memiliki jenis yang cukup beragam, diantaranya; *Atrial fibrillation* (AFIB), adalah irama jantung yang tidak teratur dan seringkali sangat cepat yang dapat menyebabkan pembekuan darah di jantung. *Ventricular Tachycardia* (VT) adalah masalah irama jantung yang disebabkan oleh sinyal listrik yang tidak teratur di bagian ventrikel. *Ventricular Fibrillation* (VF) terjadi ketika bilik jantung bagian bawah berkontraksi dengan sangat cepat dan tidak terkoordinasi dan menyebabkan jantung tidak memompa darah ke seluruh tubuh dan paling sering menjadi penyebab kematian jantung mendadak. *Premature Ventricular Contraction* (PVC) adalah detak jantung ekstra yang dimulai di ventrikel. *Premature Atrial Contraction* (PAC) terjadi karena jantung berdetak terlalu dini pada bagian atrium. *Electrocardiography* (ECG) merupakan salah satu cara untuk mendeteksi aritmia [36]. Alat Elektrokardiogram (EKG) dapat digunakan untuk merekam dan mengidentifikasi sinyal jantung. Sinyal Jantung setiap manusia itu unik, tergantung pada faktor termasuk usia, jenis kelamin, keturunan, dan masalah medis yang dimilikinya [10]. Namun, karena sifat aritmia yang intermiten, diagnosis dalam kasus ini memerlukan pemeriksaan dalam jumlah data yang besar, oleh karena itu menekankan penggunaan metode analisis otomatis,

yang biasanya terbukti akurat dan dapat diandalkan dalam mengkategorisasi individu yang terkena aritmia [6].

Saat ini cukup banyak penelitian yang mengusulkan metode deteksi aritmia berdasarkan EKG. Pada umumnya deteksi aritmia terdiri dari tiga tahapan, *pre-processing*, ekstraksi ciri, dan klasifikasi. Tahap *pre-processing* adalah tahap untuk menghilangkan *noise* yang ada pada sinyal EKG dan merubah format sinyal agar dapat diproses oleh algoritma. Tahap ekstraksi ciri untuk menghasilkan fitur-fitur yang digunakan untuk mengidentifikasi. Dan terakhir, tahap klasifikasi yang merupakan tahap untuk menentukan akurasi dari algoritma yang digunakan.

Sudah ada beberapa metode yang diusulkan untuk pengklasifikasian aritmia, seperti yang dilakukan [3] yang melakukan penelitian untuk mengklasifikasi aritmia menggunakan *Support Vector Machine* dan [13] yang melakukan penelitian menggunakan berbagai pendekatan algoritma *machine learning*.

Pada tahun 2018, [38] melakukan penelitian menggunakan algoritma *deep learning* dengan menggunakan *convolution neural networks*, dan hasil akurasi dari metode klasifikasi yang diusulkan adalah 83.5%. Hal ini terjadi karena fitur bigeminy mudah disalahartikan sebagai sinyal normal, VT, AF, dan AFIB, yang akan menyebabkan *false positive*. [29] juga melakukan penelitian menggunakan *convolution neural networks* untuk klasifikasi aritmia dengan menggunakan aktivasi fungsi *Rectified Linear Unit* (ReLU), dengan akurasi 84.44%.

Selain metode yang disebutkan, telah banyak metode klasifikasi yang dicoba untuk aritmia, tetapi hanya sedikit yang menggunakan algoritma *deep learning*. Namun, metode-metode ini belum mencapai akurasi yang signifikan. Diperlukan metode yang optimal untuk meningkatkan akurasi dan mengurangi dimensi data dengan banyak fitur. *Deep learning* menjadi pendekatan paling menjanjikan untuk deteksi aritmia.

1.2 Perumusan Masalah

Berdasarkan latar belakang di atas, rumusan masalah tugas akhir ini adalah sebagai berikut:

1. Bagaimana melakukan studi algoritma *deep learning* dalam peningkatan akurasi deteksi aritmia?
2. Bagaimana mengembangkan model deteksi aritmia berdasarkan algoritma *deep learning*?
3. Bagaimana meningkatkan performansi akurasi model algoritma *deep learning* yang diusulkan pada Elektrokardiogram untuk mendeteksi aritmia?

1.3 Pernyataan Masalah

Berdasarkan latar belakang diatas, dapat disimpulkan terdapat permasalahan pada algoritma klasifikasi dan deteksi yang sudah ada sebagai berikut :

1. Algoritma klasifikasi *deep learning* yang ada masih menghasilkan akurasi deteksi yang rendah.
2. Pengembangan model deteksi aritmia berdasarkan algoritma *deep learning* masih jarang dilakukan.
3. Performansi dari pengembangan model *deep learning* untuk deteksi aritmia masih rendah.

1.4 Tujuan

Tujuan dari penelitian ini adalah sebagai berikut :

1. Melakukan studi literatur algoritma *deep learning* dan membuat Taksonomi algoritma *deep learning*.
2. Mengembangkan model algoritma *deep learning* untuk klasifikasi dan deteksi aritmia.
3. Melakukan analisis performansi model algoritma yang dibangun dengan menggunakan metrik uji akurasi, spesifisitas, dan sensitivitas.

1.5 Batasan Masalah

Berikut adalah ruang lingkup yang ada pada penulisan tugas akhir ini :

1. Tahap analisis yang dilakukan hanya pada bagian klasifikasi.
2. Pengujian hanya dilakukan pada dataset MIT-BIH *Arrhythmia Database*

1.6 Hipotesis

1. Algoritma klasifikasi *deep learning* yang diusulkan dalam penelitian ini menghasilkan akurasi lebih dari 85% untuk mengklasifikasikan sinyal aritmia pada elektrokardiogram.
2. Performansi algoritma dari model *deep learning* yang dibangun dan dikembangkan lebih akurat dibanding yang sudah ada

1.7 Sistematika Penulisan

Tugas Akhir ini disusun dengan sistematika penulisan sebagai berikut :

- **BAB I Pendahuluan.** Bab ini membahas mengenai latar belakang, rumusan masalah, dan tujuan pengerjaan Tugas Akhir ini.
- **Bab II Kajian Pustaka.** Bab ini membahas fakta dan teori yang berkaitan dengan perancangan sistem untuk mendirikan landasan berfikir. Dengan menggunakan fakta dan teori yang dikemukakan pada bab ini penulis menganalisis kebutuhan akan rancangan arsitektur model algoritma yang dibangun.
- **BAB III Metodologi.** Bab ini menjelaskan metode penelitian, metode perancangan model algoritma dan metode pengujian yang dilakukan dalam penelitian.
- **BAB IV Hasil dan Pembahasan.** Bab ini menjelaskan hasil dari algoritma deep learning dan hasil pengujian yang dilakukan terhadap data MIT-BIH
- **BAB V Kesimpulan dan Saran.** Bab ini menjelaskan pernyataan tentang riset yang telah dilakukan.

Bab II

Kajian Pustaka

Bab ini menjelaskan riset terkait tugas akhir dan landasan teori pendukung yang digunakan. Riset Terkait diuraikan di Sub Bab 2.1, sedangkan landasan teori dapat ditemukan pada Sub Bab 2.2. Ringkasan disajikan pada bagian terakhir dari Bab 2.

2.1 Penelitian Terkait

Banyak penelitian telah dilakukan untuk mendiagnosis aritmia. Berikut adalah 15 studi terkait yang telah dipublikasikan sejak tahun 2019 hingga sekarang.

Ivanovic et al. [22] mengusulkan sebuah arsitektur gabungan *Convolutional Neural Network* dan *Reccurent Neural Network* untuk mengekstraksi fitur dari serangkaian interval RR guna mendeteksi *Atrial Fibrillation* (AF), *Atrial Flutter* (AFL), atau *Sinus Rythm* (SR). Strategi *10-fold cross-validation* juga digunakan untuk memilih desain arsitektur dan menyetel *hyperparameter* model. Setelah diujikan, model arsitektur ini memiliki 88.28% akurasi, dengan sensitivitas 93.83%, 83.60% dan 83.83% untuk SR, AF dan AFL.

Faust et al. [16] juga mengusulkan sistem pemrosesan sinyal yang ditetapkan sesuai aturan *10-fold cross-validation* dan kemudian menggunakan algoritma ResNet untuk mengklasifikasi AF, AFL, dan RS berdasarkan sinyal interval RR, algoritma tersebut diuji dengan 4051 data pasien. Algoritma ini dapat membedakan AF, AFL, dan SR dengan akurasi 97.96% dan sensitivitas 97.58%.

Romdhane et al. [37] mengusulkan langkah pengoptimalan algoritma *deep learning* berdasarkan model *Convolutional Neural Network* dengan menggunakan fungsi *Focal Loss* yang dapat meningkatkan akurasi klasifikasi untuk kelas minoritas (S dan F) serta metrik keseluruhan. Metode yang diusulkan menggunakan kumpulan data MIT-BIH dan INCART untuk mengidentifikasi lima kategori aritmia (N, S, V, Q, dan F) berdasarkan standar Association for Advancement of Medical Instrumentation (AAMI) dan mencapai akurasi keseluruhan 98.41%.

Rajkumar et al. [35] melakukan penelitian untuk mengenali berbagai jenis

aritmia menggunakan *Convolutional Neural Network*, sebuah algoritma *Deep Learning* yang efisien dalam mengklasifikasikan data sinyal EKG mentah sebagai input, data diproses dan kemudian fitur diekstraksi. Algoritma yang diusulkan dapat mengklasifikasikan sinyal dari tujuh jenis aritmia dengan akurasi 93.6% dengan menggunakan fungsi aktivasi *Exponential Linear Unit* (ELU).

Tiga tahun kemudian, Lokesh et al. [29] melakukan penelitian yang berfokus pada *Convolutional Neural Networks* (CNNs), semacam *Deep Neural Networks* (DNNs) dan mengembangkan *11 layered CNN* dengan menggunakan aktivasi fungsi *Exponential Linear Units* (ELU) dan *Rectified Linear Unit* (ReLU) untuk klasifikasi aritmia EKG yang efektif. Fungsi aktivasi ELU mengungguli fungsi aktivasi ReLU dalam skenario ini, dengan tingkat akurasi 99,2%.

Izci et al. [23] melakukan penelitian dan mengusulkan arsitektur *Two-Dimensional Convolution Neural Network* (2D-CNN) untuk klasifikasi dari lima kategori aritmia yang berbeda. Dengan menggunakan database aritmia MIT-BIH, sinyal EKG disegmentasi menjadi detak jantung (*heartbeats*) dan setiap detak jantung diubah menjadi gambar skala abu-abu 2-D (*2-D grayscale*) sebagai input data untuk struktur CNN. 2D-CNN dapat mencapai akurasi 97.42% tanpa preprosesing, ekstraksi fitur dan pemilihan fitur sinyal EKG.

Al et al. [1] juga menggunakan *Two-Dimensional Convolution Neural Network* (2D-CNN) yang dilatih dengan menggunakan gambar sinyal Elektrokardiogram (EKG), data dikumpulkan dari database MIT-BIH untuk mendiagnosis 17 tipe aritmia. CNN cocok untuk mengklasifikasikan data gambar yang didapatkan dari rekaman EKG yang masih diproduksi diatas kertas termal dan kemudian menggunakan *image processing* untuk mendigitalkan rekaman tersebut, hal ini akan mengurangi prapemrosesan. Algoritma yang diusulkan memperoleh akurasi rata-rata 96.673% untuk 17 kategori aritmia.

Li et al. [28] mengusulkan *One-Dimensional Taguchi-based Convolutional Neuro-Fuzzy Network*(1D-TCNFN) yang memiliki beberapa lapisan diantaranya; satu lapisan input, tiga lapisan *1D convolutional*, satu lapisan *feature fusion*, satu lapisan *fuzzification*, satu lapisan aturan, dan satu lapisan output untuk mendeteksi 17 sinyal aritmia pada EKG, metode Taguchi digunakan untuk mengoptimalkan arsitektur jaringan dari ID-TCNFN. dengan menggunakan database aritmia MIT-BIH, metode yang diusulkan memiliki akurasi deteksi 93.95%.

Yue et al. [45] mengumpulkan sample data dari enam database EKG dan kemudian menggunakan metode *label shuffling* untuk memperkuat sampel, yang secara efektif mengatasi ketidakseimbangan dalam sample. *Long Short-Term Memory* (LSTM) dengan menggunakan model *multi-input* dan *single-output* diusulkan untuk mendeteksi titik fitur utama dari sinyal EKG yang akan menyelesaikan segmentasi dan pemrosesan data. Dikarenakan sumber data berasal dari enam database yang berbeda, perbedaan metode pengumpulan data dapat mempengaruhi akurasi klasifikasi yaitu hanya 71%.

Khan et al. [25] melakukan penelitian untuk mengklasifikasikan aritmia menjadi 16 kelas berdasarkan karakteristik 219 atribut dataset EKG dari repository University of California, Irvine (UCI) dengan menggunakan *deep learning* dalam bentuk *Long Short-Term Memory* (LSTM) yang juga mencakup algoritma klasifikasi *Principal Components Analysis* (PAC) untuk menghilangkan *noise* pada sinyal EKG. Akurasi klasifikasi tertinggi dengan menggunakan metode yang diusulkan mencapai 93.5%.

Zheng et al. [46] melakukan pengembangan metode klasifikasi aritmia untuk mendiagnosis delapan sinyal EKG satu dimensi yang kemudian dikonversi menjadi *2D grayscale* sebagai data input dan menggunakan kombinasi *Convolutional Neural Networks* (CNN) dan *Long Short-Term Memory* (LSTM) untuk mendeteksi dan klasifikasi data input tanpa memerlukan ekstraksi fitur atau *noise filtering* pada sinyal EKG. Setelah dilakukan pengujian, metode ini memiliki 99.01% akurasi, dengan sensitivitas 97.67%.

Ullah et al. [42] melakukan riset untuk mengklasifikasikan aritmia menggunakan *deep learning* pada dua jenis kumpulan data, lima kelas database aritmia MIT-BIH (N, S, V, F, dan Q) dan dua kelas dari database PTB Diagnostic ECG. Model *deep learning* yang digunakan adalah CNN, CNN + LSTM, dan CNN + LSTM + Attention Model, attention model digunakan untuk memberikan fokus tambahan pada komponen tertentu. Dalam *deep learning*, Attention model berhubungan dengan fokus pada sesuatu yang khusus dan mencatat kepentingan spesifiknya. Model yang diusulkan mendapatkan tingkat akurasi 99.12% untuk CNN, 99.3% akurasi untuk CNN + LSTM, dan 99.29% akurasi untuk CNN + LSTM + Attention Model.

Pada tahun yang sama, Essa et al. [14] juga mengusulkan metode untuk mengklasifikasikan aritmia. Metode ini menggabungkan beberapa model *deep learning* dengan arsitektur dan level data yang berbeda. Untuk mendiagnosis semua kelainan detak jantung secara akurat, kombinasi model *deep learning* telah dikembangkan berdasarkan dua arsitektur: CNN-LSTM dan LSTM yang dikombinasikan dengan interval RR dan *Higher Order Statistic* (HOS). LSTM secara efektif mengidentifikasi detak jantung abnormal. Setelah dilakukan pengujian dan evaluasi menggunakan data EKG dari database aritmia MIT-BIH, metode yang diusulkan memperoleh akurasi keseluruhan 98.81%.

Sowmya et al. [40] melakukan riset dengan menggunakan arsitektur jaringan berdasarkan *deep learning* LSTM dan CNN untuk mengidentifikasi kelainan pada sinyal EKG karena aritmia. Metode ekstraksi ciri dan seleksi ciri dipelajari untuk mendeteksi penyakit aritmia pada seseorang. Jika dibandingkan dengan CNN yang akurasinya 96.8%, pendekatan *deep learning hybrid* CNN-LSTM untuk mengklasifikasikan sinyal EKG mencapai kompatibilitas akurasi yang tinggi yaitu mencapai 97.3%.

Islam et al. [21] mengusulkan dengan menggunakan wavelet Daubechies, dapat mengatasi masalah fraktal dan diskontinuitas sinyal dan noise da-

ri filter yang diproses sebelumnya juga dihilangkan. Pendekatan normalisasi Pan-Tompkins digunakan untuk menangani berbagai sampel dengan distribusi normal sebelum *Z-normalization*. Untuk mengatasi sampel sinyal yang tidak seimbang, sinyal sintetis dibuat menggunakan *Generative Adversarial Network* (GAN). *Bidirectional RNN with Dilated CNN* (BRDC) adalah arsitektur yang diusulkan dengan menggabungkan *bidirectional gated recurrent Units* (BiGRU) dan *bidirectional long short-term memory* (BiLSTM) untuk menyediakan fitur fusi. Terakhir, sinyal dikategorikan menggunakan fungsi aktivasi *Rectified Linear Unit* (ReLU) dan *fully connected layer*. Model BRDC yang diusulkan mengungguli model yang ada dengan menggunakan data EKG dari MIT-BIH untuk mendiagnosis aritmia dengan akurasi 99,90%.

Perbandingan hasil penelitian di atas dapat dilihat pada tabel di bawah ini:

Tabel 2.1: Ringkasan riset terkait

No.	Judul	Penulis / Tahun	Hasil Riset	Kelebihan	Kekurangan
1	Deep Learning Approach for Highly Specific Atrial Fibrillation and Flutter Detection based on RR Intervals	Tsamir-Rou LinMarija D. Ivanovic, Vladimir Atanasoski, Alexei Shvilkin, Ljupco Hadzievski and Aleksandra Maluckov / 2019	Deteksi AF dan AFL menggunakan arsitektur DNN, yang menggabungkan <i>convolutional and recurrent neural networks</i>	Akurasi klasifikasi adalah 89.67%	Hanya efektif untuk mendeteksi AF & AFL dan sensitivitas AFL yang masih rendah yaitu 77.78%
2	Automated Arrhythmia Detection Based on RR Intervals	Oliver Faust, Murthada Kareem, Ali Ali, Edward J. Ciaccio, U. Rajendra Acharya / 2021	Deteksi aritmia berdasarkan RR interval menggunakan metode ResNet 10-fold cross-validation untuk menguji model	Akurasi dari metode yang diusulkan mencapai 99.98%	Hanya efektif untuk mengklasifikasikan AFL, AFIB, and NSR sinyal interval RR
3	Electrocardiogram Heartbeat Classification Based on a Deep Convolutional Neural Network and Focal Loss	Taissir Fekih Roshanmdhanea, Haikel Alhichrib, Ridha Ounib, Mohamed Atrici / 2020	Deep CNN dan Focal Loss untuk mendeteksi lima kategori aritmia (N, S, V, Q, dan F) berdasarkan standar AAMI	Focal loss meningkatkan akurasi klasifikasi untuk kelas minoritas serta metrik keseluruhan dan Metode yang diusulkan mencapai akurasi keseluruhan 98,41%	Jumlah dataset yang digunakan cukup sedikit

Tabel 2.1: Ringkasan riset terkait

No.	Judul	Penulis / Tahun	Hasil Riset	Kelebihan	Kekurangan
4	Arrhythmia classification on ECG using Deep Learning	Rajkumar. A, Ganeshan. M, Lavanya. R / 2019	Klasifikasi aritmia menggunakan CNN dengan sistem dimana data mentah EKG sebagai input	CNN dengan aktivasi fungsi ELU memiliki akurasi 93.6%	memerlukan <i>epoch</i> yang banyak untuk akurasi yang tinggi
5	CNN based deep learning methods for precise analysis of cardiac arrhythmias	S. Lokesh, A. Priya, D. T. Sakhare, R. Manjula Devi, Dillip Narayan Sahu, Pundru Chandra Shaker Reddy / 2022	Klasifikasi aritmia menggunakan 11 layer CNN dengan aktivasi fungsi ELU dan ReLU	Aktivasi fungsi ELU memiliki 99.2% akurasi	ReLU memiliki akurasi 84.44%
6	Diagnosis of arrhythmia based on ECG analysis using CNN	Muayed S. Al-Huseiny, Noor Khudhair Abbas, Ahmed S Sajit / 2020	Diagnosis aritmia menggunakan 2D-CNN	Akurasi klasifikasi menggunakan 2D-CNN adalah 96.67%	Dataset yang digunakan untuk melatih model deep learning hanya terdiri dari 1000 fragmen sinyal ECG.

Tabel 2.1: Ringkasan riset terkait

No.	Judul	Penulis / Tahun	Hasil Riset	Kelebihan	Kekurangan
7	Cardiac Arrhythmia Detection from 2D ECG Images by Using Deep Learning Technique	Elif Izci, Mehmet Akif Odzemir, Murside Dergimenci, dan Aydin Akan / 2019	Klasifikasi lima tipe aritmia berdasarkan sinyal EKG yang tersegmentasi menjadi detak jantung dan masing-masing detak jantung itu dikonversi menjadi gambar <i>2D grayscale</i> sebagai input data untuk struktur CNN	Nilai akurasi dari model yang diusulkan adalah 97.42%	Deteksi <i>fusion arrhythmia</i> memiliki tingkat keberhasilan yang lebih rendah daripada deteksi jenis aritmia lainnya
8	Arrhythmia Detection Using a Taguchi-based Convolutional Neuro-fuzzy Network	Jiarong Li, Jyun-Yu Jhang, Cheng-Jian Lin, Xue-Qian Lin / 2022	one-dimensional Taguchi-based convolutional neuro-fuzzy network (1D-TCNFN) untuk mendeteksi 17 tipe aritmia pada sinyal EKG	Memiliki nilai akurasi yang cukup tinggi yaitu 93.95%	<i>Confusion matrix</i> 1D-TCNFN menyatakan <i>pacemaker rhythm</i> salah diklasifikasikan sebagai <i>normal sinus rhythm</i> terhadap 25% kasus

Tabel 2.1: Ringkasan riset terkait

No.	Judul	Penulis / Tahun	Hasil Riset	Kelebihan	Kekurangan
9	Using Deep Learning Detection Of Arrhythmia	Zhang Yue1, Li Feng1 / 2019	Deteksi aritmia menggunakan Long-Term and Short-Term Memory Model (LSTM) untuk deteksi titik fitur utama dari sinyal EKG dari enam database berbeda	Akurasi keseluruhan untuk klasifikasi aritmia dengan model yang diusulkan adalah 71%	Karena sumber data berasal dari enam database yang berbeda, perbedaan metode pengumpulan data dapat mempengaruhi akurasi klasifikasi
10	Cardiac Arrhythmia Disease Classification Using LSTM Deep Learning Approach	Muhammad Ashfaq Khan dan Yangwoo Kim / 2020	Klasifikasi aritmia menggunakan LSTM mencakup algoritma klasifikasi Principal Components Analysis (PAC) untuk menghilangkan <i>noise</i>	Akurasi klasifikasi tertinggi mencapai 93.5%	Masih membutuhkan penelitian lebih lanjut untuk urasi. Namun, diperlukan penelitian lebih lanjut untuk pengelompokan, penghilangan <i>noise</i> , dan teknik klasifikasi yang berbeda untuk mende-teksi aritmia lainnya

Tabel 2.1: Ringkasan riset terkait

No.	Judul	Penulis / Tahun	Hasil Riset	Kelebihan	Kekurangan
11	An Automatic Diagnosis of Arrhythmias Using a Combination of CNN and LSTM Technology	Zhenyu Zheng, Zhencheng Chen, Fangrong Hu, Jianming Zhu, Qunfeng Tang, Yongbo Liang / 2020	metode klasifikasi untuk aritmia berdasarkan kombinasi convolutional neural network dan long short-term memory, yang kemudian digunakan untuk mendiagnosis delapan sinyal EKG termasuk normal sinus rhythm	Model yang diusulkan memiliki nilai akurasi 99.01%	dengan model yang diusulkan tanpa regulisasi <i>dropout</i> , <i>training model</i> cenderung overfitting dan mempengaruhi kemampuan klasifikasi model.
12	Classification of Arrhythmia in Heartbeat Detection Using Deep Learning	Wusat Ullah, Imran Siddique, Rana Muhammad Zulqarnain, Mohammad Mahtab Alam, Irfan Ahmad, Usman Ahmad Raza / 2021	Deteksi aritmia menggunakan CNN, CNN+LSTM, and CNN+LSTM+Attention Model	Tingkat akurasi 99,12% untuk CNN, akurasi 99,3% untuk CNN+LSTM, dan akurasi 99,29% untuk CNN+LSTM+Attention Model	kemungkinan adanya overfitting data pada model yang diusulkan

Tabel 2.1: Ringkasan riset terkait

No.	Judul	Penulis / Tahun	Hasil Riset	Kelebihan	Kekurangan
13	An Ensemble of Deep Learning-Based Multi-Model for ECG Heartbeats Arrhythmia Classification	Ehab Essa, Xianghua Xie / 2021	Klasifikasi aritmia menggunakan CNN-LSTM dan RRHOS-LSTM (RR interval dan High-Order Statistic) berdasarkan 4 kelas rekomendasi AAMI	Model yang diusulkan memiliki nilai akurasi 95.81%	Nilai sensitivitas yang masih tergolong rendah
14	Contemplate on ECG signals and classification of arrhythmia signals using CNN-LSTM deep learning model	S. Sowmya, Deepa Jose / 2022	Identifikasi sinyal EKG aritmia menggunakan arsitektur berdasarakan metode <i>deep learning</i> LSTM-CNN	metode yang diusulkan memiliki akurasi 97.3%	Ukuran sample yang kecil sehingga kurang efisien untuk mende-teksi beberapa tipe aritmia lainnya

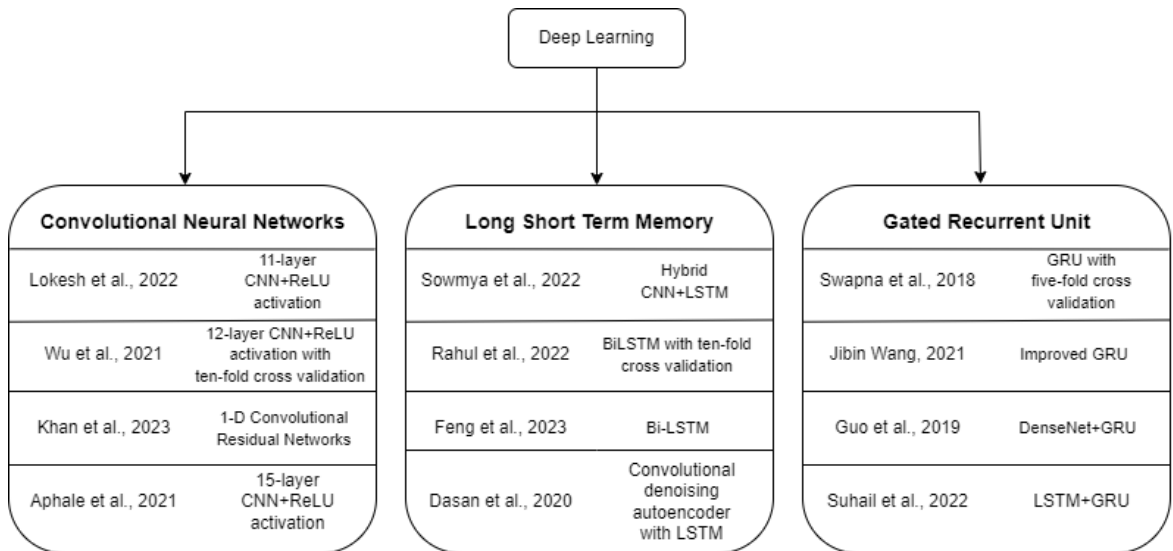
Tabel 2.1: Ringkasan riset terkait

No.	Judul	Penulis / Tahun	Hasil Riset	Kelebihan	Kekurangan
15	New Hybrid Deep Learning Approach Using BiGRU-BiLSTM and Multilayered Dilated CNN to Detect Arrhythmia	MD Shoifiquil Islam, MD Nahidul Islam, Noramiza Hashim, Mamunur Rashid, Bifta Sama Bari, Fahmid Al Farid / 2022	Deteksi aritmia menggunakan (bi-directional gated recurrent Units, BiGRU - bidirectional long short-term memory, BiLSTM) untuk menghasilkan <i>feature fusion</i> dan kemudian diklasifikasi menggunakan fungsi CNN dengan fungsi aktivasi ReLU	99.90% akurasi didapatkan dengan menggunakan model <i>hybrid</i> yang diusulkan	Hanya efektif untuk mendeteksi sinyal AF

2.2 Deep Learning

Dalam beberapa tahun terakhir, *deep learning* telah menjadi populer dan menarik di bidang pembelajaran mesin. Dalam hal *machine learning*, *deep learning* adalah metode yang paling akurat, *supervised*, dan hemat waktu. Dengan *deep learning*, pengetahuan yang dapat diperoleh tidak terbatas. Ini mencakup berbagai proses dan struktur dan sangat membantu dalam berbagai skenario sulit. Teknik *deep learning* telah mencapai kemajuan substansial dalam berbagai bidang, termasuk keamanan. Aplikasi *deep learning* yang paling sering digunakan adalah diagnosis kanker, identifikasi objek, klasifikasi dalam biomedis, dan banyak bidang lainnya [20]

Deep Learning mengacu pada kelas algoritma *Machine Learning* yang menggunakan modifikasi nonlinear secara ekstensif untuk merepresentasikan ide abstrak yang diambil dari data. Kumpulan data besar harus diberi label pada fase pelatihan (*training*), pada fase *inferring*, kesimpulan dibuat dan kumpulan data yang tidak diketahui diklasifikasikan berdasarkan informasi sebelumnya. *deep learning* digunakan untuk membantu sistem dalam memahami dan melaksanakan tugas-tugas sensorik yang kompleks dengan tingkat akurasi yang tinggi. *deep learning* metode pembelajaran mesin yang menggunakan unit pemrosesan nonlinier untuk mengekstraksi dan mengubah data dari data dalam jumlah besar [9].



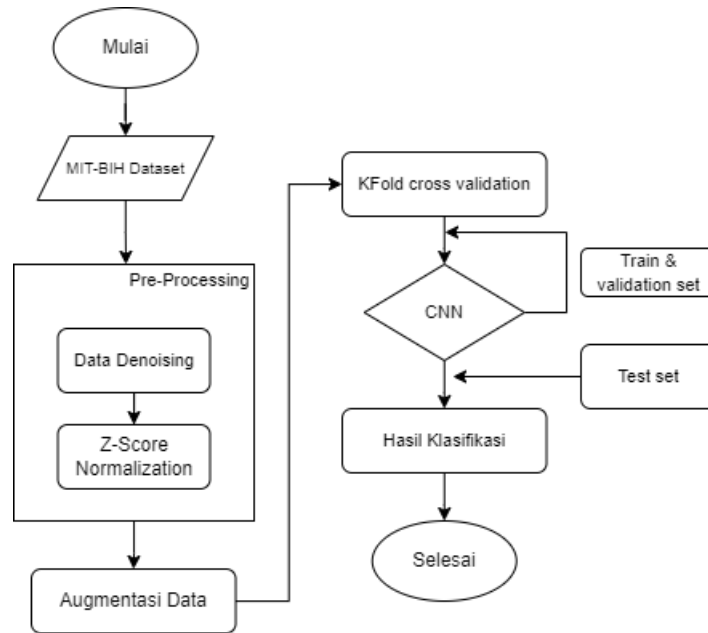
Gambar 2.1: Taksonomi *Deep Learning*

Gambar 2.1 memberikan contoh taksonomi literatur *deep learning* yaitu *Convolutional Neural Network*, *Long Short-Term Memory*, dan *Gated Recurrent Unit* beserta studi literatur terkait, contohnya pada *Convolutional Neural Network*, penelitian menggunakan *11-layer CNN + ReLU activation* oleh Lo-

kesh et al. kemudian ada *BiLSTM with ten-fold cross validation* oleh Rahul et al. lalu, *DenseNet* oleh Guo et al.

2.2.1 Convolutional Neural Network

Convolutional Neural Network (CNN) tidak dapat memanfaatkan struktur spasial atau temporal yang ada dalam data karena setiap neuron terhubung ke setiap neuron di lapisan berikutnya [34]. Salah satu arsitektur DNN yang paling sering digunakan adalah CNN, yang biasanya dilatih menggunakan pendekatan *gradient-based optimization algorithm* [27]. CNN biasanya terdiri dari banyak lapisan back-to-back yang terhubung ke *feedforward*. *Convolutional layer*, *normalization layer*, *pooling layer*, dan *fully-connected layer* adalah lapisan utama. Tiga lapisan pertama bertanggung jawab untuk mengekstraksi fitur, sementara *fully-connected layer* bertanggung jawab atas proses klasifikasi. Dan pada penelitian ini, menggunakan *Convolutional Neural Network*.

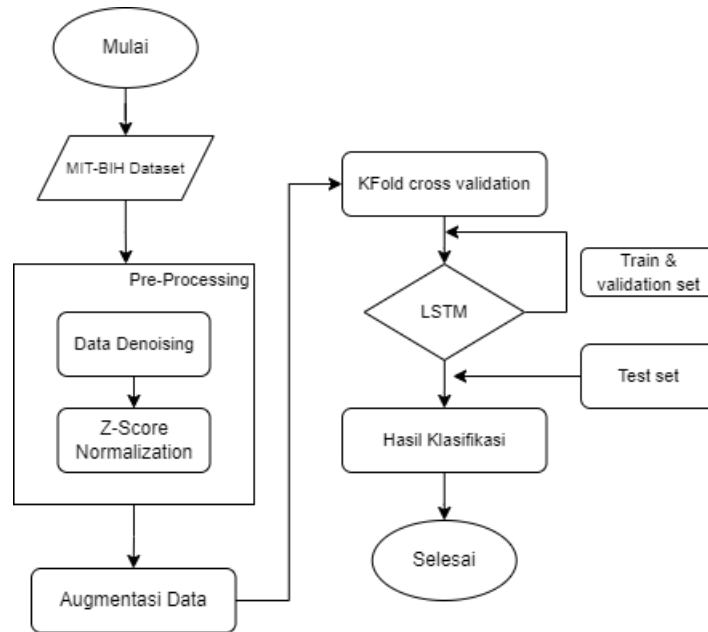


Gambar 2.2: Flowchart model CNN

2.2.2 Long Short-Term Memory

Long short-term memory (LSTM) adalah jenis RNN tradisional yang khusus dirancang untuk urutan temporal dan ketergantungan jangka panjang [8, 26]. Jaringan LSTM mengatasi masalah gradien yang tidak stabil dan memungkinkan jaringan mempelajari ketergantungan jangka panjang. Selain itu, telah dibuktikan bahwa jaringan LSTM bekerja lebih baik daripada desain RNN konvensional lainnya [18]. LSTM menggunakan blok memori, yang masing-masing terdiri dari satu atau lebih sel memori dengan gerbang pengali

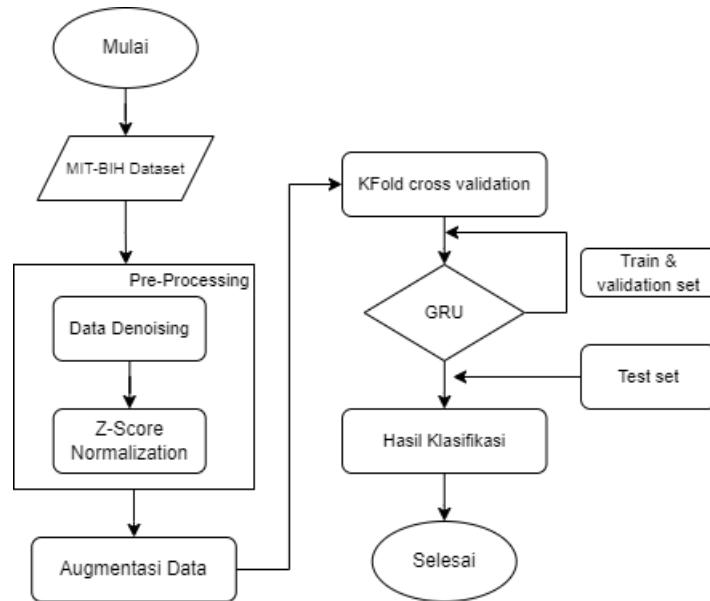
adaptif yang bertindak sebagai input dan output. Berdasarkan gerbang input dan output, blok memori menyimpan informasi dan mengubahnya seiring waktu. input dan output informasi ke sel memori diatur oleh gerbang. Dan pada penelitian ini, akan menggunakan LSTM sebagai algoritma kedua.



Gambar 2.3: Flowchart model LSTM

2.2.3 Gated Recurent Unit

Gated Recurrent Unit (GRU) memiliki kesamaan dengan model LSTM. GRU menggunakan state tersembunyi (*hidden state*) daripada state cell (*cell state*) untuk mengirimkan informasi dan mencakup dua pintu (*gate*): pintu reset dan pintu update. Pintu update dalam GRU menggabungkan fungsi dari (*forget gate*) dan pintu masukan (*input gate*) dalam LSTM, menentukan apakah informasi harus dipertahankan atau diabaikan dari memori. Pintu reset mengontrol sejauh mana informasi state sebelumnya diabaikan. Model GRU memiliki waktu pelatihan yang lebih cepat dibandingkan LSTM karena memiliki lebih sedikit pintu dan operasi tensor [11]. Pilihan antara GRU dan LSTM tergantung pada persyaratan tugas yang spesifik.



Gambar 2.4: Flowchart model GRU

2.3 Aritmia

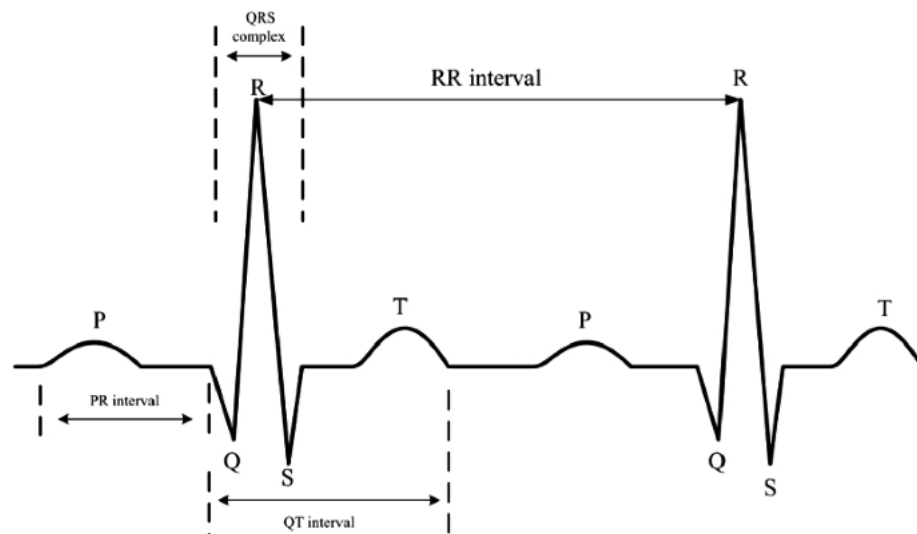
Penyakit yang dikenal sebagai aritmia jantung terjadi ketika detak jantung tidak berdetak secara teratur. Aritmia mengacu pada ritme apa pun yang tidak sesuai dengan standar normal sinus rhythm (SR). Istilah *normal sinus rhythm* mengacu pada detak jantung secara umum. Setiap denyut yang dihasilkan oleh *sinoatrial* (SA) selama sinus rhythm dan menghasilkan gelombang bioelektrik yang merambat melintasi empat ruang jantung secara tersinkronisasi. Setiap impuls berjalan melalui seluruh atrium sebelum dikirim ke ventrikel melalui *atrioventrikular* (AV). Denyut listrik ini menyebabkan proses *intracellular calcium* yang mengakibatkan kontraksi otot jantung dan memompa darah ke organ tubuh. Jantung biasanya memompa sekitar 5 liter darah setiap menit dan berdetak dengan kecepatan sekitar 60-100 denyut per menit (walaupun ada variasi individu yang signifikan). Takikardia adalah kondisi dimana detak jantung berdetak terlalu kencang, sedangkan ketika jantung berdetak terlalu lambat, disebut bradikardia.

Aritmia jantung memiliki jenis yang cukup beragam, diantaranya; *Atrial fibrillation* (AFIB), adalah irama jantung yang tidak teratur dan seringkali sangat cepat yang dapat menyebabkan pembekuan darah di jantung. AFIB meningkatkan risiko stroke, gagal jantung, dan komplikasi terkait jantung lainnya. *Ventricular Tachycardia* (VT) adalah masalah irama jantung yang disebabkan oleh sinyal listrik yang tidak teratur di ruang bawah jantung (ventrikel). *Ventricular Fibrillation* (VF) adalah jenis irama jantung yang tidak teratur. Selama *ventricular fibrillation*, bilik jantung bagian bawah berkon-

traksi dengan sangat cepat dan tidak terkoordinasi. Akibatnya, jantung tidak memompa darah ke seluruh tubuh. *ventricular fibrillation* merupakan keadaan darurat yang membutuhkan perhatian medis segera dan menjadi penyebab kematian jantung mendadak yang paling sering. *Premature Ventricular Contraction* (PVC) adalah detak jantung ekstra yang dimulai di salah satu dari dua ruang pompa bawah jantung (ventrikel). Detak ekstra ini mengganggu ritme jantung yang teratur, terkadang menyebabkan sensasi berdebar atau detak jantung yang terlalu cepat di dada, sedangkan *Premature Atrial Contraction* (PAC) terjadi karena jantung berdetak terlalu dini pada bagian atrium yang menyebabkan detak jantung ekstra dan mengganggu ritme jantung.

2.4 Elektrokardiogram (EKG)

Alat yang sering digunakan untuk merekam aktivitas jantung manusia adalah Elektrokardiogram (EKG). EKG adalah alat yang efektif untuk berbagai aplikasi biomedis seperti mengukur detak jantung, memeriksa irama detak jantung, mendiagnosa kelainan jantung, pengenalan emosi dan identifikasi biometrik. Gelombang P, Q, R, S, dan T adalah lima gelombang utama dari sinyal ECG. Gelombang elektrokardiogram dapat dilihat pada gambar 2.2.



Gambar 2.5: Gelombang Elektrokardiogram (EKG)

Masalah kardiovaskular sering didiagnosis menggunakan elektrokardiogram (EKG). Dengan menempelkan elektroda ke permukaan kulit, dimungkinkan untuk merekam aktivitas listrik jantung yang disebabkan oleh perubahan ion-ionnya. EKG adalah grafik yang menunjukkan hasil rekaman potensial listrik yang dihasilkan detak jantung.

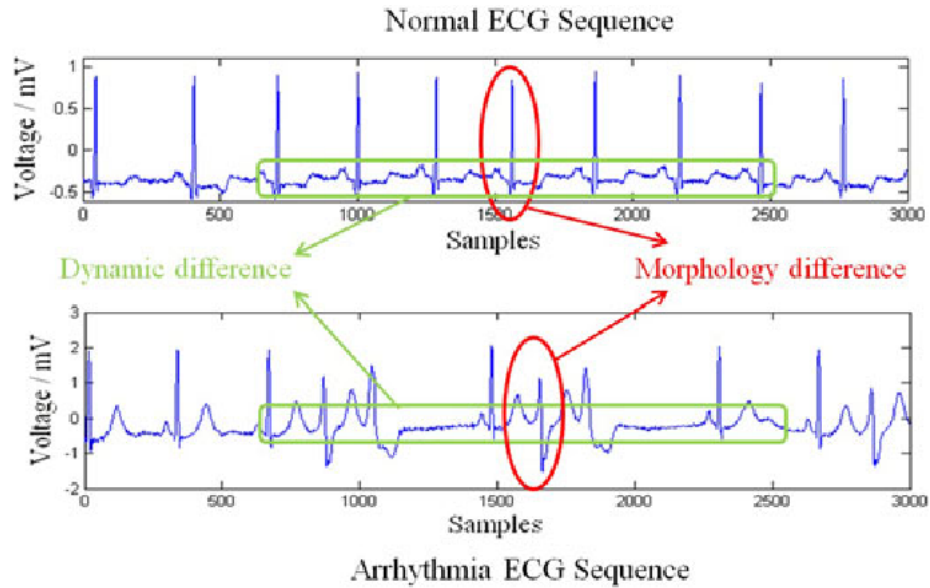
Pada sinyal EKG terdapat lima gelombang utama. Gelombang P yang merupakan defleksi kecil bertegangan rendah yang menjauh dari garis dasar, di-

sebabkan oleh depolarisasi atrium sebelum kontraksi atrium karena gelombang depan aktivasi (depolarisasi) merambat dari nodus sinoatrial melalui atrium. Gelombang Q adalah defleksi ke bawah setelah gelombang P. Gelombang R mengikuti sebagai defleksi ke atas, dan gelombang S adalah defleksi ke bawah mengikuti gelombang R. Gelombang Q, R, dan S bersamaan menunjukkan satu kejadian. Oleh karena itu, mereka biasanya dianggap sebagai QRS kompleks. Fitur berdasarkan QRS kompleks adalah salah satu fitur yang paling kuat untuk analisis EKG. QRS kompleks disebabkan oleh arus yang dihasilkan ketika ventrikel mengalami depolarisasi sebelum kontraksi. Meskipun repolarisasi atrium terjadi sebelum depolarisasi ventrikel, bentuk gelombang terakhir (yaitu, QRS-kompleks) memiliki amplitudo yang jauh lebih besar, dan oleh karena itu, repolarisasi atrium tidak terlihat pada EKG. Gelombang T, yang mengikuti gelombang S, adalah repolarisasi ventrikel, dimana otot jantung dipersiapkan untuk siklus EKG berikutnya [5].

Selain itu, interval tertentu dalam sinyal EKG membawa informasi penting dan digunakan sebagai fitur. Sebagai contoh, interval PR adalah durasi antara awal gelombang P dan QRS kompleks dari elektrokardiogram. Durasi ini mengandung sinyal antara permulaan depolarisasi atrium dan timbulnya depolarisasi ventrikel [19]. Durasi antara awal depolarisasi ventrikel dan akhir repolarisasi ventrikel dikenal sebagai interval QT. Durasi antara akhir gelombang S dan awal gelombang T dikenal sebagai ST-interval [31]. Durasi antara puncak R dari satu detak ke detak jantung berikutnya, dikenal sebagai interval RR (atau interval detak jantung)[30]. Pada penelitian ini RR interval sebagai fitur dari EKG.

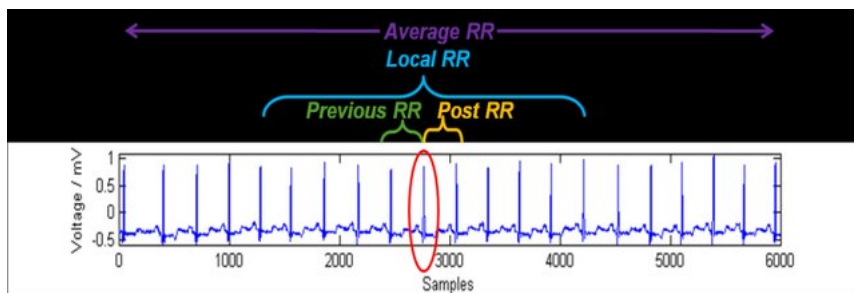
2.5 Fitur Sinyal Elektrokardiogram

Komponen struktural utama sinyal EKG seperti QRS kompleks, membantu kategorisasi karena bentuknya yang mendominasi. Struktur ini dihilangkan selama deteksi QRS, yang digunakan untuk mengekstrak RR interval [16]. RR interval hanya mencerminkan ritme detak jantung. Fitur dinamis diperkenalkan untuk mengkarakterisasi ritme detak jantung. fitur dinamis berguna untuk mendeteksi dan mengklasifikasikan aritmia dan RR interval digunakan untuk mengkarakterisasi fitur dinamis [44].



Gambar 2.6: Contoh perbedaan fitur dinamis dan morfologi

Empat jenis fitur RR interval dihitung untuk mewakili fitur dinamis detak jantung EKG yaitu, *previous RR*, *post RR*, *local RR*, dan *average RR*. *Previous RR* didefinisikan sebagai interval antara detak jantung yang diberikan dan detak jantung sebelumnya sedangkan *post RR* mengacu pada interval antara ketukan yang diberikan dan ketukan berikutnya. *local RR* didefinisikan sebagai rata-rata dari sepuluh interval RR yang berpusat pada ketukan yang diberikan. Terakhir, *average RR* ditentukan dengan rata-rata interval RR dalam periode 5 menit [44].



Gambar 2.7: *previous RR*, *post RR*, *local RR*, dan *average RR*.

Bab III

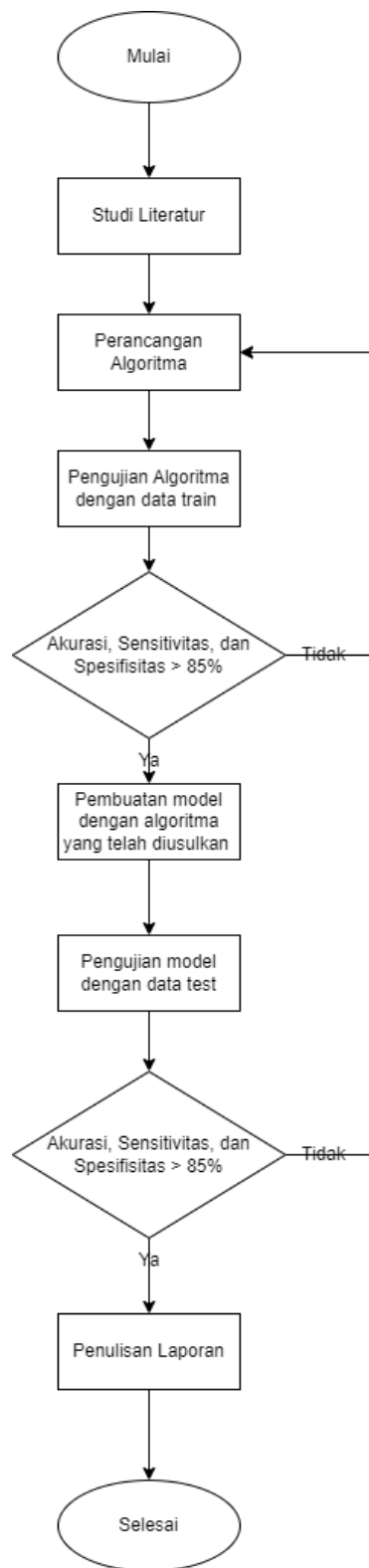
Metodologi

3.1 Metode Penelitian

Sub bab 3.1 berisi metode penelitian yang akan digunakan, dan mencakup framework penelitian, metodologi untuk mencapai tujuan penelitian, analisis kebutuhan sistem, data yang digunakan, metrik uji, metode pengujian dan perbandingan hasil penelitian.

3.1.1 Framework Penelitian

Metodologi yang dilakukan untuk penelitian ini ditunjukkan pada gambar diagram alir 3.1 dibawah ini :



Gambar 3.1: Diagram Alir penelitian *Framework*

Berikut penjelasan dari masing-masing tahapan penelitian pada gambar diagram alir 3.1 :

1. Studi Litaratur

Pada tahap ini, penulis melakukan review terhadap penelitian-penelitian sebelumnya dan kemudian merangkum fakta, hipotesis, dan metode penelitian yang diusulkan. Penulis juga mempertimbangkan kelebihan dan kekurangan dari pendekatan yang disarankan, serta menganalisis bagaimana mengatasi kekurangan tersebut.

2. Perancangan Algoritma

Pada tahap ini, penulis melakukan percobaan terhadap berbagai algoritma dan melakukan perancangan untuk mendapatkan algoritma terbaik yang dapat diusulkan. Dataset MIT-BIH *Arrhythmia Database* akan digunakan untuk melatih algoritma setelah solusi optimal ditemukan. Tahap ini mengarah pada pembuatan algoritma untuk mengklasifikasikan aritmia. Hasil dari tahap ini adalah algoritma yang paling optimal untuk mendeteksi aritmia.

3. Pengujian Algoritma

Pada tahap ini, penulis melakukan pengujian algoritma yang telah diusulkan dengan cara melakukan validasi hasil deteksi algoritma dengan menggunakan data latih dari dataset MIT-BIH *Arrhythmia Database*. Hasil dari tahap ini adalah nilai akurasi, spesifisitas, dan sensitivitas dari algoritma yang diusulkan.

4. Perancangan model dengan Menerapkan Algoritma yang diusulkan

Pada tahap ini, penulis melakukan perancangan model yang akan dibuat. Perancangan yang dilakukan meliputi perancangan model yang akan dibangun dengan menerapkan algoritma yang telah diusulkan. Hasil dari tahapan ini adalah model deteksi aritmia menggunakan algoritma klasifikasi yang telah diusulkan.

5. Pengujian dan Analisis model yang dibangun

Pada tahap ini, penulis melakukan pengujian terhadap performansi model yang dibangun dengan menggunakan data uji dari dataset MIT-BIH *Arrhythmia Database*. Hasil dari tahap ini adalah nilai akurasi, spesifisitas, dan sensitivitas dari model yang telah dibangun..

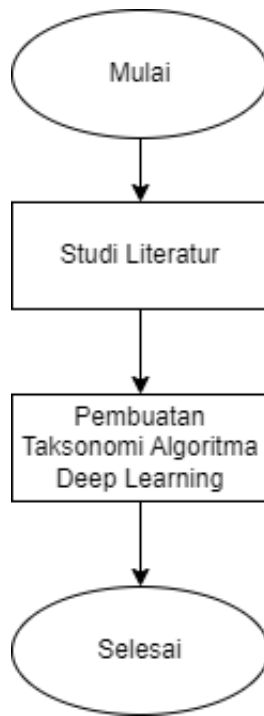
6. Penulisan Laporan

Pada tahap ini penulis menyusun laporan terkait penelitian yang dilakukan mengikuti metode perancangan tata tulis ilmiah. Hasil dari tahapan ini adalah buku tugas akhir.

3.1.2 Metodologi untuk Mencapai Tujuan Penelitian

A) Metodologi untuk mencapai objektif pertama

Metodologi yang dilakukan dalam mencapai objektif pertama digambarkan pada diagram alir 3.2 :



Gambar 3.2: Diagram Alir Metodologi Objektif Pertama

Berikut adalah penjelasan untuk setiap tahapan metodologi :

(a) **Studi Literatur**

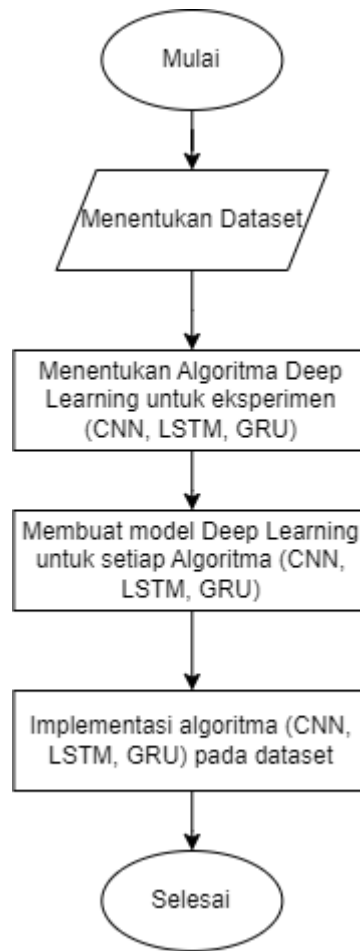
Studi algoritma dilakukan untuk mempelajari algoritma klasifikasi aritmia untuk mendeteksi aritmia yang pernah digunakan pada penelitian-penelitian sebelumnya.

(b) **Pembuatan Taksonomi Algoritma Deep Learning**

Pada tahap ini, algoritma *deep learning* dikategorikan dan taksonomi dibuat berdasarkan hasil dari tahap studi literatur.

B) Metodologi untuk mencapai objektif kedua

Berikut adalah metodologi yang dilakukan dalam mencapai objektif pertama digambarkan pada diagram alir 3.3 :



Gambar 3.3: Diagram Alir Metodologi Objektif Kedua

Berikut adalah penjelasan dari masing-masing tahapan :

(a) **Menentukan Dataset**

Pada tahap ini, dilakukan proses penentuan dataset yang akan digunakan, dan dipilihlah MIT-BIH Dataset sebagai sumber data yang akan menjadi dasar dalam pelaksanaan eksperimen.

(b) **Menentukan Algoritma Deep Learning**

Pada tahap ini, ditentukan proses penentuan model algoritma berdasarkan literatur yang telah dipelajari, yaitu algoritma *Convolutional Neural Network*, *Long Short Term Memory*, dan *Gated Recurrent Unit*.

(c) **Membuat Model Algoritma Deep Learning**

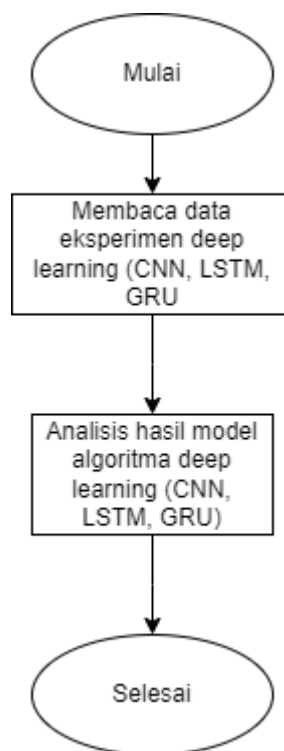
Pada tahap ini, model algoritma deep learning dibuat berdasarkan algoritma yang telah ditentukan.

(d) **Implementasi Algoritma Pada Dataset**

Pada tahap ini dilakukan implementasi dari algoritma yang telah disiapkan untuk diterapkan kedalam dataset yang dipilih. Hasil dari tahapan ini adalah algoritma dapat berjalan menggunakan dataset yang ditentukan dengan baik dan memberi hasil seperti yang diinginkan.

C) **Metodologi untuk mencapai objektif ketiga**

Metodologi yang dilakukan dalam mencapai objektif ketiga digambarkan pada diagram alir 3.4 :



Gambar 3.4: Diagram Alir Metodologi Objektif Ketiga

Berikut adalah penjelasan untuk setiap tahapan metodologi :

(a) **Membaca data eksperimen model deep learning**

Pada tahap ini, dilakukan proses membaca data eksperimen model deep learning, yaitu model *Convolutional Neural Network*, *Long Short Term Memory*, dan *Gated Recurrent Unit*.

(b) **Analisis Model yang telah dibangun**

Pada tahap ini dilakukan analisis dari akurasi yang dihasilkan oleh model pada saat mendeteksi sinyal aritmia pada data latih, Jika

akurasi, spesifisitas, dan sensitivitas deteksi yang dihasilkan tidak jauh berbeda antara deteksi menggunakan model yang diusulkan dengan data dari MIT-BIH *Arrhythmia Database*, maka implementasi algoritma pada model yang dibangun dikatakan berhasil.

3.1.3 Analisis Kebutuhan Sistem

A) Spesifikasi Perangkat Keras

- Laptop Processor Intel(R) Core(TM) i7-8750H @2.20GHz
- Memory 8GB
- Hard Drive 1TB

B) Spesifikasi Perangkat Lunak

- Windows 11 Professional
- Python 3.9
- Jupyter Notebook
- Google Colab

3.1.4 Dataset

Penelitian ini menggunakan data dan anotasi dari basis data aritmia MIT-BIH [32]. Basis data ini berisi 48 rekaman sinyal EKG, masing-masing berdurasi 30 menit, yang dikumpulkan dari 47 individu (terdiri dari 22 perempuan dan 25 laki-laki). Data tersebut diambil dengan kecepatan 360 Hz dan ditandai dengan lokasi puncak R untuk setiap detak jantung. Secara total, basis data ini mencakup kurang lebih 105.000 detak jantung. Setiap rekaman EKG terdiri dari dua derivasi sinyal: modified-lead II (MLII) dan salah satu dari lead V1, V2, V3, V4, atau V5 [17, 33]. Dalam penelitian ini, khususnya digunakan sinyal data lead II.

Sesuai dengan standar yang ditetapkan oleh *Association for the Advancement of Medical Instrumentation Standard*, beberapa label dipilih untuk klasifikasi berdasarkan anotasi yang disediakan dalam basis data aritmia MIT-BIH. Label-label ini meliputi N (mewakili ritme sinus normal - NOR), L (menunjukkan blok cabang kiri - LBBB), R (mengindikasikan blok cabang kanan - RBBB), A (mewakili detak prematur atrium - APB), dan V (menandakan kontraksi ventrikel prematur - PVC). .

Setelah itu, langkah-langkah berikutnya melibatkan pembagian data menjadi set pelatihan dan pengujian yang berbeda. Praktik standar untuk mengalokasikan 80% data untuk pelatihan dan menyisakan 20% untuk pengujian diikuti. Selanjutnya, data tersebut diubah bentuknya untuk sesuai dengan format yang diperlukan agar kompatibel dengan CNN, LSTM dan GRU.

3.1.5 Default Parameter

Convolutional Neural Networks (CNN) yang terdiri dari lapisan konvolusional, lapisan pooling, dan lapisan yang terhubung sepenuhnya. Parameter utamanya mencakup ukuran filter, jumlah filter, stride, padding (baik 'Valid' atau 'Same'), dan fungsi aktivasi, dengan ReLU [38]. *Long Short-Term Memory* (LSTM) adalah bagian dari jaringan berulang yang disesuaikan untuk urutan, yang mampu mengingat ketergantungan jangka panjang. Parameter utama untuk LSTM mencakup jumlah unit, fungsi aktivasi (*tanh* dan *sigmoid*). Regularisasi di LSTM sering kali dicapai dengan menggunakan *dropout* untuk mengatasi *overfitting* [41]. Gated Recurrent Units (GRU), mirip dengan LSTM tetapi menggunakan mekanisme *gate* yang berbeda. Parameter utama GRU mencerminkan parameter LSTM: jumlah unit, fungsi aktivasi, dan keputusan untuk mengembalikan urutan. GRU juga mempekerjakan *dropout* untuk regularisasi [12]. Bengio [4] menyatakan bahwa nilai default optimal untuk parameter *deep learning* dengan memiliki *batch size* 32, *epoch* 20, dan *learning rate* 0.001.

Tabel 3.1: Nilai Parameter

Parameter	Nilai
Learning rate	0.001
Batch Size	32
Epoch	20
Dropout rate	0.5
Regularization strength	0.0001
KFold	5
Optimizer	Adam
Activation Function	Softmax

3.1.6 Hyperparameter Tuning

Hyperparameter adalah pengaturan konfigurasi yang telah ditentukan sebelumnya yang memengaruhi cara model belajar dari data dan memiliki dampak signifikan terhadap performa model. Beberapa hyperparameter penting mencakup *learning rate* (ukuran langkah untuk pengoptimalan), *batch size* (jumlah sampel yang digunakan dalam setiap pembaruan), *epochs* (iterasi melalui data pelatihan), *dropout rate* (probabilitas penurunan neuron), *regularization strength* (penalti untuk nilai parameter yang besar), fungsi aktivasi, dan *optimizer* [24]. Pengaruh *batch size* dan *learning rate* terhadap kinerja *deep learning* yang dibahas oleh Kandel et al. [24], menemukan bahwa *learning rate*, *batch size*, dan *epochs* semuanya memiliki dampak besar terhadap kinerja *network*. Kandel et al. [24] merekomendasikan penggunaan *batch size* kecil yang dikombinasikan dengan *learning rate* yang rendah. Untuk menetapkan *batch size*

yang optimal dalam arti praktis, disarankan untuk memulai dengan *batch size* yang lebih kecil (32 atau 64), namun perlu diingat bahwa *batch size* yang lebih kecil memerlukan *learning rate* yang lebih kecil. Selain itu juga, mengusulkan untuk mencoba *batch size* yang lebih kecil terlebih dahulu. Tabel 3.2 menunjukkan *hyperparameter* yang digunakan pada tugas akhir ini.

Tabel 3.2: *Hyperparameter Tuning*

Parameter	Nilai
Learning rate	0.0001
Batch Size	64
Epoch	50
Dropout rate	0.5
Regularization strength	0.0001
KFold	5
Optimizer	Adam
Activation Function	Softmax

3.1.7 Metrik Uji

Metrik pengujian yang digunakan dalam melakukan pengujian algoritma adalah metrik yang juga digunakan pada penelitian-penelitian sebelumnya, seperti pada penelitian-penelitian yang dilakukan oleh [23, 42, 45]. Meliputi akurasi, presisi, spesifisiti, sensitiviti, dan skor F1.

Persamaan Akurasi

$$accuracy = \frac{TP + TN}{TP + FP + FN + TN} \quad (3.1)$$

Persamaan Presisi

$$precision = \frac{TP}{TP + FP} \quad (3.2)$$

Persamaan Sensitiviti

$$sensitivity = \frac{TP}{TP + FN} \quad (3.3)$$

Persamaan Spesifisiti

$$specificity = \frac{TN}{TN + FP} \quad (3.4)$$

Persamaan F1-Score

$$f1 - score = 2 \times \frac{precision \times sensitivity}{sensitivity + precision} \quad (3.5)$$

TP, TN, FP, dan FN adalah *true positive*, *true negative*, *false positive*, dan *false negative*. TP (True Positive) mengacu pada identifikasi yang akurat terhadap prediksi aritmia positif, sedangkan TN (True Negative) mengindikasikan keputusan yang benar terkait ketiadaan aritmia. Sebaliknya, FP (False Positive) mewakili prediksi palsu aritmia ketika pasien sebenarnya sehat, dan FN (False Negative) menunjukkan kesalahan klasifikasi pasien dengan aritmia sebagai sehat, yang dianggap sebagai hasil yang paling tidak menguntungkan.

3.1.8 Metode Pengujian

Untuk mengetahui keberhasilan seluruh rancangan model yang akan dibangun, diperlukan adanya pengujian, baik secara perangkat maupun algoritma. Hal ini ditujukan mengetahui apakah tujuan tugas akhir ini tercapai.

Tujuan Pengujian

Tujuan dari pengujian ini adalah untuk membandingkan algoritma *deep learning* dan mencari metode manakah yang memiliki akurasi tertinggi.

Skenario Pengujian

Berikut ini adalah algoritma *deep learning* yang digunakan :

Tabel 3.3: *Algoritma Deep Learning*

No	Metode
1	Convolutional Neural Network
2	Long Short Term Memory
3	Gated Recurrent Unit

Dengan menggunakan ketiga metode yang terdapat pada tabel 3.3 diatas, penelitian ini akan melakukan pengujian algoritma klasifikasi menggunakan skenario berikut :

1. Skenario 1 : Convolutional Neural Network

Pada skenario ini, data dibagi menjadi set pelatihan, pengujian, dan validasi, dengan 60% dialokasikan untuk pelatihan, 20% untuk pengujian, dan 20% untuk validasi. Set pengujian berisi 20.003 data points. Data mentah digunakan sebagai masukan, dan denoising dilakukan menggunakan *Sym4 wavelet function* untuk meningkatkan kompleksitas QRS. Normalisasi Z-Score diterapkan untuk membuat 360 sampel yang berpusat di sekitar puncak R yang diidentifikasi, dengan algoritma Pan-Tompkins digunakan untuk mengurangi noise dan mendeteksi

puncak R dalam setiap detak jantung. Teknik oversampling digunakan untuk mengatasi ketidakseimbangan kelas sebelum pelatihan model. Dataset diacak secara acak dan validasi silang *5-fold cross-validation* untuk meningkatkan generalisasi model dan meminimalkan bias selama proses pembagian data. Arsitektur yang diusulkan untuk *Convolutional Neural Network* (CNN) satu dimensi terdiri dari 12 lapisan, yang secara berurutan memproses tensor masukan untuk mengklasifikasikan data. Dimulai dengan lapisan konvolusi awal dengan ukuran 360 x 1, diikuti oleh lapisan rata-rata pooling dan lapisan konvolusi tambahan dengan berbagai ukuran kernel dan filter. Arsitektur ini termasuk lapisan dropout dan lapisan dense dengan regularisasi L2 untuk mengatasi *overfitting*. Lapisan softmax akhir berisi 5 neuron yang sesuai dengan 5 kelas untuk klasifikasi. Selama arsitektur ini, fungsi aktivasi ReLU digunakan dalam lapisan pooling, memfasilitasi non-linearitas dalam operasi pooling.

2. Skenario 2 : Long Short Term Memory

Pada skenario ini, Sebanyak 60% data digunakan untuk pelatihan, 20% untuk pengujian, dan 20% untuk validasi. Terdapat 20.003 titik data yang membentuk himpunan uji. Untuk meningkatkan kompleksitas QRS, dilakukan denoising menggunakan *Sym4 wavelet function* pada data mentah. Algoritma Pan-Tompkins digunakan untuk mengurangi noise dan mendeteksi puncak R pada setiap denyut jantung, dan Normalisasi Z-Score digunakan untuk menyusun 360 sampel berpusat pada puncak R yang terdeteksi. Sebelum melatih model, kelas-kelas minoritas mengalami oversampling untuk mengatasi masalah ketidakseimbangan kelas. Dataset kemudian diacak, dan untuk meningkatkan generalisasi model dan mengurangi bias selama pembagian data, diterapkan strategi *cross-validation 5-fold*. Arsitektur ini menggabungkan arsitektur LSTM dengan lapisan *fully connected* untuk klasifikasi model. Model ini mencakup lapisan LSTM awal dengan 64 neuron dan bentuk masukan (187,1). Model ini juga mencakup lapisan dense dengan 128 neuron menggunakan fungsi aktivasi *rectified linear unit* (ReLU), dan lapisan dropout dengan ambang batas 0.3 untuk menyederhanakan model. Akhirnya, ditambahkan lapisan dense dengan 5 neuron menggunakan fungsi aktivasi softmax sebagai lapisan terakhir.

3. Skenario 3 : Gated Recurrent Unit

Pada skenario ini, data dibagi menjadi tiga set: 20% untuk pengujian, 20% untuk validasi, dan 60% untuk pelatihan. Set pengujian terdiri dari 20.003 data points. Denoising diterapkan pada data deret waktu mentah menggunakan *Sym4 wavelet function* untuk meningkatkan QRS kompleks. Reduksi noise dan deteksi puncak R dilakukan pada setiap detak jantung menggunakan teknik Pan-Tompkins. Normalisasi Z-Score

digunakan untuk membuat 360 sampel yang berpusat pada puncak R yang diidentifikasi. Untuk mengatasi ketidakseimbangan kelas, teknik oversampling diterapkan pada kelas minoritas sebelum pelatihan model. Dataset dibagi secara acak, dan validasi silang *5-fold* dengan KFold digunakan untuk meningkatkan generalisasi model dan meminimalkan bias selama pemisahan data. Model menggabungkan lapisan GRU dengan *fully connected layer* untuk klasifikasi, mirip dengan arsitektur LSTM. Lapisan GRU diikuti oleh lapisan dense dengan 128 neuron menggunakan fungsi aktivasi ReLU dan bentuk input (187,1). Lapisan dropout dengan threshold 0.3 diperkenalkan untuk mengurangi kompleksitas model. Akhirnya, lapisan dense dengan lima neuron dan fungsi aktivasi softmax ditambahkan sebagai lapisan terakhir.

3.1.9 Perbandingan Hasil Penelitian

Tugas Akhir ini melakukan perbandingan hasil yang didapat dengan penelitian sejenis yang telah dilakukan oleh [15, 23, 46].

3.2 Ringkasan

Berdasarkan dari penjelasan metodologi di atas, dapat disimpulkan bahwa metode yang digunakan pada penelitian ini adalah metode *deep learning*. Alur dari penelitian ini adalah perancangan algoritma, pengujian algoritma, pembuatan model, pengujian model, dan penulisan laporan. Metode *deep learning* yang digunakan pada penelitian ini adalah *Convolutional Neural Network*, *Long Short Term Memory* dan *Gated Recurrent Unit*. Hasil dari penelitian ini akan dibandingkan dengan hasil penelitian-penelitian sejenis yang telah dilakukan oleh [15, 23, 46].

Bab IV

Hasil dan Pembahasan

Dalam bab ini, akan disajikan diskusi tentang hasil algoritma klasifikasi dan hasil yang diperoleh dari pengujian skenario yang dilakukan pada dataset MIT-BIH.

4.1 Hasil Objektif Pertama

Hasil objektif pertama telah dijelaskan dan dapat dilihat pada Bab 2.1. Bab penelitian terkait ini menguraikan hasil analisis studi literatur terkait beserta pembuatan taksonomi algoritma *deep learning* pada Gambar 2.1 yang telah dipilih berdasarkan fokus penelitian yang berkaitan dengan klasifikasi aritmia menggunakan sinyal EKG menggunakan *deep learning*. Penelitian-penelitian ini dipilih berdasarkan fokus penelitian yang berkaitan dengan klasifikasi aritmia menggunakan sinyal elektrokardiogram menggunakan *deep learning*. Penulis dapat mengenal *Convolutional Neural Network*, *Long Short Term Memory*, dan *Gated Recurrent Unit* lebih dalam berkat pemahaman yang lebih mendalam tentang taksonomi *deep learning* yang ditentukan. Penulis dapat menggunakan algoritma-algoritma tersebut untuk analisis penulis karena pemahaman perbedaan algoritma-algoritma tersebut serta fungsinya.

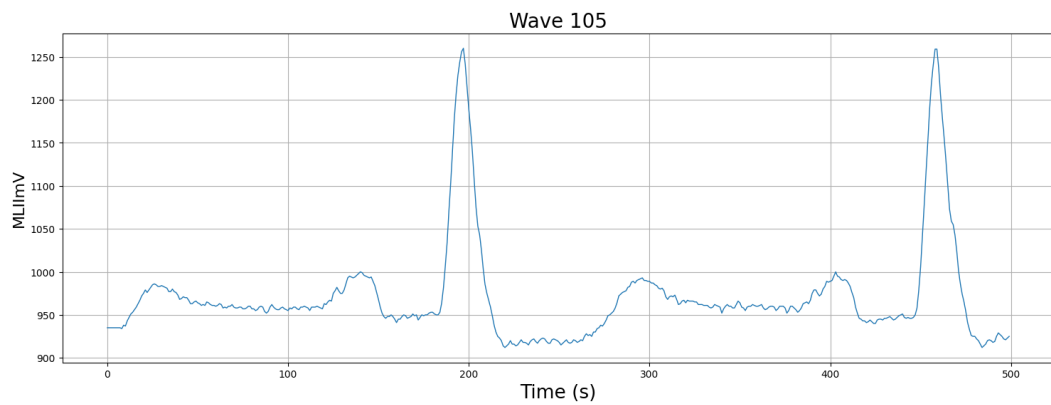
4.2 Hasil Objektif Kedua

Pada sub bab ini akan memaparkan hasil dari pengujian algoritma *deep learning*, yaitu *Convolutional Neural Network*, *Long Short Term Memory*, dan *Gated Recurrent Unit*.

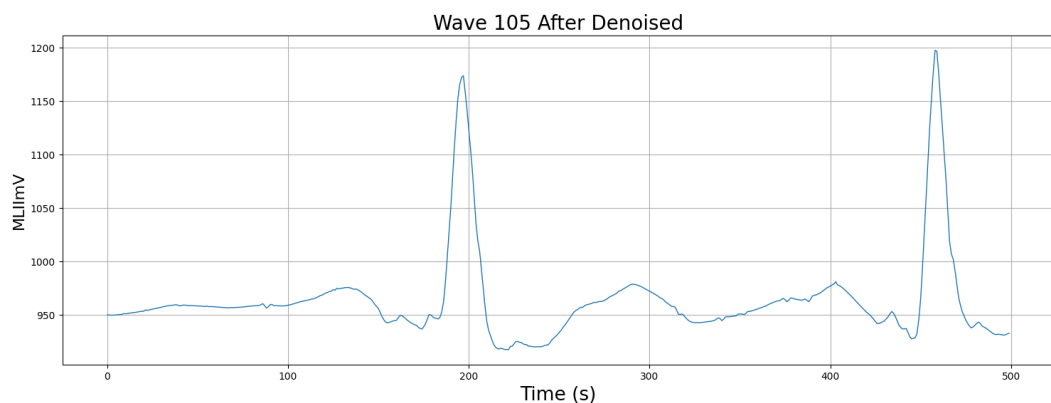
4.2.1 Preprocessing

Tugas akhir ini menggunakan basis data MIT-BIH arrhythmia, yang terdiri dari 48 rekaman sinyal EKG yang diambil dari 47 subjek (22 wanita dan 25 pria). Data diambil pada tingkat sampel 360 Hz dan diberi anotasi dengan lokasi setiap puncak R pada detak jantung. Basis data ini mengandung sekitar 109.000 detak jantung [33]. Setiap rekaman EKG memiliki dua sinyal inti, dan untuk tugas akhir ini, sinyal *lead II* digunakan. Elektrokardiografi adalah teknik untuk merekam aktivitas listrik jantung, tetapi kontaminasi sinyal oleh berbagai jenis noise umum terjadi. Metode penghilangan noise seperti filter

bandpass, filter low-pass, dan *wavelet transform* banyak digunakan. Tugas akhir ini menggunakan wavelet Sym4 dengan pendekatan *adaptive threshold* untuk pra-pemrosesan sinyal EKG, terutama untuk deteksi QRS kompleks yang akurat [3]. *Wavelet transform* menangkap informasi lokal spektral dan temporal, cocok untuk sinyal dengan osilasi yang jelas. *Adaptive threshold* menyesuaikan *threshold* secara dinamis berdasarkan karakteristik sinyal untuk deteksi fitur yang tepat. Proses ini melibatkan *discrete wavelet transformation* satu dimensi multilevel pada data input menggunakan fungsi Sym4 dan menentukan tingkat dekomposisi yang paling maksimal. Kemudian, *threshold* diterapkan pada koefisien, diikuti oleh *inverse discrete wavelet transformation* untuk mendapatkan hasil yang sudah dihilangkan noise. Gambar 4.1 dan 4.2 menggambarkan sinyal EKG sebelum dan setelah pra-pemrosesan.



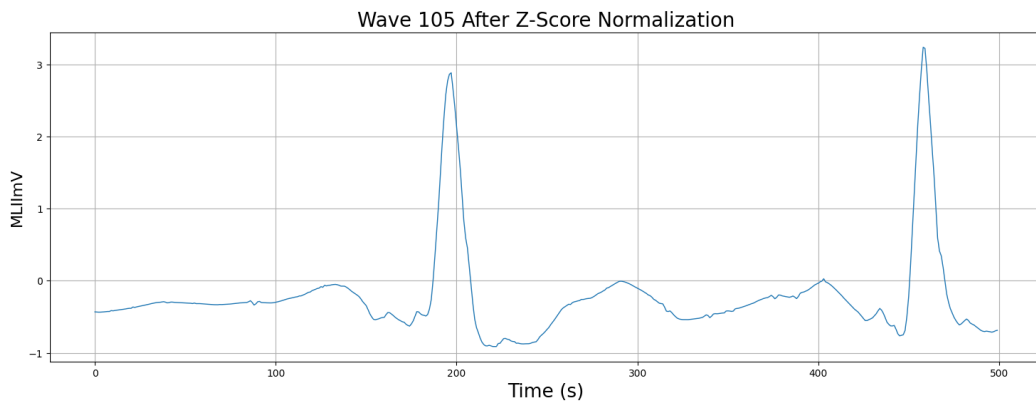
Gambar 4.1: Contoh sinyal EKG sebelum pra-pemrosesan yang menampilkan sinyal ECG mentah.



Gambar 4.2: Contoh sinyal EKG setelah pra-pemrosesan yang menggambarkan sinyal ECG setelah noise dihilangkan.

4.2.2 Ekstraksi Fitur

Algoritma Pan-Tompkins digunakan untuk mendeteksi puncak R pada setiap detak jantung menggunakan *single lead* dari kumpulan data. Normalisasi Z-Score diterapkan untuk membagi data menjadi 360 sampel yang berpusat pada puncak R yang teridentifikasi. Algoritma ini menekankan komponen frekuensi yang terkait dengan depolarisasi ventrikel yang cepat menggunakan filter bandpass dan memperbesar amplitudo sinyal untuk meningkatkan QRS kompleks. Teknik *adaptive threshold* dengan akurasi tinggi mengidentifikasi puncak pada sinyal yang difilter, sehingga mendeteksi QRS kompleks. Sinyal EKG diambil dengan kecepatan 360 sampel per detik untuk analisis yang tepat. Untuk meningkatkan akurasi klasifikasi dan identifikasi puncak R, digunakan filter bandpass, diikuti oleh filter derivatif dan filter *squaring* untuk menghilangkan noise dari sinyal. *Moving window integrator* kemudian dilakukan untuk mengidentifikasi puncak R yang sebenarnya pada sinyal-sinyal tersebut. Sinyal setelah normalisasi Z-Score ditunjukkan pada Gambar 4.3.



Gambar 4.3: Sinyal EKG setelah normalisasi Z-Score.

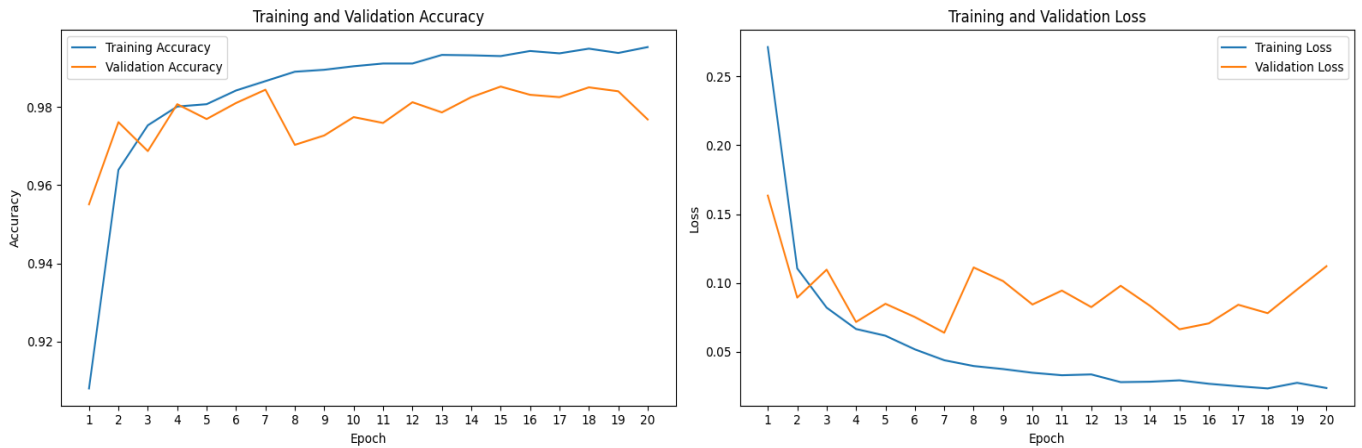
4.2.3 Hasil Algoritma Klasifikasi

Algoritma klasifikasi deep learning yang digunakan dalam penelitian ini meliputi *Convolutional Neural Network* (CNN), *Long Short Term Memory* (LSTM), dan *Gated Recurrent Unit* (GRU). Algoritma-algoritma yang diusulkan bertujuan untuk mendeteksi sinyal EKG. Dalam tugas akhir ini, algoritma klasifikasi telah dikembangkan untuk mendeteksi sinyal EKG, dan tujuan dari tugas akhir ini adalah untuk menilai kesesuaian dan efektivitasnya dalam mengklasifikasikan aritmia menggunakan basis data MIT-BIH.

1. Hasil Model Convolutional Neural Network

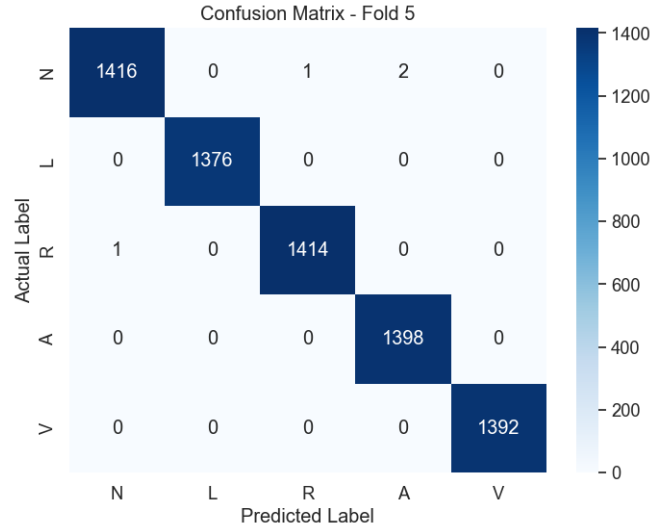
Hasil dari model *Convolutional Neural Network* yang telah diterapkan untuk mendeteksi 5 kelas aritmia yaitu N, L, R, A, dan V akan dijelaskan pada

bagian penjelasan hasil dan pembahasan model *Convolutional Neural Network*. Penjelasan hasil akan dibagi menjadi dua bagian. Hasil metode pelatihan dan validasi model *Convolutional Neural Network* akan ditinjau terlebih dahulu. Pembahasan ini akan mencakup analisis performa model pada *train set* selama fase pelatihan, serta pemeriksaan performa model pada set validasi. Kedua, kita akan membahas hasil metode *hyperparameter tuning* yang digunakan pada model *Convolutional Neural Network*.



Gambar 4.4: Grafik *accuracy* dan *loss* model *convolutional neural network* selama *training* dan *validation*.

Gambar 4.4 menunjukkan hasil selama proses *iterative optimization* yang berlangsung selama 20 *epoch*, *Convolutional Neural Network* (CNN) menunjukkan peningkatan yang dalam kemampuan prediktifnya pada kumpulan data *training*. Hal ini dibuktikan secara kuantitatif dengan peningkatan *training accuracy*, yang dimulai dari nilai awal sebesar 90,8% dan mencapai puncaknya pada angka 99,53%. Pada saat yang sama, *training loss* model, yaitu metrik yang mengukur perbedaan antara prediksi model dan label sebenarnya, menunjukkan penurunan yang signifikan, bertransisi dari nilai awal sebesar 0,2713 ke 0,0236 setelah selesainya proses *training*. Performa CNN mencapai puncaknya pada epoch ke-15 dengan akurasi 98,52% saat diuji pada kumpulan data *validation*. Namun, pada epoch ke-20, terjadi sedikit penurunan sehingga *validation accuracy* mencapai 97,68%. Variasi dalam metrik *validation* tersebut menunjukkan respons model terhadap distribusi data yang berbeda dan kapasitasnya untuk menggeneralisasi data yang tidak terlihat. *Validation loss*, yang merupakan indikasi penting lainnya mengenai performa model pada data baru, mengikuti tren non-monotonik selama beberapa *epochs*, dan mencapai puncaknya pada 0,1121 pada *epoch* terakhir.



Gambar 4.5: Fold 5 dari *confusion matrix* model *convolutional neural network*

Confusion Matrix yang ditampilkan pada Gambar 4.5, yang dihasilkan dari fold ke-5 selama proses *training*, terdapat fokus yang jelas di sepanjang diagonal utama, menyoroti kemampuan model *Convolutional Neural Network* yang lebih baik dalam klasifikasi aritmia. Khususnya kelas L, A, dan V, menampilkan akurasi klasifikasi yang sempurna, terbukti dengan tidak adanya kesalahan off-diagonal dalam kategori tersebut. Kelas N, sebaliknya, memiliki kesalahan klasifikasi yang kecil.

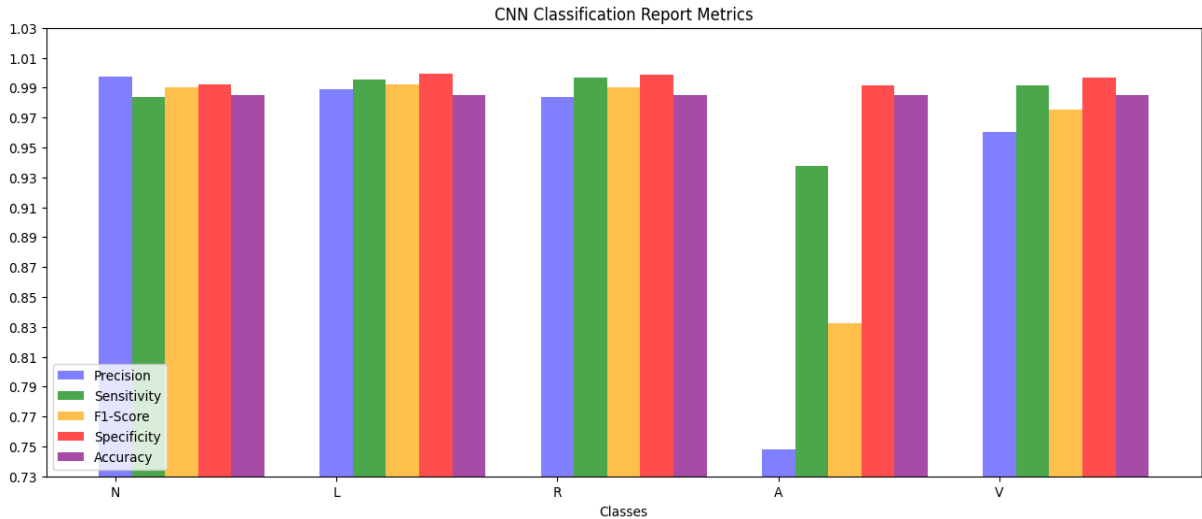
Tabel 4.1 menunjukkan hasil performansi model *Convolutional Neural Network* secara keseluruhan, dan Tabel 4.2 menunjukkan hasil performansi setiap kelas.

Tabel 4.1: Metrik performansi *Convolutional Neural Network* (%).

Accuracy	Precision	Sensitivity	Specificity	F1-Score
98.49%	98.66%	98.49%	98.97%	98.54%

Tabel 4.2: Metrik performansi *Convolutional Neural Network* untuk klasifikasi aritmia berdasarkan kelas (%).

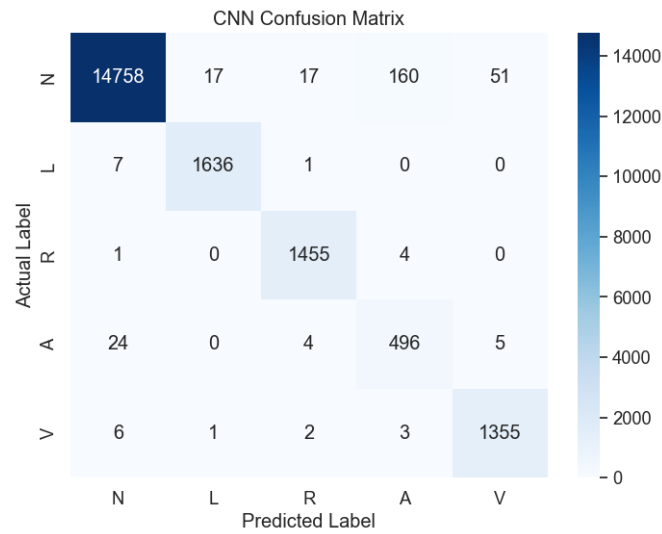
Kelas	Accuracy	Precision	Sensitivity	Specificity	F1-Score
N	98.49%	99.74%	98.37%	99.24%	99.05%
L	98.49%	98.91%	99.51%	99.90%	99.21%
R	98.49%	98.38%	99.66%	99.87%	99.01%
A	98.49%	74.81%	93.76%	99.14%	83.22%
V	98.49%	96.03%	99.12%	99.70%	97.55%



Gambar 4.6: Grafik metrik performansi model *convolutional neural network*.

Tabel 4.2 dan grafik pada Gambar 4.6 menunjukkan bahwa model CNN mencapai akurasi rata-rata 98,49%, yang berarti model tersebut mengklasifikasikan sekitar 98 dari 100 detak jantung dengan benar. Akurasi tertinggi dicapai pada kelas L sebesar 99,51% dan akurasi terendah dicapai pada kelas A sebesar 93,76%. Presisi rata-ratanya adalah 98,66%, yang berarti sekitar 97 dari 100 detak jantung positif yang diprediksi benar positif. Presisi tertinggi dicapai pada kelas N sebesar 99,74% dan presisi terendah dicapai pada kelas A sebesar 74,81%. Sensitivitas rata-rata adalah 98,49%, yang berarti sekitar 98 dari 100 detak jantung positif sebenarnya diklasifikasikan dengan benar sebagai positif. Sensitivitas tertinggi diperoleh pada kelas R sebesar 99,66% dan sensitivitas terendah diperoleh pada kelas N sebesar 98,37%. Spesifisitas

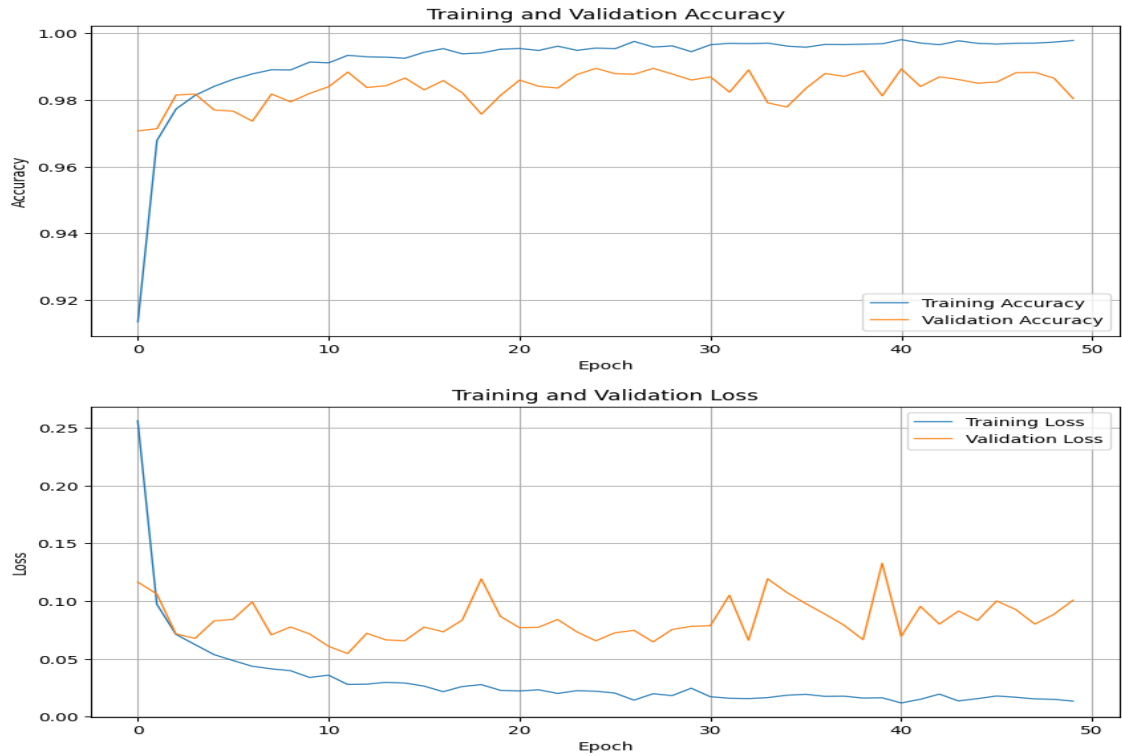
rata-rata adalah 98,97%, yang berarti sekitar 98 dari 100 detak jantung negatif sebenarnya diklasifikasikan dengan benar sebagai negatif. Spesifisitas tertinggi dicapai pada kelas L dan R sebesar 99,90%, dan spesifisitas terendah dicapai pada kelas N sebesar 99,24%. Rata-rata skor F1 adalah 98,54%, yang berarti model mencapai keseimbangan yang baik antara presisi dan sensitivitas. Nilai F1 tertinggi diraih kelas L sebesar 99,21% dan nilai F1 terendah diraih kelas A sebesar 83,22%.



Gambar 4.7: Confusion matrix dari model *convolutional neural network*

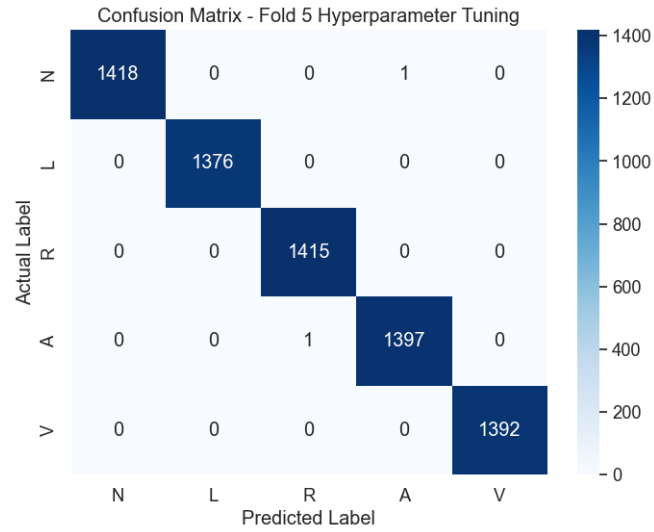
Confusion Matrix yang ditampilkan pada Gambar 4.6, yang dihasilkan dari *testing* untuk mengevaluasi kinerja model setelah dilatih. Matriks tersebut menggambarkan seberapa cocok prediksi model dengan klasifikasi sebenarnya untuk setiap jenis detak jantung. Khususnya, model ini unggul dalam mengidentifikasi kelas N dengan benar, dengan 14758 prediksi yang benar. Dengan 1636 klasifikasi yang benar, juga menunjukkan kinerja dalam mengenali kelas L. Selain itu, model tersebut menunjukkan akurasi dalam mengidentifikasi kelas R dan A, dengan masing-masing 1.455 dan 496 prediksi akurat. Model tersebut berhasil mendeteksi secara tepat 1.355 kemunculan kelas V.

Hyperparameter Tuning



Gambar 4.8: Grafik *accuracy* dan *loss* model selama *training* dan *validation* setelah *hyperparameter tuning*.

Selama 50 epoch yang ditampilkan pada Gambar 4.8, *Convolutional Neural Network* (CNN) menunjukkan peningkatan kemampuan pada data *training* dengan *training accuracy* meningkat dari 91,34% ke 99,78%. *Training loss* juga menurun dari 0,2562 ke 0,0134. Meski performa tertinggi pada data *validation* tercapai di epoch ke-12 dengan akurasi 98,84%, akurasi ini menurun menjadi 98,04% di epoch ke-50. *Validation loss* mencapai puncak 0,1008 di epoch terakhir.



Gambar 4.9: Fold 5 dari *confusion matrix* model CNN setelah *hyperparameter tuning*.

Berdasarkan *Confusion Matrix* pada Gambar 4.9 dari fold ke-5, model *Convolutional Neural Network* menunjukkan kinerja klasifikasi yang sangat baik dengan fokus pada diagonal utama. Kesalahan klasifikasi hampir tidak ada, menunjukkan performa optimal model pada fold tersebut.

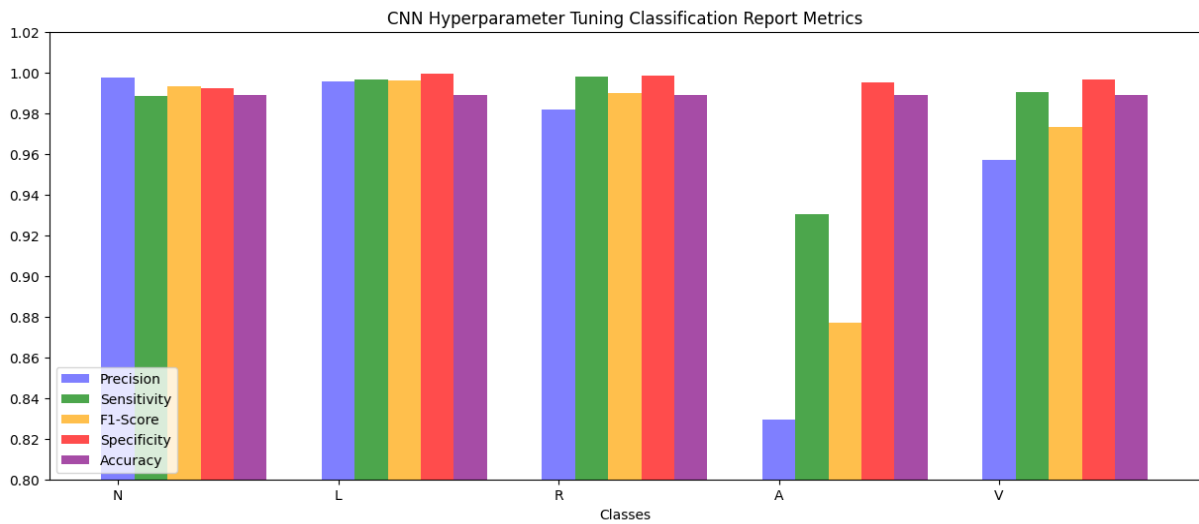
Tabel 4.5 menunjukkan hasil performansi model *Convolutional Neural Network* setelah *hyperparameter tuning* secara keseluruhan, dan Tabel 4.6 menunjukkan hasil performansi untuk setiap kelas.

Tabel 4.5: Metrik performansi *convolutional neural network* setelah *hyperparameter tuning* (%).

Accuracy	Precision	Sensitivity	Specificity	F1-Score
98.87%	98.93%	98.87%	99.56%	98.89%

Tabel 4.6: Metrik performansi *convolutional neural network* untuk klasifikasi aritmia berdasarkan kelas setelah *hyperparameter tuning* (%).

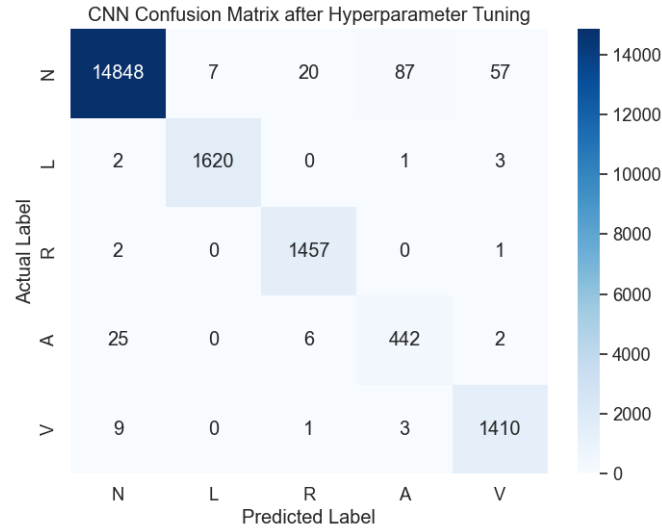
Kelas	Accuracy	Precision	Sensitivity	Specificity	F1-Score
N	98.87%	99.74%	98.86%	99.24%	99.30%
L	98.87%	99.57%	99.63%	99.96%	99.60%
R	98.87%	98.18%	99.79%	99.85%	98.98%
A	98.87%	82.93%	93.05%	99.53%	87.70%
V	98.87%	95.72%	99.02%	99.66%	97.30%



Gambar 4.10: Grafik metrik performansi model *convolutional neural network* setelah *hyperparameter tuning*.

Berdasarkan Tabel 4.5, 4.6, dan grafik pada Gambar 4.10. Model *Convolutional Neural Network* (CNN) setelah *hyperparameter tuning* menampilkan akurasi keseluruhan 98,87%. Artinya, model ini dapat mengklasifikasikan sekitar 99 dari 100 detak jantung dengan tepat. Presisi rata-rata sebesar 98,93% dan sensitivitas rata-rata sebesar 98,87% menandakan keandalan model dalam memprediksi dan mengidentifikasi detak jantung positif. Dengan spesifisitas 99,56% dan skor F1 rata-rata 98,89%, model mencapai keseimbangan antara presisi dan sensitivitas. Pada Tabel 4.7, kelas L menunjukkan performa klasifikasi yang paling unggul dengan presisi 99,57% dan skor F1 99,60%. Di

sisi lain, meskipun kelas A memiliki presisi yang lebih rendah (82,93%), skor F1-nya tetap tinggi (87,70%), menunjukkan keseimbangan antara presisi dan sensitivitas. Kelas R memiliki sensitivitas tertinggi, sebesar 99,79%.

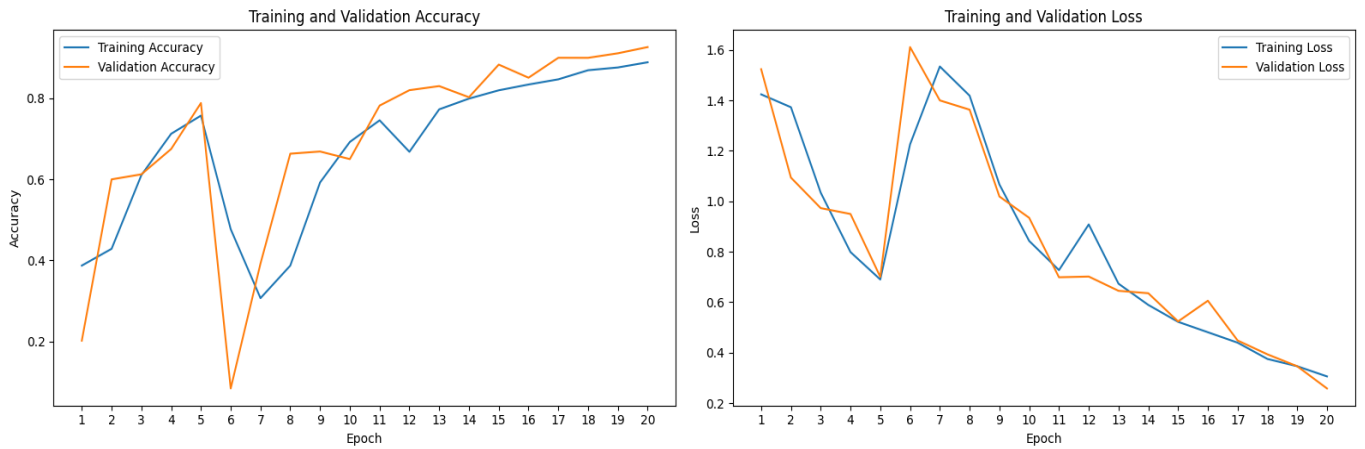


Gambar 4.11: *Confusion Matrix* model CNN setelah *hyperparameter tuning*.

Berdasarkan *Confusion Matrix* pada Gambar 4.9 setelah *hyperparameter tuning*, model menunjukkan kemampuan tinggi dalam mengklasifikasikan detak jantung. Khususnya, model berhasil mengidentifikasi kelas N dengan akurasi mendekati sempurna, dengan 14.848 detak jantung diklasifikasikan dengan benar. Performa serupa terlihat pada kelas L, R, A, dan V. Meskipun ada beberapa kesalahan klasifikasi, keseluruhan hasil menunjukkan bahwa model memiliki kemampuan klasifikasi yang impresif untuk semua kelas, dengan kesalahan yang minim.

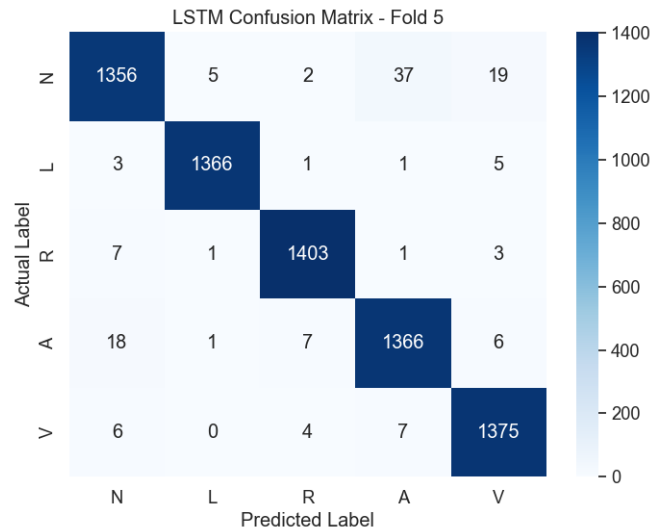
2. Hasil Model Long Short Term Memory

Hasil dari model *Long Short Term Memory* yang telah diterapkan untuk mendeteksi 5 kelas aritmia yaitu N, L, R, A, dan V akan dijelaskan pada bagian penjelasan hasil dan pembahasan model *Long Short Term Memory*. Penjelasan hasil akan dibagi menjadi dua bagian. Hasil metode pelatihan dan validasi model *Long Short Term Memory* akan ditinjau terlebih dahulu. Pembahasan ini akan mencakup analisis performa model pada *train set* selama fase pelatihan, serta pemeriksaan performa model pada set validasi. Kedua, kita akan membahas hasil metode *hyperparameter tuning* yang digunakan pada model *Long Short Term Memory*.



Gambar 4.12: Grafik *accuracy* dan *loss* model *long short term memory* selama *training* dan *validation*.

Gambar 4.12 yang menampilkan *training loss* dari model LSTM selama proses *iterative optimization* menunjukkan bahwa model mengalami peningkatan dalam kemampuan prediktifnya pada kumpulan data *training*. Secara kuantitatif, *training loss*, yaitu metrik yang mengukur perbedaan antara prediksi model dan label sebenarnya, bertransisi dari nilai awal yang relatif tinggi pada epoch pertama, menuju ke nilai yang jauh lebih rendah pada epoch ke-20. Penurunan ini menandakan bahwa model semakin mendekati prediksi yang sesuai dengan label sebenarnya saat berlangsungnya proses pelatihan. Dalam gambar yang menampilkan *training accuracy*, kita dapat melihat bahwa model LSTM menunjukkan peningkatan yang konsisten dalam akurasinya. Dimulai dari nilai awal sekitar 38,69% pada epoch pertama, akurasi pelatihan meningkat signifikan, mencapai puncaknya di angka 88,88% pada epoch ke-20. Peningkatan ini menggambarkan bagaimana model semakin memahami pola dalam data *training* dan menjadi lebih handal dalam membuat prediksi yang benar. Namun, saat kita evaluasi performa model pada kumpulan data validasi, terlihat adanya variasi dalam metrik validasi selama proses pelatihan. Pada beberapa epoch, terjadi penurunan dalam *validation accuracy*, menunjukkan bahwa model mungkin sedang mengalami kesulitan dalam menggeneralisasi data yang tidak terlihat. Namun, kecenderungan umumnya adalah peningkatan, dengan *validation accuracy* mencapai 92,63% pada epoch ke-20. Ini menunjukkan bahwa, meskipun ada beberapa fluktuasi, model secara keseluruhan semakin baik dalam mengenali dan mengklasifikasikan data yang tidak dikenal.



Gambar 4.13: Fold 5 *confusion matrix* model *long short term memory*.

Pada Gambar 4.13 dapat dilihat bahwa diagonal utama matriks dari fold ke-5, yang merepresentasikan prediksi yang benar untuk setiap kelas, memiliki nilai yang cukup tinggi, menunjukkan performa klasifikasi yang sangat baik. Terutama kelas R dengan 1403 prediksi yang benar dan hanya sedikit kesalahan klasifikasi. Namun, kelas pertama memiliki beberapa kesalahan off-diagonal, terutama dengan 37 kesalahan klasifikasi ke kelas A dan 19 kesalahan ke kelas V. Meskipun demikian, jika kita melihat proporsi kesalahan ini terhadap jumlah total prediksi untuk kelas pertama, kesalahan ini relatif kecil.

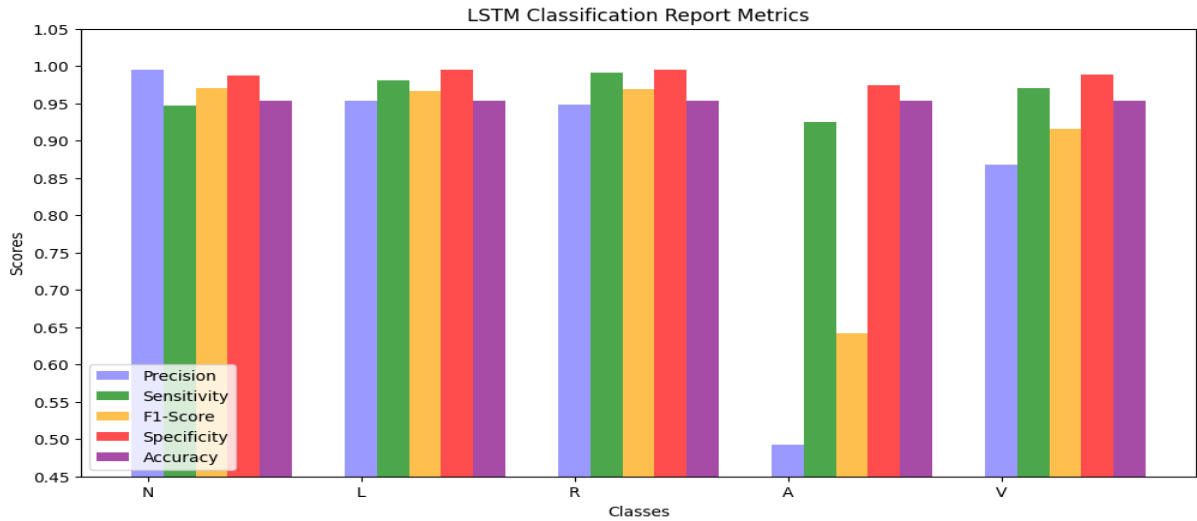
Tabel 4.9 menunjukkan hasil performansi model *Long Short Term Memory* secara keseluruhan, dan Tabel 4.10 menunjukkan hasil performansi setiap kelas.

Tabel 4.9: Metrik performansi *long short term memory* (%).

Accuracy	Precision	Sensitivity	Specificity	F1-Score
95.36%	96.66%	95.36%	96.18%	95.76%

Tabel 4.10: Metrik performansi *long short term memory* untuk klasifikasi aritmia berdasarkan kelas (%).

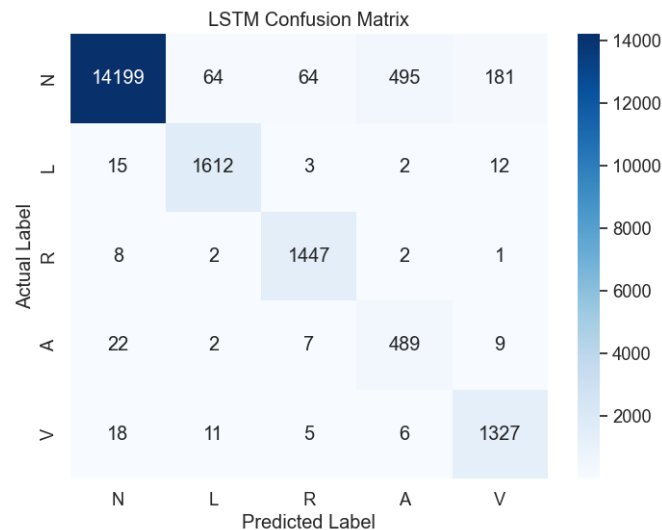
Kelas	Accuracy	Precision	Sensitivity	Specificity	F1-Score
N	95.36%	99.56%	94.64%	98.74%	97.04%
L	95.36%	95.33%	98.05%	99.57%	96.67%
R	95.36%	94.82%	99.11%	99.57%	96.92%
A	95.36%	49.20%	92.44%	97.41%	64.22%
V	95.36%	86.73%	97.07%	98.91%	91.61%



Gambar 4.14: Grafik metrik performansi model *long short term memory*.

Model *Long Short Term Memory* (LSTM) yang ditunjukkan pada Tabel 4.9 dan grafik pada Gambar 4.14 mencapai akurasi sebesar 95,36%, yang berarti model tersebut mengklasifikasikan sekitar 95 dari 100 detak jantung dengan benar. Presisi adalah 96,66%, mengimplikasikan bahwa sekitar 97 dari 100 detak jantung positif yang diprediksi adalah benar positif. Sensitivitasnya adalah 95,36%, artinya sekitar 95 dari 100 detak jantung positif sebenarnya diklasifikasikan dengan benar sebagai positif. Spesifisitasnya adalah 96,18%, yang menandakan sekitar 96 dari 100 detak jantung negatif sebenarnya diklasifikasikan dengan benar sebagai negatif. Skor F1 adalah 95,76%, menunjukkan bahwa model mencapai keseimbangan yang baik antara presisi dan sensitivitas.

Berdasarkan Tabel 4.11, kelas N memiliki akurasi sebesar 95,36% dengan presisi tertinggi sebesar 99,56% dan skor F1 sebesar 97,04%. Kelas L mencatatkan akurasi yang sama, 95,36%, dengan sensitivitas tertinggi di antara semua kelas yaitu 98,05% dan skor F1 sebesar 96,67%. Kelas R, dengan akurasi 95,36%, memiliki sensitivitas sebesar 99,11% dan skor F1 sebesar 96,92%. Namun, kelas A menunjukkan presisi terendah yaitu 49,20% dan skor F1 terendah yaitu 64,22%, meskipun akurasinya tetap stabil di 95,36%. Sementara itu, kelas V memiliki presisi sebesar 86,73%, sensitivitas 97,07%, dan skor F1 sebesar 91,61% dengan akurasi yang sama, 95,36%. Dari data di atas, dapat disimpulkan bahwa kelas N unggul dalam presisi, sementara kelas L dan R menonjol dalam sensitivitas dan spesifisitas. Sebaliknya, kelas A tampaknya menjadi titik lemah dalam model dengan presisi dan skor F1 yang relatif rendah.

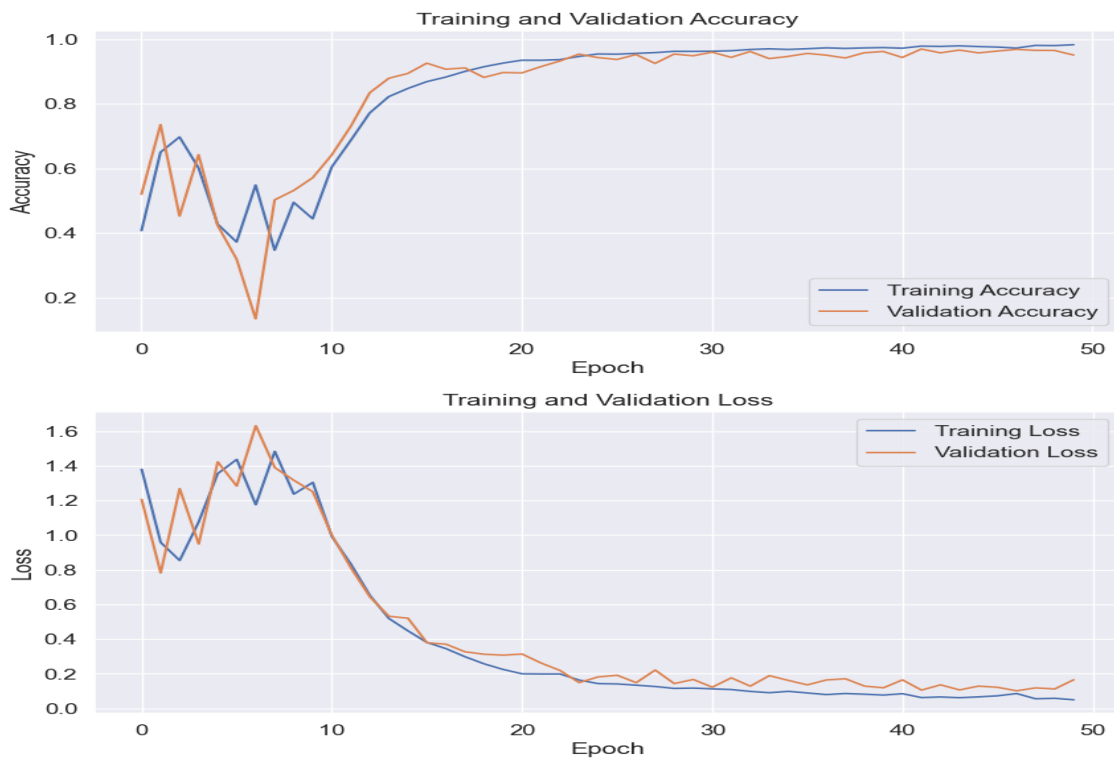


Gambar 4.15: *Confusion matrix* model long short term memory

Confusion Matrix pada Gambar 4.15 yang dihasilkan dari *testing* untuk mengevaluasi kinerja model setelah dilatih, menunjukkan bagaimana model LSTM memprediksi kelas aritmia. Model ini menunjukkan keunggulannya dalam mengidentifikasi kelas N dengan 14,199 prediksi yang benar. Meskipun demikian, terdapat beberapa kesalahan, dengan 64 detak yang salah diklasifikasikan ke kelas L, 64 detak ke kelas R, 495 detak ke kelas A, dan 181 detak ke kelas V. Pada kelas L, model berhasil mendeteksi 1,612 detak. Namun, beberapa detak mengalami kesalahan klasifikasi, seperti 15 detak yang dikategorikan sebagai kelas L, 3 detak ke kelas R, 2 detak ke kelas A, dan 12 detak ke kelas V. Pada kelas R, model memprediksi 1,447 detak dengan presisi. Hanya ada sedikit kesalahan, dengan 8 detak yang diklasifikasikan sebagai kelas N, 2 detak ke kelas L, serta 2 detak dan 1 detak yang salah ditempatkan di

kelas A dan V. Dalam kelas A, model menunjukkan kemampuannya dengan mengidentifikasi 489 detak dengan benar. Beberapa kesalahan terjadi, yaitu 22 detak diklasifikasikan sebagai kelas N, 2 detak ke kelas L, 7 detak ke kelas R, dan 9 detak ke kelas V. Untuk kelas V, model cukup akurat dengan 1,327 detak yang benar dikenali. Namun, terdapat beberapa kesalahan, dengan 18 detak yang dikategorikan sebagai kelas N, 11 detak ke kelas L, 5 detak ke kelas R, dan 6 detak ke kelas A.

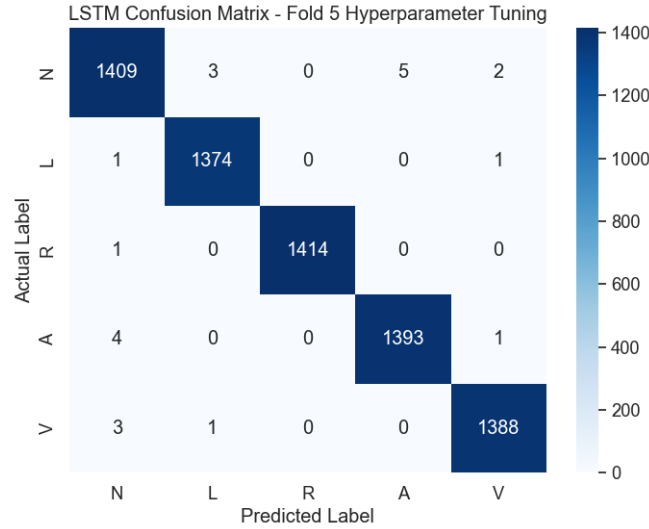
Hyperparameter Tuning



Gambar 4.16: Grafik *accuracy* dan *loss* model *long short term memory* selama *training* dan *validation* setelah *hyperparameter tuning*.

Gambar 4.16 menggambarkan proses optimalisasi model LSTM selama 50 epoch pasca-*hyperparameter tuning*. Pada awalnya, model memiliki akurasi pelatihan sebesar 40,77% dan *loss* sebesar 1,3799. Namun, di akhir iterasi ke-50, akurasi meningkat signifikan menjadi 98,35%, dengan *loss* menurun drastis menjadi 0,0498. Dalam konteks data validasi, puncak performa model dicapai pada epoch ke-42 dengan akurasi 97,02%, namun sedikit menurun menjadi 95,15% di epoch ke-50. Selain itu, *validation loss* mencapai nilai terendah di epoch ke-47 sebesar 0,1018, tetapi meningkat menjadi 0,1652 di epoch terakhir. Setelah melakukan *hyperparameter tuning*, model LSTM menunjukkan

peningkatan kinerja yang signifikan, meski ada fluktuasi pada metrik validasi. Model secara keseluruhan menunjukkan tren positif dalam meningkatkan akurasi dan menurunkan *loss*.



Gambar 4.17: Fold 5 dari *Confusion Matrix* LSTM setelah *hyperparameter tuning*.

Dari *Confusion Matrix* untuk Fold ke-5 pada Gambar 4.17, model LSTM setelah *hyperparameter tuning* menunjukkan kemampuan klasifikasi yang cukup baik. Kelas N berhasil mengklasifikasikan 1409 sampel dengan benar namun memiliki 10 kesalahan klasifikasi. Kelas L memiliki 1374 klasifikasi yang benar dengan hanya 2 kesalahan. Menonjol di antara semua kelas, kelas R menunjukkan kinerja yang hampir sempurna dengan 1414 klasifikasi yang benar dan hanya satu kesalahan. Kelas A dan V juga menunjukkan kinerja yang kuat dengan 1393 dan 1388 klasifikasi yang benar serta 5 dan 4 kesalahan klasifikasi masing-masing.

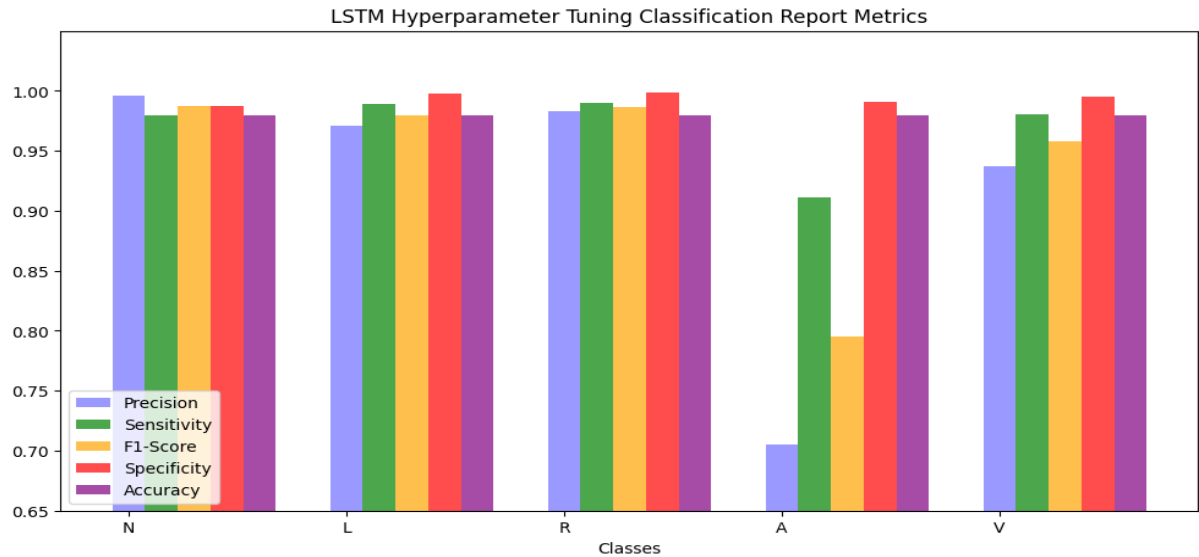
Tabel 4.13 menunjukkan hasil performansi model *Long Short Term Memory* setelah *hyperparameter tuning* secara keseluruhan, dan Tabel 4.14 menunjukkan hasil performansi untuk setiap kelas.

Tabel 4.13: Metrik performansi *long short term memory* setelah *hyperparameter tuning* (%).

Accuracy	Precision	Sensitivity	Specificity	F1-Score
97.94%	98.17%	97.94%	97.27%	98.01%

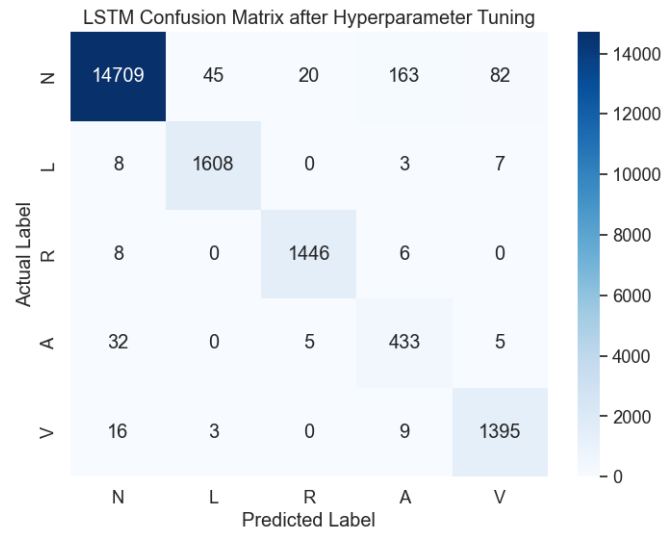
Tabel 4.14: Metrik performansi *long short term memory* untuk klasifikasi aritmia berdasarkan kelas setelah *hyperparameter tuning* (%).

Kelas	Accuracy	Precision	Sensitivity	Specificity	F1-Score
N	97.94%	99.57%	97.94%	98.72%	98.74%
L	97.94%	97.10%	98.89%	99.74%	97.99%
R	97.94%	98.30%	99.04%	99.87%	98.67%
A	97.94%	70.52%	91.16%	99.07%	79.52%
V	97.94%	93.69%	98.03%	99.49%	95.81%



Gambar 4.18: Grafik metrik performansi model *long short term memory* setelah *hyperparameter tuning*.

Model *Long Short Term Memory* (LSTM) setelah *hyperparameter tuning* mencapai akurasi rata-rata 97,94% dengan presisi 98,17%, seperti yang ditunjukkan pada Tabel 4.13. Skor F1 rata-ratanya adalah 98,01%, menunjukkan keseimbangan yang baik antara presisi dan sensitivitas. Dari Tabel 4.15 dan grafik pada Gambar 4.18 kelas N memiliki presisi tertinggi yaitu 99,57%. Kelas L menonjol dengan sensitivitas tertinggi yaitu 98,89%, sedangkan kelas R memiliki spesifisitas tertinggi, 99,87%. Namun, kelas A menjadi titik lemah dengan presisi hanya 70,52% dan skor F1 terendah yaitu 79,52%. Meskipun presisi kelas A meningkat signifikan setelah tuning, ini tetap merupakan presisi terendah di antara semua kelas, peningkatannya menunjukkan bahwa *tuning* membantu model dalam mengidentifikasi kelas A dengan lebih baik.

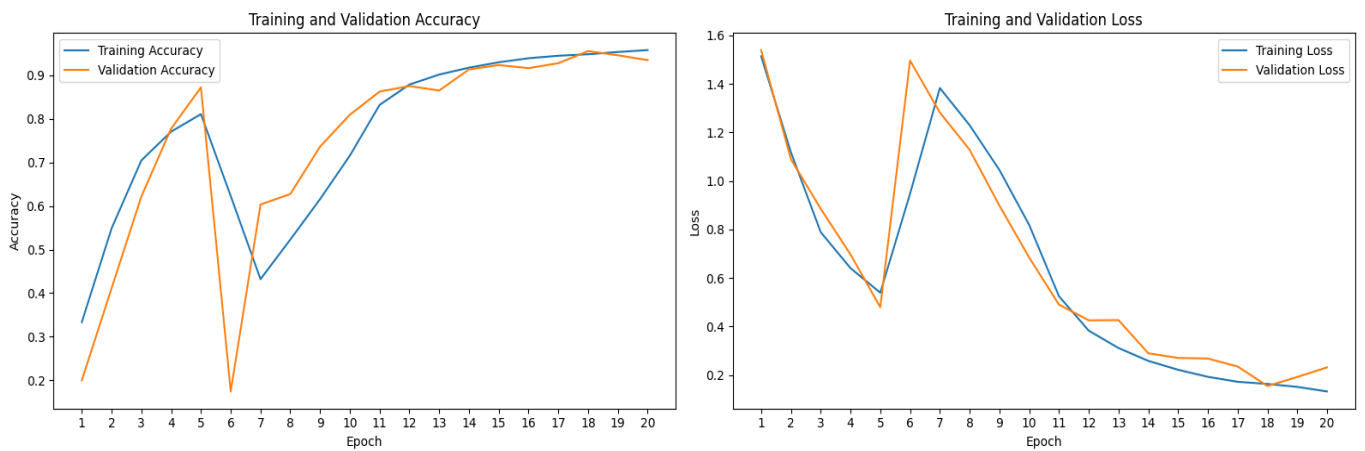


Gambar 4.19: *Confusion matrix* LSTM setelah *hyperparameter tuning*.

Setelah melalui proses *hyperparameter tuning*, *confusion matrix* yang ditampilkan pada Gambar 4.19 menyoroti peningkatan kinerja model LSTM dalam memprediksi jenis detak jantung. Model ini kini lebih unggul dalam mengidentifikasi kelas N dengan 14,709 prediksi yang benar. Namun, masih ada beberapa kesalahan; misalnya, 45 detak dari kelas N yang diklasifikasikan sebagai kelas L. Di kelas L, model menunjukkan akurasi yang lebih meningkat dengan 1,608 prediksi benar, meskipun terdapat 8 detak yang diklasifikasikan sebagai kelas N. Kelas R menunjukkan performa yang solid dengan 1,446 prediksi benar dan kesalahan yang minimal. Sementara kelas A dan V, setelah penyesuaian, masing-masing berhasil memprediksi dengan benar sebanyak 433 dan 1,395 detak. Keseluruhan hasil, setelah *tuning*, menegaskan bahwa model LSTM kini lebih efisien dan akurat dalam mengenali jenis detak jantung.

3. Hasil Model Gated Recurrent Unit

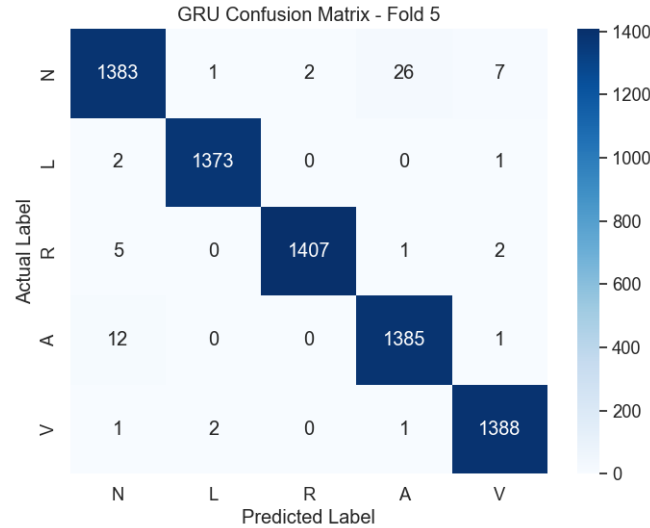
Hasil dari model *Gated Recurrent Unit* yang telah diterapkan untuk mendeteksi 5 kelas aritmia yaitu N, L, R, A, dan V akan dijelaskan pada bagian penjelasan hasil dan pembahasan model *Gated Recurrent Unit*. Penjelasan hasil akan dibagi menjadi dua bagian. Hasil metode pelatihan dan validasi model *Gated Recurrent Unit* akan ditinjau terlebih dahulu. Pembahasan ini akan mencakup analisis performa model pada *train set* selama fase pelatihan, serta pemeriksaan performa model pada set validasi. Kedua, kita akan membahas hasil metode *hyperparameter tuning* yang digunakan pada model *Gated Recurrent Unit*.



Gambar 4.20: Grafik *accuracy* dan *loss* model *gated recurrent unit* selama *training* dan *validation*.

Gambar 4.20 menunjukkan hasil selama proses *iterative optimization* yang berlangsung selama 20 epoch menunjukkan bahwa model *Gated Recurrent Unit* (GRU) mengalami berbagai tingkat kinerja pelatihan. Mulai dengan *training accuracy*, model memulai dengan akurasi sebesar 33,34% pada epoch pertama. Selama proses *training*, dan mencapai akurasi maksimum sebesar 95,81% pada epoch ke-20. Bersamaan dengan itu, *training loss*, yang mengukur perbedaan antara prediksi model dan label sebenarnya, menunjukkan tren menurun secara umum. Model memulai dengan *loss* sebesar 1,5142 pada epoch awal dan berhasil menguranginya menjadi 0,1324 pada epoch terakhir. Ketika mengevaluasi kinerja model pada dataset validasi, model mencapai puncak akurasinya sebesar 95,58% pada epoch ke-18. Namun, pada epoch ke-20, terjadi sedikit penurunan sehingga *validation accuracy* menjadi 93,50%. Fluktuasi yang diamati pada metrik validasi menunjukkan respons model terhadap distribusi data yang berbeda dan kemampuannya untuk menggeneralisasi pada data yang belum dilihat. *Validation loss*, indikator penting lainnya mengenai ki-

nerja model pada data baru, mengikuti tren non-monotonik selama beberapa epoch. Dimulai dari 1,5395 pada epoch pertama, dan meskipun ada beberapa naik turun, berakhir pada 0,2316 pada epoch ke-20.



Gambar 4.21: Fold 5 dari *confusion matrix* model *gated recurrent unit*.

Confusion Matrix yang ditampilkan pada Gambar 4.21, yang dihasilkan dari *testing* untuk mengevaluasi kinerja model setelah dilatih. Matriks tersebut menggambarkan seberapa cocok prediksi model dengan klasifikasi sebenarnya untuk setiap jenis detak jantung. Khususnya, model ini unggul dalam mengidentifikasi kelas N dengan benar, dengan 1,383 prediksi yang benar. Dengan 1,373 klasifikasi yang benar, juga menunjukkan kinerja dalam mengenali kelas L. Selain itu, model tersebut menunjukkan akurasi dalam mengidentifikasi kelas R dan A, dengan masing-masing 1,407 dan 1,385 prediksi akurat. Model tersebut berhasil mendeteksi secara tepat 1,388 kemunculan kelas V.

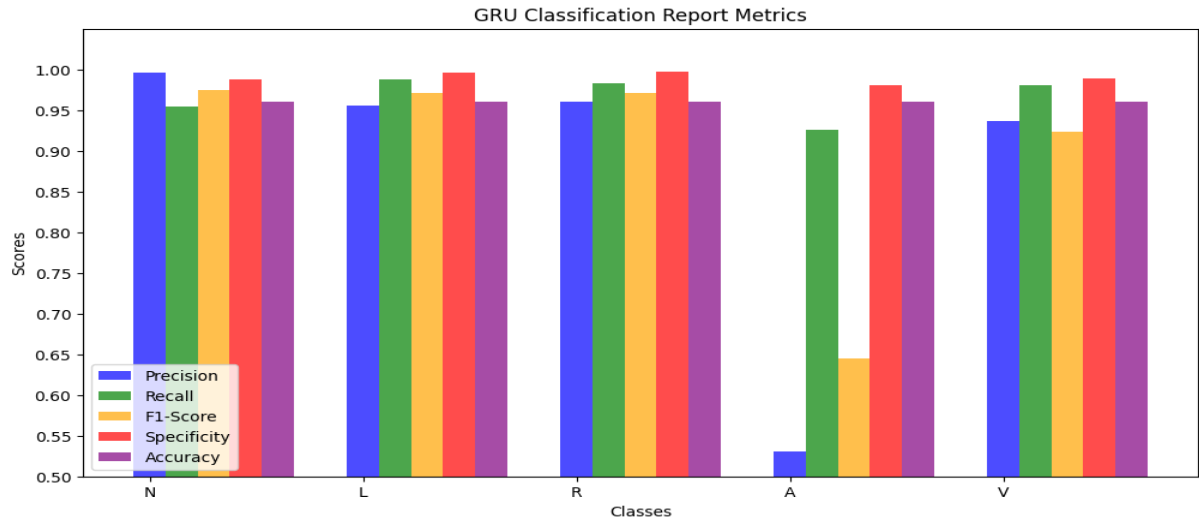
Tabel 4.17 menunjukkan hasil performansi model *Gated Recurrent Unit* (GRU) secara keseluruhan, dan Tabel 4.18 menunjukkan hasil performansi untuk setiap kelas.

Tabel 4.17: Metrik performansi *gated recurrent unit* (%).

Accuracy	Precision	Sensitivity	Specificity	F1-Score
96.07%	97.02%	96.07%	96.16%	96.36%

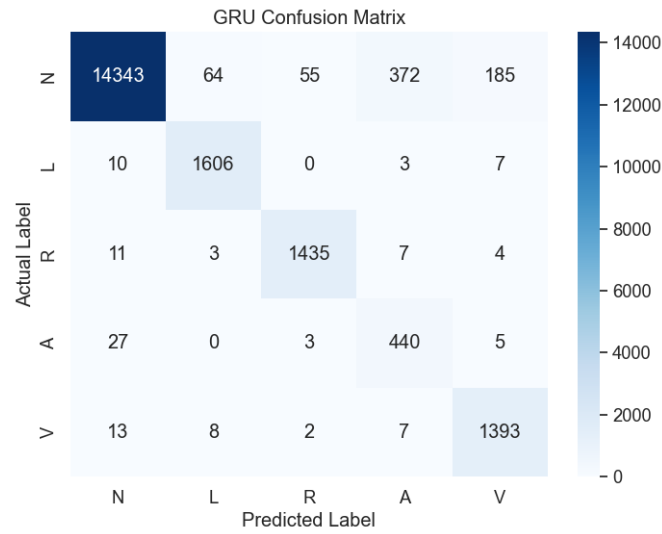
Tabel 4.18: Metrik performansi *gated recurrent unit* untuk klasifikasi aritmia berdasarkan kelas (%).

Kelas	Accuracy	Precision	Sensitivity	Specificity	F1-Score
N	96.07%	99.58%	95.50%	98.78%	97.50%
L	96.07%	95.54%	98.77%	99.59%	97.13%
R	96.07%	95.99%	98.29%	99.68%	97.12%
A	96.07%	53.08%	92.63%	98.01%	64.48%
V	96.07%	93.69%	98.03%	98.92%	92.34%



Gambar 4.22: Grafik metrik performansi model *gated recurrent unit*.

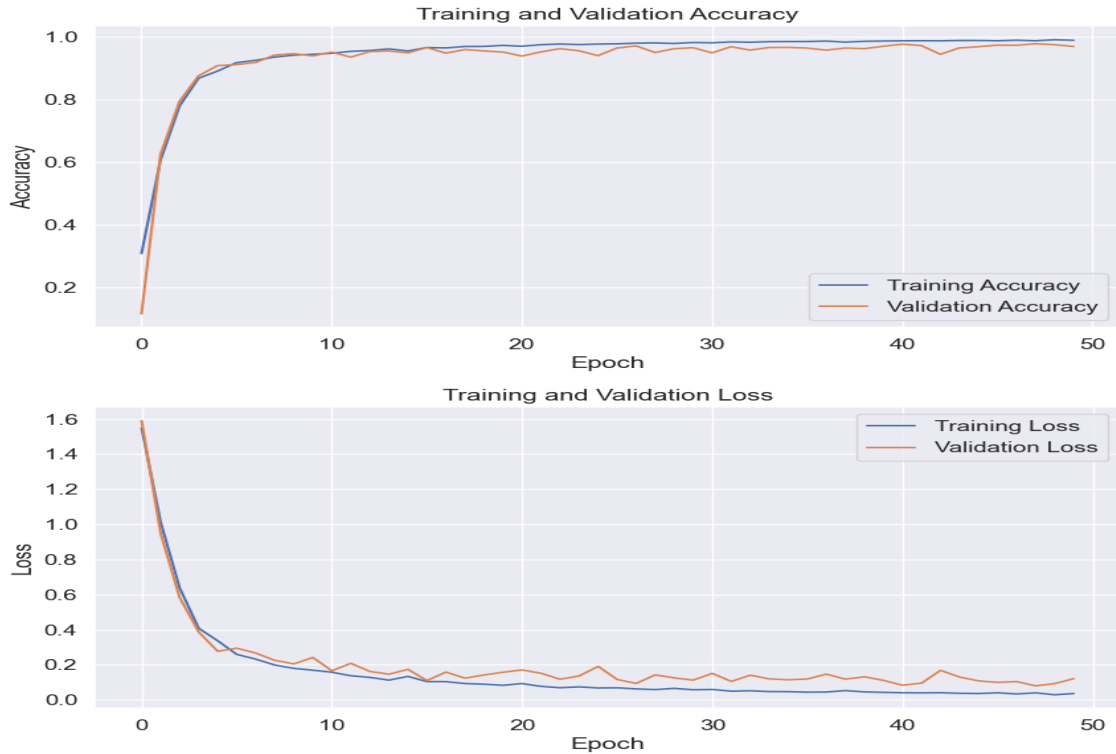
Tabel 4.17 dan grafik pada Gambar 4.22 menunjukkan model *Gated Recurrent Unit* (GRU) memiliki akurasi rata-rata 96,07%, presisi 97,02%, sensitivitas 96,07%, spesifisitas 96,16%, dan skor F1 96,36%, menandakan keseimbangan yang baik antara presisi dan sensitivitas. Pada Table 4.18 dijelaskan presisi tertinggi dimiliki kelas N sebesar 99,58% dan presisi terendah pada kelas A sebesar 53,08%. Sensitivitas tertinggi diperoleh pada kelas L sebesar 98,77% dan sensitivitas terendah diperoleh pada kelas N sebesar 95,50%. Spesifisitas tertinggi dicapai pada kelas R sebesar 99,68% dan spesifisitas terendah dicapai pada kelas A sebesar 98,01%. Nilai F1 tertinggi diraih kelas N sebesar 97,50% dan nilai F1 terendah diraih kelas A sebesar 64,48%.



Gambar 4.23: *Confusion matrix* model *gated recurrent unit*

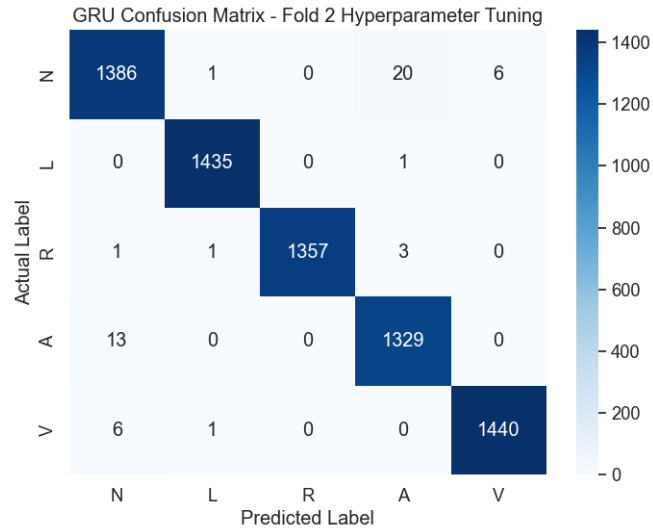
Confusion matrix pada Gambar 4.23 memberikan representasi rinci tentang kinerja prediksi model untuk setiap kategori detak jantung. Dari data yang disajikan, tampak bahwa model ini memiliki kemampuan yang sangat baik dalam mengidentifikasi kelas N, dengan total 14.343 prediksi yang akurat. Pada kelas L, model berhasil menghasilkan 1.606 identifikasi yang benar, menunjukkan efisiensi yang tinggi. Kelas R juga diidentifikasi dengan baik dengan 1.435 prediksi yang tepat. Sementara itu, kelas A memiliki 440 prediksi yang akurat dan kelas V dengan 1.393.

Hyperparameter Tuning



Gambar 4.24: Grafik *accuracy* dan *loss* model *gated recurrent unit* selama *training* dan *validation* setelah *hyperparameter tuning*.

Hyperparameter tuning pada model GRU yang ditampilkan pada Gambar 4.24 berhasil meningkatkan kinerja proses optimalisasi selama 50 epoch. *Training accuracy* mengalami peningkatan signifikan dari 30,86% pada awal pelatihan hingga mencapai puncak 99,03% pada epoch ke-49. Demikian pula, nilai *training loss* menunjukkan tren penurunan yang signifikan, turun dari 1,5489 pada awal menjadi 0,0360 pada akhir pelatihan. Saat dievaluasi pada dataset validasi, model mencapai akurasi puncak sebesar 97,76% pada epoch ke-48, dan meskipun mengalami penurunan sedikit pada epoch ke-50, akurasi tetap tinggi dengan nilai 96,86%. *Validation loss* juga mengikuti tren penurunan sepanjang pelatihan, mulai dari 1,5901 hingga mencapai 0,1207 pada akhir epoch. Fluktuasi dalam metrik validasi menunjukkan respons model terhadap variasi data, sementara tren penurunan loss mengindikasikan kemampuan model untuk menggeneralisasi pada data yang tidak terlihat sebelumnya.



Gambar 4.25: Fold 2 dari *confusion matrix* model GRU setelah *hyperparameter tuning*

Confusion matrix yang ditampilkan pada Gambar 4.25, yang dihasilkan dari fold ke-2 selama proses *training*, terdapat fokus yang jelas di sepanjang diagonal utama, menyoroti kemampuan model GRU yang telah dioptimalkan dengan *hyperparameter tuning* yang lebih baik dalam klasifikasi detak jantung. Khususnya kelas L dan V, menampilkan akurasi klasifikasi yang sempurna, terbukti dengan tidak adanya kesalahan off-diagonal dalam kategori tersebut. Kelas N dan R, sebaliknya, memiliki kesalahan klasifikasi yang sangat kecil.

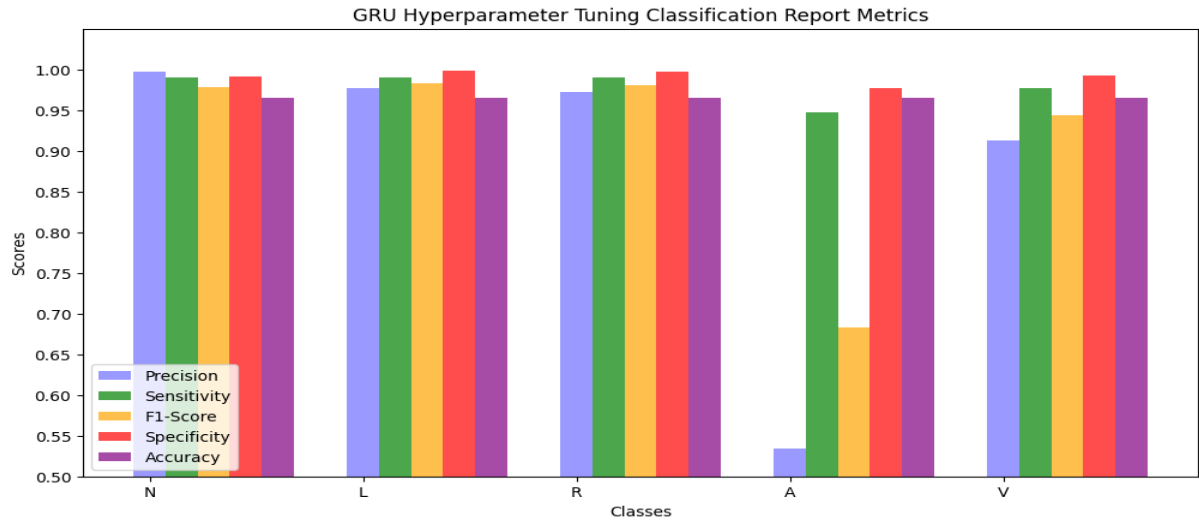
Tabel 4.21 menunjukkan hasil performansi model *Gated Recurrent Unit* setelah *hyperparameter tuning* secara keseluruhan, dan Tabel 4.22 menunjukkan hasil performansi untuk setiap kelas.

Tabel 4.21: Metrik performansi *gated recurrent unit* setelah *hyperparameter tuning* (%).

Accuracy	Precision	Sensitivity	Specificity	F1-Score
96.55%	97.53%	96.55%	97.95%	96.85%

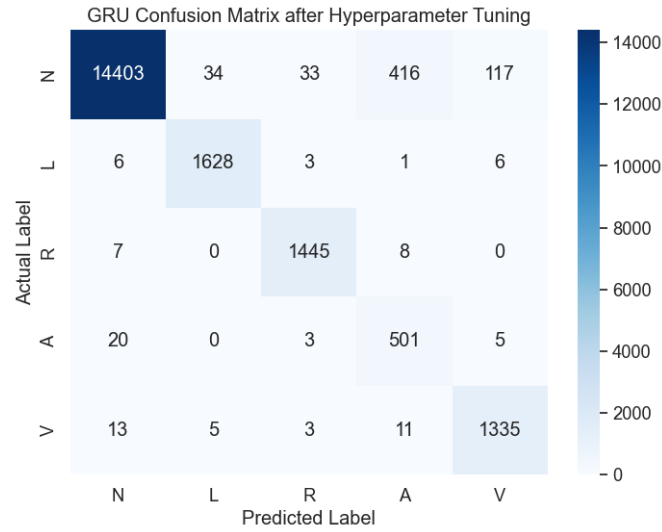
Tabel 4.22: Metrik performansi *gated recurrent unit* untuk klasifikasi aritmia berdasarkan kelas setelah *hyperparameter tuning* (%).

Kelas	Accuracy	Precision	Sensitivity	Specificity	F1-Score
N	96.55%	99.68%	96.00%	99.08%	97.81%
L	96.55%	97.66%	99.03%	99.79%	98.34%
R	96.55%	97.18%	98.97%	99.77%	98.07%
A	96.55%	53.47%	94.71%	97.76%	68.35%
V	96.55%	91.25%	97.66%	99.31%	94.35%



Gambar 4.26: Grafik metrik performansi model *gated recurrent unit* setelah *hyperparameter tuning*.

Tabel 4.21, 4.21, dan grafik pada Gambar 4.26 menunjukkan model *Gated Recurrent Unit* (GRU) setelah *hyperparameter tuning* memiliki akurasi rata-rata 96,55%, presisi 97,53%, sensitivitas 96,55%, spesifisitas 97,95%, dan skor F1 96,85%. Dari Tabel 4.23, presisi tertinggi diraih kelas N (99,68%) dan terendah pada kelas A (53,47%). Sensitivitas tertinggi diperoleh kelas L (99,03%) dan terendah pada kelas N (96,00%). Spesifisitas tertinggi dicapai oleh kelas L (99,79%). Nilai F1 tertinggi diraih oleh kelas L (98,34%) dan terendah oleh kelas A (68,35%).



Gambar 4.27: *Confusion matrix* GRU setelah *hyperparameter tuning*

Gambar 4.27 menunjukkan hasil setelah *hyperparameter tuning* pada model GRU, *Confusion Matrix* menunjukkan kinerja prediksi model untuk setiap jenis detak jantung. Model berhasil mengidentifikasi kelas N dengan 14.403 prediksi akurat, kelas L dengan 1.628, kelas R dengan 1.445, kelas A dengan 501, dan kelas V dengan 1.335. Meskipun ada beberapa kesalahan klasifikasi di antara kelas-kelas, keseluruhan hasil menunjukkan bahwa model memiliki kinerja yang kuat dalam prediksi detak jantung.

4.3 Hasil Objektif Ketiga

Algoritma klasifikasi deep learning yang digunakan dalam penelitian ini meliputi *Convolutional Neural Network* (CNN), *Long Short Term Memory* (LSTM), dan *Gated Recurrent Unit* (GRU). Algoritma-algoritma yang diusulkan bertujuan untuk mendeteksi sinyal EKG. Dalam tugas akhir ini, algoritma klasifikasi telah dikembangkan untuk mendeteksi sinyal EKG, dan tujuan dari tugas akhir ini adalah untuk menilai kesesuaian dan efektivitasnya dalam mengklasifikasikan aritmia menggunakan basis data MIT-BIH. Pada sub bab ini akan dilakukan analisis terkait performansi tiap algoritma.

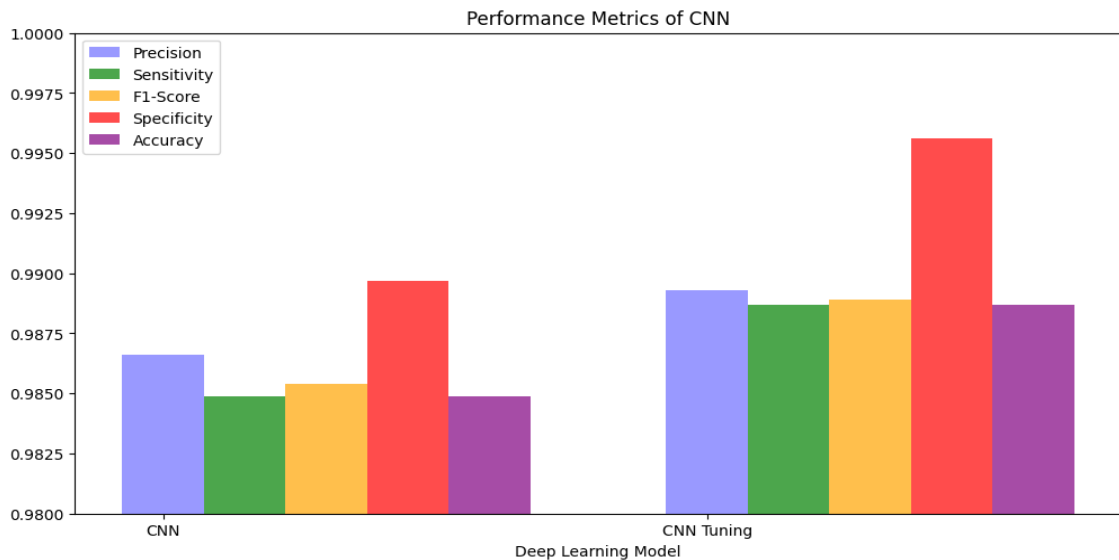
1. Convolutional Neural Network

Tabel 4.23 dan Gambar 4.28 menyajikan perbandingan metrik kinerja untuk *Convolutional Neural Network* (CNN) yang digunakan dalam klasifikasi aritmia, sebelum dan sesudah *hyperparameter tuning*. Metrik diukur dalam persentase. Tabel dan Gambar ini menunjukkan hasil untuk

dua skenario berbeda: satu dengan CNN sebelum *hyperparameter tuning* dan skenario lainnya dengan CNN setelah *hyperparameter tuning*.

Tabel 4.23: Perbandingan metrik performansi *convolutional neural network* untuk klasifikasi aritmia sebelum dan setelah *hyperparameter tuning* (%).

Algoritma	Accuracy	Precision	Sensitivity	Specificity	F1-Score
CNN	98.49%	98.66%	98.49%	98.97%	98.54%
CNN Tuning	98.87%	98.93%	98.87%	99.56%	98.89%



Gambar 4.28: Grafik perbandingan metrik performansi model *convolutional neural network* sebelum dan setelah *hyperparameter tuning*.

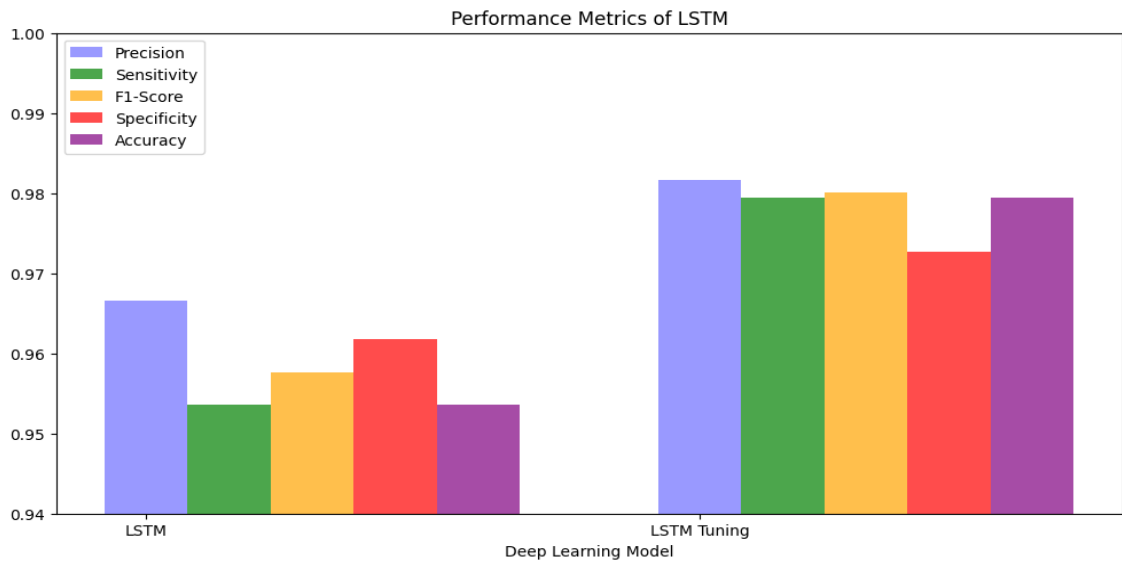
Dengan membandingkan kedua skenario tersebut pada Table 4.23 dan Gambar 4.28, dapat diamati bahwa setelah *hyperparameter tuning*, metrik kinerja CNN secara umum meningkat. Akurasi, presisi, sensitivitas, dan Skor F1 telah meningkat, yang menunjukkan bahwa model membuat prediksi yang lebih baik setelah *hyperparameter tuning*. Spesifisitas juga meningkat, menunjukkan peningkatan dalam mengidentifikasi aritmia negatif dengan benar. Hal ini menunjukkan efektivitas *hyperparameter tuning* dalam meningkatkan performa klasifikasi model untuk deteksi aritmia.

2. Long Short Term Memory

Tabel 4.24 dan Gambar 4.29 menyajikan perbandingan metrik kinerja untuk *Long Short Term Memory* (LSTM) yang digunakan dalam klasifikasi aritmia, sebelum dan sesudah *hyperparameter tuning*. Metrik diukur dalam persentase. Tabel dan Gambar ini menunjukkan hasil untuk dua skenario berbeda: satu dengan LSTM sebelum *hyperparameter tuning* dan skenario lainnya dengan LSTM setelah *hyperparameter tuning*.

Tabel 4.24: Perbandingan metrik performansi *long short term memory* untuk klasifikasi aritmia sebelum dan setelah *hyperparameter tuning* (%).

Algoritma	Accuracy	Precision	Sensitivity	Specificity	F1-Score
LSTM	95.36%	96.66%	95.36%	96.18%	95.76%
LSTM Tuning	97.94%	98.17%	97.94%	97.27%	98.01%



Gambar 4.29: Grafik perbandingan metrik performansi model *long short term memory* sebelum dan setelah *hyperparameter tuning*.

Membandingkan kedua skenario, terlihat jelas bahwa setelah *hyperparameter tuning*, metrik kinerja model LSTM telah meningkat secara keseluruhan. Akurasi, presisi, sensitivitas, dan Skor F1 semuanya meningkat, yang menunjukkan bahwa model membuat prediksi yang lebih akurat

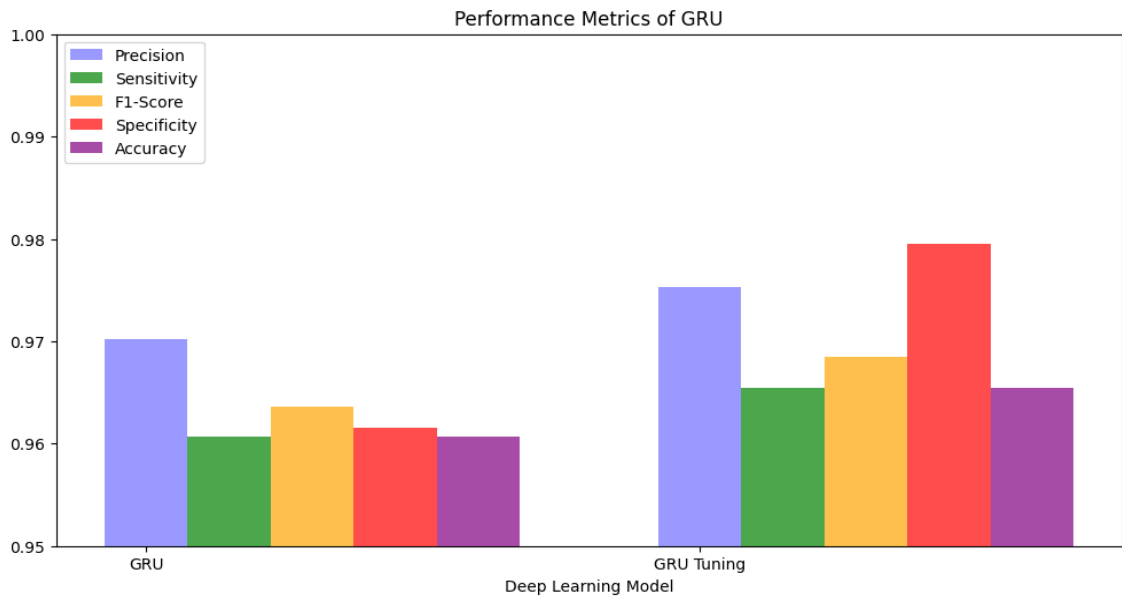
setelah *hyperparameter tuning*. Peningkatan kinerja secara keseluruhan menunjukkan bahwa *hyperparameter tuning* telah meningkatkan kemampuan model LSTM untuk mengklasifikasikan kasus aritmia secara efektif.

3. Gated Recurrent Unit

Tabel 4.25 dan Gambar 4.30 menyajikan perbandingan metrik kinerja untuk *Gated Recurrent Unit* (GRU) yang digunakan dalam klasifikasi aritmia, sebelum dan sesudah *hyperparameter tuning*. Metrik diukur dalam persentase. Tabel dan Gambar ini menunjukkan hasil untuk dua skenario berbeda: satu dengan GRU sebelum *hyperparameter tuning* dan skenario lainnya dengan GRU setelah *hyperparameter tuning*.

Tabel 4.25: Perbandingan metrik performansi *gated recurrent unit* untuk klasifikasi aritmia sebelum dan setelah *hyperparameter tuning* (%).

Algoritma	Accuracy	Precision	Sensitivity	Specificity	F1-Score
GRU	96.07%	97.02%	96.07%	96.16%	96.36%
GRU Tuning	96.55%	97.53%	96.55%	97.95%	96.85%



Gambar 4.30: Grafik perbandingan metrik performansi model *gated recurrent unit* sebelum dan setelah *hyperparameter tuning*.

Tabel menunjukkan seberapa baik algoritma GRU dapat membedakan antara detak jantung normal dan abnormal. Algoritma ini memiliki beberapa pengaturan yang dapat diubah untuk membuatnya bekerja lebih baik. Ini disebut *hyperparameter tuning*. Tabel membandingkan hasil algoritma sebelum dan sesudah *hyperparameter tuning*. Hasilnya diukur dengan beberapa angka, seperti akurasi, presisi, sensitivitas, dan skor F1. Angka-angka ini menunjukkan seberapa akurat algoritma dalam membuat prediksi. Tabel menunjukkan bahwa setelah *hyperparameter tuning*, semua angka lebih tinggi, yang berarti bahwa algoritma lebih akurat dan dapat membedakan antara detak jantung normal dan abnormal dengan lebih baik.

4.4 Pembahasan

Dalam eksperimen ini, telah dilakukan evaluasi terhadap tiga model deep learning, yaitu *Convolutional Neural Network*, *Long Short Term Memory*, *Gated Recurrent Unit*, menggunakan lima metrik performansi utama: akurasi, presisi, sensitivitas, spesifisitas dan F1-Score. Dengan memeriksa kinerja model-model ini dalam dua skenario: saat menggunakan parameter default dan setelah melakukan *hyperparameter tuning*.

4.4.1 Perbandingan Hasil Model Default dan Hasil Model Hyperparameter Tuning

Tiga algoritma *deep learning* dievaluasi: CNN, LSTM, dan GRU. Tujuannya adalah untuk mengklasifikasikan sinyal EKG dari basis data MIT-BIH.

a. Convolutional Neural Network (CNN)

Training dan *Validation* CNN menunjukkan peningkatan konsisten dalam kemampuan prediksinya selama 20 epoch (Gambar 4.4). Dimulai dari *training accuracy* awal 90,8%, model mencapai puncak yang mengesankan sebesar 99,53%. Namun, kinerja model pada set validasi menunjukkan sedikit penurunan pada epoch ke-20, mencapai akurasi 97,68%. Variasi ini menunjukkan respons model terhadap distribusi data yang berbeda dan kemampuannya dalam generalisasi. Dalam hal Hasil Klasifikasi, *confusion matrix* dari fold ke-5 (Gambar 4.5) menyoroti kemampuan CNN, khususnya dalam kelas L, A, dan V. Metrik akurasi untuk setiap kelas aritmia (Tabel 4.1 & 4.2) lebih menekankan efisiensi model. Pada tahap *hyperparameter tuning*, setelah *tuning*, CNN menampilkan kinerja yang meningkat selama 50 epoch (Gambar 4.7). Meskipun ada sedikit penurunan dalam akurasi validasi pada epoch ke-50, tren keseluruhan menampilkan pembelajaran model yang ditingkatkan. *confusion matrix* pasca-*tuning* (Gambar 4.8) dan metrik kinerja (Tabel 4.5 & 4.6)

lebih menonjolkan perbaikan yang dihasilkan oleh optimalisasi hyperparameter.

b. Long Short Term Memory (LSTM)

Training dan *Validation* model LSTM menunjukkan peningkatan signifikan dalam *training accuracy* dari awal 38,69% menjadi 88,88% pada epoch ke-20 (Gambar 4.10). Meskipun ada fluktuasi dalam metrik validasi, tren keseluruhan adalah positif, mencapai *validation accuracy* 92,63%. Mengenai Hasil Klasifikasi, *confusion matrix* untuk fold ke-5 (Gambar 4.11) menekankan kecakapan klasifikasi LSTM, terutama untuk kelas R. Metrik kinerja untuk setiap kelas aritmia (Tabel 4.9 & 4.10) lebih mencerminkan kemampuan dan area perbaikan model. Pada tahap *hyperparameter tuning*, pasca-*tuning*, model LSTM menunjukkan pertumbuhan signifikan dalam *training accuracy*, mencapai 98,35% pada epoch ke-50 (Gambar 4.13). *Confusion matrix* setelah *hyperparameter tuning* (Gambar 4.14) dan metrik kinerja (Tabel 4.13 & 4.14) menyoroti kemampuan model dalam mengklasifikasikan aritmia pasca-optimalisasi.

c. Gated Recurrent Unit (GRU)

Training dan *Validation* model GRU, selama 20 epoch, menunjukkan peningkatan yang stabil dalam *training accuracy*, dimulai dari 42,15% dan berakhir pada 91,21% (Gambar 4.16). Metrik validasi juga menunjukkan pertumbuhan, mencapai akurasi 89,47% pada epoch ke-20, dengan fluktuasi kecil yang menunjukkan adaptabilitas model terhadap distribusi data yang berbeda. Mengenai Hasil Klasifikasi, *confusion matrix* untuk fold ke-5 (Gambar 4.17) memberikan gambaran tentang kemampuan klasifikasi GRU. Kelas seperti L dan V menunjukkan akurasi yang mengesankan. Detail metrik kinerja untuk setiap kelas aritmia (Tabel 4.17 & 4.18) lebih menjelaskan kekuatan model dan area yang memerlukan penyempurnaan. Dalam hal *hyperparameter tuning*, model GRU menjalani penyetelan untuk mengoptimalkan kinerjanya. Pasca-*hyperparameter tuning*, model menampilkan peningkatan tajam dalam akurasi pelatihan sepanjang 50 epoch, mencapai puncak 96,89% (Gambar 4.20). *Confusion matrix* pasca-*hyperparameter tuning* (Gambar 4.21) dan metrik kinerja (Tabel 4.21 & 4.22) menekankan perbaikan signifikan yang diwujudkan melalui optimalisasi hyperparameter.

Ketika membandingkan tiga model—CNN, LSTM, dan GRU—bahwa sementara CNN menampilkan metrik kinerja tertinggi, baik LSTM dan GRU menampilkan potensi yang luar biasa, terutama pasca-*hyperparameter tuning*. GRU, dengan arsitekturnya yang lebih sederhana dibandingkan dengan LSTM, mencapai hasil yang sebanding, menekankan pentingnya memilih model yang tepat berdasarkan dataset.

Berdasarkan pemaparan pada Section 4.4.1 dan Tabel 4.23, 4.24, dan 4.25, terlihat bahwa CNN mengungguli LSTM dan GRU untuk klasifikasi aritmia. Tabel tersebut menunjukkan perbandingan metrik performa seperti akurasi, presisi, sensitivitas, spesifisitas, dan skor F1 untuk CNN, LSTM, dan GRU sebelum dan sesudah *hyperparameter tuning*. CNN mengungguli LSTM dan GRU untuk klasifikasi aritmia dikarenakan,

CNN mampu menangkap *local features* dan *patterns* dari sinyal *input* dengan menerapkan filter *convolutional* dengan ukuran kernel berbeda [38]. Hal ini dapat membantu mengurangi dimensi dan *noise* data, serta mengekstrak informasi yang relevan untuk tugas klasifikasi [38]. LSTM dan GRU, sebaliknya, memproses sinyal masukan secara berurutan [35], yang mungkin tidak dapat menangkap *local features* seefektif CNN.

CNN juga dapat mempelajari fitur hierarki dengan menumpuk beberapa lapisan *convolutional* dengan *filter sizes* dan *pooling operations* [38]. Hal ini dapat membantu meningkatkan kekuatan representasi dan kemampuan generalisasi model, serta mengurangi risiko *overfitting* [29]. LSTM dan GRU, di sisi lain, memiliki *hidden units* yang tetap di setiap lapisan, yang mungkin membatasi kapasitas mereka untuk mempelajari fitur dan pola kompleks dari data [35].

CNN juga dapat memperoleh manfaat dari teknik regularisasi seperti *dropout* dan *weight decay*, yang dapat membantu mencegah *overfitting* dan meningkatkan performa model [38]. LSTM dan GRU, di sisi lain, mungkin mengalami masalah hilangnya gradien atau *exploding*, yang dapat memengaruhi efisiensi dan stabilitas pembelajarannya [35]. Selain itu, LSTM dan GRU memiliki lebih banyak parameter dibandingkan CNN, sehingga dapat meningkatkan biaya komputasi dan kompleksitas model [29].

4.4.2 Penjelasan Hasil Hyperparameter Tuning

Metrik menunjukkan kenaikan untuk semua model (CNN, LSTM, dan GRU) setelah *hyperparameter tuning* karena *hyperparameter tuning* adalah proses menemukan nilai optimal untuk parameter yang memengaruhi performa model, seperti *learning rate*, *batch size*, *number of epoch*, *dropout rate*, dan *regularization strength* [29]. *Hyperparameter tuning* dapat membantu meningkatkan performa model dengan menemukan kombinasi terbaik dari parameter ini yang dapat meminimalkan hilangnya fungsi dan memaksimalkan akurasi. *Hyperparameter tuning* juga dapat membantu mencegah *overfitting* atau *underfitting* dengan menyesuaikan kompleksitas dan kapasitas model [29].

Misalnya, CNN *tuning* memiliki akurasi 98,87%, lebih tinggi dibandingkan CNN tanpa *tuning* (98,49%). Artinya, CNN *tuning* telah menemukan kumpulan hyperparameter yang lebih baik dibandingkan CNN tanpa *hyperparameter tuning*, yang dapat membuat model lebih akurat dan andal untuk klasifikasi aritmia [38]. Demikian pula LSTM *tuning* memiliki akurasi 97,94%, lebih

tinggi dibandingkan LSTM tanpa *tuning* (95,36%). Artinya, penyetelan LSTM telah meningkatkan performa model dengan menemukan *learning rate*, *batch size*, *number of epoch*, *dropout rate*, dan *regularization strength* yang lebih baik dibandingkan LSTM tanpa *tuning* [35]. GRU *tuning* memiliki akurasi sebesar 96,55%, lebih tinggi dibandingkan GRU tanpa *tuning* (96,07%). Artinya, GRU *tuning* juga telah meningkatkan performa model dengan menemukan kombinasi *hyperparameter* yang lebih baik dibandingkan GRU tanpa *tuning* [29].

4.4.3 Perbandingan Hasil Akurasi dengan Penelitian Sebelumnya

Tabel 4.31: Perbandingan state-of-the-art menggunakan database yang sama (MIT-BIH Database)

Author	Methodology	Accuracy
Singh et al.	RNN dan GRU	82.5%
Savalia et al.	CNN dan fungsi ReLU	83.5%
Lokesh et al.	11-layer CNN dan fungsi ReLU	84.44%
Alphale et al.	15-layer CNN dengan SMOTE	92.7%
Li et al.	GRU dan Attention Mechanism	92.8%
Metode Penulis	CNN dan fungsi ReLU dengan SMOTE	98%

Tabel 4.31 menyajikan metode-metode terkini yang komprehensif antara model *deep learning* yang diusulkan dan teknik-teknik terdepan yang berlaku dalam area ini. Penting untuk dicatat bahwa berbagai metodologi telah mengintegrasikan *deep learning* dalam klasifikasi aritmia. Ketika diukur dengan metrik akurasi, pendekatan yang kami usulkan menonjol sebagai yang terdepan, melampaui semua metodologi yang telah ada. Sebagai contoh, Lokesh et al. [29] mengembangkan CNN 11-layer yang berhasil mencapai tingkat akurasi 84,44% dalam mengklasifikasikan aritmia EKG secara efektif. Demikian pula, Singh et al. [39] mencapai akurasi 82,5% dengan menggunakan arsitektur model yang menggabungkan tiga lapisan RNN-GRU. Sebaliknya, Li et al. [28] meraih akurasi 92,8% dengan menggabungkan GRU dengan mekanisme perhatian untuk mengatasi asimetri data. Alphale et al. [2] merancang sebuah CNN yang menggunakan teknik seperti filter low-pass untuk menghilangkan *noise* dan SMOTE untuk menangani ketidakseimbangan dataset. Hasil dari pendekatan ini adalah akurasi yang mengesankan sebesar 92,73%. Selain itu, Savalia et al. [38] mengusulkan algoritma yang menggabungkan multi-layer perceptron (MLP) dengan empat lapisan tersembunyi dan jaringan saraf konvolusi empat lapisan untuk memetakan sampel EKG ke berbagai kelas aritmia, mencapai akurasi 83,5%.

4.5 Ringkasan

Ada beberapa hal yang perlu menjadi catatan dari Bab ini sebagai berikut:

1. Tugas akhir ini menggunakan basis data aritmia MIT-BIH dengan 48 rekaman EKG dari 47 individu (22 perempuan, 25 laki-laki) dan memilih sinyal *lead II* karena relevansinya dalam analisis ritme jantung untuk kompleks QRS.
2. Mengatasi gangguan sinyal/*noise* dengan menggunakan gelombang *Sym4* yang dikombinasikan dengan *threshold* adaptif dan menunjukkan efikasi yang signifikan dalam deteksi kompleks QRS.
3. Menggunakan algoritma Pan-Tompkins untuk deteksi puncak R dalam denyut jantung. dengan kombinasi filter *bandpass*, filter *derivative*, dan filter *squaring* untuk mengurangi noise dan meningkatkan kompleks QRS. Kemudian normalisasi *Z-Score* memusatkan analisis pada puncak R yang terdeteksi, memastikan konsistensi di seluruh dataset.
4. Dataset ini dibagi menjadi set pelatihan, pengujian, dan validasi, dengan 80% untuk pelatihan dan 20% untuk pengujian.
5. Model *deep learning* yang diusulkan, yang mencakup CNN, LSTM, dan GRU, dievaluasi menggunakan berbagai metrik. Model CNN menunjukkan performa yang unggul dalam klasifikasi aritmia, mencapai tingkat akurasi sebesar 98,87% setelah *hyperparameter tuning* dan berhasil menangkap pola dan fitur penting. Baik model LSTM maupun GRU juga menunjukkan kemampuan yang efisien dan bisa menjadi alternatif yang layak tergantung pada kebutuhan dan pertimbangan tertentu.
6. *Hyperparameter tuning* secara signifikan meningkatkan kinerja model dan menyoroti pentingnya optimalisasi.

Bab V

Kesimpulan dan Saran

5.1 Kesimpulan

Tugas akhir ini telah mencapai semua obyektif yang disebutkan pada Bab I, sebagai berikut:

1. Obyektif Pertama berhasil dicapai. Bukti capaian yang berupa taksonomi algoritma *deep learning* dapat yang dapat dilihat pada Bab 2.2. Taksonomi literatur *deep learning* yaitu *Convolutional Neural Network*, *Long Short-Term Memory*, dan *Gated Recurrent Unit* beserta studi literatur terkait, contohnya pada *Convolutional Neural Network*, penelitian menggunakan *11-layer CNN + ReLU activation* oleh Lokesh et al. kemudian ada *BiLSTM with ten-fold cross validation* oleh Rahul et al. lalu, *DenseNet* oleh Guo et al.
2. Obyektif Kedua berhasil dicapai. Bukti dari capaian ada pada Bab 4.2, algoritma *deep learning* yang dikembangkan adalah *Convolutional Neural Network*, *Long Short Term Memory*, dan *Gated Recurrent Unit*. Berdasarkan Tabel 4.23, 4.24, dan 4.25 terlihat bahwa CNN mengungguli LSTM dan GRU dalam klasifikasi aritmia. Tabel ini menyajikan perbandingan metrik performa seperti akurasi, presisi, sensitivitas, spesifisitas, dan skor F1 untuk CNN, LSTM, dan GRU sebelum dan sesudah *hyperparameter tuning*. Keunggulan CNN dalam klasifikasi aritmia dapat dikaitkan dengan kemampuannya mempelajari fitur dari sinyal *input* dengan menerapkan filter konvolusional dengan berbagai kernel *size*. Selain itu, CNN dapat mempelajari fitur hierarki dengan menumpuk beberapa lapisan konvolusional dengan *filter size* dan *pooling operations*. Hal ini membantu dalam pengurangan dimensi, pengurangan *noise*, dan ekstraksi informasi yang relevan untuk tugas klasifikasi. CNN juga mendapat manfaat dari teknik regularisasi seperti *dropout* dan *dropout rate*, yang membantu mencegah *overfitting* dan meningkatkan performa model.
3. Obyektif Ketiga berhasil dicapai dengan bukti ada pada Bab 4.3 yang

membahas mengenai analisa performansi dari model yang telah dikembangkan, baik sebelum maupun setelah *hyperparameter tuning*. Model CNN menunjukkan performa yang unggul dalam klasifikasi aritmia, namun baik model LSTM maupun GRU juga menunjukkan kemampuan yang kuat dan bisa menjadi alternatif yang layak tergantung pada kebutuhan dan pertimbangan tertentu.

5.2 Saran

Berdasarkan proses perancangan dan pengujian sistem, penulis melihat beberapa pengembangan rancangan dan langkah pengujian yang dapat dilakukan, antara lain:

1. Untuk mengatasi ketidakseimbangan kelas dan meningkatkan kemampuan generalisasi model, dapat menerapkan teknik augmentasi data yang lain seperti *random rotations, flips, and translations*. Augmentasi dataset dapat membantu model untuk lebih baik menangkap karakteristik kelas minoritas.
2. Jelajahi kemungkinan penggunaan teknik ensemble learning untuk menggabungkan prediksi dari beberapa model, termasuk CNN, LSTM, dan GRU. Model ensemble dapat meningkatkan kinerja secara keseluruhan dan memberikan klasifikasi yang lebih tahan terhadap variasi data.
3. Validasi model *deep learning* yang diusulkan pada dataset eksternal dan skenario klinis dunia nyata untuk menilai generalisabilitas dan efektivitasnya dalam beragam data EKG. Bekerjasama dengan rumah sakit untuk mendapatkan data asli untuk menguji akurasi algoritma

Daftar Pustaka

- [1] AL-HUSEINY, M. S., ABBAS, N. K., AND SAJIT, A. S. Diagnosis of arrhythmia based on ecg analysis using cnn. *Bulletin of Electrical Engineering and Informatics* 9, 3 (2020), 988–995.
- [2] APHALE, S. S., JOHN, E., AND BANERJEE, T. Arrhynet: a high accuracy arrhythmia classification convolutional neural network. In *2021 IEEE International Midwest Symposium on Circuits and Systems (MWSCAS)* (2021), IEEE, pp. 453–457.
- [3] AZIZ, S., AHMED, S., AND ALOUINI, M.-S. Ecg-based machine-learning algorithms for heartbeat classification. *Scientific reports* 11, 1 (2021), 18738.
- [4] BENGIO, Y. Practical recommendations for gradient-based training of deep architectures. In *Neural Networks: Tricks of the Trade: Second Edition*. Springer, 2012, pp. 437–478.
- [5] BERKAYA, S. K., UYSAL, A. K., GUNAL, E. S., ERGIN, S., GUNAL, S., AND GULMEZOGLU, M. B. A survey on ecg analysis. *Biomedical Signal Processing and Control* 43 (2018), 216–235.
- [6] BIAGETTI, G., CRIPPA, P., FALASCETTI, L., AND TURCHETTI, C. A multi-channel electromyography, electrocardiography and inertial wireless sensor module using bluetooth low-energy. *Electronics* 9, 6 (2020), 934.
- [7] BOTE, J. M., RECAS, J., RINCÓN, F., ATIENZA, D., AND HERMIDA, R. A modular low-complexity ecg delineation algorithm for real-time embedded systems. *IEEE journal of biomedical and health informatics* 22, 2 (2017), 429–441.
- [8] CHUNG, J., GULCEHRE, C., CHO, K., AND BENGIO, Y. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555* (2014).
- [9] DARGAN, S., KUMAR, M., AYYAGARI, M. R., AND KUMAR, G. A survey of deep learning and its applications: a new paradigm to machine

- learning. *Archives of Computational Methods in Engineering* 27, 4 (2020), 1071–1092.
- [10] DESTIANI, D. K., ADIWIJAYA, A., AND UTAMA, D. Q. Klasifikasi sinyal ecg gagal jantung menggunakan wavelet dan jst propagasi balik dengan modifikasi gradien konjugat. *eProceedings of Engineering* 5, 1 (2018).
 - [11] DEY, R., AND SALEM, F. M. Gate-variants of gated recurrent unit (gru) neural networks. In *2017 IEEE 60th international midwest symposium on circuits and systems (MWSCAS)* (2017), IEEE, pp. 1597–1600.
 - [12] DEY, R., AND SALEM, F. M. Gate-variants of gated recurrent unit (gru) neural networks. In *2017 IEEE 60th international midwest symposium on circuits and systems (MWSCAS)* (2017), IEEE, pp. 1597–1600.
 - [13] DHYANI, S., KUMAR, A., AND CHOUDHURY, S. Analysis of ecg-based arrhythmia detection system using machine learning. *MethodsX* 10 (2023), 102195.
 - [14] ESSA, E., AND XIE, X. An ensemble of deep learning-based multi-model for ecg heartbeats arrhythmia classification. *IEEE Access* 9 (2021), 103452–103464.
 - [15] FALASCHETTI, L., ALESSANDRINI, M., BIAGETTI, G., CRIPPA, P., AND TURCHETTI, C. Ecg-based arrhythmia classification using recurrent neural networks in embedded systems. *Procedia Computer Science* 207 (2022), 3479–3487.
 - [16] FAUST, O., KAREEM, M., ALI, A., CIACCIO, E. J., AND ACHARYA, U. R. Automated arrhythmia detection based on rr intervals. *Diagnostics* 11, 8 (2021), 1446.
 - [17] GOLDBERGER, A. L., AMARAL, L. A., GLASS, L., HAUSDORFF, J. M., IVANOV, P. C., MARK, R. G., MIETUS, J. E., MOODY, G. B., PENG, C.-K., AND STANLEY, H. E. Physiobank, physiotoolkit, and physionet: components of a new research resource for complex physiologic signals. *circulation* 101, 23 (2000), e215–e220.
 - [18] GRAVES, A., AND JAITLEY, N. Towards end-to-end speech recognition with recurrent neural networks. In *International conference on machine learning* (2014), PMLR, pp. 1764–1772.
 - [19] HOMAEINEZHAD, M. R., ERFANIANMOSHIRI-NEJAD, M., AND NASERI, H. A correlation analysis-based detection and delineation of ecg characteristic events using template waveforms extracted by ensemble averaging

- of clustered heart cycles. *Computers in biology and medicine* 44 (2014), 66–75.
- [20] ILYAS, M. A., AND FAROOQ, U. Architectural paradigm of deep learning.
- [21] ISLAM, M. S., ISLAM, M. N., HASHIM, N., RASHID, M., BARI, B. S., AND AL FARID, F. New hybrid deep learning approach using bigru-bilstm and multilayered dilated cnn to detect arrhythmia. *IEEE Access* 10 (2022), 58081–58096.
- [22] IVANOVIC, M. D., ATANASOSKI, V., SHVILKIN, A., HADZIEVSKI, L., AND MALUCKOV, A. Deep learning approach for highly specific atrial fibrillation and flutter detection based on rr intervals. In *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)* (2019), IEEE, pp. 1780–1783.
- [23] IZCI, E., OZDEMIR, M. A., DEGIRMENCI, M., AND AKAN, A. Cardiac arrhythmia detection from 2d ecg images by using deep learning technique. In *2019 Medical Technologies Congress (TIPTEKNO)* (2019), IEEE, pp. 1–4.
- [24] KANDEL, I., AND CASTELLI, M. The effect of batch size on the generalizability of the convolutional neural networks on a histopathology dataset. *ICT express* 6, 4 (2020), 312–315.
- [25] KHAN, M. A., AND KIM, Y. Cardiac arrhythmia disease classification using lstm deep learning approach. *CMC-COMPUTERS MATERIALS & CONTINUA* 67, 1 (2021), 427–443.
- [26] LECUN, Y., BENGIO, Y., AND HINTON, G. Deep learning. *nature* 521, 7553 (2015), 436–444.
- [27] LECUN, Y., BOTTOU, L., BENGIO, Y., AND HAFFNER, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* 86, 11 (1998), 2278–2324.
- [28] LI, J., JHANG, J.-Y., LIN, C.-J., AND LIN, X.-Q. Arrhythmia detection using a taguchi-based convolutional neuro-fuzzy network. *Sensors and Materials* 34, 7 (2022), 2853–2867.
- [29] LOKESH, S., PRIYA, A., SAKHARE, D., DEVI, R. M., SAHU, D. N., AND REDDY, P. C. S. Cnn based deep learning methods for precise analysis of cardiac arrhythmias. *International journal of health sciences* 6 (2022).

- [30] LUZ, E. J. D. S., SCHWARTZ, W. R., CÁMARA-CHÁVEZ, G., AND MENOTTI, D. Ecg-based heartbeat classification for arrhythmia detection: A survey. *Computer methods and programs in biomedicine* 127 (2016), 144–164.
- [31] MCSHARRY, P. E., CLIFFORD, G. D., TARASSENKO, L., AND SMITH, L. A. A dynamical model for generating synthetic electrocardiogram signals. *IEEE transactions on biomedical engineering* 50, 3 (2003), 289–294.
- [32] MOODY, G. B., AND MARK, R. G. The impact of the mit-bih arrhythmia database. *IEEE Engineering in Medicine and Biology Magazine* 20, 3 (2001), 45–50.
- [33] MOODY, G. B., AND MARK, R. G. The impact of the mit-bih arrhythmia database. *IEEE engineering in medicine and biology magazine* 20, 3 (2001), 45–50.
- [34] NIELSEN, M. A. *Neural networks and deep learning*, vol. 25. Determination press San Francisco, CA, USA, 2015.
- [35] RAJKUMAR, A., GANESAN, M., AND LAVANYA, R. Arrhythmia classification on ecg using deep learning. In *2019 5th international conference on advanced computing & communication systems (ICACCS)* (2019), IEEE, pp. 365–369.
- [36] RAMADHANI, G. T., ADIWIJAYA, A., AND UTAMA, D. Q. Klasifikasi penyakit aritmia melalui sinyal elektrokardiogram (ekg) menggunakan metode local features dan support vector machine. *eProceedings of Engineering* 5, 1 (2018).
- [37] ROMDHANE, T. F., AND PR, M. A. Electrocardiogram heartbeat classification based on a deep convolutional neural network and focal loss. *Computers in Biology and Medicine* 123 (2020), 103866.
- [38] SAVALIA, S., AND EMAMIAN, V. Cardiac arrhythmia classification by multi-layer perceptron and convolution neural networks. *Bioengineering* 5, 2 (2018), 35.
- [39] SINGH, S., PANDEY, S. K., PAWAR, U., AND JANGHEL, R. R. Classification of ecg arrhythmia using recurrent neural networks. *Procedia computer science* 132 (2018), 1290–1297.
- [40] SOWMYA, S., AND JOSE, D. Contemplate on ecg signals and classification of arrhythmia signals using cnn-lstm deep learning methods. *Measurement: Sensors* (2022), 100558.

- [41] SWAPNA, G., SOMAN, K., AND VINAYAKUMAR, R. Automated detection of cardiac arrhythmia using deep learning techniques. *Procedia computer science* 132 (2018), 1192–1201.
- [42] ULLAH, W., SIDDIQUE, I., ZULQARNAIN, R. M., ALAM, M. M., AHMAD, I., AND RAZA, U. A. Classification of arrhythmia in heartbeat detection using deep learning. *Computational Intelligence and Neuroscience* 2021 (2021).
- [43] WHO. Noncommunicable diseases. <https://www.who.int/news-room/fact-sheets/detail/noncommunicable-diseases>, 2022. Accessed: 2022-11-27.
- [44] YE, C., COIMBRA, M. T., AND KUMAR, B. V. Investigation of human identification using two-lead electrocardiogram (ecg) signals. In *2010 Fourth IEEE International Conference on Biometrics: Theory, Applications and Systems (BTAS)* (2010), IEEE, pp. 1–8.
- [45] YUE, Z., AND FENG, L. Using deep learning detection of arrhythmia. In *ITM Web of Conferences* (2019), vol. 27, EDP Sciences, p. 03001.
- [46] ZHENG, Z., CHEN, Z., HU, F., ZHU, J., TANG, Q., AND LIANG, Y. An automatic diagnosis of arrhythmias using a combination of cnn and lstm technology. *Electronics* 9, 1 (2020), 121.