



TUGAS AKHIR - EC184801

KLASIFIKASI ARITMIA PADA SINYAL ECG BERBASIS *DEEP LEARNING*

Firdaus Nanda Pradanggapasti
NRP 0721164000015

Dosen Pembimbing
Arief Kurniawan, ST., MT.
Dr. Reza Fuad Rachmadi, ST., MT.

DEPARTEMEN TEKNIK KOMPUTER
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember
Surabaya 2020



ITS
Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR - EC184801

**KLASIFIKASI ARITMIA PADA SINYAL ECG BERBASIS
*DEEP LEARNING***

Firdaus Nanda Pradanggapasti
NRP 0721164000015

Dosen Pembimbing
Arief Kurniawan, ST., MT.
Dr. Reza Fuad Rachmadi, ST., MT.

DEPARTEMEN TEKNIK KOMPUTER
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember
Surabaya 2020

[Halaman ini sengaja dikosongkan].



FINAL PROJECT - EC184801

**ARRHYTHMIA CLASSIFICATION IN THE ECG SIGNAL
BASED ON DEEP LEARNING**

Firdaus Nanda Pradangapasti
NRP 07211640000015

Advisor
Arief Kurniawan, ST., MT.
Dr. Reza Fuad Rachmadi, ST., MT.

Departement of Computer Engineering
Faculty of Intelligent Electrical And Informatics Technology
Sepuluh Nopember Institute of Technology
Surabaya 2020

[Halaman ini sengaja dikosongkan].

PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul “**Klasifikasi Aritmia pada Sinyal ECG Berbasis *Deep Learning***” adalah benar-benar hasil karya intelektual sendiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diizinkan dan bukan merupakan karya orang lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka.

Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai dengan peraturan yang berlaku.

Surabaya, Juli 2020

Firdaus Nanda Pradangapasti
NRP. 0721164000015

[Halaman ini sengaja dikosongkan].

LEMBAR PENGESAHAN

Klasifikasi Aritmia pada Sinyal ECG Berbasis *Deep Learning*
Tugas Akhir ini disusun untuk memenuhi salah satu syarat memperoleh gelar Sarjana Teknik di Institut Teknologi Sepuluh Nopember Surabaya

Oleh: Firdaus Nanda Pradangapasti (NRP: 07211640000015)

Tanggal Ujian : 8 Juli 2020

Periode Wisuda : September 2020

Disetujui oleh:

Arief Kurniawan, ST., MT.
NIP: 19740907 200212 1 001

(Pembimbing I)

Dr. Reza Fuad Rachmadi, ST., MT.
NIP: 19850403 201212 1 001

(Pembimbing II)

Eko Pramunanto, ST., MT.
NIP. 19661203 199412 1 001

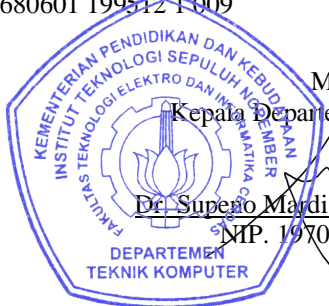
(Penguji I)

Susi Juniastuti, ST., M.Eng.
NIP. 19650618 199903 2 001

(Penguji II)

Dr. Eko Mulyanto Yuniarno, ST., MT.
NIP. 19680601 199512 1 009

(Penguji III)



Mengetahui

Kepala Departemen Teknik Komputer

Dr. Supero Mardi Susiki Nugroho, ST., MT.
NIP. 19700313 1995121 001

[Halaman ini sengaja dikosongkan].

ABSTRAK

Nama Mahasiswa : Firdaus Nanda Pradanggapasti
Judul Tugas Akhir : Klasifikasi Aritmia pada Sinyal ECG Berbasis *Deep Learning*
Pembimbing : 1. Arief Kurniawan, ST., MT.
2. Dr. Reza Fuad Rachmadi, ST., MT.

Penyakit jantung merupakan penyebab utama kematian di dunia. Untuk mengetahui penyakit jantung sejak dini, dapat dideteksi dengan memeriksa ada tidaknya aritmia. Aritmia merupakan suatu kelainan irama detak jantung, bisa berdetak terlalu cepat, terlalu lambat, ataupun berdetak dengan pola yang tidak beraturan, sehingga aritmia memiliki banyak jenisnya. Untuk mendiagnosis aritmia, salah satu metode yang dapat digunakan adalah dengan cara menganalisis sinyal ECG (*Elektrokardiogram*). Saat ini, dokter maupun tenaga medis menganalisis sinyal ECG dengan cara manual. Dengan berkembangnya teknologi pada zaman ini, terdapat teknologi yang bernama *Deep Learning*. *Deep Learning* merupakan suatu perkembangan dari *Machine Learning*. Pada tugas akhir ini, salah satu metode dari *Deep Learning*, yaitu *Convolutional Neural Network*, digunakan untuk mengklasifikasikan 5 jenis aritmia pada sinyal ECG. Akurasi tertinggi yang dihasilkan pada tugas akhir ini sebesar 98.6%.

Kata Kunci : Aritmia, ECG, Deep Learning, Pengolahan Sinyal

Halaman ini sengaja dikosongkan

ABSTRACT

Name : Firdaus Nanda Pradangapasti
Title : *Arrhythmia Classification in the ECG
Signal Based on Deep Learning*
Advisors : 1. Arief Kurniawan, ST., MT.
2. Dr. Reza Fuad Rachmadi, ST., MT.

Heart disease is the leading cause of death in the world. To find out heart disease early, it can be detected by examining the presence or absence of arrhythmias. Arrhythmia is an abnormal heart beat rhythm, can beat too fast, too slow, or beat with irregular patterns, so that the arrhythmia has many types. To diagnose arrhythmias, one method that can be used is by analyzing ECG (Electrocardiogram) signals. Currently, doctors and medical personnel analyze ECG signals manually. With the development of technology in this era, there is a technology called Deep Learning. Deep Learning is a development of Machine Learning. In this final project, one method of Deep Learning, namely Convolutional Neural Network, is used to classify 5 types of arrhythmias on ECG signals. The highest accuracy produced in this final project is 98.6%.

Keywords : *Arrhythmia, ECG, Deep Learning, Signal Processing*

Halaman ini sengaja dikosongkan

KATA PENGANTAR

Puji dan syukur kehadiran Tuhan Yang Maha Esa atas segala karunia-Nya, penulis dapat menyelesaikan penelitian ini dengan judul **Klasifikasi Aritmia pada Sinyal ECG Berbasis *Deep Learning***.

Penelitian ini disusun dalam rangka pemenuhan bidang riset di Departemen Teknik Komputer ITS, Bidang Studi Telematika, serta digunakan sebagai persyaratan menyelesaikan pendidikan S1. Oleh karena itu, penulis mengucapkan terima kasih kepada:

1. Keluarga, Ibu, Ayah, dan Saudara tercinta yang telah memberikan dorongan spiritual dan material dalam penyelesaian buku penelitian ini.
2. Bapak Dr. Supeno Mardi Susiki Nugroho, ST., MT. selaku Kepala Departemen Teknik Komputer, Fakultas Teknik Elektro dan Informatika Cerdas, Institut Teknologi Sepuluh Nopember.
3. Bapak Arief Kurniawan, ST., MT. selaku dosen pembimbing I dan Bapak Dr. Reza Fuad Rachmadi, ST., MT. selaku dosen pembimbing II yang selalu memberikan arahan selama mengerjakan penelitian tugas akhir ini.
4. Bapak-ibu dosen pengajar Departemen Teknik Komputer, atas pengajaran, bimbingan, serta perhatian yang diberikan kepada penulis selama ini.
5. Seluruh teman-teman dari angkatan e56, Teknik Komputer, Laboratorium B401, dan B201 Teknik Komputer ITS.

Kesempurnaan hanya milik Allah SWT, untuk itu penulis memohon segenap kritik dan saran yang membangun. Semoga penelitian ini dapat memberikan manfaat bagi kita semua. Amin.

Surabaya, Maret 2020

Firdaus Nanda Pradanggapasti

Halaman ini sengaja dikosongkan

DAFTAR ISI

Abstrak	i
Abstract	iii
KATA PENGANTAR	v
DAFTAR ISI	vii
DAFTAR GAMBAR	xi
DAFTAR TABEL	xv
NOMENKLATUR	xvii
1 PENDAHULUAN	1
1.1 Latar belakang	1
1.2 Perumusan Masalah	2
1.3 Tujuan	2
1.4 Batasan masalah	2
1.5 Sistematika Penulisan	2
2 TINJAUAN PUSTAKA	5
2.1 Dasar Teori	5
2.1.1 <i>Machine Learning</i>	5
2.1.1.1 <i>Supervised Learning</i>	6
2.1.1.2 <i>Unsupervised Learning</i>	6
2.1.1.3 <i>Semi Supervised Learning</i>	6
2.1.1.4 <i>Reinforcement Learning</i>	6
2.1.2 <i>Deep Learning</i>	7
2.1.3 <i>Convolutional Neural Network</i>	7
2.1.3.1 <i>Convolutional Layer</i>	7
2.1.3.2 <i>Padding</i>	8
2.1.3.3 <i>Pooling Layer</i>	9
2.1.3.4 <i>Fully Connected Layer</i>	10
2.1.4 Aritmia	10
2.1.5 Elektrokardiogram	11

2.1.6	<i>Classification Performance</i>	11
2.1.6.1	<i>Accuracy</i>	13
2.1.6.2	<i>Precision</i>	13
2.1.6.3	<i>Recall</i>	13
2.1.6.4	<i>F1 Score</i>	14
2.2	Penelitian Terkait	14
2.2.1	Perbandingan Metode SVM, FUZZY-KNN, Dan BDT-SVM Untuk Klasifikasi Detak Jantung Hasil Elektrokardiografi	14
2.2.2	Klasifikasi Sinyal Ecg Menggunakan Deep Learning Dengan Stacked Denoising Autoencoders	14
2.2.3	<i>Cardiac arrhythmia detection using deep learning</i>	15
2.2.4	<i>A deep convolutional neural network model to classify heartbeats</i>	15
2.2.5	<i>Arrhythmia Detection Using Deep Convolutional Neural Network With Long Duration ECG Signals</i>	16
2.2.6	<i>Electrocardiogram Classification Based on Faster Regions with Convolutional Neural Network</i>	16
3	DESAIN DAN IMPLEMENTASI SISTEM	17
3.1	Desain Sistem	17
3.2	Pengambilan Data ECG	18
3.3	Transformasi Sinyal ECG menjadi Data 2 Dimensi .	18
3.3.1	Spektogram	18
3.3.2	Plot Sinyal	19
3.4	Pembagian <i>Dataset</i>	20
3.5	Pembuatan Arsitektur CNN	21
3.6	Proses <i>Training</i>	22
3.7	Proses <i>Validation</i>	22
3.8	Proses <i>Testing</i>	23
4	PENGUJIAN DAN ANALISA	25
4.1	Hasil Pengambilan Data ECG	25
4.2	Hasil Transformasi Sinyal ECG Menjadi Data 2 Dimensi	26
4.2.1	Spektogram	26

4.2.1.1	Panjang <i>Window</i> 4 dan <i>Overlap</i> 2	26
4.2.1.2	Panjang <i>Window</i> 8 dan <i>Overlap</i> 4	27
4.2.1.3	Panjang <i>Window</i> 16 dan <i>Overlap</i> 8	27
4.2.1.4	Panjang <i>Window</i> 32 dan <i>Overlap</i> 16	28
4.2.1.5	Panjang <i>Window</i> 64 dan <i>Overlap</i> 32	29
4.2.1.6	Panjang <i>Window</i> 64 dan <i>Overlap</i> 48	29
4.2.1.7	Panjang <i>Window</i> 64 dan <i>Overlap</i> 60	30
4.2.1.8	Panjang <i>Window</i> 128 dan <i>Overlap</i> 64	31
4.2.1.9	Panjang <i>Window</i> 256 dan <i>Overlap</i> 128	31
4.2.2	Plot Sinyal ECG	32
4.3	Pembagian <i>Dataset</i>	33
4.4	Arsitektur CNN	34
4.5	Hasil <i>Training</i> dan <i>Validation</i>	35
4.5.1	Pengujian dengan Menggunakan <i>Dataset</i> Model Pertama	36
4.5.1.1	Pengujian dengan Menggunakan Plot Sinyal ECG	36
4.5.1.2	Pengujian dengan Menggunakan Spektogram dengan Panjang <i>Window</i> 64 dan <i>Overlap</i> 32	37
4.5.1.3	Pengujian dengan Menggunakan Spektogram dengan Panjang <i>Window</i> 64 dan <i>Overlap</i> 48	37
4.5.1.4	Pengujian dengan Menggunakan Spektogram dengan Panjang <i>Window</i> 64 dan <i>Overlap</i> 60	38
4.5.2	Pengujian dengan Menggunakan <i>Dataset</i> Model Kedua	39
4.5.2.1	Pengujian dengan Menggunakan Plot Sinyal ECG	39
4.5.2.2	Pengujian dengan Menggunakan Spektogram dengan Panjang <i>Window</i> 64 dan <i>Overlap</i> 32	40
4.5.2.3	Pengujian dengan Menggunakan Spektogram dengan Panjang <i>Window</i> 64 dan <i>Overlap</i> 48	41

4.5.2.4	Pengujian dengan Menggunakan Spektogram dengan Panjang <i>Window</i> 64 dan <i>Overlap</i> 60	41
4.6	Hasil <i>Testing</i>	42
4.6.1	Pengujian dengan Menggunakan <i>Dataset</i> Model Pertama	43
4.6.1.1	Pengujian dengan Menggunakan Plot Sinyal ECG	43
4.6.1.2	Pengujian dengan Menggunakan Spektogram dengan Panjang <i>Window</i> 64 dan <i>Overlap</i> 32	44
4.6.1.3	Pengujian dengan Menggunakan Spektogram dengan Panjang <i>Window</i> 64 dan <i>Overlap</i> 48	45
4.6.1.4	Pengujian dengan Menggunakan Spektogram dengan Panjang <i>Window</i> 64 dan <i>Overlap</i> 60	46
4.6.2	Pengujian dengan Menggunakan <i>Dataset</i> Model Kedua	47
4.6.2.1	Pengujian dengan Menggunakan Plot Sinyal ECG	48
4.6.2.2	Pengujian dengan Menggunakan Spektogram dengan Panjang <i>Window</i> 64 dan <i>Overlap</i> 32	49
4.6.2.3	Pengujian dengan Menggunakan Spektogram dengan Panjang <i>Window</i> 64 dan <i>Overlap</i> 48	50
4.6.2.4	Pengujian dengan Menggunakan Spektogram dengan Panjang <i>Window</i> 64 dan <i>Overlap</i> 60	51
5	PENUTUP	53
5.1	Kesimpulan	53
5.2	Saran	53
	DAFTAR PUSTAKA	55
	Biografi Penulis	57

DAFTAR GAMBAR

2.1	Visualisasi Percabangan Ilmu <i>Artificial Intelligence</i>	5
2.2	Visualisasi Proses <i>Convolutional Layer</i>	8
2.3	Visualisasi Proses <i>Pooling Layer</i>	9
2.4	Visualisasi Proses <i>Fully Connected Layer</i>	10
2.5	Bentuk Sinyal ECG	11
2.6	<i>Confusion Matrix</i>	12
3.1	Blok Diagram Kerja Sistem	17
3.2	Bagan Umum Proses Pembuatan Spektogram	19
3.3	Visualisasi Spektogram	19
3.4	Visualisasi Plot Sinyal	20
3.5	Visualisasi Pembagian <i>Dataset</i>	20
3.6	Blok Diagram Alur Kerja CNN	22
3.7	Visualisasi <i>Cross Validation</i>	23
4.1	Spektogram dari <i>Normal Beat</i> dengan Panjang <i>Window</i> 4 dan <i>Overlap</i> 2	26
4.2	Spektogram dari <i>Normal Beat</i> dengan Panjang <i>Window</i> 8 dan <i>Overlap</i> 4	27
4.3	Spektogram dari <i>Normal Beat</i> dengan Panjang <i>Window</i> 16 dan <i>Overlap</i> 8	28
4.4	Spektogram dari <i>Normal Beat</i> dengan Panjang <i>Window</i> 32 dan <i>Overlap</i> 16	28
4.5	Spektogram dari <i>Normal Beat</i> dengan Panjang <i>Window</i> 64 dan <i>Overlap</i> 32	29
4.6	Spektogram dari <i>Normal Beat</i> dengan Panjang <i>Window</i> 64 dan <i>Overlap</i> 48	30
4.7	Spektogram dari <i>Normal Beat</i> dengan Panjang <i>Window</i> 64 dan <i>Overlap</i> 60	30
4.8	Spektogram dari <i>Normal Beat</i> dengan Panjang <i>Window</i> 128 dan <i>Overlap</i> 64	31
4.9	Spektogram dari <i>Normal Beat</i> dengan Panjang <i>Window</i> 256 dan <i>Overlap</i> 128	32
4.10	Plot Sinyal ECG dari Setiap Kelas Aritmia	33

4.11	Grafik Hasil <i>Training</i> dan <i>Validation</i> pada Pengujian Menggunakan Plot Sinyal ECG	36
4.12	Grafik Hasil <i>Training</i> dan <i>Validation</i> pada Pengujian Menggunakan Spektogram dengan Panjang <i>Window</i> 64 dan <i>Overlap</i> 32	37
4.13	Grafik Hasil <i>Training</i> dan <i>Validation</i> pada Pengujian dengan Menggunakan Spektogram dengan Panjang <i>Window</i> 64 dan <i>Overlap</i> 48	38
4.14	Grafik Hasil <i>Training</i> dan <i>Validation</i> pada Pengujian dengan Menggunakan Spektogram dengan Panjang <i>Window</i> 64 dan <i>Overlap</i> 60	39
4.15	Grafik Hasil <i>Training</i> dan <i>Validation</i> pada Pengujian Menggunakan Plot Sinyal ECG	40
4.16	Grafik Hasil <i>Training</i> dan <i>Validation</i> pada Pengujian dengan Menggunakan Spektogram dengan Panjang <i>Window</i> 64 dan <i>Overlap</i> 32	40
4.17	Grafik Hasil <i>Training</i> dan <i>Validation</i> pada Pengujian dengan Menggunakan Spektogram dengan Panjang <i>Window</i> 64 dan <i>Overlap</i> 48	41
4.18	Grafik Hasil <i>Training</i> dan <i>Validation</i> pada PPengujian dengan Menggunakan Spektogram dengan Panjang <i>Window</i> 64 dan <i>Overlap</i> 60	42
4.19	<i>Confusion Matrix</i> Hasil Pengujian dengan Menggunakan Plot Sinyal ECG	43
4.20	<i>Confusion Matrix</i> Hasil Pengujian dengan Menggunakan Spektogram dengan Panjang <i>Window</i> 64 dan <i>Overlap</i> 32	45
4.21	<i>Confusion Matrix</i> Hasil Pengujian dengan Menggunakan Spektogram dengan Panjang <i>Window</i> 64 dan <i>Overlap</i> 48	46
4.22	<i>Confusion Matrix</i> Hasil <i>Testing</i> Ketiga	47
4.23	<i>Confusion Matrix</i> Hasil Pengujian dengan Menggunakan Plot Sinyal ECG	48
4.24	<i>Confusion Matrix</i> Hasil Pengujian dengan Menggunakan Spektogram dengan Panjang <i>Window</i> 64 dan <i>Overlap</i> 32	50

4.25	<i>Confusion Matrix</i> Hasil Pengujian dengan Menggunakan Spektogram dengan Panjang <i>Window</i> 64 dan <i>Overlap</i> 48	51
4.26	<i>Confusion Matrix</i> Hasil Pengujian dengan Menggunakan Spektogram dengan Panjang <i>Window</i> 64 dan <i>Overlap</i> 60	52

Halaman ini sengaja dikosongkan

DAFTAR TABEL

4.1	Spesifikasi PC yang Digunakan	25
4.2	Jumlah Sampel ECG pada Setiap Kelas Aritmia . .	26
4.3	Pembagian Sampel ECG untuk Dataset Model Pertama	34
4.4	Pembagian Sampel ECG untuk Dataset Model Kedua	34
4.5	Spesifikasi Arsitektur CNN yang Digunakan	35
4.6	<i>Classification Report</i> Hasil Pengujian dengan Menggunakan Plot Sinyal ECG	44
4.7	<i>Classification Report</i> Hasil Pengujian dengan Menggunakan Spektogram dengan Panjang <i>Window</i> 64 dan <i>Overlap</i> 32	44
4.8	<i>Classification Report</i> Hasil Pengujian dengan Menggunakan Spektogram dengan Panjang <i>Window</i> 64 dan <i>Overlap</i> 48	46
4.9	<i>Classification Report</i> Hasil Pengujian dengan Menggunakan Spektogram dengan Panjang <i>Window</i> 64 dan <i>Overlap</i> 60	47
4.10	<i>Classification Report</i> Hasil Pengujian dengan Menggunakan Plot Sinyal ECG	49
4.11	<i>Classification Report</i> Hasil Pengujian dengan Menggunakan pektogram dengan Panjang <i>Window</i> 64 dan <i>Overlap</i> 32	49
4.12	<i>Classification Report</i> Hasil Pengujian dengan Menggunakan pektogram dengan Panjang <i>Window</i> 64 dan <i>Overlap</i> 48	51
4.13	<i>Classification Report</i> Hasil Pengujian dengan Menggunakan pektogram dengan Panjang <i>Window</i> 64 dan <i>Overlap</i> 60	52

Halaman ini sengaja dikosongkan

NUMENKLATUR

TP : *True Positive*
TN : *True Negative*
FP : *False Positive*
FN : *False Negative*

Halaman ini sengaja dikosongkan

BAB 1

PENDAHULUAN

Penelitian ini dilatarbelakangi oleh berbagai kondisi yang menjadi acuan. Selain itu juga terdapat beberapa permasalahan yang akan dijawab sebagai luaran dari penelitian.

1.1 Latar belakang

Kesehatan merupakan hal terpenting bagi seseorang karena bersangkutan pada kualitas hidupnya. Salah satu organ tubuh yang harus dijaga agar tetap sehat adalah jantung karena jantung merupakan organ tubuh yang sangat vital yang berfungsi untuk memompa darah ke seluruh tubuh. Jika jantung mengalami kelainan dalam menjalankan fungsinya, kemungkinan terburuk yang dapat dialami orang tersebut adalah meninggal. Menurut data *World Health Organization* (WHO), pada tahun 2016, sebanyak 17.9 juta orang meninggal disebabkan oleh penyakit jantung atau sekitar 31% dari populasi seluruh dunia dan sekitar 17 juta orang yang meninggal pada usia di bawah 70 tahun.[1]

Penyakit jantung dapat dideteksi sejak dini dengan cara mendiagnosa Aritmia. Aritmia merupakan suatu kondisi dimana irama detak jantung tidak beraturan, bisa berdetak terlalu cepat, terlalu lambat, ataupun berdetak membentuk pola yang tidak beraturan [2]. Tidak semua aritmia berbahaya, karena ketika kita sedang melakukan aktivitas berat, jantung kita bisa berdetak lebih cepat. Sehingga untuk memeriksa aritmia pada seseorang, dibutuhkan waktu minimal 24 jam untuk merekam aktivitas jantung [3]. Aktivitas jantung dapat direkam dengan menggunakan alat yang disebut dengan Elektrokardiograf yang menghasilkan sinyal Elektrokardiogram (ECG). Pada sinyal ECG, dalam setiap satu detakan jantung, terdapat satu gelombang P, satu kompleks QRS, dan satu gelombang T. Gelombang-gelombang tersebutlah yang digunakan sebagai parameter untuk mengklasifikasikan jenis-jenis aritmia [4].

Deep Learning merupakan salah satu cabang dari *Machine Learning* yang memiliki kemampuan untuk mengekstraksi fitur secara otomatis. Salah satu metode dari *deep learning* adalah *Convolutio-*

nal Neural Network atau biasa disebut dengan CNN. CNN digunakan untuk memproses data yang memiliki bentuk dua dimensi atau berbentuk suatu citra.

Pada penelitian sebelumnya [12], telah dilakukan klasifikasi 5 kelas aritmia dengan menggunakan *Faster* R-CNN dengan plot sinyal ECG sebagai *input* dan didapatkan hasil akurasi sebesar 99.21%. Pada tugas akhir ini, digunakan metode CNN dengan dua kondisi input yang berbeda, yaitu plot sinyal ECG dan spektrogram dengan harapan menggunakan spektrogram sebagai *input* memiliki tingkat akurasi yang lebih tinggi.

1.2 Perumusan Masalah

Berdasarkan data yang telah dipaparkan di latar belakang, penyakit jantung merupakan salah satu penyebab utama kematian di dunia yang dapat dideteksi sejak dini dengan mengetahui aritmia jenis apa yang diderita pasien melalui sinyal ECG sehingga diperlukan sebuah algoritma komputer yang dapat mengklasifikasikan aritmia secara otomatis.

1.3 Tujuan

Tujuan dari pembuatan tugas akhir ini adalah untuk mengklasifikasi 5 jenis aritmia pada sinyal ECG yang telah diubah menjadi data citra menggunakan metode *deep learning*.

1.4 Batasan masalah

Batasan masalah yang timbul dari permasalahan Tugas Akhir ini adalah:

1. Data training dan testing menggunakan data yang diambil dari MIT-BIH Arrhythmia Database (physionet.org).
2. Karena data yang digunakan hanya dari satu sumber, maka kelas aritmia yang digunakan hanya kelas yang memiliki lima data terbanyak.

1.5 Sistematika Penulisan

Laporan penelitian Tugas akhir ini tersusun dalam sistematika dan terstruktur sehingga mudah dipahami dan dipelajari oleh pembaca maupun seseorang yang ingin melanjutkan penelitian ini. Alur

sistematika penulisan laporan penelitian ini yaitu:

1. BAB I Pendahuluan

Bab ini berisi uraian tentang latar belakang permasalahan, penegasan dan alasan pemilihan judul, sistematika laporan, tujuan dan metodologi penelitian.

2. BAB II Dasar Teori

Pada bab ini berisi tentang uraian secara sistematis teori-teori yang berhubungan dengan permasalahan yang dibahas pada penelitian ini. Teori-teori ini digunakan sebagai dasar dalam penelitian, yaitu *Deep Learning*, aritmia, dan elektrokardiogram.

3. BAB III Perancangan Sistem dan Impementasi

Bab ini berisi tentang penjelasan-penjelasan terkait eksperimen yang akan dilakukan dan langkah-langkah data diolah hingga menghasilkan visualisasi. Guna mendukung eksperimen pada penelitian ini, digunakanlah blok diagram atau *work flow* agar sistem yang akan dibuat dapat terlihat dan mudah dibaca untuk implementasi pada pelaksanaan tugas akhir.

4. BAB IV Pengujian dan Analisa

Bab ini menjelaskan tentang pengujian eksperimen yang dilakukan terhadap data dan analisisnya. Beberapa teknik visualisasi akan ditunjukkan hasilnya pada bab ini dan dilakukan analisa terhadap hasil visualisasi dan informasi yang didapat dari hasil mengamati visualisasi yang tersaji.

5. BAB V Penutup

Bab ini merupakan penutup yang berisi kesimpulan yang diambil dari penelitian dan pengujian yang telah dilakukan. Saran dan kritik yang membangun untuk pengembangkan lebih lanjut juga dituliskan pada bab ini.

Halaman ini sengaja dikosongkan

BAB 2

TINJAUAN PUSTAKA

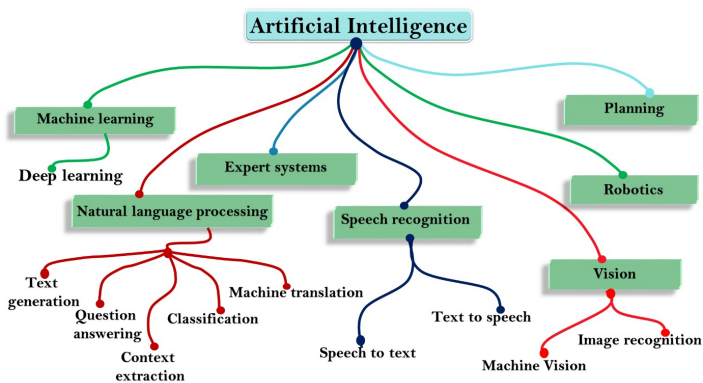
Demi mendukung penelitian ini, dibutuhkan beberapa teori penunjang sebagai bahan acuan dan referensi. Dengan demikian penelitian ini menjadi lebih terarah.

2.1 Dasar Teori

2.1.1 *Machine Learning*

Machine Learning menurut Arthur Samuel, seorang ilmuwan komputer yang mempelopori kecerdasan buatan, adalah sebuah bidang yang memberi komputer kemampuan untuk belajar tanpa diprogram secara eksplisit.

Machine Learning adalah sebuah cabang dari *Artificial Intelligence* (AI). Kecerdasan buatan memiliki pengertian yang sangat luas, umumnya memiliki arti bagaimana komputer bisa memiliki kecerdasan seperti manusia. Sedangkan *Machine Learning* memiliki arti yang lebih spesifik, yaitu menggunakan metode statistika untuk membuat komputer dapat mempelajari pola pada data. Gambar 2.1 merupakan visualisasi percabangan ilmu *Artificial Intelligence*



Gambar 2.1: Visualisasi Percabangan Ilmu *Artificial Intelligence*

Machine Learning terbagi menjadi empat kategori umum:

2.1.1.1 *Supervised Learning*

Pada *supervised learning*, *dataset* yang digunakan memiliki label. Label adalah *tag* atau pengenalan dari sebuah data. Misalnya sebuah buah memiliki atribut berwarna hijau, berat lebih dari 500 gram, memiliki kulit keras, berduri, memiliki bau yang menyengat, dan daging buahnya manis. Buah yang memiliki karakteristik seperti yang disebutkan dikenali sebagai durian, maka label dari atribut tersebut adalah durian. Klasifikasi email spam adalah contoh dari *supervised learning*.

2.1.1.2 *Unsupervised Learning*

Pada *unsupervised learning*, *dataset* yang digunakan tidak memiliki label. Model *unsupervised* melakukan belajar sendiri untuk melabeli atau mengelompokkan data. Contoh kasus *unsupervised* adalah dari data 100 pengunjung sebuah situs web, model akan belajar sendiri untuk mengelompokkan pengunjung. Dapat berdasarkan waktu kunjungan, lama kunjungan, jumlah klik, dan sebagainya.

2.1.1.3 *Semi Supervised Learning*

Semi Supervised merupakan gabungan dari *supervised learning* dan *unsupervised learning*. Pada model *semi supervised*, *dataset* untuk pelatihan sebagian memiliki label dan sebagian tidak. Implementasi dari model *semi supervised* yang sering digunakan misalnya pada Google Photos. Pada Google Photos bisa memberi *tag* untuk setiap orang yang terdapat pada sebuah foto. Alhasil, ketika mengunggah foto baru dengan wajah orang yang sudah diberi *tag* sebelumnya, Google Photos akan secara otomatis mengenali orang tersebut. Salah satu contoh dari model *semi supervised* adalah *Deep Belief Network* (DBN).

2.1.1.4 *Reinforcement Learning*

Reinforcement Learning adalah model yang belajar menggunakan sistem *reward* dan *penalties*. Model belajar agar terus mendapatkan *reward* dan menghindari *penalties*. Alpha Go adalah contoh terkenal dari *reinforcement learning*. Alpha Go adalah sebuah program yang dikembangkan Google DeepMind untuk memainkan

permainan Go, sebuah permainan papan yang berasal dari Tiongkok. Alpha Go mempelajari setiap langkah dalam jutaan permainan go untuk terus mendapatkan *reward* yaitu memenangkan sebuah permainan. Alpha Go menjadi terkenal setelah menjadi program komputer pertama yang berhasil mengalahkan pemain profesional.

2.1.2 Deep Learning

Deep Learning merupakan salah satu cabang dari *Machine Learning*. *Deep Learning* juga merupakan salah satu algoritma dari Jaringan Syaraf Tiruan yang terinspirasi dari sistem otak manusia.

Algoritma pada *Deep Learning* mempunyai kemampuan yang unik, yaitu dapat mengekstraksi fitur secara otomatis. Lapisan tersembunyi (*hidden layer*) pada *Deep Learning* lebih banyak daripada Jaringan Syaraf Tiruan, sehingga pada Jaringan Syaraf Tiruan membutuhkan lebih banyak informasi tentang data masukan untuk menentukan model yang cocok. Berbeda dengan *Deep Learning* yang tidak membutuhkan informasi apapun terhadap data yang akan dipelajarinya karena secara mandiri dapat memilih model yang optimal [5]. Beberapa contoh dari *deep learning* adalah *Deep Neural Network* (DNN), *Recurrent Neural Network* (RNN), dan *Convolutional Neural Network* (CNN).

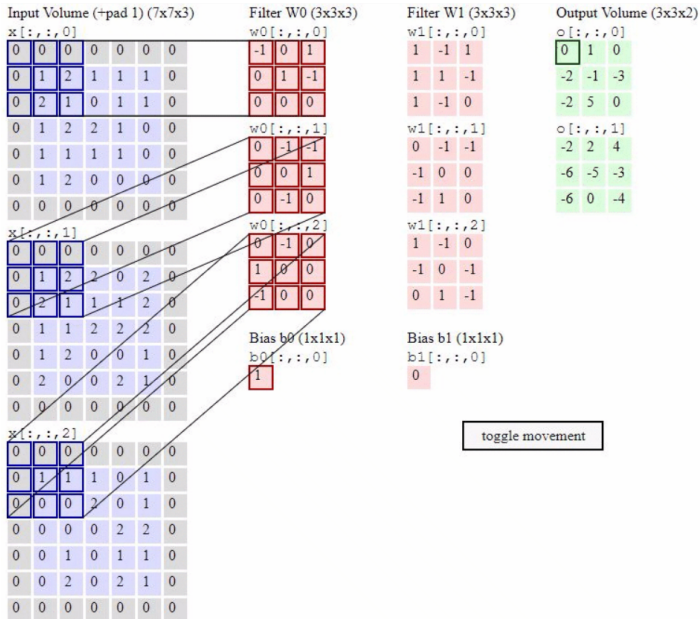
2.1.3 Convolutional Neural Network

Convolutional Neural Network merupakan metode *deep learning* yang paling sering digunakan untuk mengolah data dua dimensi. *Convolutional Neural Network* memiliki beberapa *layer* antara lain:

2.1.3.1 Convolutional Layer

Convolutional Layer merupakan sebuah *neural network* yang tersusun sedemikian rupa sehingga membentuk suatu filter dengan ukuran panjang, lebar, dan kedalaman yang menggunakan satuan piksel. *Input* dari *convolutional layer* berbentuk suatu citra yang memiliki panjang, lebar, dan kedalaman citra (jumlah kanal). Lapisan pertama pada sebuah CNN selalu merupakan *convolutional layer*. Sebagai contoh apabila *input* citra berukuran 7x7x3 dengan *filter* berukuran 3x3x3. Pada *filter* juga terdapat nilai berupa angka yang biasa disebut dengan *weight*, kedalaman (kanal) pada *filter*

yang digunakan harus sesuai dengan kedalaman pada *input* citra. Ketiga filter tersebut akan bergeser ke seluruh bagian dari citra. Setiap pergeseran akan dilakukan operasi *dot* matriks antara *input* dan nilai dari filter tersebut sehingga menghasilkan sebuah *output* yang biasa disebut dengan *feature map* seperti yang divisualisasikan pada Gambar 2.2



Gambar 2.2: Visualisasi Proses *Convolutional Layer*

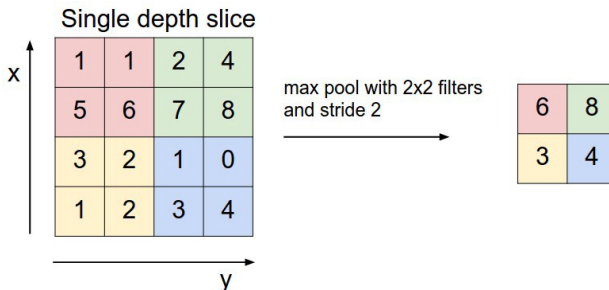
2.1.3.2 *Padding*

Padding adalah parameter yang menentukan jumlah piksel (berisi nilai 0) yang akan ditambahkan pada setiap sisi matriks citra. Hal ini bertujuan untuk memanipulasi dimensi *output* yang dihasilkan dari *convolutional layer* (*feature map*). Tujuan dari penggunaan *padding* adalah karena dimensi *output* dari *convolutional layer* lebih kecil dari *input*-nya. *Output* ini akan digunakan kembali sebagai *input* dari *convolutional layer* berikutnya, sehingga semakin banyak data yang terbuang. Dengan menggunakan *padding*, dimensi *ou-*

tput akan tetap sama seperti dimensi *input* atau setidaknya tidak berkurang secara drastis sehingga bisa menggunakan *convolutional layer* yang lebih dalam yang menghasilkan lebih banyak fitur yang berhasil didapat. Pada Gambar 2.2, dimensi dari *input* sebenarnya adalah 5×5 , jika dilakukan *convolution* dengan *filter* 3×3 dan *stride* sebesar 2, maka *feature map* yang dihasilkan memiliki ukuran 2×2 . Namun jika ditambahkan *zero padding* sebanyak satu, maka *feature map* yang dihasilkan memiliki ukuran 3×3 yang berarti lebih banyak informasi yang dihasilkan.

2.1.3.3 Pooling Layer

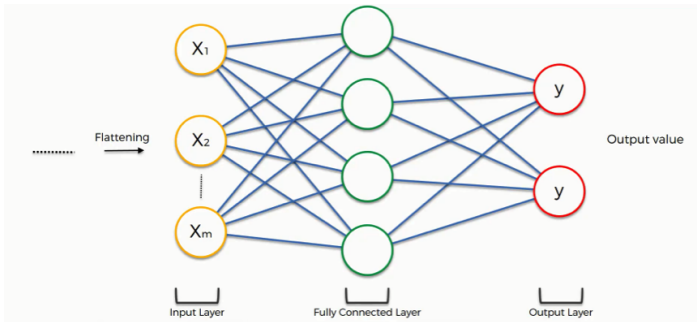
Pooling Layer biasanya berada setelah *convolutional layer*. Pada prinsipnya, *pooling layer* terdiri dari sebuah *filter* dengan ukuran dan *stride* tertentu yang akan bergeser pada seluruh area *feature map*. Tujuan dari *pooling layer* adalah untuk mengurangi dimensi dari *feature map* (*downsampling*), sehingga mempercepat komputasi karena parameter yang harus diperbarui semakin sedikit dan menghindari *overfitting*. Operasi *pooling* yang biasa digunakan adalah *average pooling* dan *max pooling*. Pada *Convolutional Neural Network*, operasi *pooling* yang kebanyakan digunakan adalah *max pooling*. *Max Pooling* membagi *feature map* menjadi beberapa matriks kecil lalu mengambil nilai terbesar dari setiap matriks dengan ukuran dan *stride* tertentu yang akan bergeser pada seluruh area *feature map* untuk menyusun matriks citra seperti pada gambar 2.3.



Gambar 2.3: Visualisasi Proses *Pooling Layer*

2.1.3.4 Fully Connected Layer

Fully Connected Layer biasanya bagian akhir dari proses *Convolutional Neural Network*. *Layer* ini biasanya terdiri dari satu atau lebih. Sebelum *feature map* masuk pada *fully connected layer*, *feature map* yang berupa matriks dua dimensi terlebih dahulu menjadi sebuah matriks satu dimensi agar bisa digunakan sebagai *input* dari *fully connected layer*. Proses tersebut dinamakan *flattening*. Pada *layer* ini tidak ada informasi spasial citra yang disimpan. *Output* dari proses *flattening* akan menjadi *input* pada *fully connected layer*. Elemen pada setiap *input* akan terhubung pada semua *neuron* pada *fully connected layer* seperti pada gambar 2.4



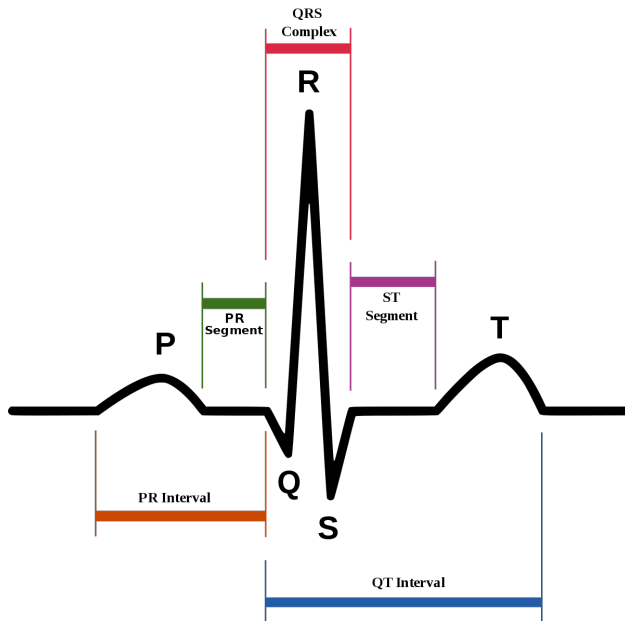
Gambar 2.4: Visualisasi Proses *Fully Connected Layer*

2.1.4 Aritmia

Aritmia merupakan suatu kelainan pada irama detak jantung, bisa berdetak terlalu cepat, terlalu lambat, ataupun berdetak dengan pola yang tidak beraturan, sehingga aritmia memiliki jenis yang bermacam-macam. Aritmia dapat terjadi ketika kita melakukan aktivitas berat, sehingga aritmia bisa dikatakan tidak selalu berbahaya. Oleh karena itu, seseorang harus memeriksa ke dokter secara berkala agar aritmia dapat dideteksi. Salah satu cara untuk merekam aktivitas jantung adalah dengan cara Elektrokardiograf yang akan menghasilkan sinyal Elektrokardiogram (ECG). Dengan menganalisis sinyal tersebut, aritmia dapat dideteksi dan diklasifikasikan.

2.1.5 Elektrokardiogram

Elektrokardiogram atau ECG adalah suatu sinyal yang dihasilkan oleh elektrokardiograf, yang merekam aktivitas jantung dengan menggunakan aliran listrik dalam kurun waktu tertentu. Sebuah sinyal ECG terdiri atas 1 gelombang P, 1 kompleks QRS, dan 1 gelombang T seperti pada gambar 2.5. bentuk dan panjang gelombang-gelombang tersebutlah yang dianalisis untuk mendeteksi aritmia.



Gambar 2.5: Bentuk Sinyal ECG

2.1.6 Classification Performance

Setelah menjalankan proses klasifikasi, langkah selanjutnya adalah menghitung seberapa efektif model yang dibuat berdasarkan beberapa pengukuran menggunakan *test set*. Salah satu perhitungan yang digunakan dalam kasus klasifikasi adalah *Confusion Matrix*. *Confusion Matrix* merupakan salah satu pengukuran yang paling

mudah untuk mencari tingkat kebenaran dan akurasi dari model.[6]

Confusion Matrix adalah sebuah tabel berbentuk dua dimensi yang terdiri dari data aktual dan data prediksi yang masing-masing memiliki kelas. Data aktual terletak pada bagian kolom, sedangkan data prediksi terletak pada bagian baris. Gambar 2.6 merupakan representasi visual dari *Confusion Matrix*.

		Actual Values	
		1 (Positive)	0 (Negative)
Predicted Values	1 (Positive)	TP (True Positive)	FP (False Positive) <small>Type I Error</small>
	0 (Negative)	FN (False Negative) <small>Type II Error</small>	TN (True Negative)

Gambar 2.6: *Confusion Matrix*

Pada *Confusion Matrix* terdapat empat istilah yang digunakan untuk merepresentasikan perbandingan dari kelas aktual dengan kelas yang diprediksi. Keempat istilah tersebut antara lain:

1. *True Positive*

True Positive (TP) merupakan suatu kondisi dimana data positif (1) diprediksi benar sebagai data positif (1). Pada tugas akhir ini dicontohkan sebagai data kelas *normal beat* diprediksi sebagai kelas *normal beat*.

2. *True Negative*

True Negative (TN) merupakan suatu kondisi dimana data negatif (0) diprediksi benar sebagai data negatif (0). Pada tugas akhir ini dicontohkan sebagai data kelas *normal beat* yang tidak diprediksi sebagai kelas *right bundle branch block*.

3. *False Positive*

False Positive (FP) merupakan suatu kondisi dimana data negatif (0) diprediksi sebagai data positif (1). Pada tugas akhir

ini dicontohkan sebagai data kelas *normal beat* yang diprediksi sebagai kelas *right bundle branch block*.

4. *False Negative*

False Negative (FN) merupakan suatu kondisi dimana data positif (1) diprediksi sebagai data negatif (0). Pada tugas akhir ini dicontohkan sebagai data kelas *normal beat* yang salah diprediksi sebagai kelas *right bundle branch block*.

Dari hasil *Confusion Matrix*, dapat dihitung berbagai pengukuran performa yang lebih mendetail, yaitu *accuracy*, *precision*, *recall*, dan *F1 Score*.

2.1.6.1 *Accuracy*

Accuracy menggambarkan seberapa akurat model dapat mengklasifikasikan dengan benar. Maka, *accuracy* merupakan rasio prediksi benar (TP dan TN) terhadap keseluruhan data. Dengan kata lain, *accuracy* merupakan tingkat kedekatan nilai prediksi dengan nilai aktual. Nilai *accuracy* dapat diperoleh dengan persamaan 2.1.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (2.1)$$

2.1.6.2 *Precision*

Precision menggambarkan tingkat keakuratan antara data yang diminta dengan hasil prediksi yang diberikan oleh model. Maka, *precision* merupakan rasio prediksi benar positif (TP) dibandingkan dengan keseluruhan hasil yang diprediksi positif (TP dan FP). Nilai *precision* dapat diperoleh dengan persamaan 2.2.

$$Precision = \frac{TP}{TP + FP} \quad (2.2)$$

2.1.6.3 *Recall*

Recall menggambarkan keberhasilan model dalam menemukan kembali sebuah informasi. Maka, *recall* merupakan rasio prediksi benar positif (TP) terhadap keseluruhan dari jumlah total data aktual positif (TP+FN). Nilai *recall* dapat diperoleh dengan persa-

maan 2.3.

$$Recall = \frac{TP}{TP + FN} \quad (2.3)$$

2.1.6.4 F1 Score

F1 Score adalah sebuah nilai antara 0 sampai 1 yang didapatkan dari rata-rata tertimbang (*harmonic mean*) dari *precision* dan *recall*. Nilai F1 Score dapat diperoleh dengan persamaan 2.4.

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (2.4)$$

2.2 Penelitian Terkait

2.2.1 Perbandingan Metode SVM, FUZZY-KNN, Dan BDT-SVM Untuk Klasifikasi Detak Jantung Hasil Elektrokardiografi

Pada tahun 2016, Penelitian yang dilakukan oleh Uswatun Hasanah, Lintang Resita M, Andhica Pratama, dan Colissodin ini memiliki tujuan untuk membandingkan metode SVM, Fuzzy-KNN, dan BDT-SVM untuk klasifikasi 3 jenis aritmia. Ketiga jenis aritmia itu adalah *Atrial Fibrillation*, *PVC Bigeminy*, dan *Ventricular Tachycardia* yang memiliki jumlah sampel yang sama, yaitu sebanyak 35 sampel pada setiap kelasnya. Penelitian ini memiliki tingkat akurasi sebesar 81.30% untuk metode SVM, 81.25% untuk metode Fuzzy-KNN, dan 70% untuk metode BDT-SVM.[7]

2.2.2 Klasifikasi Sinyal Ecg Menggunakan Deep Learning Dengan Stacked Denoising Autoencoders

Pada tahun 2017, Ilham Muhamamd, Jondri, M.Si, dan Untari Novia Wisety, S.T., M.T. melakukan sebuah penelitian yang bertujuan untuk mengklasifikasikan aritmia dari sinyal ECG. Terdapat lima kelas aritmia yang diklasifikasikan, yaitu *Normal Beat* dengan jumlah 2274 sampel, *Left Bundle Branch Block* dengan jumlah 706

sampel, *Right Bundle Branch Block* dengan jumlah 1967 sampel, *Atrial Premature Contraction* dengan jumlah 789 sampel, dan *Ventricular Premature Contraction* dengan jumlah 1951 sampel. Metode yang digunakan adalah *Stacked Denoising Autoencoders* (SDAE) yang merupakan salah satu metode dari *Deep Learning*. SDAE digunakan untuk merekonstruksi suatu data yang telah dirusak agar tetap memiliki informasi yang mendekati data yang sebenarnya. Hasil penelitian tersebut memiliki tingkat akurasi sebesar 98.91%. [8]

2.2.3 Cardiac arrhythmia detection using deep learning

Pada tahun 2017, Ali Isin dan Selen Ozdalili melakukan penelitian yang bertujuan untuk mengklasifikasikan aritmia. Terdapat tiga kelas aritmia yang diklasifikasikan, yaitu *Normal Beat*, *Right Bundle Branch Block*, dan *Paced Beats*. Ketiga kelas tersebut memiliki jumlah sampel yang sama, yaitu sebanyak 21600 sampel. Metode pada penelitian ini terlebih dahulu mendeteksi QRS, kemudian mengekstrak fitur QRS dengan ANN, dan akhirnya mengklasifikasikannya. Penelitian ini memiliki tingkat akurasi sebesar 92%. [9]

2.2.4 A deep convolutional neural network model to classify heartbeats

Penelitian yang dilakukan U Rajendra Acharya, Shu Lih Oh, Yuki Hagiwara, Jen Hong Tan, Muhammad Adam, Arkadiusz Gerlach, dan Tan Ru San pada tahun 2017 ini memiliki tujuan untuk mengklasifikasikan lima jenis aritmia berdasarkan pengkategorian oleh ANSI/AAMI EC57 standar tahun 2012. Kelima jenis aritmia tersebut adalah *non-ectopic* (N), *supraventricular ectopic* (S), *ventricular ectopic* (V), *fusion* (F), dan *unkown* (Q). Untuk kelas N memiliki sampel sebanyak 90592 sampel, kelas S memiliki 2781 sampel, kelas V memiliki 7235 sampel, kelas F memiliki 802 sampel, dan kelas Q memiliki 8039 sampel. Penelitian ini menggunakan metode CNN dan memiliki tingkat akurasi sebesar 94.03%. [10]

2.2.5 Arrhythmia Detection Using Deep Convolutional Neural Network With Long Duration ECG Signals

Pada tahun 2018, Penelitian yang dilakukan oleh Ozal Yildirim, Pawel Plawiak, Ru-San Tan, dan U Rajendra Acharya memiliki tujuan untuk mengklasifikasikan 17 jenis aritmia dari sinyal ECG. Penelitian ini menggunakan metode 1D CNN tanpa mendeteksi QRS dan tanpa mengekstraksi fitur QRS tersebut. Metode tersebut memiliki bentuk masukan berupa sinyal ECG berdurasi 10 detik atau sebanyak 3600 sampel. Penelitian ini melakukan percobaan dengan tiga kondisi yang memiliki jumlah kelas yang berbeda, yaitu 13 kelas, 15 kelas, dan 17 kelas. Penelitian ini menghasilkan tingkat akurasi sebesar 95.2% untuk 13 kelas, 92.51% untuk 15 kelas, dan 91.33% untuk 17 kelas.[11]

2.2.6 Electrocardiogram Classification Based on Faster Regions with Convolutional Neural Network

Pada tahun 2019, Yinsheng Ji, Sen Zhang, dan Wendong Xiao melakukan sebuah penelitian yang bertujuan untuk mengklasifikasikan lima jenis aritmia, yaitu *Normal Beat*, *Left Bundle Branch Block*, *Right Bundle Branch Block*, *Premature Ventricular Contraction*, dan *Fusion of Ventricular and Normal Beat*. Kelima kelas tersebut menggunakan data ECG dari MIT-BIH *Arrhythmia Database* dan data dari pasien. Untuk Kelas *Normal Beat* memiliki 10500 sampel, *Left Bundle Branch Block* memiliki 10074 sampel, *Right Bundle Branch Block* memiliki 9259 sampel, *Premature Ventricular Contraction* memiliki 9129 sampel, dan kelas *Fusion of Ventricular and Normal Beat* memiliki 10030 sampel. Penelitian ini menggunakan metode *Faster R-CNN* dengan *input* berupa citra ECG yang memiliki tingkat akurasi sebesar 99.21%.[12]

BAB 3

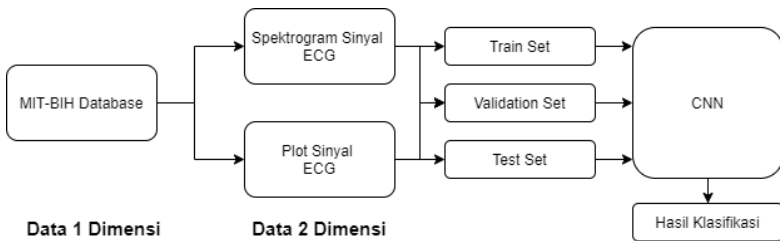
DESAIN DAN IMPLEMENTASI SISTEM

Penelitian ini dilaksanakan sesuai dengan desain sistem berikut dengan implementasinya. Desain sistem merupakan konsep dari pembuatan dan perancangan infrastruktur kemudian diwujudkan dalam bentuk blok-blok alur yang harus dikerjakan. Pada bagian implementasi merupakan pelaksanaan teknis untuk setiap blok pada desain sistem.

3.1 Desain Sistem

Tugas akhir ini merupakan salah satu bentuk penerapan dari disiplin ilmu *Deep Learning* yang bertujuan untuk mengklasifikasikan jenis-jenis aritmia dengan menggunakan metode *Convolutional Neural Network* (CNN). Gambar 3.1 menunjukkan blok diagram kerja sistem. Sistem yang dibangun menggunakan data ECG yang didapat dari MIT-BIH *Arrhythmia Database* (<https://physionet.org>). Terdapat lima jenis aritmia yang diklasifikasikan pada Tugas Akhir ini, antara lain:

1. *Normal Beat* (NOR)
2. *Right Bundle Branch Block* (RBBB)
3. *Left Bundle Branch Block* (LBBB)
4. *Premature Ventricular Contraction* (PVC)
5. *Fusion of Ventricular and Normal* (FVN)



Gambar 3.1: Blok Diagram Kerja Sistem

Bentuk data ECG dari MIT-BIH *Arrhythmia Database* merupakan sebuah sinyal yang memiliki frekuensi 360 Hz dengan durasi rekaman rata-rata 30 menit (648000 data). Tiap rekaman akan dipotong-potong sebesar 1 detik (360 data) dan dikelompokkan berdasarkan jenisnya. Kemudian sinyal ditransformasikan menjadi bentuk dua dimensi yang berupa sebuah spektogram dan sebuah plot sinyal ECG. Setelah itu, data dibagi menjadi tiga bagian, yaitu data untuk *training* sebanyak 80%, data *validation* sebanyak 10%, dan data *testing* sebanyak 10%. Data kemudian diolah menggunakan CNN yang akan menghasilkan klasifikasi aritmia.

3.2 Pengambilan Data ECG

Data yang digunakan pada proses pengerjaan Tugas Akhir ini diambil dari MIT-BIH *Arrhythmia Database*. Pada *database* tersebut terdapat 48 rekaman sinyal ECG yang diambil dari 47 pasien. Setiap rekaman memiliki frekuensi 360 Hz dengan durasi rekaman rata-rata 30 menit (648000 data).

Agar data dapat diolah menggunakan metode CNN, maka rekaman ECG dipotong-potong sepanjang 1 detik (360 data). Pemotongan 360 data dilakukan dengan acuan puncak R sebagai titik tengah. Data ECG kemudian dipotong sepanjang 180 data sebelum dan sesudah puncak R sehingga mempunyai panjang 360 data.

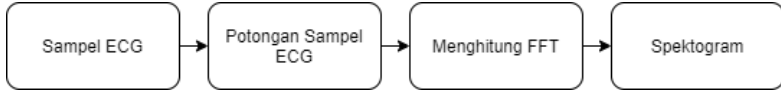
3.3 Transformasi Sinyal ECG menjadi Data 2 Dimensi

CNN merupakan metode *Deep Learning* yang paling sering digunakan untuk mengolah data dua dimensi, termasuk gambar. Dibanding dengan metode yang lain, CNN mempunyai hasil yang lebih baik untuk mengolah gambar. Sehingga, pada Tugas Akhir ini, sinyal ECG ditransformasikan menjadi data dua dimensi, yaitu berupa spektogram dan plot sinyal ECG.

3.3.1 Spektogram

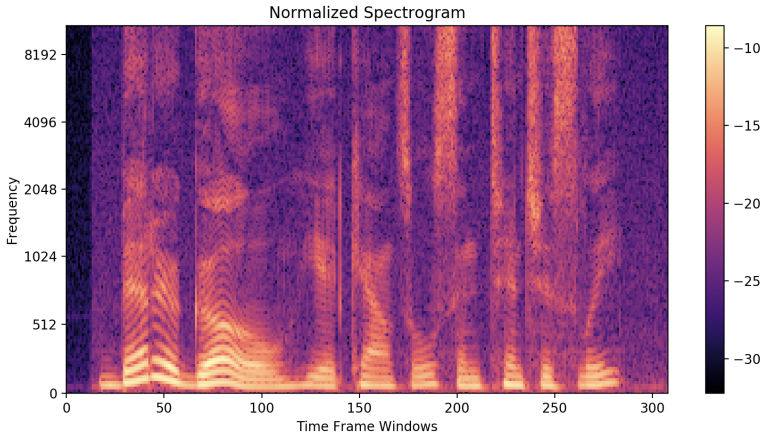
Spektogram merupakan representasi visual dari spektrum frekuensi sinyal terhadap waktu. Gambar 3.2 merupakan bagan umum proses pembuatan spektogram.

Cara untuk membuat spektogram adalah memotong sampel ECG menjadi lebih kecil (*window*) dan menghitung FFT untuk se-



Gambar 3.2: Bagan Umum Proses Pembuatan Spektrogram

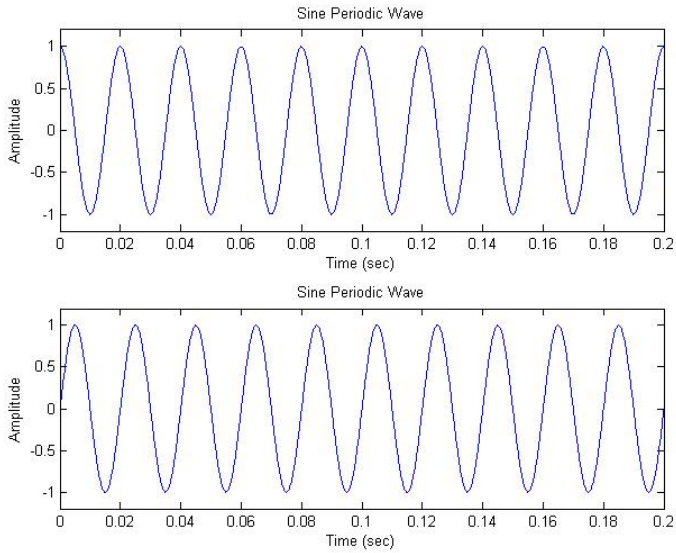
tiap *window*. Panjang *window* pada spektrogram bervariasi, semakin mirip bentuk spektrogram dengan sinyal aslinya, maka semakin bagus hasilnya. Langkah ini akan menghasilkan domain frekuensi pada setiap *window* dan nomor *window* merepresentasikan waktu. Untuk menghindari kehilangan beberapa frekuensi, maka setiap *window* harus tumpang tindih (*overlap*)[13]. Nilai *overlap* yang digunakan bervariasi, mulai dari 25% sampai 75% dari panjang *window*, tetapi yang digunakan pada umumnya adalah 50%. Gambar 3.3 merupakan contoh dari spektrogram.



Gambar 3.3: Visualisasi Spektrogram

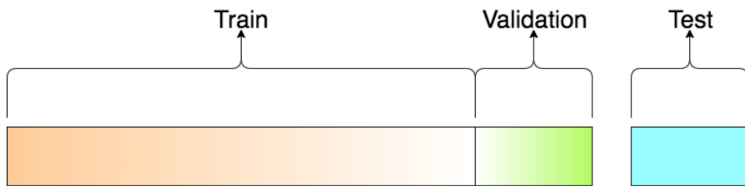
3.3.2 Plot Sinyal

Selain spektrogram, data dua dimensi bisa didapatkan dari plot sinyal. Plot ini merupakan sebuah representasi visual dari nilai sinyal terhadap waktu. Gambar 3.4 merupakan contoh dari plot sinyal.



Gambar 3.4: Visualisasi Plot Sinyal

3.4 Pembagian *Dataset*



Gambar 3.5: Visualisasi Pembagian *Dataset*

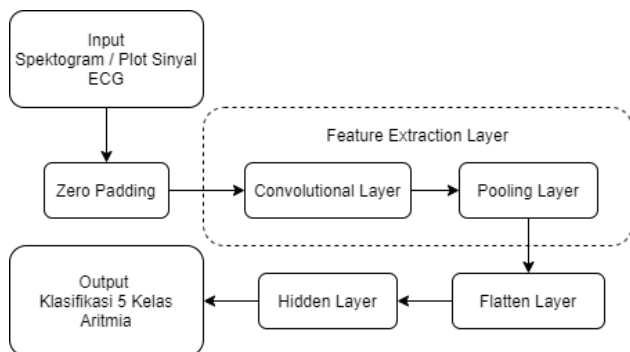
Pada *Machine Learning*, sebelum melakukan proses algoritma, *dataset* dibagi menjadi tiga *subset* seperti pada Gambar 3.5. Pada umumnya, pembagian *dataset* memiliki komposisi sebagai berikut:

- 80% *Training Set*, 10% *Validation Set*, dan 10% *Testing Set*
- 70% *Training Set*, 20% *Validation Set*, dan 10% *Testing Set*

1. *Training Set*
Training Set merupakan bagian dari *dataset* yang digunakan pada proses *training* untuk melatih model dari sebuah algoritma *Machine Learning*.
2. *Validation Set*
Validation Set merupakan bagian dari *training set* yang digunakan pada pengujian tahap awal.
3. *Testing Set*
Testing Set merupakan bagian dari *dataset* yang digunakan pada proses *testing* untuk menguji sebuah model.

3.5 Pembuatan Arsitektur CNN

CNN merupakan salah satu dari jenis *Neural Network* yang biasa digunakan untuk mengolah data dua dimensi. CNN biasa digunakan untuk mendeteksi dan mengenali suatu objek pada sebuah gambar. Secara garis besar, CNN hampir sama dengan *neural network* biasanya. Hal yang membedakannya terletak pada arsitekturnya. CNN memiliki sebuah proses untuk mengekstraksi fitur dari sebuah gambar. Proses tersebut terdiri dari dua bagian, yaitu *Convolutional Layer* dan *Pooling Layer*. Keluaran dari proses ekstraksi fitur kemudian diubah menjadi matriks 1 dimensi. Proses tersebut dinamakan sebagai *Flatten*. Setelah berbentuk matriks 1 dimensi, matriks tersebut akan dimasukkan ke *hidden layer* yang nantinya akan mengeluarkan hasil klasifikasi. Gambar 3.6 merupakan blok diagram alur kerja CNN pada tugas akhir ini.



Gambar 3.6: Blok Diagram Alur Kerja CNN

3.6 Proses *Training*

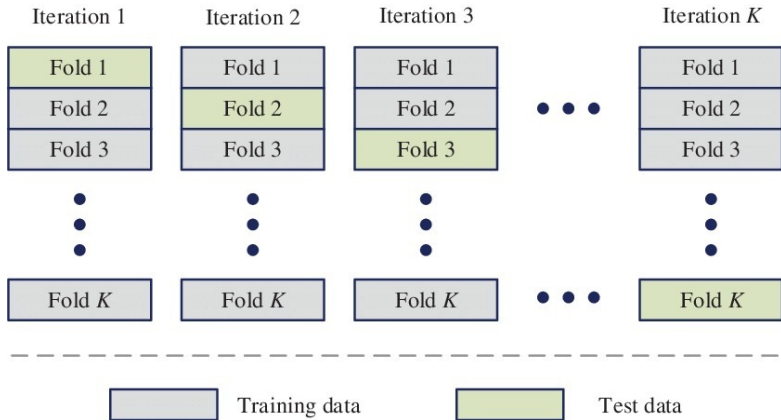
Sebelum melakukan proses *training*, perlu dilakukan penentuan ukuran *batch size*, *iteration* dan *epoch*.

1. *Batch size* mendefinisikan banyaknya sampel yang akan dibagikan ke *neural network* dalam sekali iterasi. Pada proses *training* ini, *batch size* yang digunakan sebesar 80.
2. *Iteration* merupakan jumlah *batch* yang diperlukan untuk menyelesaikan satu *epoch*.
3. *Epoch* adalah ketika seluruh dataset sudah melalui proses *training* pada *neural network* sampai dikembalikan ke awal (*Backpropagation*) untuk sekali putaran. Beberapa *epoch* diperlukan pada proses *training* agar mendapatkan kesalahan (*loss*) sekecil mungkin. Pada proses ini, *epoch* yang digunakan sebanyak 100 kali.

3.7 Proses *Validation*

Proses *Validation* sebenarnya berjalan bersamaan dengan proses *training*. Setiap *epoch*, proses *training* dan *validation* dilakukan secara berurutan. Proses ini dilakukan untuk mencegah *overfitting*. *Cross validation* adalah salah satu teknik yang populer digunakan dalam evaluasi model *Machine Learning*. Pada *cross validation*, *dataset* dibagi sebanyak K lipatan. Pada setiap iterasi, setiap lipatan akan dipakai satu kali sebagai data uji dan lipatan sisanya dipa-

kai sebagai data latih. Dengan menggunakan *cross validation* akan mendapat hasil evaluasi yang lebih akurat karena model dievaluasi dengan seluruh data. Gambar 3.7 merupakan visualisasi dari *cross validation*.



Gambar 3.7: Visualisasi *Cross Validation*

3.8 Proses *Testing*

Setelah proses *validation* dilakukan, langkah selanjutnya adalah proses *testing*. Proses ini merupakan sebuah proses untuk memprediksi hasil dari apa yang diinginkan melalui arsitektur CNN yang dibuat.

Halaman ini sengaja dikosongkan

BAB 4

PENGUJIAN DAN ANALISA

Pada bab ini dipaparkan hasil pengujian serta analisa dari desain sistem dan implementasi .Pengujian dilakukan guna mengetahui tingkat kesalahan dan menarik kesimpulan dari sistem yang telah dibuat.

Pada pengujian ini digunakan *Personal Computer* (PC) dengan spesifikasi hardware seperti pada tabel 4.1.

Processor	Intel(R) Core(TM) i5-9300H CPU @ 2.40GHz
RAM	16 GB SODIMM DDR4 2666 MHz Dual Channel
Storage	SSD M.2 NVMe 512GB
Graphic Card	NVIDIA GeForce GTX 1660 Ti 6GB
Operating System	Windows 10 Home 64-bit

Tabel 4.1: Spesifikasi PC yang Digunakan

4.1 Hasil Pengambilan Data ECG

Data ECG diambil dari MIT-BIH *Arrhythmia Database*. *Database* tersebut memiliki sejumlah 48 rekaman sinyal ECG dari 47 pasien. Terdapat dua kanal sinyal pada MIT-BIH *Arrhythmia Database*. Pada Tugas Akhir ini, sinyal ECG yang digunakan adalah kanal MLII. Sinyal tersebut mempunyai frekuensi 360 Hz dan durasi rekaman selama 30 menit.

Sebelum sinyal ECG diproses menggunakan CNN, sinyal tersebut dipotong-potong selama satu detik (360 data). Pemotongan sinyal dilakukan dengan puncak R sebagai titik tengah. Setelah sinyal dipotong, diperoleh tabel 4.2 yang menunjukkan jumlah sampel dari setiap jenis aritmia.

Kelas Aritmia	Jumlah Sampel
NOR	75011
RBBB	7255
LBBB	8071
FVN	802
PVC	7129

Tabel 4.2: Jumlah Sampel ECG pada Setiap Kelas Aritmia

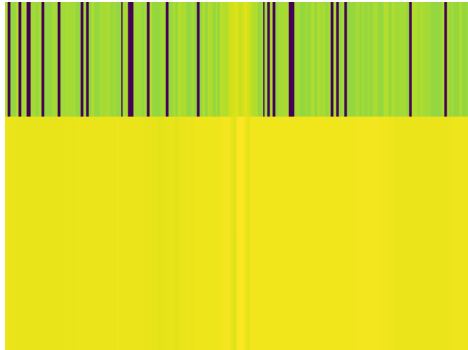
4.2 Hasil Transformasi Sinyal ECG Menjadi Data 2 Dimensi

4.2.1 Spektrogram

Pada pengujian transformasi sinyal ECG menjadi spektrogram, dilakukan beberapa pengujian dengan panjang *window* dan *overlap* yang berbeda-beda.

4.2.1.1 Panjang *Window* 4 dan *Overlap* 2

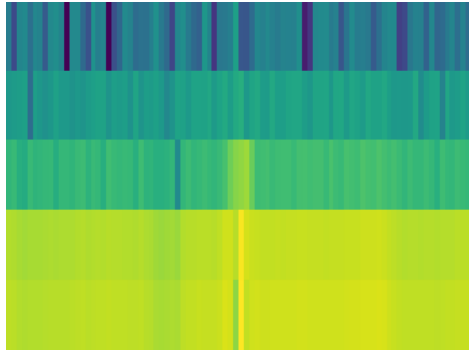
Pengujian pertama dilakukan dengan panjang *window* sebesar 4 dan *overlap* sebesar 2. Gambar 4.1 merupakan spektrogram dari kelas *Normal Beat* dengan panjang *window* sebesar 4 dan *overlap* sebesar 2.



Gambar 4.1: Spektrogram dari *Normal Beat* dengan Panjang *Window* 4 dan *Overlap* 2

4.2.1.2 Panjang *Window* 8 dan *Overlap* 4

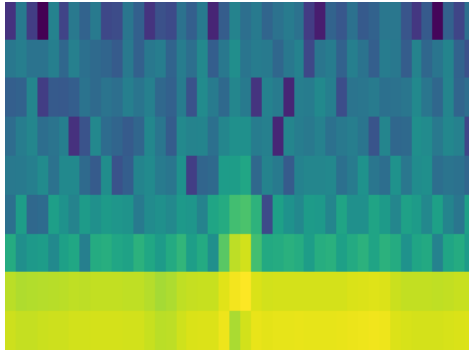
Pengujian kedua dilakukan dengan panjang *window* sebesar 8 dan *overlap* sebesar 4. Gambar 4.2 merupakan spektrogram dari kelas *Normal Beat* dengan panjang *window* sebesar 8 dan *overlap* sebesar 4.



Gambar 4.2: Spektrogram dari *Normal Beat* dengan Panjang *Window* 8 dan *Overlap* 4

4.2.1.3 Panjang *Window* 16 dan *Overlap* 8

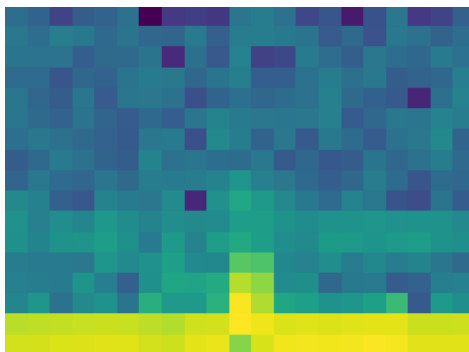
Pengujian ketiga dilakukan dengan panjang *window* sebesar 16 dan *overlap* sebesar 8. Gambar 4.3 merupakan spektrogram dari kelas *Normal Beat* dengan panjang *window* sebesar 16 dan *overlap* sebesar 8.



Gambar 4.3: Spektrogram dari *Normal Beat* dengan Panjang *Window* 16 dan *Overlap* 8

4.2.1.4 Panjang *Window* 32 dan *Overlap* 16

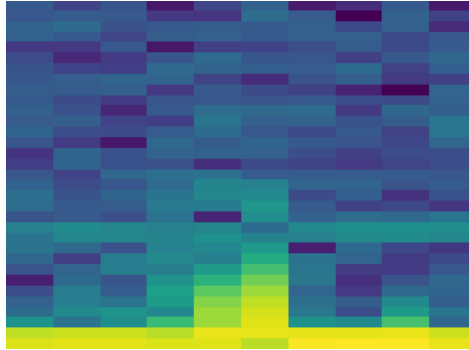
Pengujian keempat dilakukan dengan panjang *window* sebesar 32 dan *overlap* sebesar 16. Gambar 4.4 merupakan spektrogram dari kelas *Normal Beat* dengan panjang *window* sebesar 32 dan *overlap* sebesar 16.



Gambar 4.4: Spektrogram dari *Normal Beat* dengan Panjang *Window* 32 dan *Overlap* 16

4.2.1.5 Panjang *Window* 64 dan *Overlap* 32

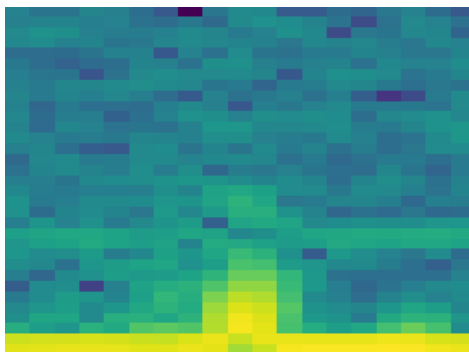
Pengujian kelima dilakukan dengan panjang *window* sebesar 64 dan *overlap* sebesar 32. Gambar 4.5 merupakan spektrogram dari kelas *Normal Beat* dengan panjang *window* sebesar 64 dan *overlap* sebesar 32.



Gambar 4.5: Spektrogram dari *Normal Beat* dengan Panjang *Window* 64 dan *Overlap* 32

4.2.1.6 Panjang *Window* 64 dan *Overlap* 48

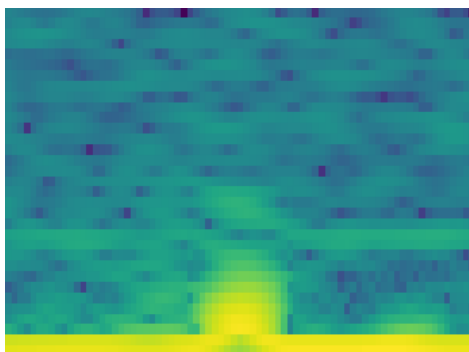
Pengujian keenam dilakukan dengan panjang *window* sebesar 64 dan *overlap* sebesar 48. Gambar 4.6 merupakan spektrogram dari kelas *Normal Beat* dengan panjang *window* sebesar 64 dan *overlap* sebesar 48.



Gambar 4.6: Spektrogram dari *Normal Beat* dengan Panjang *Window* 64 dan *Overlap* 48

4.2.1.7 Panjang *Window* 64 dan *Overlap* 60

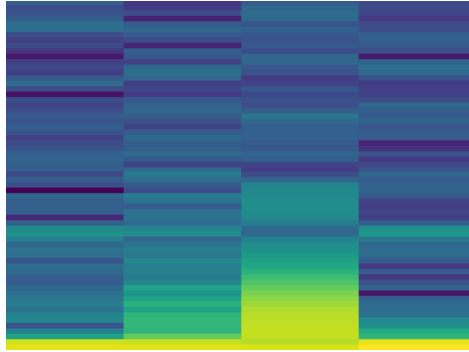
Pengujian ketujuh dilakukan dengan panjang *window* sebesar 64 dan *overlap* sebesar 60. Gambar 4.7 merupakan spektrogram dari kelas *Normal Beat* dengan panjang *window* sebesar 64 dan *overlap* sebesar 60.



Gambar 4.7: Spektrogram dari *Normal Beat* dengan Panjang *Window* 64 dan *Overlap* 60

4.2.1.8 Panjang *Window* 128 dan *Overlap* 64

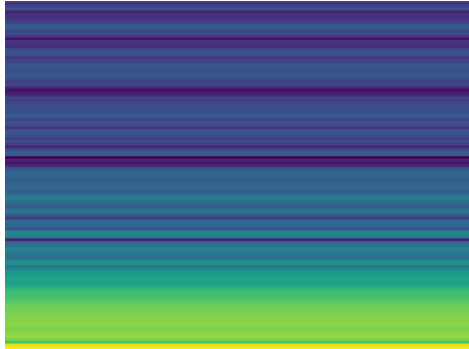
Pengujian kedelapan dilakukan dengan panjang *window* sebesar 128 dan *overlap* sebesar 64. Gambar 4.8 merupakan spektrogram dari kelas *Normal Beat* dengan panjang *window* sebesar 128 dan *overlap* sebesar 64.



Gambar 4.8: Spektrogram dari *Normal Beat* dengan Panjang *Window* 128 dan *Overlap* 64

4.2.1.9 Panjang *Window* 256 dan *Overlap* 128

Pengujian kesembilan dilakukan dengan panjang *window* sebesar 256 dan *overlap* sebesar 128. Gambar 4.9 merupakan spektrogram dari kelas *Normal Beat* dengan panjang *window* sebesar 256 dan *overlap* sebesar 128.

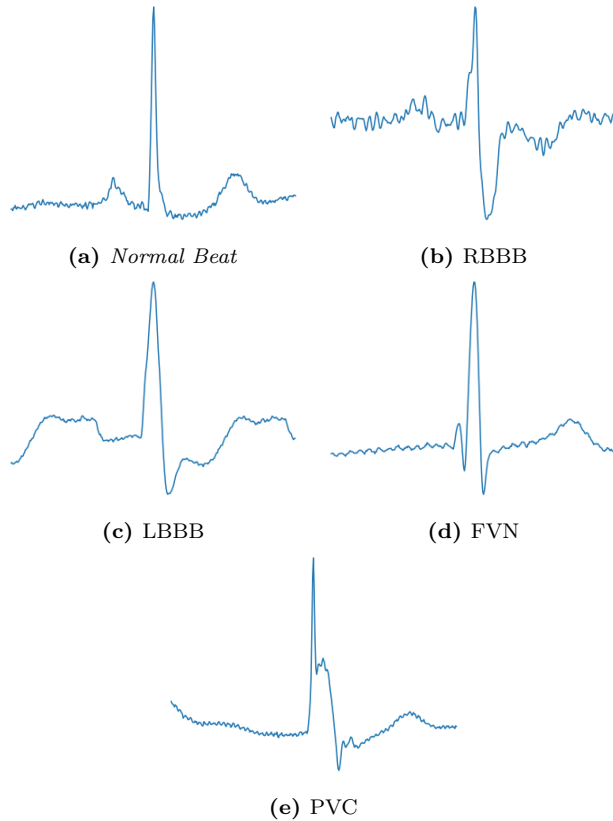


Gambar 4.9: Spektrogram dari *Normal Beat* dengan Panjang *Window* 256 dan *Overlap* 128

Dari hasil pengujian di atas, terdapat tiga spektrogram yang menyerupai sinyal ECG kelas *Normal Beat*, yaitu spektrogram dengan panjang *window* 64 dan *overlap* 32, panjang *window* 64 dan *overlap* 48, dan panjang *window* 64 dan *overlap* 60.

4.2.2 Plot Sinyal ECG

Pada pengujian plot sinyal ECG, dilakukan satu kali pengujian dan menyimpan data citra dengan ukuran 347x256 piksel. Gambar 4.10 merupakan hasil plot sinyal dari setiap kelas aritmia.



Gambar 4.10: Plot Sinyal ECG dari Setiap Kelas Aritmia

4.3 Pembagian *Dataset*

Pembagian *dataset* pada Tugas Akhir ini menggunakan komposisi 80% sebagai data *training*, 10% data *validation*, dan 10% data *testing*. Agar tidak terjadi *overfitting* pada kelas aritmia tertentu, maka jumlah data pada setiap kelas dibuat sama. *Dataset* pada Tugas Akhir ini dibagi menjadi dua model. Model pertama adalah jumlah *dataset* yang digunakan tiap kelas sebanyak 802 data yang diperoleh dari kelas aritmia FVN karena memiliki jumlah sampel

yang paling sedikit, maka jumlah tersebut digunakan sebagai acuan untuk penyetaraan jumlah data. Tabel 4.3 menunjukkan pembagian sampel ECG untuk *dataset* pada model pertama.

Kelas Aritmia	<i>Dataset</i>		
	<i>Training</i>	<i>Validation</i>	<i>Testing</i>
NOR	642	80	80
RBBB	642	80	80
LBBB	642	80	80
FVN	642	80	80
PVC	642	80	80

Tabel 4.3: Pembagian Sampel ECG untuk Dataset Model Pertama

Model kedua adalah melakukan augmentasi data pada kelas FVN agar kelas FVN memiliki sampel lebih banyak. Pada augmentasi ini, titik tengah digeser ke kiri dan ke kanan setiap satu data masing-masing sebanyak lima kali sehingga memiliki jumlah sampel sebanyak 8020 data. Sehingga kelas PVC memiliki sampel paling sedikit, yaitu 7129 sampel, dan dijadikan sebagai acuan untuk penyetaraan jumlah data.

Kelas Aritmia	<i>Dataset</i>		
	<i>Training</i>	<i>Validation</i>	<i>Testing</i>
NOR	5703	713	713
RBBB	5703	713	713
LBBB	5703	713	713
FVN	5703	713	713
PVC	5703	713	713

Tabel 4.4: Pembagian Sampel ECG untuk Dataset Model Kedua

4.4 Arsitektur CNN

Pada pengujian Tugas Akhir ini, sebuah *Convolutional Neural Network* yang memiliki 19 lapis digunakan untuk mengklasifikasikan 5 kelas aritmia. Tabel 4.5 merupakan spesifikasi arsitektur CNN yang digunakan pada Tugas Akhir ini.

Layer (type)	Output Shape	Param
InputLayer	[(None, 258, 347, 3)]	0
ZeroPadding2D	(None, 262, 351, 3)	0
Conv2D	(None, 258, 347, 16)	1216
MaxPooling2D	(None, 129, 173, 16)	0
Conv2D	(None, 127, 171, 32)	4640
MaxPooling2D	(None, 63, 85, 32)	0
Conv2D	(None, 61, 83, 32)	9248
Conv2D	(None, 59, 81, 32)	9248
MaxPooling2D	(None, 29, 40, 32)	0
Conv2D	(None, 27, 38, 32)	9248
Conv2D	(None, 25, 36, 32)	9248
MaxPooling2D	(None, 12, 18, 32)	0
Conv2D	(None, 10, 16, 64)	18496
Conv2D	(None, 8, 14, 64)	36928
MaxPooling2D	(None, 4, 7, 64)	0
Flatten	(None, 1792)	0
Dense	(None, 1024)	1836032
Dense	(None, 2048)	2099200
Dense	(None, 5)	10245

Tabel 4.5: Spesifikasi Arsitektur CNN yang Digunakan

Arsitektur tersebut memiliki *input* berupa citra berukuran 256 x 347 x 3 yang akan diproses melalui *Feature Extraction Layer* dan dilanjutkan dengan *flattening* yang akan digunakan sebagai *input* pada 2 lapis *Fully Connected Layer* dan akan menghasilkan *output* berupa presentase dari 5 kelas aritmia.

4.5 Hasil *Training* dan *Validation*

Proses *training* dan *validation* pada setiap pengujian dilakukan dengan jumlah *epoch* sebanyak 100 dan *batch size* sebesar 80. Terdapat enam pengujian pada tahap *training* dan *validation*, yaitu:

1. Pengujian dengan menggunakan *dataset* model pertama.
2. Pengujian dengan menggunakan *dataset* model kedua.

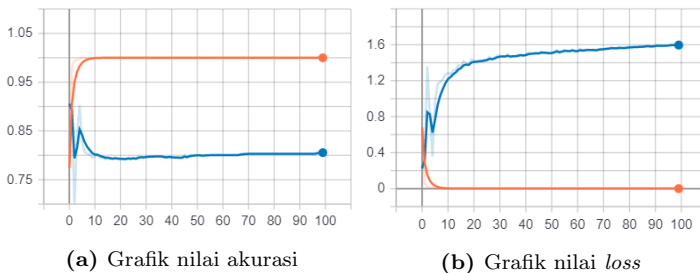
4.5.1 Pengujian dengan Menggunakan *Dataset* Model Pertama

Pada pengujian ini, dilakukan *training* dan *validation* dengan menggunakan *dataset* model pertama, yaitu dengan jumlah sampel *training* sebanyak 642, sampel *validation* sebanyak 80, dan sampel *testing* sebanyak 80. Terdapat empat pengujian pada pengujian ini, antara lain:

1. Pengujian dengan menggunakan plot sinyal ECG.
2. Pengujian dengan menggunakan spektrogram dengan panjang *window* 64 dan *overlap* 32.
3. Pengujian dengan menggunakan spektrogram dengan panjang *window* 64 dan *overlap* 48.
4. Pengujian dengan menggunakan spektrogram dengan panjang *window* 64 dan *overlap* 60.

4.5.1.1 Pengujian dengan Menggunakan Plot Sinyal ECG

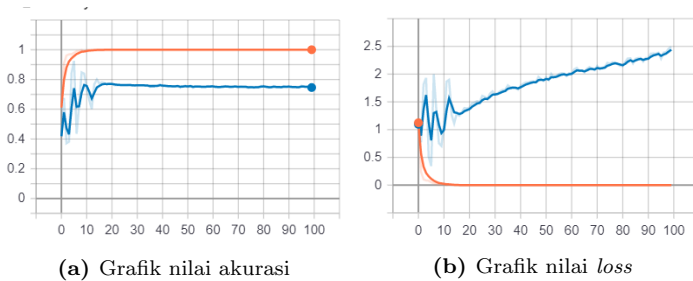
Pada pengujian ini, dilakukan *training* dan *validation* dengan menggunakan plot sinyal ECG sebagai *input*. Didapatkan hasil dengan tingkat akurasi *training* sebesar 100% dan tingkat akurasi *validation* sebesar 80.61% seperti yang ditunjukkan pada gambar 4.11a dan memiliki *loss* pada *training* sebesar $3.9203e-8$ atau mendekati 0, sedangkan *loss* pada *validation* sebesar 1.596. Gambar 4.11b merupakan grafik nilai *loss* pada proses *training* dan *validation*.



Gambar 4.11: Grafik Hasil *Training* dan *Validation* pada Pengujian Menggunakan Plot Sinyal ECG

4.5.1.2 Pengujian dengan Menggunakan Spektogram dengan Panjang *Window* 64 dan *Overlap* 32

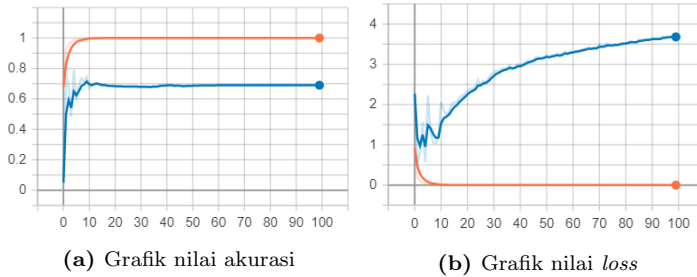
Pada pengujian ini, dilakukan *training* dan *validation* dengan menggunakan spektogram dengan panjang *window* 64 dan *overlap* 32. Didapatkan hasil dengan tingkat akurasi *training* sebesar 100% dan tingkat akurasi *validation* sebesar 74.24% seperti yang ditunjukkan pada gambar 4.12a dan memiliki *loss* pada *training* sebesar $3.7982e-5$ atau mendekati 0, sedangkan *loss* pada *validation* sebesar 2.492. Gambar 4.12b merupakan grafik nilai *loss* pada proses *training* dan *validation*.



Gambar 4.12: Grafik Hasil *Training* dan *Validation* pada Pengujian Menggunakan Spektogram dengan Panjang *Window* 64 dan *Overlap* 32

4.5.1.3 Pengujian dengan Menggunakan Spektogram dengan Panjang *Window* 64 dan *Overlap* 48

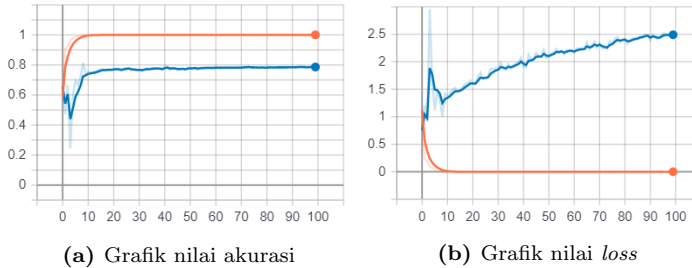
Pada pengujian ini, dilakukan *training* dan *validation* dengan menggunakan spektogram dengan panjang *window* 64 dan *overlap* 48. Didapatkan hasil dengan tingkat akurasi *training* sebesar 100% dan tingkat akurasi *validation* sebesar 68.98% seperti yang ditunjukkan pada gambar 4.13a dan memiliki *loss* pada *training* sebesar $8.2004e-8$ atau mendekati 0, sedangkan *loss* pada *validation* sebesar 3.679. Gambar 4.13b merupakan grafik nilai *loss* pada proses *training* dan *validation*.



Gambar 4.13: Grafik Hasil *Training* dan *Validation* pada Pengujian dengan Menggunakan Spektogram dengan Panjang *Window* 64 dan *Overlap* 48

4.5.1.4 Pengujian dengan Menggunakan Spektogram dengan Panjang *Window* 64 dan *Overlap* 60

Pada pengujian ini, dilakukan *training* dan *validation* dengan menggunakan spektogram dengan panjang *window* 64 dan *overlap* 60. Didapatkan hasil dengan tingkat akurasi *training* sebesar 100% dan tingkat akurasi *validation* sebesar 78.67% seperti yang ditunjukkan pada gambar 4.14a dan memiliki *loss* pada *training* sebesar $1.0431e-7$ atau mendekati 0, sedangkan *loss* pada *validation* sebesar 2.504. Gambar 4.14b merupakan grafik nilai *loss* pada proses *training* dan *validation*.



Gambar 4.14: Grafik Hasil *Training* dan *Validation* pada Pengujian dengan Menggunakan Spektogram dengan Panjang *Window* 64 dan *Overlap* 60

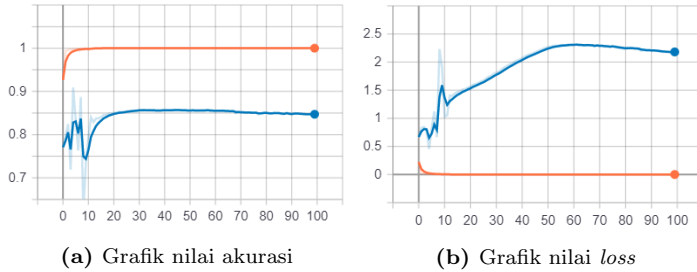
4.5.2 Pengujian dengan Menggunakan *Dataset* Model Kedua

Pada pengujian ini, dilakukan *training* dengan menggunakan *dataset* model pertama, yaitu dengan jumlah sampel *training* sebanyak 5703, sampel *validation* sebanyak 713, dan sampel *testing* sebanyak 713. Terdapat empat pengujian pada pengujian ini, antara lain:

1. Pengujian dengan menggunakan plot sinyal ECG.
2. Pengujian dengan menggunakan spektogram dengan panjang *window* 64 dan *overlap* 32.
3. Pengujian dengan menggunakan spektogram dengan panjang *window* 64 dan *overlap* 48.
4. Pengujian dengan menggunakan spektogram dengan panjang *window* 64 dan *overlap* 60.

4.5.2.1 Pengujian dengan Menggunakan Plot Sinyal ECG

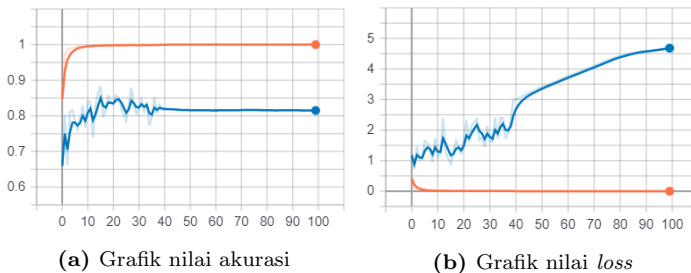
Pada pengujian ini, dilakukan *training* dan *validation* dengan menggunakan plot sinyal ECG sebagai *input*. Didapatkan hasil dengan tingkat akurasi *training* sebesar 100% dan tingkat akurasi *validation* sebesar 84.71% seperti yang ditunjukkan pada gambar 4.15a dan memiliki *loss* pada *training* sebesar $1.2404e-11$ atau mendekati 0, sedangkan *loss* pada *validation* sebesar 2.188. Gambar 4.15b merupakan grafik nilai *loss* pada proses *training* dan *validation*.



Gambar 4.15: Grafik Hasil *Training* dan *Validation* pada Pengujian Menggunakan Plot Sinyal ECG

4.5.2.2 Pengujian dengan Menggunakan Spektogram dengan Panjang *Window* 64 dan *Overlap* 32

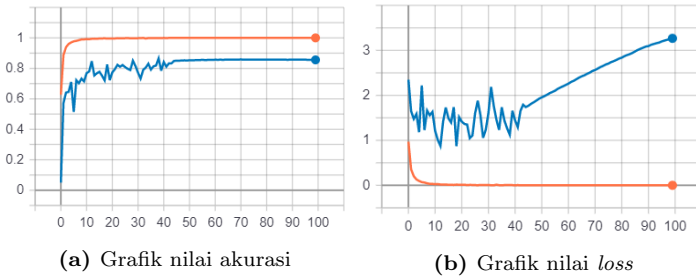
Pada pengujian ini, dilakukan *training* dan *validation* dengan menggunakan spektogram dengan panjang *window* 64 dan *overlap* 32. Didapatkan hasil dengan tingkat akurasi *training* sebesar 100% dan tingkat akurasi *validation* sebesar 81.51% seperti yang ditunjukkan pada gambar 4.16a dan memiliki *loss* pada *training* sebesar $5.9187e-14$ atau mendekati 0, sedangkan *loss* pada *validation* sebesar 4.703. Gambar 4.16b merupakan grafik nilai *loss* pada proses *training* dan *validation*.



Gambar 4.16: Grafik Hasil *Training* dan *Validation* pada Pengujian dengan Menggunakan Spektogram dengan Panjang *Window* 64 dan *Overlap* 32

4.5.2.3 Pengujian dengan Menggunakan Spektogram dengan Panjang *Window* 64 dan *Overlap* 48

Pada pengujian ini, dilakukan *training* dan *validation* dengan menggunakan spektogram dengan panjang *window* 64 dan *overlap* 48. Didapatkan hasil dengan tingkat akurasi *training* sebesar 100% dan tingkat akurasi *validation* sebesar 88.06% seperti yang ditunjukkan pada gambar 4.17a dan memiliki *loss* pada *training* sebesar $9.3632e-15$ atau mendekati 0, sedangkan *loss* pada *validation* sebesar 3.266. Gambar 4.17b merupakan grafik nilai *loss* pada proses *training* dan *validation*.

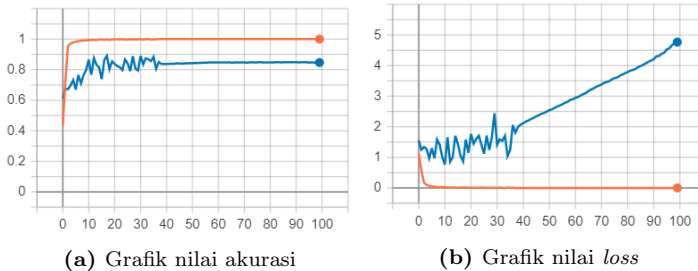


Gambar 4.17: Grafik Hasil *Training* dan *Validation* pada Pengujian dengan Menggunakan Spektogram dengan Panjang *Window* 64 dan *Overlap* 48

4.5.2.4 Pengujian dengan Menggunakan Spektogram dengan Panjang *Window* 64 dan *Overlap* 60

Pada pengujian keenam, dilakukan *training* dengan menggunakan *dataset* model kedua, yaitu dengan jumlah sampel *training* sebanyak 5703, sampel *validation* sebanyak 713, dan sampel *testing* sebanyak 713. Serta menggunakan spektogram dengan panjang *window* 64 dan *overlap* 60. Didapatkan hasil dengan tingkat akurasi *training* sebesar 100% dan tingkat akurasi *validation* sebesar 87.87% seperti yang ditunjukkan pada gambar 4.18a dan memiliki *loss* pada *training* sebesar $4.1289e-12$ atau mendekati 0, sedangkan *loss* pada

validation sebesar 2.451. Gambar 4.18b merupakan grafik nilai *loss* pada proses *training* dan *validation*.



Gambar 4.18: Grafik Hasil *Training* dan *Validation* pada PPengujian dengan Menggunakan Spektrogram dengan Panjang *Window* 64 dan *Overlap* 60

4.6 Hasil *Testing*

Proses *testing* dilakukan dengan menggunakan hasil kalkulasi *weight* pada setiap *training* pada pengujian sebelumnya sehingga terdapat delapan pengujian pada pengujian kali ini, antara lain:

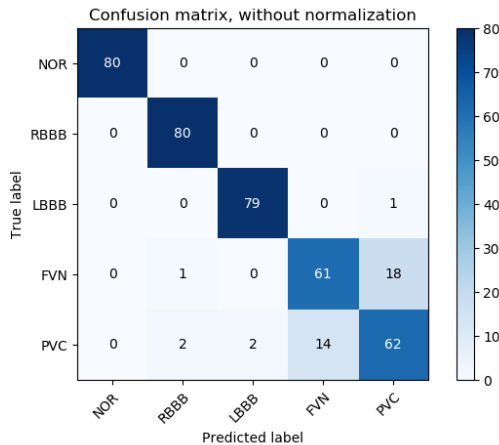
1. Pengujian dengan menggunakan *dataset* model pertama.
 - (a) Pengujian dengan menggunakan plot sinyal ECG.
 - (b) Pengujian dengan menggunakan spektrogram dengan panjang *window* 64 dan *overlap* 32.
 - (c) Pengujian dengan menggunakan spektrogram dengan panjang *window* 64 dan *overlap* 48.
 - (d) Pengujian dengan menggunakan spektrogram dengan panjang *window* 64 dan *overlap* 60.
2. Pengujian dengan menggunakan *dataset* model kedua.
 - (a) Pengujian dengan menggunakan plot sinyal ECG.
 - (b) Pengujian dengan menggunakan spektrogram dengan panjang *window* 64 dan *overlap* 32.
 - (c) Pengujian dengan menggunakan spektrogram dengan panjang *window* 64 dan *overlap* 48.
 - (d) Pengujian dengan menggunakan spektrogram dengan panjang *window* 64 dan *overlap* 60.

4.6.1 Pengujian dengan Menggunakan *Dataset* Model Pertama

Pada pengujian ini, dilakukan *testing* dengan menggunakan *dataset* model pertama, yaitu dengan jumlah sampel *training* sebanyak 642, sampel *validation* sebanyak 80, dan sampel *testing* sebanyak 80.

4.6.1.1 Pengujian dengan Menggunakan Plot Sinyal ECG

Pada pengujian ini, digunakan plot sinyal ECG sebagai *input*. Didapatkan hasil akurasi sebesar 90.5% dengan rincian deteksi kelas *Normal Beat* dengan benar sebanyak 80 sampel (100%), kelas RBBB sebanyak 80 sampel (100%), kelas LBBB sebanyak 79 sampel (98.75%), kelas FVN sebanyak 61 sampel (76.25%), dan kelas PVC sebanyak 62 sampel (77.5%). Gambar 4.20 merupakan *Confusion Matrix* dan tabel 4.6 merupakan *classification report* dari hasil pengujian dengan menggunakan plot sinyal ECG.



Gambar 4.19: *Confusion Matrix* Hasil Pengujian dengan Menggunakan Plot Sinyal ECG

Kelas	<i>Precision</i>	<i>Recall</i>	<i>F1 Score</i>
NOR	1.00	1.00	1.00
RBBB	0.96	1.00	0.98
LBBB	0.98	0.99	0.98
FVN	0.81	0.76	0.79
PVC	0.77	0.78	0.77
<i>Accuracy</i>			0.91

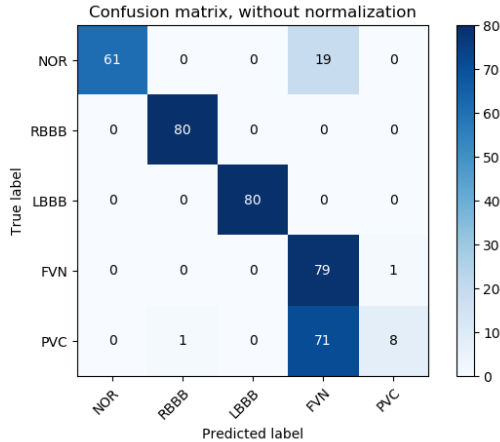
Tabel 4.6: *Classification Report* Hasil Pengujian dengan Menggunakan Plot Sinyal ECG

4.6.1.2 Pengujian dengan Menggunakan Spektogram dengan Panjang *Window* 64 dan *Overlap* 32

Pada pengujian ini, digunakan spektogram dengan panjang *window* 64 dan *overlap* 32. Didapatkan hasil akurasi sebesar 77% dengan rincian deteksi kelas *Normal Beat* dengan benar sebanyak 61 sampel (76.25%), kelas RBBB sebanyak 80 sampel (100%), kelas LBBB sebanyak 80 sampel (100%), kelas FVN sebanyak 79 sampel (98.75%), dan kelas PVC sebanyak 8 sampel (10%). Gambar 4.20 merupakan *Confusion Matrix* dan tabel 4.7 merupakan *classification report* dari hasil pengujian dengan menggunakan spektogram dengan panjang *window* 64 dan *overlap* 32.

Kelas	<i>Precision</i>	<i>Recall</i>	<i>F1 Score</i>
NOR	1.00	0.76	0.87
RBBB	0.99	1.00	0.99
LBBB	1.00	1.00	1.00
FVN	0.47	0.99	0.63
PVC	0.89	0.10	0.18
<i>Accuracy</i>			0.77

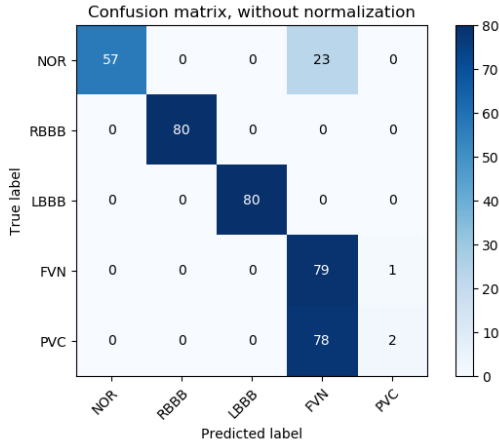
Tabel 4.7: *Classification Report* Hasil Pengujian dengan Menggunakan Spektogram dengan Panjang *Window* 64 dan *Overlap* 32



Gambar 4.20: *Confusion Matrix* Hasil Pengujian dengan Menggunakan Spektrogram dengan Panjang *Window* 64 dan *Overlap* 32

4.6.1.3 Pengujian dengan Menggunakan Spektrogram dengan Panjang *Window* 64 dan *Overlap* 48

Pada pengujian ini, digunakan spektrogram dengan panjang *window* 64 dan *overlap* 48. Didapatkan hasil akurasi sebesar 74.5% dengan rincian deteksi kelas *Normal Beat* dengan benar sebanyak 57 sampel (71.25%), kelas RBBB sebanyak 80 sampel (100%), kelas LBBB sebanyak 80 sampel (100%), kelas FVN sebanyak 79 sampel (98.75%), dan kelas PVC sebanyak 2 sampel (2.5%). Gambar 4.21 merupakan *Confusion Matrix* dan tabel 4.8 merupakan *classification report* dari hasil pengujian dengan menggunakan spektrogram dengan panjang *window* 64 dan *overlap* 48.



Gambar 4.21: *Confusion Matrix* Hasil Pengujian dengan Menggunakan Spektrogram dengan Panjang *Window* 64 dan *Overlap* 48

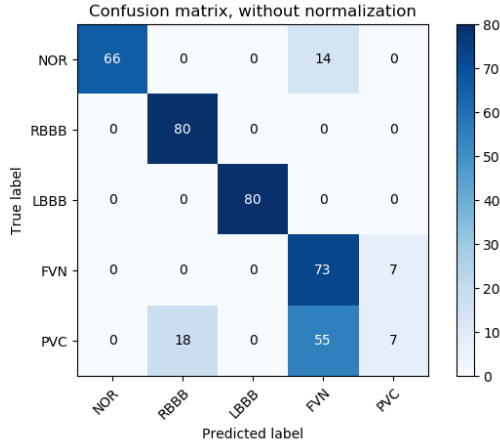
Kelas	<i>Precision</i>	<i>Recall</i>	<i>F1 Score</i>
NOR	1.00	0.71	0.83
RBBB	1.00	1.00	1.00
LBBB	1.00	1.00	1.00
FVN	0.47	0.99	0.61
PVC	0.67	0.03	0.05
<i>Accuracy</i>			0.74

Tabel 4.8: *Classification Report* Hasil Pengujian dengan Menggunakan Spektrogram dengan Panjang *Window* 64 dan *Overlap* 48

4.6.1.4 Pengujian dengan Menggunakan Spektrogram dengan Panjang *Window* 64 dan *Overlap* 60

Pada pengujian ini, digunakan spektrogram dengan panjang *window* 64 dan *overlap* 60. Didapatkan hasil akurasi sebesar 76.5% dengan rincian deteksi kelas *Normal Beat* dengan benar sebanyak 66 sampel (82.5%), kelas RBBB sebanyak 80 sampel (100%), kelas LBBB sebanyak 80 sampel (100%), kelas FVN sebanyak 73 sampel

(91.25%), dan kelas PVC sebanyak 7 sampel (8.75%). Gambar 4.22 merupakan *Confusion Matrix* dan tabel 4.9 merupakan *classification report* dari hasil pengujian dengan menggunakan spektrogram dengan panjang *window* 64 dan *overlap* 60.



Gambar 4.22: *Confusion Matrix* Hasil *Testing* Ketiga

Kelas	<i>Precision</i>	<i>Recall</i>	<i>F1 Score</i>
NOR	1.00	0.82	0.90
RBBB	0.82	1.00	0.90
LBBB	1.00	1.00	1.00
FVN	0.51	0.91	0.66
PVC	0.50	0.09	0.15
<i>Accuracy</i>			0.77

Tabel 4.9: *Classification Report* Hasil Pengujian dengan Menggunakan Spektrogram dengan Panjang *Window* 64 dan *Overlap* 60

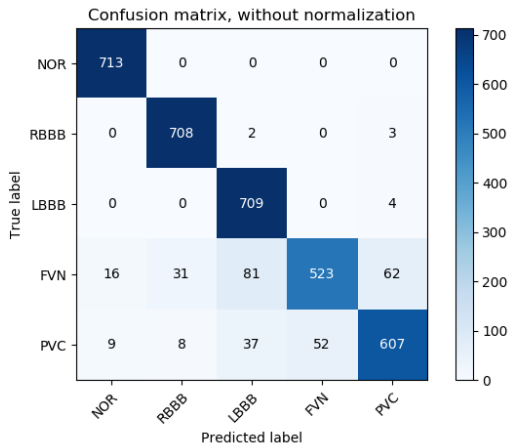
4.6.2 Pengujian dengan Menggunakan *Dataset* Model Kedua

Pada pengujian ini, dilakukan *testing* dengan menggunakan *dataset* model pertama, yaitu dengan jumlah sampel *training* se-

banyak 5703, sampel *validation* sebanyak 713, dan sampel *testing* sebanyak 713.

4.6.2.1 Pengujian dengan Menggunakan Plot Sinyal ECG

Pada pengujian ini, digunakan plot sinyal ECG sebagai *input*. Didapatkan hasil akurasi sebesar 91.58% dengan rincian deteksi kelas *Normal Beat* dengan benar sebanyak 713 sampel (100%), kelas RBBB sebanyak 708 sampel (100%), kelas LBBB sebanyak 709 sampel (99.43%), kelas FVN sebanyak 523 sampel (73.35%), dan kelas PVC sebanyak 607 sampel (85.13%). Gambar 4.23 merupakan *Confusion Matrix* dan tabel 4.10 merupakan *classification report* dari hasil pengujian dengan menggunakan plot sinyal ECG.



Gambar 4.23: *Confusion Matrix* Hasil Pengujian dengan Menggunakan Plot Sinyal ECG

Kelas	<i>Precision</i>	<i>Recall</i>	<i>F1 Score</i>
NOR	0.97	1.00	0.98
RBBB	0.95	0.99	0.97
LBBB	0.86	0.99	0.92
FVN	0.91	0.73	0.81
PVC	0.90	0.85	0.87
<i>Accuracy</i>			0.91

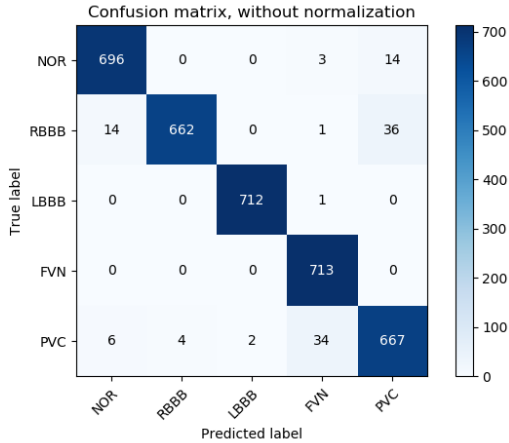
Tabel 4.10: *Classification Report* Hasil Pengujian dengan Menggunakan Plot Sinyal ECG

4.6.2.2 Pengujian dengan Menggunakan Spektogram dengan Panjang *Window* 64 dan *Overlap* 32

Pada pengujian ini, digunakan spektogram dengan panjang *window* 64 dan *overlap* 32. Didapatkan hasil akurasi sebesar 96.78% dengan rincian deteksi kelas *Normal Beat* dengan benar sebanyak 696 sampel (97.62%), kelas RBBB sebanyak 694 sampel (92.85%), kelas LBBB sebanyak 712 sampel (99.86%), kelas FVN sebanyak 713 sampel (100%), dan kelas PVC sebanyak 667 sampel (93.55%). Gambar 4.24 merupakan *Confusion Matrix* dan tabel 4.11 merupakan *classification report* dari hasil pengujian dengan menggunakan spektogram dengan panjang *window* 64 dan *overlap* 32.

Kelas	<i>Precision</i>	<i>Recall</i>	<i>F1 Score</i>
NOR	0.97	0.98	0.97
RBBB	0.99	0.93	0.96
LBBB	1.00	1.00	1.00
FVN	0.95	1.00	0.97
PVC	0.93	0.94	0.93
<i>Accuracy</i>			0.97

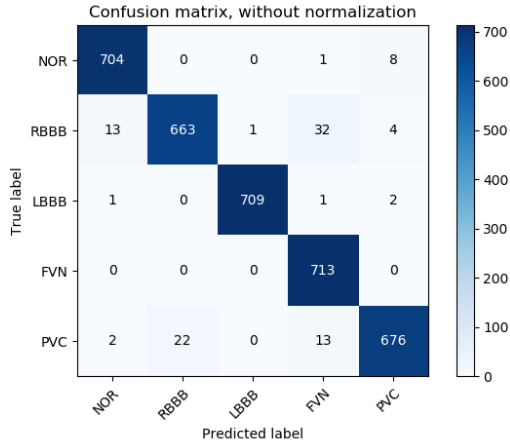
Tabel 4.11: *Classification Report* Hasil Pengujian dengan Menggunakan pektogram dengan Panjang *Window* 64 dan *Overlap* 32



Gambar 4.24: *Confusion Matrix* Hasil Pengujian dengan Menggunakan Spektrogram dengan Panjang *Window* 64 dan *Overlap* 32

4.6.2.3 Pengujian dengan Menggunakan Spektrogram dengan Panjang *Window* 64 dan *Overlap* 48

Pada pengujian ini, digunakan spektrogram dengan panjang *window* 64 dan *overlap* 48. Didapatkan hasil akurasi sebesar 97.2% dengan rincian deteksi kelas *Normal Beat* dengan benar sebanyak 704 sampel (98.74%), kelas RBBB sebanyak 663 sampel (92.99%), kelas LBBB sebanyak 709 sampel (99.44%), kelas FVN sebanyak 713 sampel (100%), dan kelas PVC sebanyak 676 sampel (94.81%). Gambar 4.25 merupakan *Confusion Matrix* dan tabel 4.12 merupakan *classification report* dari hasil pengujian dengan menggunakan spektrogram dengan panjang *window* 64 dan *overlap* 48.



Gambar 4.25: *Confusion Matrix* Hasil Pengujian dengan Menggunakan Spektrogram dengan Panjang *Window* 64 dan *Overlap* 48

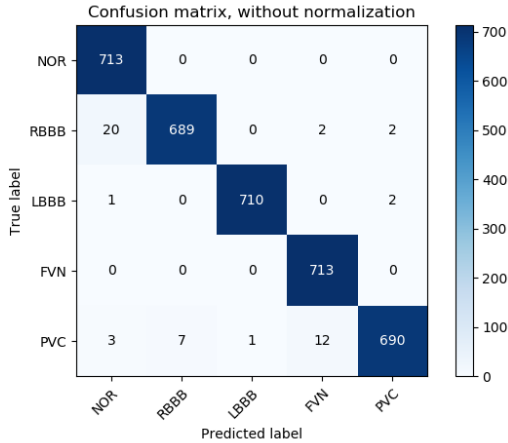
Kelas	<i>Precision</i>	<i>Recall</i>	<i>F1 Score</i>
NOR	0.98	0.99	0.98
RBBB	0.97	0.93	0.95
LBBB	1.00	0.99	1.00
FVN	0.94	1.00	0.97
PVC	0.98	0.95	0.96
<i>Accuracy</i>			0.97

Tabel 4.12: *Classification Report* Hasil Pengujian dengan Menggunakan pektrogram dengan Panjang *Window* 64 dan *Overlap* 48

4.6.2.4 Pengujian dengan Menggunakan Spektrogram dengan Panjang *Window* 64 dan *Overlap* 60

Pada pengujian ini, digunakan spektrogram dengan panjang *window* 64 dan *overlap* 60. Didapatkan hasil akurasi sebesar 98.6% dengan rincian deteksi kelas *Normal Beat* dengan benar sebanyak 713 sampel (100%), kelas RBBB sebanyak 689 sampel (96.63%), kelas LBBB sebanyak 710 sampel (99.58%), kelas FVN sebanyak

713 sampel (100%), dan kelas PVC sebanyak 690 sampel (96.77%). Gambar 4.26 merupakan *Confusion Matrix* dan tabel 4.13 merupakan *classification report* dari hasil pengujian dengan menggunakan spektrogram dengan panjang *window* 64 dan *overlap* 60.



Gambar 4.26: *Confusion Matrix* Hasil Pengujian dengan Menggunakan Spektrogram dengan Panjang *Window* 64 dan *Overlap* 60

Kelas	<i>Precision</i>	<i>Recall</i>	<i>F1 Score</i>
NOR	0.97	1.00	0.98
RBBB	0.99	0.97	0.98
LBBB	1.00	1.00	1.00
FVN	0.98	1.00	0.99
PVC	0.99	0.97	0.98
<i>Accuracy</i>			0.99

Tabel 4.13: *Classification Report* Hasil Pengujian dengan Menggunakan Spektrogram dengan Panjang *Window* 64 dan *Overlap* 60

BAB 5

PENUTUP

5.1 Kesimpulan

Pada penelitian ini, dilakukan klasifikasi lima kelas aritmia pada sinyal ECG dengan metode Convolutional Neural Network. Kelima kelas tersebut adalah *Normal Beat*, *Right Bundle Branch Block*, *Left Bundle Branch Block*, *Premature Ventricular Contraction*, dan *Fusion of Ventricular and Normal Beat*. Sinyal ECG yang sudah didapatkan akan dipotong sepanjang satu detik yang memiliki 360 data. Hasil pemotongan tersebut selanjutnya ditransformasi menjadi spektogram dengan metode FFT. Spektogram tersebut kemudian dijadikan sebagai masukan pada CNN.

Dari hasil pengujian yang sudah dilakukan pada bab sebelumnya, dapat ditarik beberapa kesimpulan sebagai berikut:

1. Hasil *testing* yang menggunakan spektogram sebagai *input* memiliki hasil yang lebih baik daripada menggunakan plot sinyal ECG.
2. Dari hasil *testing* yang menggunakan *dataset* model kedua, diketahui bahwa hasil pengujian menggunakan spektogram dengan panjang *window* 64 dan *overlap* 60 memiliki tingkat akurasi yang paling tinggi, sehingga dapat disimpulkan bahwa spektogram dengan panjang *window* 64 dan *overlap* 60 merupakan bentuk spektogram yang paling baik dibandingkan dengan bentuk spektogram lain yang diujikan.
3. Klasifikasi aritmia pada sinyal ECG yang menggunakan metode CNN sudah dapat dilakukan dengan baik, dengan tingkat akurasi sebesar 98.6% dengan rincian deteksi kelas *Normal Beat* dengan benar sebanyak 713 sampel (100%), kelas RBBB sebanyak 689 sampel (96.63%), kelas LBBB sebanyak 710 sampel (99.58%), kelas FVN sebanyak 713 sampel (100%), dan kelas PVC sebanyak 690 sampel (96.77%).

5.2 Saran

Untuk pengembangan penelitian selanjutnya terdapat beberapa saran sebagai berikut :

1. Menambah jumlah sampel pada kelas aritmia yang masih me-

miliki jumlah sampel yang sedikit yang akan digunakan pada klasifikasi aritmia.

2. Menambah kelas aritmia selain kelas *Normal Beat*, RBBB, LBBB, PVC, dan FVN.

DAFTAR PUSTAKA

- [1] World Health Organisation, “Cardiovascular diseases (cvds),” 2017. [https://www.who.int/en/news-room/fact-sheets/detail/cardiovascular-diseases-\(cvds\)](https://www.who.int/en/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds)), Last accessed on 2019-11-30. (Dikutip pada halaman 1).
- [2] MedlinePlus, “Arrhythmia,” 2016. <https://medlineplus.gov/arrhythmia.html>, Last accessed on 2019-11-30. (Dikutip pada halaman 1).
- [3] MedlinePlus, “Holter monitor (24h),” 2018. <https://medlineplus.gov/ency/article/003877.htm>, Last accessed on 2019-11-30. (Dikutip pada halaman 1).
- [4] A. Isin and S. Ozdalili, “Cardiac arrhythmia detection using deep learning,” Procedia computer science, vol. 120, pp. 268–275, 2017. (Dikutip pada halaman 1).
- [5] MathWorks, “What is deep learning?.” <https://ww2.mathworks.cn/en/discovery/deep-learning.html>, Last accessed on 2019-12-14. (Dikutip pada halaman 7).
- [6] Mohammed Sunasra, “Performance metrics for classification problems in machine learning,” 2017. <https://medium.com/@MohammedS/performance-metrics-for-classification-problems-in-mach> Last accessed on 2020-07-25. (Dikutip pada halaman 12).
- [7] U. Hasanah, L. R. Mayangsari, A. Pratama, and I. Cholissodin, “Perbandingan metode svm, fuzzy-knn, dan bdt-svm untuk klasifikasi detak jantung hasil elektrokardiografi,” Jurnal Teknologi Informasi dan Ilmu Komputer, vol. 3, no. 3, pp. 201–207, 2016. (Dikutip pada halaman 14).
- [8] I. Muhammad, J. Jondri, and U. Wisesty, “Klasifikasi sinyal ecg menggunakan deep learning dengan stacked denoising autoencoders,” eProceedings of Engineering, vol. 4, no. 3, 2017. (Dikutip pada halaman 15).

- [9] A. Isin and S. Ozdalili, “Cardiac arrhythmia detection using deep learning,” Procedia computer science, vol. 120, pp. 268–275, 2017. (Dikutip pada halaman 15).
- [10] U. R. Acharya, S. L. Oh, Y. Hagiwara, J. H. Tan, M. Adam, A. Gertych, and R. San Tan, “A deep convolutional neural network model to classify heartbeats,” Computers in biology and medicine, vol. 89, pp. 389–396, 2017. (Dikutip pada halaman 15).
- [11] Ö. Yildırım, P. Pławiak, R.-S. Tan, and U. R. Acharya, “Arrhythmia detection using deep convolutional neural network with long duration ecg signals,” Computers in biology and medicine, vol. 102, pp. 411–420, 2018. (Dikutip pada halaman 16).
- [12] Y. Ji, S. Zhang, and W. Xiao, “Electrocardiogram classification based on faster regions with convolutional neural network,” Sensors, vol. 19, no. 11, p. 2558, 2019. (Dikutip pada halaman 2, 16).
- [13] Kartik Chaudhary, “Understanding audio data, fourier transform, fft and spectrogram features for a speech recognition system,” 2019. <https://towardsdatascience.com/understanding-audio-data-fourier-transform-fft-spectrogram>. Last accessed on 2020-02-20. (Dikutip pada halaman 19).

BIOGRAFI PENULIS



Firdaus Nanda Pradangapasti, atau biasa dipanggil Firdaus, lahir di Pasuruan Jawa Timur pada tanggal 16 Maret 1998. Merupakan anak pertama dari dua bersaudara. Penulis lulus dari SMP Negeri 1 Purwosari dan melanjutkan ke SMA Negeri 1 Purwosari. Penulis melanjutkan ke jenjang strata satu di Departemen Teknik Komputer Fakultas Teknologi Elektro dan Informatika Cerdas ITS. Dalam masa kuliah, penulis tertarik dengan pengembangan *Web Apps* dan *Internet of*

Things (IoT). Penulis pernah aktif menjadi Staf Departemen Komunikasi dan Informasi Himpunan Mahasiswa Teknik Elektro ITS serta menjadi Koordinator Acara MAGE 4. Bagi pembaca yang memiliki kritik, saran, atau pertanyaan mengenai tugas akhir ini dapat menghubungi penulis melalui email firdausnp16@gmail.com.

Halaman ini sengaja dikosongkan