



ITS
Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR - EC184801

**DETEKSI *R PEAK* DAN KLASIFIKASI
ARITMIA MENGGUNAKAN *HYBRID QVAT* DAN CNN
BERBASIS *SINGLE BOARD COMPUTER***

Mpu Hambyah Syah Bagaskara Aji
NRP 07211740000010

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

DEPARTEMEN TEKNIK KOMPUTER
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember
Surabaya 2021

Halaman ini sengaja dikosongkan



ITS

Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR - EC184801

**DETEKSI *R PEAK* DAN KLASIFIKASI
ARITMIA MENGGUNAKAN *HYBRID QVAT* DAN CNN
BERBASIS *SINGLE BOARD COMPUTER***

Mpu Hambyah Syah Bagaskara Aji
NRP 0721174000010

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

DEPARTEMEN TEKNIK KOMPUTER
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember
Surabaya 2021

Halaman ini sengaja dikosongkan



FINAL PROJECT - EC184801

***DETECTION OF R PEAK AND ARRHYTHMIA
CLASSIFICATION USING HYBRID QVAT AND CNN
BASED ON SINGLE BOARD COMPUTER***

Mpu Hambyah Syah Bagaskara Aji
NRP 0721174000010

Advisor
Arief Kurniawan, ST., MT.
Dr. Reza Fuad Rachmadi, S.T., M.T.

DEPARTEMENT OF COMPUTER ENGINEERING
Faculty of Intelligent Electrical And Informatics Technology
Sepuluh Nopember Institute of Technology
Surabaya 2021

Halaman ini sengaja dikosongkan

PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul “**Deteksi R Peak dan Klasifikasi Aritmia Menggunakan Hybrid QVAT dan CNN Berbasis Single Board Computer**” adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak 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 peraturan yang berlaku.

Surabaya, Juli 2021



Mpu Hambyah Syah Bagaskara Aji
NRP. 0721174000010

Halaman ini sengaja dikosongkan

LEMBAR PENGESAHAN

Deteksi *R Peak* dan Klasifikasi Aritmia Menggunakan *Hybrid* QVAT dan CNN Berbasis *Single Board Computer*

Tugas Akhir ini disusun untuk memenuhi salah satu syarat memperoleh gelar Sarjana Teknik di Institut Teknologi Sepuluh Nopember Surabaya

Oleh: Mpu Hambyah Syah Bagaskara Aji (NRP: 07211740000010)

Tanggal Ujian: 23 Juli 2021

Periode Wisuda: Oktober 2021

Disetujui oleh:

Arief Kurniawan, S.T., M.T.

(Pembimbing I)

NIP: 197409072002121001

Dr. Reza Fuad Rachmadi, S.T., M.T.

(Pembimbing II)

NIP: 198504032012121001

Eko Pramunanto, ST., MT.

(Penguji I)

NIP: 196612031994121001

Ahmad Zaini, ST., M.Sc.

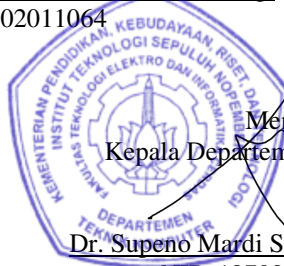
(Penguji II)

NIP: 197504192002121003

Dion Hayu Fandiantoro, S.T., M.Eng.

(Penguji III)

NPP: 1994202011064



Mengetahui
Kepala Departemen Teknik Komputer

Dr. Supeno Mardi Susiki Nugroho, ST., MT.

NIP. 197003131995121001

Halaman ini sengaja dikosongkan

ABSTRAK

Nama Mahasiswa : Mpu Hambyah Syah Bagaskara Aji
Judul Tugas Akhir : Deteksi *R Peak* dan Klasifikasi Aritmia Menggunakan *Hybrid QVAT* dan CNN Berbasis *Single Board Computer*
Pembimbing : 1. Arief Kurniawan, ST., MT.
2. Dr. Reza Fuad Rachmadi, ST., MT.

Aritmia adalah suatu gejala dari gangguan irama jantung yang bisa mengindikasikan pada penyakit kardiovaskular yang menjadi penyebab utama kematian di dunia. Untuk mendiagnosis aritmia, salah satu cara yang dapat digunakan adalah dengan menganalisis sinyal ECG (*Electrocardiogram*). Paramedis masih menggunakan cara manual untuk menganalisis sinyal ECG. Pada tugas akhir ini akan dikembangkan alat untuk mengklasifikasikan jenis aritmia berbasis *single board computer*. Penggunaan *single board computer* dalam memproses klasifikasi aritmia menggunakan metode *deep learning* yaitu secara *portable*. Metode QVAT digunakan untuk segmentasi kompleks QRS pada sinyal ECG. Hasil segmentasi berupa *r peak* yang dijadikan titik anotasi pada tahap *preprocessing* agar dapat digunakan sebagai *input* tahap klasifikasi jenis aritmia menggunakan metode CNN (*Convolutional Neural Network*). Hasil dari tugas akhir ini adalah berupa model *weight* yang dapat mengklasifikasikan jenis aritmia dengan akurasi 99.55% yang dapat diimplementasikan dengan baik menggunakan alat berbasis *single board computer* sehingga dapat membantu paramedis untuk mendiagnosis penyakit jantung sedini mungkin pada pasien agar dapat menekan angka kematian akibat penyakit jantung.

Kata Kunci : Aritmia, *Deep Learning*, *Electrocardiogram*, *Single Board Computer*

Halaman ini sengaja dikosongkan

ABSTRACT

Name : Mpu Hambyah Syah Bagaskara Aji
Title : *R Peak Detection and Arrhythmia Classification Using Hybrid QVAT and CNN Based on Single Board Computer*
Advisors : 1. Arief Kurniawan, ST., MT.
2. Dr. Reza Fuad Rachmadi, ST., MT.

Arrhythmia is a symptom of a heart rhythm disorder that can indicate cardiovascular disease which is the leading cause of death in the world. To diagnose arrhythmia, one way that can be used is to analyze the ECG (Electrocardiogram) signal. Paramedic still use manual methods to analyze ECG signals. In this final project, a tool is developed to classify the types of arrhythmias using single board computer based with deep learning method. The QVAT method is used to segment the QRS complex on the ECG signal. The r peak from the segmentation results will be used as an annotation at the preprocessing step so that it can be used as input for the classification step of arrhythmia types using the CNN (Convolutional Neural Network) method. The expected result is a weight model that can classify the type of arrhythmia with 99.55% accuracy so that it can be implemented properly using single board computer based so that it can help paramedics to diagnose heart disease as early as possible in patients in order to reduce the number death from the disease.

Keywords : Arrhythmia, Deep Learning, Electrocardiogram, Single Board Computer

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 **Deteksi *R Peak* dan Klasifikasi Aritmia Menggunakan *Hybrid QVAT* dan *CNN* Berbasis *Single Board Computer***. Penelitian ini disusun dalam rangka pemenuhan bidang riset di Departemen Teknik Komputer ITS, 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 e57, Teknik Komputer, Laboratorium B401 dan B201 Teknik Komputer ITS serta Alumni yang telah banyak memberikan banyak bantuan.

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, Juli 2021

Mpu Hambyah Syah Bagaskara Aji

Halaman ini sengaja dikosongkan

DAFTAR ISI

ABSTRAK	i
ABSTRACT	iii
KATA PENGANTAR	v
DAFTAR ISI	vii
DAFTAR GAMBAR	xi
DAFTAR TABEL	xiii
NOMENKLATUR	xv
1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Perumusan Masalah	3
1.3 Tujuan	3
1.4 Batasan Masalah	3
1.5 Sistematika Penulisan	4
2 TINJAUAN PUSTAKA	7
2.1 <i>State of The Art</i>	7
2.1.1 <i>IOT Based Detection of Cardiac Arrhythmias Using Raspberry PI</i>	7
2.1.2 <i>Portable ECG Monitor with Low Power Consumption and Small Size Based on AD8232 Chip</i>	8
2.1.3 <i>Cardiac Arrhythmia Detection using Deep Learning</i>	8
2.1.4 <i>Arrhythmia Detection Using Deep Convolutional Neural Network with Long Duration</i>	8
2.1.5 <i>A Deep Convolutional Neural Network Model to Classify Heartbeats</i>	8

2.1.6	Perbandingan Metode <i>SVM</i> , <i>FUZZY-KNN</i> , Dan <i>BDT-SVM</i> untuk Klasifikasi Detak Jantung Hasil Elektrokardiografi	9
2.1.7	<i>QVAT: QRS Complex Detection based on Variance Analysis and Adaptive Threshold for Electrocardiogram Signal</i>	9
2.1.8	Pendeteksi Penyakit Aritmia Berbasis Prediksi Fitur QRS pada Sinyal ECG dengan Metode <i>Feed Forward Backpropagation</i> menggunakan Sistem Tertanam	9
2.1.9	Klasifikasi Aritmia pada Sinyal ECG Berbasis <i>Deep Learning</i>	9
2.2	Teori Penunjang	9
2.2.1	Segmen Sinyal Jantung	10
2.2.2	Elektrokardiogram	10
2.2.3	Kompleks QRS	10
2.2.4	Aritmia	11
2.2.5	<i>Machine Learning</i>	12
2.2.6	<i>Deep Learning</i>	12
2.2.7	<i>Convolutional Neural Network</i>	13
	2.2.7.1 <i>Convolutional Layer</i>	14
	2.2.7.2 <i>Padding</i>	14
	2.2.7.3 <i>Pooling Layer</i>	15
	2.2.7.4 <i>Fully Connected Layer</i>	15
2.2.8	<i>Classification Performance</i>	16
	2.2.8.1 <i>Accuracy</i>	17
	2.2.8.2 <i>Precision</i>	17
	2.2.8.3 <i>Recall</i>	18
	2.2.8.4 <i>F1 Score</i>	18
2.2.9	Metode QVAT	18
2.2.10	Raspberry Pi 3 Model B+	19
2.2.11	Sensor Sparkfun AD8232	19
2.2.12	Intel Neural Compute Stick 2	20
2.2.13	ADS1115	21
3	DESAIN DAN IMPLEMENTASI SISTEM	23
3.1	Desain Sistem	23
3.2	Desain dan Perancangan Perangkat ECG	24

3.3	Pengambilan Data ECG sebagai <i>Input</i>	25
3.4	<i>Preprocessing</i> - Pembagian <i>Dataset</i>	28
3.5	<i>Preprocessing</i> - Augmentasi <i>Dataset</i>	29
3.6	<i>Preprocessing</i> - Deteksi <i>R Peak</i> (Puncak R)	30
3.7	<i>Preprocessing</i> - Pemotongan Sinyal	31
3.8	<i>Preprocessing</i> - Spektogram	31
3.9	<i>Preprocessing</i> - Pembagian Variasi <i>Dataset</i>	32
3.10	<i>Preprocessing</i> - <i>Filtering Dataset</i>	33
3.11	<i>Preprocessing</i> - Penambahan <i>Dataset</i>	36
3.12	Arsitektur CNN	37
3.13	Proses <i>Training</i>	37
3.14	Proses <i>Testing</i> Menggunakan <i>Dataset</i>	39
3.15	Proses <i>Testing</i> Menggunakan Data Akuisisi	39
3.16	Visualisasi Hasil Menggunakan <i>User Interface</i>	39
4	HASIL DAN PENGUJIAN	41
4.1	Implementasi <i>Hardware</i>	41
4.2	Pengujian Hasil <i>Training</i> dan <i>Validation</i> Model	42
4.2.1	Hasil <i>Training</i> dan <i>Validation</i> Menggunakan Variasi Pembagian <i>Dataset</i>	42
4.2.2	Hasil <i>Training</i> dan <i>Validation</i> Menggunakan <i>Filter Dataset</i>	43
4.2.3	Hasil <i>Training</i> dan <i>Validation</i> Menggunakan Penambahan <i>Dataset</i>	44
4.3	Pengujian <i>Testing</i> Model	45
4.3.1	Pengujian <i>Testing</i> Model Menggunakan Variasi Pembagian <i>Dataset</i>	46
4.3.2	Pengujian <i>Testing</i> Model Menggunakan <i>Filter Dataset</i>	48
4.3.3	Pengujian <i>Testing</i> Model Menggunakan Penambahan <i>Dataset</i>	50
4.4	Pengujian <i>R Peak</i>	52
4.5	Pengujian Sistem Menggunakan <i>Dataset</i>	53
4.6	Pengujian Sistem Menggunakan Data Akuisisi	55
4.6.1	Pengujian Pada Pasien	55
4.6.2	Pengujian Pada Simulator	57
4.7	Pengujian <i>Execution Time</i> Analisis	58
4.8	Pengujian Menggunakan <i>User Interface</i>	60

5 PENUTUP	63
5.1 Kesimpulan	63
5.2 Saran	64
DAFTAR PUSTAKA	65
BIOGRAFI PENULIS	69

DAFTAR GAMBAR

2.1	Bentuk Sinyal ECG	11
2.2	Diagram <i>Machine Learning</i>	13
2.3	Visualisasi dari Arsitektur CNN	14
2.4	<i>Confusion Matrix</i>	16
2.5	Raspberry Pi 3 Model B+	19
2.6	Sparkfun AD8232	20
2.7	Intel Neural Compute Stick 2	21
2.8	ADS1115	22
3.1	Blok Diagram Kerja Sistem	24
3.2	Diagram Perangkat ECG	25
3.3	Desain Perangkat ECG	25
3.4	Desain <i>Nesting Box</i>	26
3.5	Penempatan Posisi Elektroda ECG	28
3.6	Visualisasi Pembagian <i>Dataset</i>	29
3.7	Blok Diagram Metode QVAT	30
3.8	Blok Diagram Transformasi Spektogram	31
3.9	Sampel Hasil Transformasi Sinyal Berupa Spektogram <i>Window 64 dan Overlap 60</i>	32
3.10	<i>Plot</i> Sinyal Pada Kelas LBBB	33
3.11	Visualisasi Pembagian <i>Dataset</i>	34
3.12	<i>Plot</i> Sinyal Pada Kelas Normal	35
3.13	<i>Plot</i> Sinyal Pada Kelas LBBB	35
3.14	Hasil Proses <i>Filtering</i> Pada <i>Record-118</i>	35
3.15	<i>Plot</i> Hasil Akuisisi Data Simulator	36
3.16	<i>Plot</i> Hasil Proses Konversi <i>Range</i>	36
3.17	<i>Plot</i> Hasil Deteksi <i>R Peak</i>	37
3.18	Blok Diagram Alur Kerja CNN	39
3.19	Desain <i>User Interface</i>	40
4.1	Foto Perangkat	41
4.2	Foto Perangkat Setiap Sudut	42
4.3	Grafik Hasil <i>Training</i> dan <i>Validation</i> Pada Penelitian Oleh Firdaus Nanda Pradangapasti	43

4.4	Grafik Hasil <i>Training</i> dan <i>Validation</i> Pada Pengujian Dengan Menggunakan Variasi Pembagian <i>Dataset</i>	44
4.5	Grafik Hasil <i>Training</i> dan <i>Validation</i> Pada Pengujian Dengan Menggunakan Variasi Pembagian <i>Dataset</i> dan <i>Filtering Dataset</i>	45
4.6	Grafik Hasil <i>Training</i> dan <i>Validation</i> Pada Pengujian Dengan Menggunakan Variasi Pembagian <i>Dataset</i> , <i>Filtering Dataset</i> dan Penambahan <i>Dataset</i>	46
4.7	<i>Confusion Matrix</i> Hasil Pengujian Oleh Firdaus Nanda Pradangapasti	47
4.8	<i>Confusion Matrix</i> Hasil Pengujian Dengan Menggunakan Variasi Pembagian <i>Dataset</i>	48
4.9	<i>Confusion Matrix</i> Hasil Pengujian Dengan Menggunakan Variasi Pembagian <i>Dataset</i> dan <i>Filtering Dataset</i>	49
4.10	<i>Confusion Matrix</i> Hasil Pengujian Dengan Menggunakan Variasi Pembagian <i>Dataset</i> , <i>Filtering Dataset</i> dan Penambahan <i>Dataset</i>	51
4.11	Hasil Proses Akuisisi Data Pasien selama 10 Detik	56
4.12	Hasil Proses Konversi Nilai <i>Range</i> dan Deteksi <i>R Peak</i>	56
4.13	Hasil <i>Filtering</i>	56
4.14	Hasil Proses Konversi Frekuensi	57
4.15	Hasil Proses Akuisisi Data Simulator selama 10 Detik	57
4.16	Hasil Proses Konversi Nilai <i>Range</i> dan Deteksi <i>R Peak</i>	58
4.17	Hasil Proses <i>Filtering</i>	58
4.18	Hasil Proses Konversi Frekuensi	58
4.19	Tampilan Hasil <i>Record</i>	60
4.20	Tampilan Hasil Analisis	61

DAFTAR TABEL

3.1	Konfigurasi Perangkat ECG	26
3.2	Jumlah Sampel ECG pada Setiap Kelas Aritmia . .	27
3.3	Pembagian Sampel ECG	29
3.4	Pembagian Sampel ECG Setelah Augmentasi Data .	30
3.5	Pembagian <i>Dataset</i> Tanpa Menggunakan Variasi . .	34
3.6	Pembagian <i>Dataset</i> Menggunakan Variasi	34
3.7	Arsitektur CNN yang digunakan	38
4.1	<i>Classification Report</i> Hasil Pengujian Oleh Firdaus Nanda Pradanggapasti	47
4.2	<i>Classification Report</i> Hasil Pengujian Dengan Menggunakan Variasi Pembagian <i>Dataset</i>	48
4.3	<i>Classification Report</i> Hasil Pengujian Menggunakan Variasi Pembagian <i>Dataset</i> dan <i>Filtering Dataset</i> . .	50
4.4	<i>Classification Report</i> Hasil Pengujian Menggunakan Variasi Pembagian <i>Dataset</i> , <i>Filtering Dataset</i> dan Penambahan <i>Dataset</i>	51
4.5	Perbandingan Hasil Pengujian <i>Testing Model</i>	52
4.6	<i>Confusion Matrix</i> dari Deteksi <i>R Peak</i>	52
4.7	Nilai <i>Accuracy</i> , <i>Precision</i> , <i>Recall</i> , <i>F1-Score</i> dari Deteksi <i>R Peak</i>	53
4.8	Hasil Pengujian Sistem Menggunakan <i>Dataset</i> . . .	55
4.9	Spesifikasi <i>Personal Computer</i>	59
4.10	Spesifikasi Raspberry Pi 3 Model B+	59
4.11	Hasil <i>Execution Time Personal Computer</i> dan Raspberry Pi 3 Model B+	59

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

Aritmia merupakan suatu kondisi dimana irama detak jantung tidak beraturan, bisa berdetak terlalu cepat, terlalu lambat, ataupun berdetak membentuk pola yang tidak beraturan. Kebanyakan aritmia tidak berbahaya, tetapi beberapa bisa serius atau bahkan mengancam jiwa. Pada penderita aritmia, jantung tidak dapat memompa cukup darah ke seluruh tubuh. Kekurangan aliran darah dapat merusak otak, jantung, dan organ lainnya [1].

Menurut data WHO (*World Health Organization*) 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 [2]. Berdasarkan data dari *American Heart Association*, penyakit kardiovaskular terdata sebagai penyebab utama kematian, terhitung sebanyak hampir 836.546 kematian di Amerika Serikat dan menjadi penyebab kematian secara global yang dilaporkan sudah lebih dari 17,9 juta kematian setiap tahun pada tahun 2015, angka tersebut diperkirakan akan tumbuh menjadi lebih dari 23,6 juta pada tahun 2030 [3]. Salah satu cara untuk mendiagnosis penyakit kardiovaskular yaitu dengan menganalisis sinyal ECG untuk melihat aktivitas kelistrikan yang dihasilkan oleh jantung. Di Indonesia, jumlah dokter dengan pasien kardiovaskular tidak sebanding diantaranya memiliki rasio perbandingan mencapai 1:250.000 [4] sehingga pasien akan masuk *waiting list* untuk menerima pemeriksaan atau penanganan medis oleh dokter spesialis jantung. Aktivitas jantung dapat direkam dengan menggunakan alat yang disebut dengan elektrokardiograf yang menghasilkan sinyal ECG. Pada sinyal ECG, dalam setiap satu detakan jantung, terdapat satu gelombang P, satu kompleks QRS dan satu gelombang T. Gelombang-gelombang tersebutlah yang diguna-

kan sebagai parameter untuk mengklasifikasikan jenis-jenis aritmia [5]. Penyakit jantung dapat dideteksi sejak dini dengan cara mendiagnosis aritmia. Tidak semua aritmia berbahaya, karena ketika sedang melakukan aktivitas berat, jantung bisa berdetak lebih cepat. Sehingga untuk memeriksa aritmia pada seseorang, dibutuhkan waktu minimal 24 jam untuk merekam aktivitas jantung [6].

Pada saat ini dengan majunya teknologi banyak alat-alat dikembangkan untuk mempermudah pekerjaan di berbagai bidang terutama di bidang kesehatan, salah satu cara untuk mendiagnosis penyakit kardiovaskular yaitu dengan menganalisa sinyal ECG untuk melihat aktivitas kelistrikan yang dihasilkan oleh jantung. Oleh karena itu, analisis sinyal ECG sangat diperlukan dengan memanfaatkan teknologi *deep learning* untuk mengenali pola kompleks yang ada pada sinyal ECG sehingga dapat mendeteksi dan mendiagnosis aritmia yang kemudian diklasifikasikan berdasarkan jenis aritmianya.

Deep Learning merupakan salah satu cabang dari *machine learning* yang memiliki kemampuan untuk mengekstraksi fitur secara otomatis. Salah satu metode dari *deep learning* adalah *convolutional neural network* atau biasa disebut dengan CNN. CNN digunakan untuk memproses data yang memiliki bentuk dua dimensi atau berbentuk suatu citra. Metode QVAT adalah algoritma yang digunakan secara otomatis mendeteksi kompleks QRS, kemudian menemukan puncak R dari sinyal ECG. Algoritma yang digunakan terdiri dari beberapa langkah, yaitu: *band-pass filter*, *analysis of variance*, *adaptif threshold* dan *local maxima*. *Filter band-pass* digunakan untuk mengurangi *noise* yang dapat menyebabkan kesalahan dalam pendeteksian gelombang QRS. Kemungkinan *noise* terjadi dikarenakan *noise* tegangan gelombang elektromagnetik dan *noise* gerakan otot. Rata-rata varians digunakan untuk memperkuat fitur Kompleks QRS pada koordinat x positif atau ambang adaptif digunakan untuk melokalisasi kompleks QRS. Hasil ambang adaptif berupa *region of interest* (ROI) dari kompleks QRS yang digunakan untuk mencari posisi puncak R. Ambang adaptif digunakan karena fitur magnitudo dan kemiringan sinyal ECG setiap subjek berbeda [7].

Penggunaan metode *deep learning* dan QVAT dilukan secara

ra *hybrid*. Hasil deteksi *r peak* menggunakan metode QVAT akan digunakan sebagai titik anotasi pada pemotongan data dan transformasi menjadi spektogram. Hasil spektogram digunakan pada tahap *training* dan *testing* menggunakan *deep learning*.

Penyakit jantung dapat dideteksi lebih dini dengan cara mendeteksi kelainan irama detak jantung (aritmia) yang didapatkan dari sinyal ECG, sehingga dibutuhkannya sebuah algoritma komputer yang dapat mengklasifikasikan aritmia secara otomatis yang dapat diimplementasikan ke perangkat. Perangkat didesain secara *portable* menggunakan *single board computer* untuk melakukan klasifikasi aritmia. Hal tersebut diharapkan dapat membantu paramedis untuk mendiagnosis sedini mungkin aritmia pada pasien sehingga dapat menekan angka kematian karena penyakit jantung.

1.2 Perumusan Masalah

Berdasarkan data yang telah dipaparkan di latar belakang, meningkatnya angka penderita penyakit jantung dan kurang adanya penanganan yang berpotensi terjadi kematian yang seharusnya dapat dideteksi sejak dini dengan mengetahui jenis denyut jantung yang tidak normal atau aritmia yang diderita oleh pasien melalui sinyal ECG sehingga diperlukan perangkat berbasis *single board computer* yang dapat mengklasifikasikan aritmia secara otomatis menggunakan metode *hybrid* QVAT dan CNN.

1.3 Tujuan

Adapun tujuan dari penelitian tugas akhir ini adalah berupa model hasil *training* menggunakan metode CNN untuk mengklasifikasi denyut jantung yang tidak normal atau aritmia yang dapat diimplementasikan dengan baik pada perangkat berbasis *single board computer* dengan menggunakan metode *hybrid* QVAT dan CNN.

1.4 Batasan Masalah

Batasan masalah yang timbul dari permasalahan pada tugas akhir ini adalah:

1. Dataset menggunakan data yang diambil dari *MIT-BIH Arrhythmia Database* (<http://www.physionet.org>).
2. Kelas aritmia yang digunakan hanya kelas yang memiliki lima

data sampel terbanyak karena data yang digunakan hanya dari satu sumber.

3. Akuisisi data dan pemrosesan sinyal ECG menggunakan Raspberry Pi 3 Model B+, Intel Neural Compute Stick 2, Modul ADC ADS1115 dan sensor AD8232.
4. Data akuisisi untuk data *testing* diambil dari pasien yang belum dapat dipastikan memiliki penyakit aritmia dikarenakan tidak bisa mendapatkan izin untuk melakukan pengambilan data oleh pasien aritmia di rumah sakit.
5. Subjek pasien dalam kondisi relaks saat pengambilan data ECG.

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 Tinjauan Pustaka
Pada bab ini berisi tentang penjelasan mengenai *state of the art* dari penelitian terkait dan uraian secara sistematis teori-teori yang berhubungan dengan permasalahan yang dibahas pada penelitian ini. Teori-teori ini digunakan sebagai dasar dalam penelitian, yaitu Segmen Sinyal Jantung, Elektrokardiogram, Kompleks QRS, Aritmia, *Machine Learning*, *Deep Learning*, *Convolutional Neural Network*, *Classification Performance*, Metode QVAT, Raspberry Pi 3 Model B+, Sensor Sparkfun AD8232, Intel Neural Compute Stick 2 dan ADS1115.
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

men 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 Hasil dan Pengujian

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 pengembangan lebih lanjut juga dituliskan pada bab ini.

Halaman ini sengaja dikosongkan

BAB 2

TINJAUAN PUSTAKA

Demi mendukung penelitian ini, dibutuhkan beberapa *state of the art* dan teori penunjang sebagai bahan acuan. Dengan demikian penelitian ini menjadi lebih terarah.

2.1 *State of The Art*

Pada bagian bab ini merupakan beberapa penelitian terkait yang digunakan untuk menjadi acuan pada penelitian tugas akhir ini, dimana metode QVAT digunakan untuk mendeteksi *r peak* serta metode *deep learning* menggunakan *convolutional neural network* yang digunakan untuk melakukan klasifikasi aritmia yang dapat diimplementasikan menggunakan *single board computer*. Berikut adalah beberapa penelitian terkait yang digunakan pada tugas akhir ini, antara lain:

2.1.1 *IOT Based Detection of Cardiac Arrhythmias Using Raspberry PI*

Akash Chatlani, Shradha Kabu, R. D. Komati melakukan penelitian ini. Pada penelitian ini melakukan klasifikasi penyakit aritmia menjadi 3 kelas yaitu kelas normal (N), denyut cepat atau *tachycardia* (T), denyut lambat atau *bradycardia* (B) menggunakan data sinyal ECG dari *MIT-BIH ECG Signal Database*. Tahapan pada penelitian ini yaitu sinyal ECG dilakukan *Pre-Processing Feature Extraction Classification, Online Database Management Display Result*.

Secara garis besar, sinyal ECG akan dilakukan *filtering* untuk menghilangkan *noise* menggunakan bantuan *software MATLAB*, kemudian sinyal ECG akan diekstraksi fitur pola kompleks sinyal menggunakan *multi-resolution analysis nature of discrete wavelet transform* dan sinyal tersebut akan diklasifikasikan termasuk dalam kelas normal, *tachycardia*, *bradycardia*. Sinyal ECG dan hasil klasifikasi akan dikirim ke *database*, kemudian Raspberry PI akan membaca hasil tersebut secara serial menggunakan RS232 dan ditampilkan pada *web server*. [8]

2.1.2 Portable ECG Monitor with Low Power Consumption and Small Size Based on AD8232 Chip

Tancheng Lu, Peng Liu, Xiang Gao, dan Qiyong Lu melakukan penelitian ini, penelitian ini mendesain alat *portable* untuk mengamati sinyal ECG dan ditampilkan pada *mobile phone*. Sinyal ECG didapat dari sensor AD8232 yang tersambung dengan *microcontroller* yang kemudian akan dikirim menggunakan *bluetooth* ke *mobile phone*. [9]

2.1.3 Cardiac Arrhythmia Detection using Deep Learning

A. Isin dan S. Ozdalili melakukan penelitian terkait klasifikasi aritmia. Pada penelitian ini mengklasifikasikan tiga jenis aritmia, yaitu *Normal Beat*, *Right Bundle Branch Block*, dan *Paced Beats*. Metode pada penelitian terlebih dahulu mendeteksi QRS, kemudian mengekstrak fitur QRS dengan ANN, dan mengklasifikasikannya. Penelitian ini memiliki tingkat akurasi sebesar 92%. [10]

2.1.4 Arrhythmia Detection Using Deep Convolutional Neural Network with Long Duration

Özal yldırım, Pawel Plawiak, Ru San Tan& Acharya, U Rajendra melakukan penelitian mengenai klasifikasi aritmia dengan mengklasifikasikan 17 jenis aritmia menggunakan 1D-CNN tanpa mendeteksi QRS dan tanpa mengekstraksi fitur QRS tersebut. Penelitian ini memiliki tingkat akurasi sebesar 91.33%. [11]

2.1.5 A Deep Convolutional Neural Network Model to Classify Heartbeats

Pada tahun 2017, Acharya, Hagiwara , Ru San Tan, Adam dan Gertych melakukan penelitian mengklasifikasikan lima jenis aritmia berdasarkan pengkategorian oleh ANSI/AAMI EC57 standar tahun 2012. Penelitian ini menggunakan metode *convolutional neural network* dan memiliki tingkat akurasi sebesar 94.03%. [12]

2.1.6 Perbandingan Metode *SVM*, *FUZZY-KNN*, Dan *BDT-SVM* untuk Klasifikasi Detak Jantung Hasil Elektrokardiografi

Penelitian ini membandingkan metode SVM, Fuzzy-KKN, dan BDT-SVM untuk klasifikasi 3 jenis aritmia, yaitu *Atrial Fibrillation*, *PVC Bigeminy*, dan *Ventricular Tachycardia*. Penelitian ini memiliki tingkat akurasi sebesar 81.30%, 81.25%, dan 70% pada masing-masing metode. [13]

2.1.7 *QVAT: QRS Complex Detection based on Variance Analysis and Adaptive Threshold for Electrocardiogram Signal*

Penelitian ini menguji akurasi metode yang diusulkan menggunakan 20 catatan saluran MLII dari *Database Aritmia MIT-BIH*. Hasil percobaan eksperimen deteksi kompleks QRS memiliki sensitivitas Se 99,79%. QVAT memiliki beberapa langkah dalam mendeteksi kompleks QRS diantaranya *data acquisition*, *noise removal*, *variance*, *adaptive threshold*, *r peak detection*. [7]

2.1.8 Pendeteksi Penyakit Aritmia Berbasis Prediksi Fitur QRS pada Sinyal ECG dengan Metode *Feed Forward Backpropagation* menggunakan Sistem Tertanam

Pada penelitian oleh Izzatul Masrurroh ini membuat pendeteksi penyakit aritmia dengan metode *Machine Learning* menggunakan sistem tertanam. [14]

2.1.9 Klasifikasi Aritmia pada Sinyal ECG Berbasis *Deep Learning*

Pada penelitian oleh Firdaus Nanda Pradangapasti mengklasifikasikan 5 jenis aritmia menggunakan metode *Convolutional neural network* dengan tingkat akurasi 98,6%. [15]

2.2 Teori Penunjang

Pada bagian bab ini merupakan dasaran teori yang digunakan untuk referensi dalam penelitian diantaranya Segmen Sinyal Jan-

tung, Elektrokardiogram, Kompleks QRS, Aritmia, *Machine Learning, Deep Learning, Convolutional Neural Network, Classification Performance*, Metode QVAT, Raspberry Pi 3 Model B+, Sensor Sparkfun AD8232, Intel Neural Compute Stick 2 dan ADS1115.

2.2.1 Segmen Sinyal Jantung

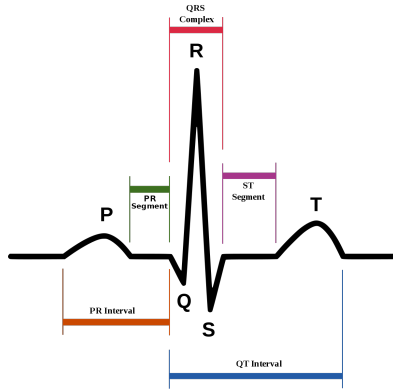
Segmen sinyal jantung dapat disadap dengan menggunakan elektroda dan digunakan untuk analisa bentuk maupun ritme jantung menggunakan elektrokardiograf. sinyal jantung yang terbentuk memiliki 5 segmen yaitu segment P, Q, R, S dan T. Gelombang sinyal P diawal, bersamaan dengan depolarisasi dari atrium. Atrium mulai berkontraksi sekitar 25 milidetik setelah gelombang P dimulai. Interval Kompleks QRS yang mana timbul akibat depolarisasi ventrikel. Sinyal elektrik relatif kuat karena otot ventrikel jauh lebih besar dari pada atrium. Ventrikel mulai berkontraksi tidak lama setelah mencapai puncak gelombang R. Gelombang T yang kecil diakhir, mengindikasi repolarisasi pada ventrikel atau disebut dengan segmen ST. Repolarisasi atrium tidak tampak, karena bersamaan saat ventrikel melakukan depolarisasi, dan QRS complex menutupi kejadian elektrik yang lain. [16]

2.2.2 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 2.1. Bentuk dan panjang gelombang-gelombang tersebutlah yang dianalisis untuk mendeteksi aritmia.

2.2.3 Kompleks QRS

Kombinasi dari tiga defleksi grafis yang terlihat pada elektrokardiogram dan merupakan bagian penelusuran yang paling sentral dan paling jelas secara visual karena lonjakan utama yang terlihat pada garis ECG. Segmen ini sesuai dengan depolarisasi ventrikel kanan dan kiri jantung manusia dan kontraksi otot ventrikel besar. Durasi, amplitudo dan morfologi kompleks QRS berguna untuk mendiagnosis aritmia jantung, abnormalitas konduksi, hipertro-



Gambar 2.1: Bentuk Sinyal ECG

fi ventrikel, infark otot jantung, gangguan elektrolit, dan keadaan sakit lainnya.

2.2.4 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. [17] Berikut 17 jenis klasifikasi dari aritmia:

- | | |
|--|---|
| 1. <i>Normal Beat</i> | 7. <i>Premature ventricular contraction</i> |
| 2. <i>Atrial premature beat</i> | 8. <i>Ventricular bigeminy</i> |
| 3. <i>Atrial flutter</i> | 9. <i>Ventricular trigeminy</i> |
| 4. <i>Atrial fibrillation</i> | 10. <i>Ventricular tachycardia</i> |
| 5. <i>Supraventricular tachyarrhythmia</i> | 11. <i>Idioventricular rhythm</i> |
| 6. <i>Pre-excitation (WPW)</i> | 12. <i>Ventricular flutter</i> |

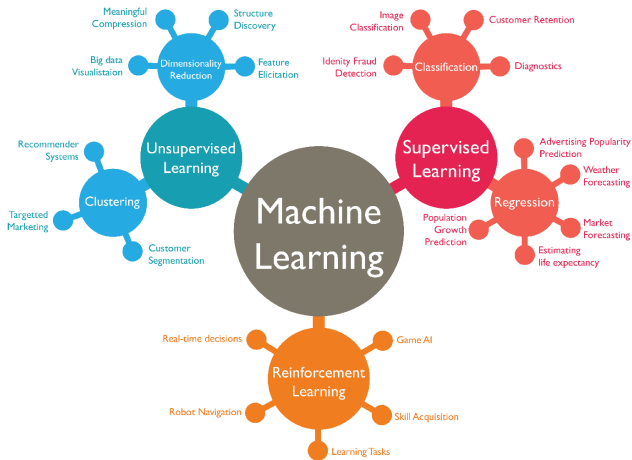
- | | |
|--|---|
| 13. <i>Fusion of ventricular and normal beat</i> | 15. <i>Right bundle branch block beat</i> |
| 14. <i>Left bundle branch block beat</i> | 16. <i>Second-degree heart block</i> |
| | 17. <i>Pacemaker rhythm</i> |

2.2.5 *Machine Learning*

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 2.2 merupakan diagram dari *machine learning*. *Machine learning* terbagi menjadi empat yaitu *supervised learning*, *unsupervised learning*, *semi supervised learning* dan *reinforcement 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*.

2.2.6 *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 *input* 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. Beberapa contoh dari *deep learning* adalah DNN (*Deep Neural Network*),



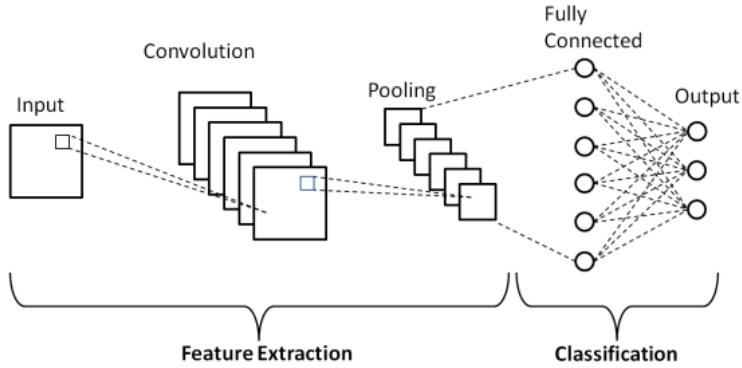
Gambar 2.2: Diagram *Machine Learning*

RNN (*Recurrent Neural Network*), dan CNN (*Convolutional Neural Network*). [18]

CNN merupakan *deep learning* yang paling sering digunakan untuk mengolah data dua dimensi. CNN memiliki beberapa *hidden layer* antara lain: *convolutional layer*, *pooling layer*, *fully connected layer*, dan *normalization layer*. Dibandingkan dengan *deep learning* lainnya, CNN memiliki hasil yang sangat bagus dalam *image processing* dan *speech recognition*. *Deep learning* menjadi semakin populer dan banyak digunakan untuk mengklasifikasi penyakit jantung dan aritmia. [19] Visualisasi dari arsitektur CNN ditampilkan pada Gambar 2.3.

2.2.7 *Convolutional Neural Network*

Convolutional neural network (CNN) adalah salah satu jenis neural network yang biasa digunakan pada data image atau mengolah data dua dimensi. *Convolutional neural network* memiliki beberapa *layer* antara lain:



Gambar 2.3: Visualisasi dari Arsitektur CNN

2.2.7.1 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 $7 \times 7 \times 3$ dengan *filter* berukuran $3 \times 3 \times 3$. 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*.

2.2.7.2 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 *in-*

put dari *convolutional layer* berikutnya, sehingga semakin banyak data yang terbuang. Dengan menggunakan *padding*, dimensi *output* 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. 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.

2.2.7.3 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.

2.2.7.4 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*.

2.2.8 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.[16]

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 2.4 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 2.4: *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*.

2.2.8.1 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.

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

2.2.8.2 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.

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

2.2.8.3 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 persamaan 2.3.

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

2.2.8.4 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.

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

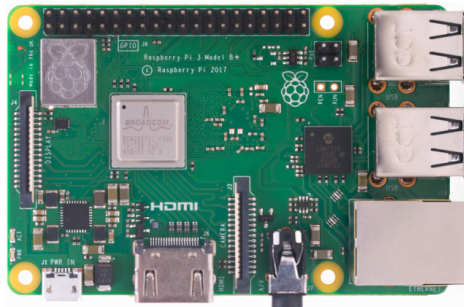
2.2.9 Metode QVAT

Salah satu parameter untuk menilai kondisi jantung adalah jarak dari puncak R ke puncak R lainnya, R adalah puncak gelombang kompleks QRS. Algoritma QVAT digunakan secara otomatis untuk mendeteksi kompleks QRS, kemudian menemukan puncak R dari sinyal ECG. Algoritma yang digunakan terdiri dari beberapa langkah yaitu: *band-pass filter*, *analysis of variance*, *adaptif threshold* dan *local maxima*.

Filter band-pass digunakan untuk mengurangi *noise* yang dapat menyebabkan kesalahan dalam pendeteksian gelombang QRS. Kemungkinan *noise* terjadi dikarenakan *noise* tegangan gelombang elektromagnetik dan *noise* gerakan otot. Rata-rata varians digunakan untuk memperkuat fitur Kompleks QRS pada koordinat x positif atau ambang adaptif digunakan untuk melokalisasi kompleks QRS. Hasil ambang adaptif berupa *region of interest* (ROI) dari kompleks QRS yang digunakan untuk mencari posisi puncak R. Ambang adaptif digunakan karena fitur magnitudo dan kemiringan sinyal ECG setiap subjek berbeda [7].

2.2.10 Raspberry Pi 3 Model B+

Raspberry Pi merupakan salah satu *single board computer* (SBC) atau komputer lengkap yang dibangun di atas papan sirkuit tunggal, dengan mikroprosesor, memori, *input / output* (I / O) dan fitur lain yang diperlukan dari komputer fungsional. SBC dibuat sebagai sistem demonstrasi atau pengembangan, untuk sistem pendidikan, atau untuk digunakan sebagai pengontrol komputer tertanam. Raspberry Pi 3 Model B + adalah produk terbaru dalam jajaran Raspberry Pi 3, menawarkan prosesor *quad core 64-bit* yang berjalan pada 1.4GHz, *dual-band* 2.4GHz dan 5GHz LAN nirkabel, *bluetooth* 4.2 / BLE, *ethernet* lebih cepat, dan PoE kemampuan melalui HAT PoE terpisah. Raspberry Pi 3 Model B + mempertahankan jejak mekanis yang sama seperti Raspberry Pi 2 Model B dan Raspberry Pi 3 Model B+ [20]. Gambar 2.5 menunjukkan Raspberry Pi 3 Model B+.



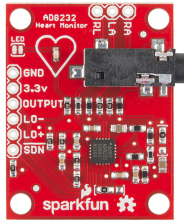
Gambar 2.5: Raspberry Pi 3 Model B+

2.2.11 Sensor Sparkfun AD8232

AD8232 adalah sebuah modul sensor yang digunakan untuk mengakuisisi sinyal ECG dengan *pin input / output* serta catu dayanya tersambung pada *microcontroller* ataupun *single board computer*. AD8232 adalah blok pengkondisi sinyal terintegrasi untuk ECG dan aplikasi pengukuran biopotensial lainnya. Ini dirancang untuk mengekstraksi, memperkuat, dan memfilter sinyal biopotensial kecil di hadapan kondisi bising, seperti yang diciptakan oleh gerakan atau penempatan elektroda jarak jauh. Desain ini memungkinkan *Ana-*

log to Digital Converter (ADC) daya *ultralow* atau mikrokontroler tertanam untuk memperoleh sinyal output dengan mudah.

AD8232 dapat menerapkan *filter high-pass* dua tiang untuk menghilangkan artefak gerak dan potensial setengah sel elektroda. *Filter* ini dipasangkan dengan erat dengan arsitektur instrumentasi dari penguat untuk memungkinkan dilakukan *filter* besar dan tinggi dalam satu tahap, sehingga menghemat ruang dan biaya. Penguat operasional tanpa komitmen memungkinkan AD8232 untuk membuat *filter low-pass* tiga kutub untuk menghilangkan kebisingan tambahan. Pengguna dapat memilih *cutoff* frekuensi semua *filter* untuk disesuaikan dengan berbagai jenis aplikasi. Untuk meningkatkan penolakan dari frekuensi saluran dalam sistem dan gangguan lain yang tidak diinginkan, AD8232 menyertakan penguat untuk aplikasi *lead* yang digerakkan, seperti *right leg drive* (RLD). [21].



Gambar 2.6: Sparkfun AD8232

2.2.12 Intel Neural Compute Stick 2

Intel Movidius Neural Compute Stick (NCS) adalah perangkat pembelajaran mendalam tanpa kipas kecil yang dapat digunakan untuk mempelajari pemrograman *artificial intelligence* di bagian ujung. NCS ditenagai oleh Unit Pemrosesan Intel Movidius Vision dengan daya rendah dan kinerja tinggi yang dapat ditemukan di jutaan kamera keamanan pintar, *drone* yang dikendalikan gerakan, peralatan visi mesin industri, dan banyak lagi. Kerangka kerja yang didukung adalah *TensorFlow*, *Caffe*, *Apache MXNet*, *Open Neural Network Exchange* (ONNX), *PyTorch*, dan *PaddlePaddle* melalui sebuah konversi ONNX. Dapat dilihat melalui Gambar 2.7 bahwa konektivitas nya menggunakan USB 3.0 *Type-A*, dan ukuran dimen-

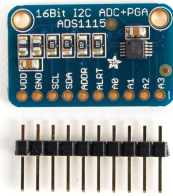
si 72.5 mm x 27 mm x 14 mm.



Gambar 2.7: Intel Neural Compute Stick 2

2.2.13 ADS1115

ADS1115 ditunjukkan pada Gambar 2.8) adalah konverter analog ke digital presisi dengan resolusi *16 bit* yang ditawarkan dalam paket QFN-10 sangat kecil (2mm x 1,5mm x 0,4mm), tanpa timbal atau paket MSOP-10. ADS1115 dirancang dengan presisi, kekuatan, dan kemudahan implementasi. ADS1115 menampilkan referensi *onboard* dan osilator. Data ditransfer melalui antarmuka serial yang kompatibel dengan I2C (empat alamat I2C dapat dipilih). ADS1115 beroperasi dari catu daya tunggal mulai dari 2.0V hingga 5.5V. ADS1115 dapat melakukan konversi dengan laju hingga 860 sampel per detik. PGA *onboard* tersedia pada ADS1115 yang menawarkan rentang *input* dari pasokan hingga serendah $\pm 256\text{mV}$, yang memungkinkan sinyal besar dan kecil diukur dengan resolusi tinggi. ADS1115 juga memiliki fitur *multiplexer input* (MUX) yang menyediakan dua *input* diferensial atau empat ujung tunggal. ADS1115 beroperasi baik dalam mode konversi kontinu atau sekali pakai yang secara otomatis mati setelah konversi dan sangat mengurangi konsumsi saat ini selama periode *idle*. Suhu pengoperasian ADS1115 dalam rentang nilai $-40\text{ }^{\circ}\text{C}$ hingga $+125\text{ }^{\circ}\text{C}$ [22].



Gambar 2.8: ADS1115

BAB 3

DESAIN DAN IMPLEMENTASI SISTEM

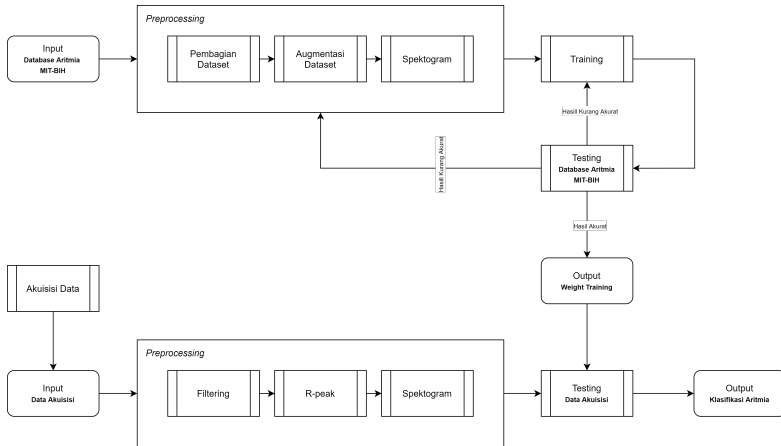
Penelitian ini dilaksanakan sesuai dengan desain sistem berikut dengan implementasinya. Desain sistem merupakan konsep dari pembuatan dan perancangan infrastruktur kemudian diwujudkan dalam bentuk blok-blok alur yang harus dikerjakan. Pada bagian implementasi merupakan pelaksanaan teknis untuk setiap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

3.1 Desain Sistem

Pada sistem yang dibangun akan melalui beberapa tahapan agar mendapatkan hasil prediksi klasifikasi aritmia seperti yang ditunjukkan pada Gambar 3.1 sebagai blok diagram kerja sistem yang akan digunakan pada penelitian. *Input* didapatkan dari MIT-BIH *Arrhythmia Database* (<https://physionet.org>) yang digunakan sebagai data pada proses *training* dan hasil akuisisi data yang digunakan pada proses *testing*. 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)

Data ECG dari MIT-BIH *Arrhythmia Database* yang digunakan sebagai data *training* merupakan 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. Sinyal ditransformasikan menjadi dua dimensi berupa spektrogram. Pada proses *training*, *dataset* tersebut dibagi menjadi tiga bagian yaitu data untuk *training* sebanyak 80%, data untuk *validation* sebanyak 10%, dan data untuk *testing* sebanyak 10%. Data tersebut dilakukan proses *training* menggunakan CNN yang akan menghasilkan model klasifikasi aritmia.



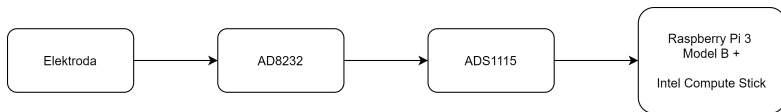
Gambar 3.1: Blok Diagram Kerja Sistem

Model klasifikasi aritmia akan digunakan untuk proses *testing* menggunakan data ECG dari hasil akuisisi data ECG menggunakan perangkat yang dibangun. Pada tahap akuisisi data dilakukan *filtering* dan konversi nilai sensor. Untuk melakukan pemotongan sinyal hasil akuisisi data, perlu dilakukan deteksi kompleks QRS dengan menggunakan metode QVAT agar didapatkan *r peak* pada sinyal tersebut. Kemudian *r peak* digunakan sebagai titik anotasi untuk pemotongan data dan ditransformasikan ke bentuk spektrogram. Hasil spektrogram dari data akuisisi digunakan untuk proses *testing* menggunakan model yang sudah dibangun pada proses *training*. Hasil *output* berupa visualisasi hasil prediksi klasifikasi aritmia menggunakan *user interface*.

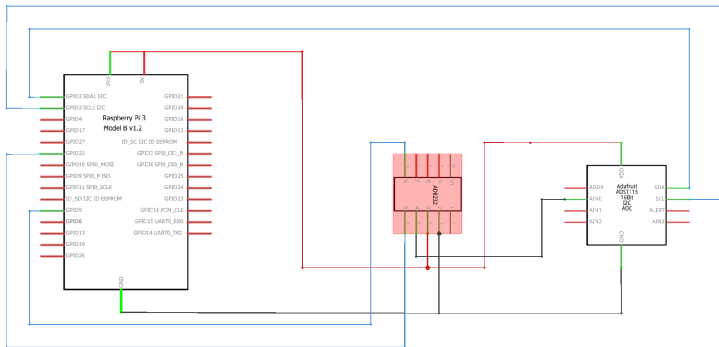
3.2 Desain dan Perancangan Perangkat ECG

Pada Gambar 3.3 merupakan desain dari perangkat ECG terdiri dari modul sparkfun AD8232, modul ADS1115, Raspberry Pi 3 Model B+ dan Intel Compute Stick yang dikonfigurasi sesuai dengan Tabel 3.1. Komponen ADS1115 berfungsi untuk mengkonversi dari nilai analog yang didapat oleh modul AD8232 menjadi nilai digital yang akan diteruskan ke *pin* Raspberry. Pada Gambar 3.2 menunjukkan diagram alur dari perangkat pada setiap komponen.

Hal ini dilakukan karena raspberry tidak memiliki *pin* analog. Intel neural compute stick berfungsi untuk mempercepat tahap komputasi pada saat *testing* yang dilakukan karena terbatasnya kemampuan raspberry dalam komputasi. Catu daya yang digunakan pada perangkat menggunakan *battery lithium polymer* 2S dengan menggunakan regulator UBEC untuk menurunkan tegangan menjadi 5V dengan arus maksimal 5A. Gambar 3.4 merupakan desain *nesting* kotak yang akan dicetak menggunakan akrilik menyerupai *box* agar mudah dibawa yang dilengkapi LCD 7 inch, tombol *on/off*, terminal USB *port* dan *hole* untuk memasang elektroda.



Gambar 3.2: Diagram Perangkat ECG



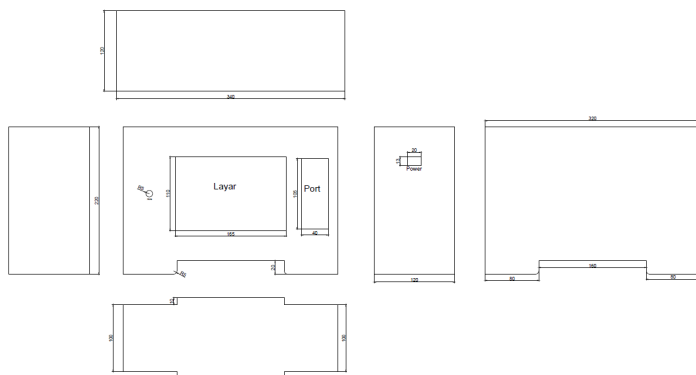
Gambar 3.3: Desain Perangkat ECG

3.3 Pengambilan Data ECG sebagai *Input*

Data yang digunakan sebagai *input* pada proses pengerjaan tugas akhir ini diambil dari MIT-BIH *Arrhythmia Database* untuk proses *training* dan hasil akuisisi data ECG menggunakan sensor AD8232 dan *single board computer* untuk proses *testing*. Pada data

Tabel 3.1: Konfigurasi Perangkat ECG

Raspberry	AD8232	ADS1115
GPIO5	LO+	-
GPIO22	LO-	-
3,3v	3,3v	VDD
GND	GND	GND
GND	-	ALRT
SDA	-	SDA
SCL	-	SCL
-	OUTPUT	A0



Gambar 3.4: Desain *Nesting Box*

dari MIT-BIH *Arrhythmia Database* 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). Tabel 3.2 menunjukkan jumlah sampel dari setiap jenis aritmia pada *dataset* tersebut.

Pada *dataset* MIT-BIH *Arrhythmia Database* perlu dilakukan *filter* agar sinyal menjadi lebih baik. *Filter* dilakukan menggunakan algoritma dari *package biospy*. *Filtering* dilakukan menggunakan tipe *Finite Impulse Response* adalah *filter* yang respons impulsnya

Tabel 3.2: Jumlah Sampel ECG pada Setiap Kelas Aritmia

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

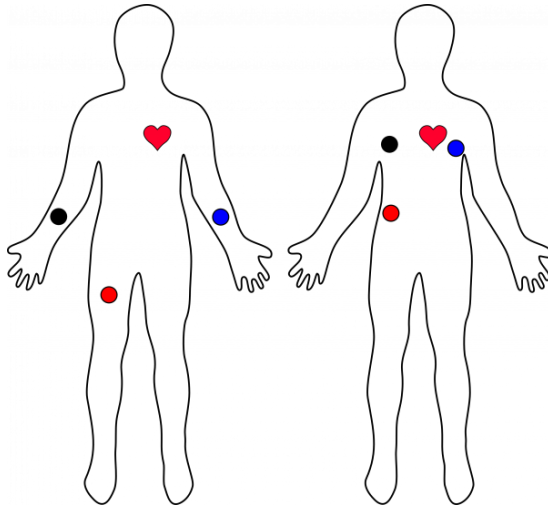
atau responsnya terhadap input panjang terhingga memiliki durasi terhingga, karena *filter* tersebut menjadi nilai nol dalam waktu terhingga. *Band-pass filter* adalah *filter* frekuensi yang melewatkan sinyal frekuensi dalam rentang frekuensi tertentu yaitu melewatkan sinyal yang berada diantara frekuensi batas bawah hingga frekuensi batas atasnya.

Untuk penempatan elektroda ECG tidak bisa pada semua bagian badan, penempatan elektroda ECG memiliki standar diantaranya seperti yang ditunjukkan pada Gambar 3.5. Pada gambar tersebut terdapat 2 pilihan untuk penempatan elektroda ECG.

Konversi nilai sensor perlu dilakukan dengan mengubah *range* nilai data yang didapatkan dari tahap akuisisi data menjadi *range* mulai dari -0.5 sampai dengan 0.5 sesuai dengan Persamaan 3.1 agar mendapatkan konstanta sehingga dapat dijadikan data *input* saat proses deteksi *r peak*.

$$y[i] = \frac{x[i] - \max(x)}{|\min(x) - \max(x)|} + 0.5 \quad (3.1)$$

Data dari hasil akuisisi menggunakan sensor AD8232 dan *single board computer* digunakan untuk proses *testing* dengan model *weight* yang telah dibangun. Sensor AD8232 menggunakan 3 *pin* pada *single board computer* yaitu 1 *pin* analog untuk membaca data sinyal ECG pada pengguna dan 2 *pin* digital untuk memastikan elektroda sudah tertempel pada titik tubuh pengguna. Hasil akuisisi data tersebut akan dilakukan deteksi kompleks QRS agar didapatkan *r peak* menggunakan metode QVAT. Titik *r peak* tersebut digunakan sebagai titik tengah pada proses pemotongan sinyal.



Gambar 3.5: Penempatan Posisi Elektroda ECG

Hasil pemotongan diubah menjadi spektrogram yang akan dilakukan proses *testing* menggunakan model yang telah dibangun pada *Single Board Computer*.

3.4 *Preprocessing* - Pembagian *Dataset*

Dataset dari MIT-BIH *Arrhythmia Database* dibagi menjadi tiga *subset*. Komposisi pembagian *dataset* tersebut adalah 80% *train set*, 10% *validation set*, and 10% *test set*. Pada setiap kelas perlu dilakukan penyetaraan jumlah data agar tidak terjadi *overfitting* pada kelas tertentu.

1. *Training Set*

Training set adalah bagian dari *dataset* yang digunakan pada proses *training* untuk melatih model dari sebuah algoritma pada *Machine Learning*.

2. *Validation Set*

Validation set adalah bagian dari *dataset* yang digunakan untuk melakukan validasi dan mencegah *overfitting*.

3. *Testing Set*

Testing set adalah bagian dari *dataset* yang digunakan pada proses *testing* untuk menguji sebuah model yang dibangun.



Gambar 3.6: Visualisasi Pembagian *Dataset*

Sebelum sinyal ECG diproses menggunakan CNN, sinyal tersebut dipotong-potong selama satu detik (360 data). Pemotongan sinyal dilakukan dengan puncak R sebagai titik tengah. Kelas FVN merupakan kelas yang memiliki jumlah data yang paling sedikit yaitu 802 data, maka jumlah tersebut digunakan sebagai acuan dalam penyetaraan jumlah data. Tabel 3.3 menunjukkan pembagian sampel ECG untuk *dataset* pada model pertama.

Tabel 3.3: Pembagian Sampel ECG

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

3.5 *Preprocessing* - *Augmentasi Dataset*

Augmentasi data meningkatkan variasi data dengan memanipulasi transformasi dari data tersebut. Augmentasi dilakukan pada kelas FVN agar kelas tersebut memiliki variasi data lebih banyak. Metode augmentasi yang digunakan yaitu dengan menggeser titik tengah ke kiri dan ke kanan pada setiap satu data masing-masing sebanyak lima kali sehingga jumlah data kelas FVN menjadi 8020

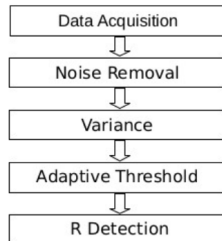
data. Setelah dilakukan augmentasi, kelas PVC memiliki jumlah data paling sedikit yaitu 7129 data, maka dijadikan sebagai acuan untuk penyetaraan jumlah data pada setiap kelas seperti yang ditunjukkan pada Tabel 3.4.

Tabel 3.4: Pembagian Sampel ECG Setelah Augmentasi Data

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

3.6 *Preprocessing* - Deteksi *R Peak* (Puncak R)

Metode QVAT digunakan untuk melakukan segmentasi bagian kompleks QRS pada sinyal ECG. Hasil segmentasi kompleks QRS akan dapat dijadikan *input* untuk tahap klasifikasi jenis aritmia menggunakan metode CNN (*Convolutional Neural Network*). Metode ini memiliki beberapa langkah dalam mendeteksi kompleks QRS diantaranya *data acquisition*, *noise removal*, *variance*, *adaptive threshold*, *r peak detection* seperti yang ditunjukkan pada Gambar 3.7. [7]



Gambar 3.7: Blok Diagram Metode QVAT

3.7 Preprocessing - Pemotongan Sinyal

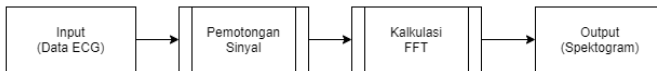
Pada *dataset* dari MIT-BIH *Arrhythmia Database* memiliki frekuensi sebesar 360 Hz. Data tersebut akan digunakan untuk proses *training* menggunakan CNN sehingga perlu dilakukan pemotongan sinyal sepanjang 1 detik (360 data). Pemotongan tersebut dilakukan dengan acuan *r peak* (puncak R) sebagai titik tengah. Pada *dataset* tersebut sudah diketahui jenis penyakit aritmia sebagai label pada titik anotasi. Data ECG kemudian dipotong sepanjang 360 data.

Frekuensi pada hasil akuisisi data ECG yang akan digunakan sebagai data *testing* perlu disesuaikan dengan frekuensi *dataset* dari MIT-BIH *Arrhythmia Database*. *Sample rate* yang didapatkan pada proses akuisisi data yaitu 300 Hz maka perlu dilakukan konversi frekuensi sinyal menjadi 360 Hz dengan melakukan *resample* menggunakan *package samplerate*. Pada *package* tersebut konversi dilakukan menggunakan fungsi *sinc interpolation*. Persamaan 3.2 menunjukkan fungsi dari *sinc interpolation*.

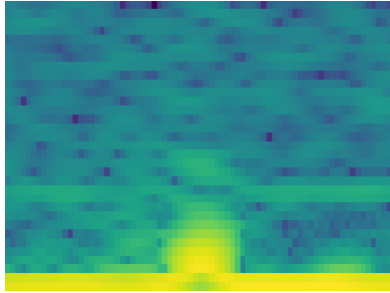
$$x(t) = \sum_{n=-\infty}^{\infty} x[n] \text{sinc} \left(\frac{t - nT}{T} \right) \quad (3.2)$$

3.8 Preprocessing - Spektrogram

Spektrogram merupakan representasi visual dari spektrum frekuensi sinyal terhadap waktu. Gambar 3.8 adalah blok diagram proses spektrogram. Mengubah bentuk sinyal menjadi spektrogram yaitu dengan memotong sampel ECG menjadi lebih kecil (*window*) dan menghitung FFT untuk setiap *window* yang menghasilkan domain frekuensi pada setiap *window* dan nomor *window* merepresentasikan waktu. Untuk menghindari kehilangan beberapa frekuensi, maka setiap *window* dilakukan proses *overlapping* atau tumpang tindih. Proses pembuatan spektrogram memiliki panjang *window* sebesar 64 data dan *overlap* sebesar 60 data.



Gambar 3.8: Blok Diagram Transformasi Spektrogram

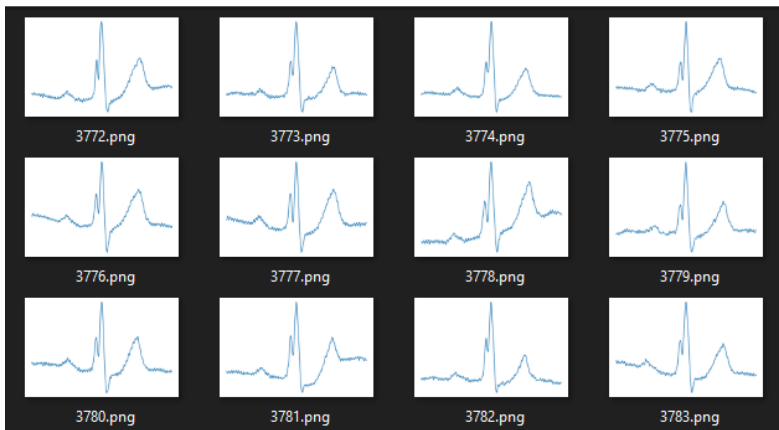
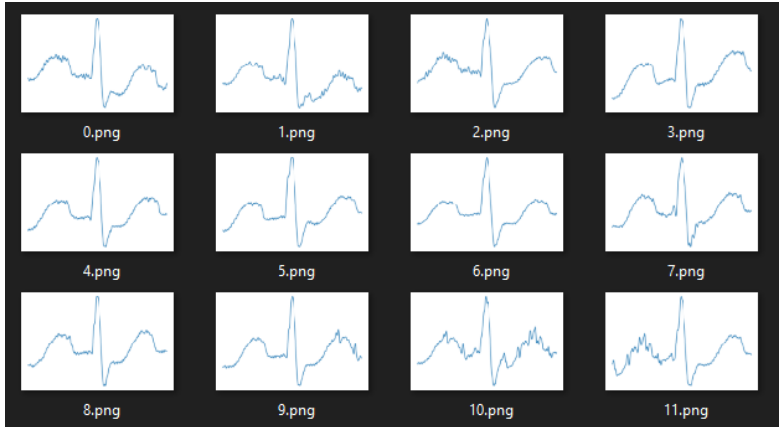


Gambar 3.9: Sampel Hasil Transformasi Sinyal Berupa Spektrogram *Window* 64 dan *Overlap* 60

3.9 *Preprocessing* - Pembagian Variasi *Dataset*

Pada penelitian yang dilakukan oleh Firdaus Nanda Pradangapasti menggunakan *dataset* dari MIT-BIH *Arrhythmia Database*. Komposisi pembagian *dataset* tersebut adalah 80% *train set*, 10% *validation set*, and 10% *test set*. Dapat dilihat pada Gambar 3.10 menunjukkan *dataset* dari hasil *plot* sinyal kelas LBBB. Dari hasil tersebut dapat dilihat jika terdapat variasi bentuk sinyal pada kelas yang sama namun memiliki bentuk yang variatif.

Metode pembagian *dataset* berupa spektrogram yang dilakukan oleh Firdaus Nanda Pradangapasti dengan berurutan seperti ditunjukkan pada Gambar 3.11a dan Tabel 3.5. Maka dari itu perlu dilakukan variasi pada setiap *subset training*, *validation* dan *testing* dengan cara menggunakan fungsi modulus untuk melakukan pembagian data agar lebih bervariasi pada setiap *subset*. Tabel 3.6 menunjukkan pembagian *dataset* setelah dilakukan variasi dan 3.11 menunjukkan hasil pembagian *dataset*. Pada Gambar 3.11b menunjukkan hasil pembagian *dataset* berupa spektrogram setelah dilakukan variasi.



Gambar 3.10: *Plot Sinyal Pada Kelas LBBB*

3.10 *Preprocessing - Filtering Dataset*

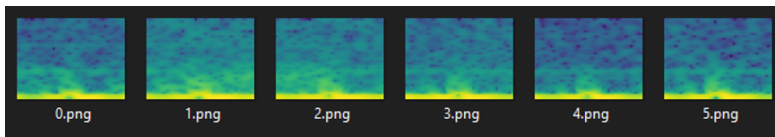
Pada Gambar 3.12 menunjukkan *plot* sinyal dari kelas normal dan Gambar 3.13 menunjukkan *plot* sinyal dari kelas LBBB, dimana dapat dilihat terjadi *baseline wonder* yaitu apabila sinyal ECG terjadi naik turun dan tidak lurus pada sumbu x. *Filtering* digunakan untuk menghilangkan *baseline wonder* dan *noise*. *Packa-*

Tabel 3.5: Pembagian *Dataset* Tanpa Menggunakan Variasi

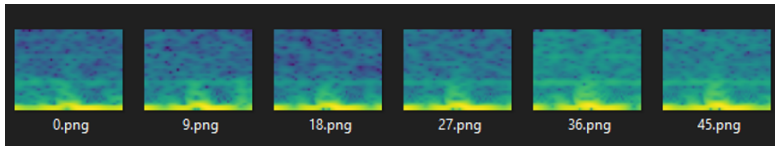
<i>Subset</i>	<i>Data</i>	<i>Jumlah Data</i>
<i>Testing Set</i>	0, 1, 2, ..., 712	713
<i>Validation Set</i>	713, 715, 716, ..., 1426	713
<i>Training Set</i>	1427, 1428, 1429, ..., 7129	5703

Tabel 3.6: Pembagian *Dataset* Menggunakan Variasi

<i>Subset</i>	<i>Data</i>	<i>Jumlah Data</i>
<i>Testing Set</i>	0, 9, 18, ..., 6408	713
<i>Validation Set</i>	1, 10, 19, ..., 6409	713
<i>Training Set</i>	2, 3, 4, ..., 6407 (sisa data)	5703



(a) Tanpa Menggunakan Variasi

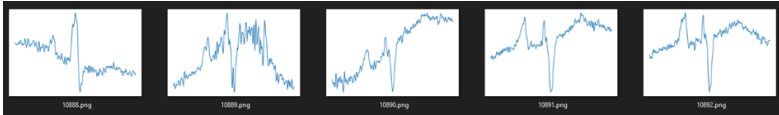


(b) Menggunakan Variasi

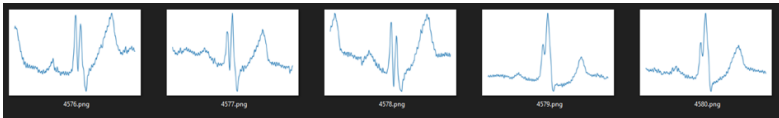
Gambar 3.11: Visualisasi Pembagian *Dataset*

ge dari *biospsy* digunakan untuk melakukan proses *filtering* dengan menggunakan *band-pass filter* dan *finite impulse response*. *Filter band-pass* digunakan untuk mengurangi *noise* yang dapat menyebabkan kesalahan dalam pendeteksian gelombang QRS. *Band-pass filter* adalah *filter* frekuensi yang melewati sinyal frekuensi dalam rentang frekuensi tertentu yaitu melewati sinyal yang berada diantara frekuensi batas bawah hingga frekuensi batas atasnya. *Finite impulse response* adalah *filter* yang respons impulsnya (atau

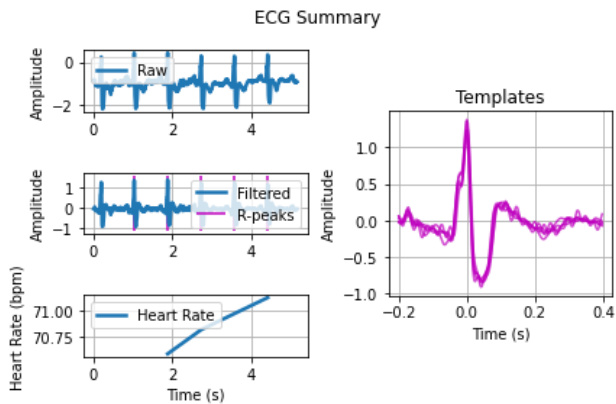
responsnya terhadap input panjang terhingga) memiliki durasi terhingga, karena filter tersebut menjadi nol dalam waktu terhingga. Pada Gambar 3.14 merupakan hasil proses *filtering* pada *record-118*.



Gambar 3.12: *Plot Sinyal Pada Kelas Normal*



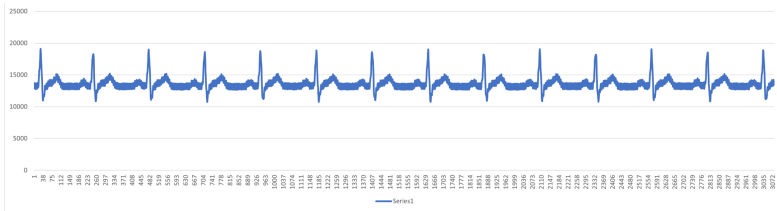
Gambar 3.13: *Plot Sinyal Pada Kelas LBBB*



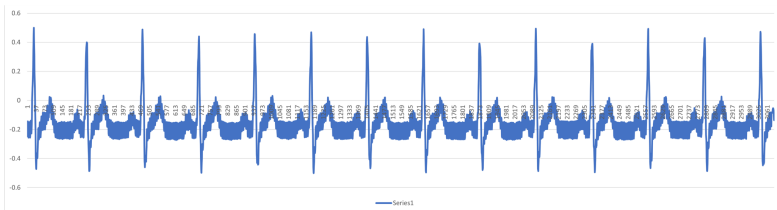
Gambar 3.14: Hasil Proses *Filtering* Pada *Record-118*

3.11 *Preprocessing* - Penambahan *Dataset*

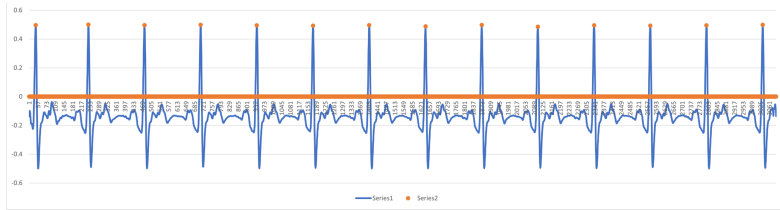
Pengujian *testing* model yang telah dibangun menggunakan *dataset* dan data dari hasil akuisisi data. Proses penambahan *dataset* diambil dari hasil akuisisi data menggunakan simulator yang telah dikonfigurasi diagnosis normal dengan *heart rate* sebesar 60 bpm. Data akuisisi yang telah diambil dilakukan *preprocessing* agar hasilnya bisa sesuai dengan *dataset* MIT-BIH *Arrhythmia Database*. Pada Gambar 3.15 menunjukkan hasil dari proses akuisisi data simulator, Gambar 3.16 menunjukkan hasil dari proses konversi *range* nilai dari hasil akuisisi data dan Gambar 3.17 menunjukkan hasil deteksi *r peak*. Setelah dilakukan pemotongan data sepanjang 300 data dan dilakukan konversi frekuensi seperti yang dilakukan pada tahap pemotongan sinyal hasil akuisisi data. Penambahan *dataset* dengan menimpa *dataset* pada kelas normal dengan dataset baru. Pembagian dataset juga dilakukan dengan variatif ke setiap *subset*.



Gambar 3.15: *Plot* Hasil Akuisisi Data Simulator



Gambar 3.16: *Plot* Hasil Proses Konversi *Range*



Gambar 3.17: *Plot Hasil Deteksi R Peak*

3.12 Arsitektur CNN

Model *convolutional neural network* yang digunakan memiliki 24 lapis digunakan untuk mengklasifikasikan 5 kelas aritmia. Tabel 3.7 merupakan spesifikasi arsitektur CNN yang digunakan.

3.13 Proses *Training*

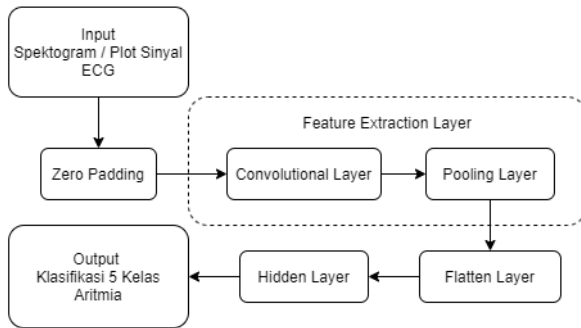
CNN merupakan salah satu dari jenis *neural network* yang biasa digunakan untuk mengolah data dua dimensi. CNN biasa digunakan untuk mendeteksi dan mengenali suatu objek pada sebuah gambar. Secara garis besar, CNN hampir sama dengan *neural network* biasanya. Hal yang membedakannya terletak pada arsitekturnya. 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 3.18 merupakan blok diagram alur kerja CNN pada tugas akhir ini. Sebelum melakukan proses *training*, perlu dilakukan penentuan ukuran *batch size*, iteration dan *epoch*.

1. *Batch size* mendefinisikan banyaknya sampel yang akan dibarkan 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*.

Tabel 3.7: Arsitektur CNN yang digunakan

<i>Layer (type)</i>	<i>Output Shape</i>	<i>Parameter</i>
<i>InputLayer</i>	[(None, 258, 347, 3)]	0
<i>ZeroPadding2D</i>	(None, 262, 351, 3)	0
<i>Conv2D</i>	(None, 258, 347, 16)	1216
<i>Batch Normalization</i>	(None, 258, 347, 16)	64
<i>MaxPooling2D</i>	(None, 129, 173, 16)	0
<i>Conv2D</i>	(None, 127, 171, 32)	4640
<i>Batch Normalization</i>	(None, 127, 171, 32)	128
<i>MaxPooling2D</i>	(None, 63, 85, 32)	0
<i>Conv2D</i>	(None, 61, 83, 32)	9248
<i>Conv2D</i>	(None, 59, 81, 32)	9248
<i>Batch Normalization</i>	(None, 59, 81, 32)	128
<i>MaxPooling2D</i>	(None, 29, 40, 32)	0
<i>Conv2D</i>	(None, 27, 38, 32)	9248
<i>Conv2D</i>	(None, 25, 36, 32)	9248
<i>Batch Normalization</i>	(None, 25, 36, 32)	128
<i>MaxPooling2D</i>	(None, 12, 18, 32)	0
<i>Conv2D</i>	(None, 10, 16, 64)	18496
<i>Conv2D</i>	(None, 8, 14, 64)	36928
<i>Batch Normalization</i>	(None, 8, 14, 64)	256
<i>MaxPooling2D</i>	(None, 4, 7, 64)	0
<i>Flatten</i>	(None, 1792)	0
<i>Dense</i>	(None, 1024)	1836032
<i>Dense</i>	(None, 2048)	2099200
<i>Dense</i>	(None, 5)	10245

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.



Gambar 3.18: Blok Diagram Alur Kerja CNN

3.14 Proses *Testing* Menggunakan *Dataset*

Setelah proses *validation* dilakukan, langkah selanjutnya adalah proses *testing*. Proses ini merupakan sebuah proses untuk memprediksi hasil klasifikasi melalui model *weight* yang telah dibangun menggunakan arsitektur CNN. Hasil dapat divisualisasikan dalam bentuk *confusion matrix*.

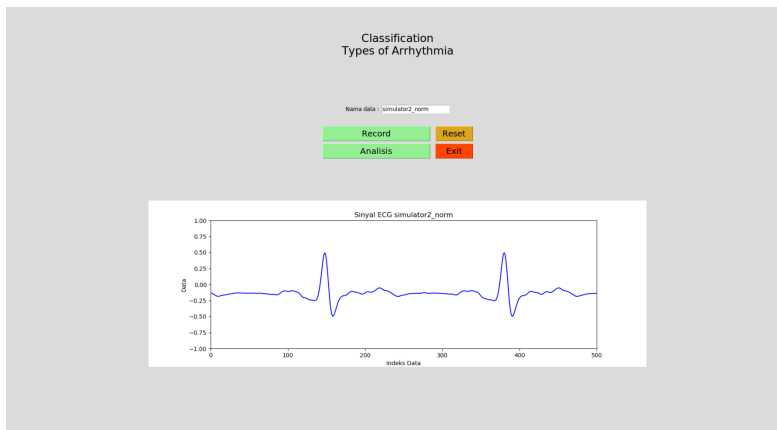
3.15 Proses *Testing* Menggunakan Data Akuisisi

Sinyal ECG didapatkan dari hasil akuisisi akan melalui tahap *preprocessing* dengan mendeteksi *r peak*. Frekuensi pada hasil akuisisi data ECG yang akan digunakan sebagai data *testing* perlu disesuaikan dengan frekuensi *dataset* dari MIT-BIH *Arrhythmia Database*. Namun maksimal frekuensi yang didapatkan pada proses akuisisi data yaitu 300 Hz maka perlu dilakukan konversi frekuensi menjadi 360 Hz. *R peak* akan digunakan sebagai titik anotasi pada pemotongan data sepanjang 360 data. Hasil tersebut ditransformasikan menjadi spektrogram agar dapat dilakukan *testing* menggunakan model *weight* yang telah dibangun pada proses *training*.

3.16 Visualisasi Hasil Menggunakan *User Interface*

Output dari proses tersebut adalah visualisasi hasil prediksi klasifikasi penyakit aritmia menggunakan *user interface*. *User inter-*

face digunakan untuk menampilkan akuisisi data ECG dan prediksi hasil klasifikasi jenis artimia. Pada Gambar 3.19 menunjukkan desain *user interface*. Terdapat tombol *record* untuk melakukan akuisisi data ECG dan tombol analisis untuk melakukan analisa dari hasil akuisisi data. Terdapat grafik visual dari hasil akuisisi data.



Gambar 3.19: Desain *User Interface*

BAB 4

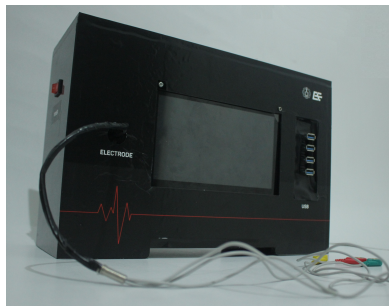
HASIL DAN PENGUJIAN

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. Terdapat 8 implementasi dan pengujian yang dilakukan, antara lain :

1. Implementasi *Hardware*
2. Pengujian Hasil *Training* dan *Validation* Model
3. Pengujian Hasil *Testing* Model
4. Pengujian Deteksi *R Peak*
5. Pengujian Sistem Menggunakan *Dataset*
6. Pengujian Sistem Menggunakan Data Akuisisi
7. Pengujian *Execution Time* Analisis
8. Pengujian Sistem Menggunakan *User Interface*

4.1 Implementasi *Hardware*

Perangkat yang ditampilkan pada Gambar 4.2 dilakukan *packaging* menggunakan akrilik dengan tebal 10 mm. Ukuran dari perangkat yaitu 34 x 10 x 10 cm. Pada Gambar 4.2a adalah perangkat bagian depan, Gambar 4.2b adalah perangkat bagian samping, Gambar 4.2c adalah perangkat bagian belakang dan Gambar 4.2d adalah perangkat bagian dalam.



Gambar 4.1: Foto Perangkat



(a) Perangkat Bagian Depan



(b) Perangkat Bagian Samping



(c) Perangkat Bagian Belakang



(d) Perangkat Bagian Dalam

Gambar 4.2: Foto Perangkat Setiap Sudut

4.2 Pengujian Hasil *Training* dan *Validation* Model

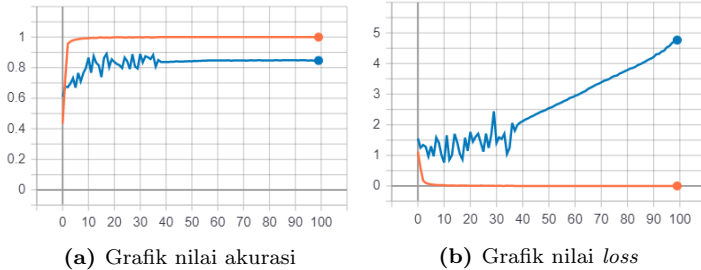
Proses *training* dan *validation* pada setiap pengujian dilakukan dengan jumlah *epoch* sebanyak 100 dan *batch size* sebesar 80. Terdapat tiga pengujian pada tahap *training* dan *validation*, yaitu:

1. Pengujian menggunakan variasi pembagian *dataset*.
2. Pengujian menggunakan *filter dataset*.
3. Pengujian menggunakan penambahan *dataset*.

4.2.1 Hasil *Training* dan *Validation* Menggunakan Variasi Pembagian *Dataset*

Pada penelitian yang telah dilakukan oleh Firdaus Nanda Pradanggapasti menggunakan *training* dengan menggunakan *dataset* yaitu dengan jumlah sampel *training* sebanyak 5703, sampel *validation* sebanyak 713, dan sampel *testing* sebanyak 713. Pada pembagian *dataset* tidak dilakukan variasi pada setiap *subset training*, *validation* dan *testing*. Didapatkan hasil dengan tingkat akurasi *tra-*

ining sebesar 100% dan tingkat akurasi *validation* sebesar 87.87% seperti yang ditunjukkan pada gambar 4.3a dan memiliki *loss* pada *training* sebesar $4.1289e-12$ atau mendekati 0, sedangkan *loss* pada *validation* sebesar 2.451. Gambar 4.3b merupakan grafik nilai *loss* pada proses *training* dan *validation*.

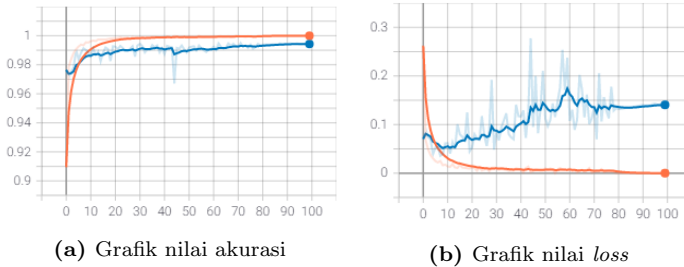


Gambar 4.3: Grafik Hasil *Training* dan *Validation* Pada Penelitian Oleh Firdaus Nanda Pradanggapasti

Pada pengujian ini dilakukan *training* dengan menggunakan *dataset* yaitu dengan jumlah sampel *training* sebanyak 5703, sampel *validation* sebanyak 713, dan sampel *testing* sebanyak 713 dengan melakukan pembagian *dataset* menggunakan variasi pada setiap *subset training*, *validation* dan *testing* dengan cara menggunakan fungsi modulus untuk melakukan pembagian data agar lebih bervariasi pada setiap *subset*. Didapatkan hasil dengan tingkat akurasi *training* sebesar 100% dan tingkat akurasi *validation* sebesar 99.17% seperti yang ditunjukkan pada gambar 4.4a dan memiliki *loss* pada *training* sebesar $1.049e-12$ atau mendekati 0, sedangkan *loss* pada *validation* sebesar 0.1077. Gambar 4.4b merupakan grafik nilai *loss* pada proses *training* dan *validation*.

4.2.2 Hasil *Training* dan *Validation* Menggunakan *Filter Dataset*

Pada pengujian sebelumnya dilakukan *training* dengan menggunakan *dataset* yaitu dengan jumlah sampel *training* sebanyak 5703, sampel *validation* sebanyak 713, dan sampel *testing* sebanyak 713 dengan melakukan pembagian *dataset* menggunakan variasi pada setiap *subset training*, *validation* dan *testing* namun tanpa meng-



Gambar 4.4: Grafik Hasil *Training* dan *Validation* Pada Pengujian Dengan Menggunakan Variasi Pembagian *Dataset*

gunakan proses *filtering* pada *dataset*.

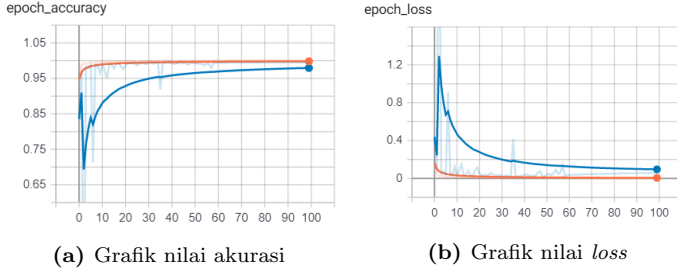
Pada pengujian ini dilakukan *training* dengan menggunakan *dataset* yaitu dengan jumlah sampel *training* sebanyak 5703, sampel *validation* sebanyak 713, dan sampel *testing* sebanyak 713 dengan melakukan pembagian *dataset* menggunakan variasi pada setiap *subset training*, *validation*, *testing* dan menggunakan proses *filtering* pada *dataset*. Proses *filtering* dilakukan menggunakan algoritma dari *biosspy*.

Didapatkan hasil dengan tingkat akurasi *training* sebesar 100% dan tingkat akurasi *validation* sebesar 99.47% seperti yang ditunjukkan pada Gambar 4.5a dan memiliki *loss* pada *training* sebesar $2.968e-12$ atau mendekati 0, sedangkan *loss* pada *validation* sebesar 0.0582. Gambar 4.5b merupakan grafik nilai *loss* pada proses *training* dan *validation*.

4.2.3 Hasil *Training* dan *Validation* Menggunakan Penambahan *Dataset*

Pada pengujian sebelumnya dilakukan *training* dengan menggunakan *dataset* yaitu dengan jumlah sampel *training* sebanyak 5703, sampel *validation* sebanyak 713, dan sampel *testing* sebanyak 713 dengan melakukan pembagian *dataset* menggunakan variasi pada setiap *subset training*, *validation* dan *testing* serta menggunakan proses *filtering* pada *dataset* tanpa melakukan penambahan *dataset*.

Pada pengujian ini dilakukan *training* dengan menggunakan



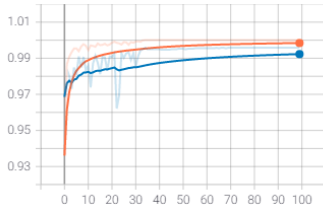
Gambar 4.5: Grafik Hasil *Training* dan *Validation* Pada Pengujian Dengan Menggunakan Variasi Pembagian *Dataset* dan *Filtering Dataset*

dataset yaitu dengan jumlah sampel *training* sebanyak 5703, sampel *validation* sebanyak 713, dan sampel *testing* sebanyak 713 dengan melakukan pembagian *dataset* menggunakan variasi pada setiap *subset training, validation, testing* serta menggunakan proses *filtering* dan penambahan *dataset*. Proses penambahan *dataset* diambil dari hasil akuisisi data menggunakan simulator yang telah dikonfigurasi diagnosis normal dengan *heart rate* sebesar 60 bpm. Data akuisisi yang telah diambil dilakukan *preprocessing* agar hasilnya bisa sesuai dengan *dataset MIT-BIH Arrhythmia Database*. Didapatkan hasil dengan tingkat akurasi *training* sebesar 100% dan tingkat akurasi *validation* sebesar 99.58% seperti yang ditunjukkan pada Gambar 4.6a dan memiliki *loss* pada *training* sebesar 4.835e-12 atau mendekati 0, sedangkan *loss* pada *validation* sebesar 0.0482. Gambar 4.6b merupakan grafik nilai *loss* pada proses *training* dan *validation*.

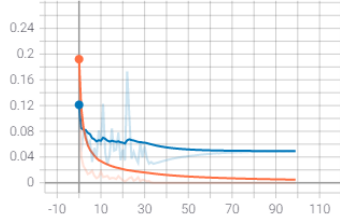
4.3 Pengujian *Testing* Model

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

1. Pengujian menggunakan variasi pembagian *dataset*.
2. Pengujian menggunakan *filter dataset*.
3. Pengujian menggunakan penambahan *dataset*.



(a) Grafik nilai akurasi



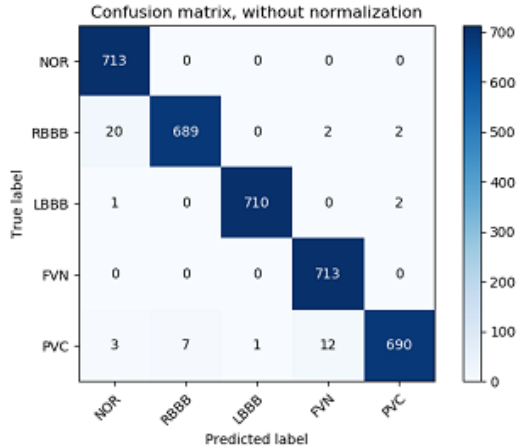
(b) Grafik nilai loss

Gambar 4.6: Grafik Hasil *Training* dan *Validation* Pada Pengujian Dengan Menggunakan Variasi Pembagian *Dataset*, *Filtering Dataset* dan Penambahan *Dataset*

4.3.1 Pengujian *Testing* Model Menggunakan Variasi Pembagian *Dataset*

Pada penelitian yang dilakukan oleh Firdaus Nanda Pradanggapasti menggunakan model hasil *training* menggunakan *dataset* yaitu dengan jumlah sampel *training* sebanyak 5703, sampel *validation* sebanyak 713, dan sampel *testing* sebanyak 713. Pada pembagian *dataset* tidak dilakukan variasi pada setiap *subset training*, *validation* dan *testing*. Didapatkan hasil akurasi sebesar 98.60% 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 4.7 merupakan *confusion matrix* dan Tabel 4.1 merupakan *classification report* dari hasil pengujian dari penelitian yang dilakukan oleh Firdaus Nanda Pradanggapasti.

Pada pengujian ini dilakukan *testing* dengan model hasil *training* menggunakan *dataset* yaitu dengan jumlah sampel *training* sebanyak 5703, sampel *validation* sebanyak 713, dan sampel *testing* sebanyak 713 dengan melakukan pembagian *dataset* menggunakan variasi pada setiap *subset training*, *validation* dan *testing*. Didapatkan hasil akurasi sebesar 98.93% dengan rincian deteksi kelas *Normal Beat* dengan benar sebanyak 712 sampel (99.85%), kelas RBBB sebanyak 712 sampel (99.85%), kelas LBBB sebanyak 707

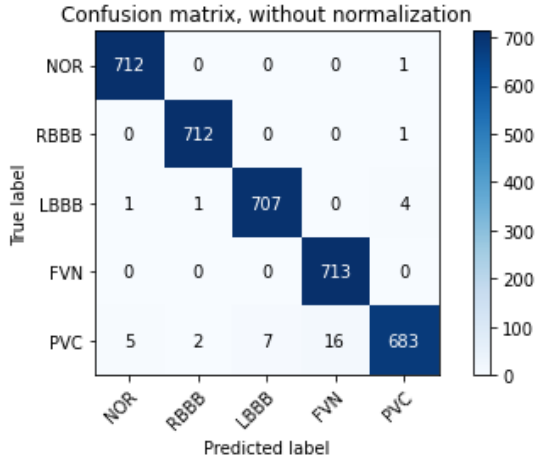


Gambar 4.7: *Confusion Matrix* Hasil Pengujian Oleh Firdaus Nanda Pradangapasti

Tabel 4.1: *Classification Report* Hasil Pengujian Oleh Firdaus Nanda Pradangapasti

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>			0.98

sampel (99.15%), kelas FVN sebanyak 713 sampel (100%), dan kelas PVC sebanyak 683 sampel (95.79%). Gambar 4.8 merupakan *confusion matrix* dan Tabel 4.2 *classification report* dari hasil pengujian.



Gambar 4.8: *Confusion Matrix* Hasil Pengujian Dengan Menggunakan Variasi Pembagian *Dataset*

Tabel 4.2: *Classification Report* Hasil Pengujian Dengan Menggunakan Variasi Pembagian *Dataset*

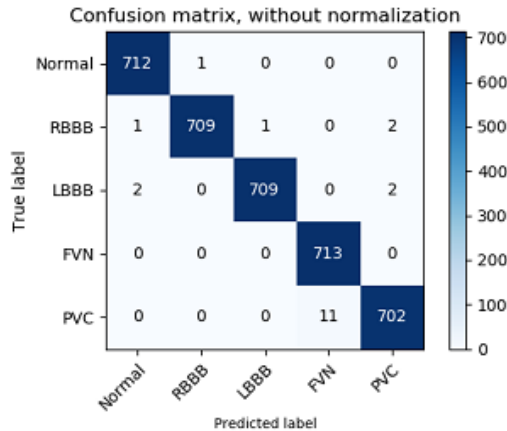
Kelas	<i>Precision</i>	<i>Recall</i>	<i>F1 Score</i>
NOR	0.99	1.00	1.00
RBBB	1.00	1.00	1.00
LBBB	0.99	0.99	0.99
FVN	0.98	1.00	0.99
PVC	0.98	0.97	0.97
<i>Accuracy</i>			0.99

4.3.2 Pengujian *Testing* Model Menggunakan *Filter Dataset*

Pada pengujian sebelumnya dilakukan *testing* dengan model hasil *training* menggunakan *dataset* yaitu dengan jumlah sampel *training* sebanyak 5703, sampel *validation* sebanyak 713, dan sampel *testing* sebanyak 713 dengan melakukan pembagian *dataset* menggunakan variasi pada setiap *subset training*, *validation* dan *testing*

namun tanpa menggunakan proses *filtering* pada *dataset*.

Pada pengujian ini dilakukan *testing* dengan menggunakan *dataset* yaitu dengan jumlah sampel *training* sebanyak 5703, sampel *validation* sebanyak 713, dan sampel *testing* sebanyak 713 dengan melakukan pembagian *dataset* menggunakan variasi pada setiap *subset training, validation* dan *testing* serta menggunakan proses *filtering*. Didapatkan hasil akurasi sebesar 99.44% dengan rincian deteksi kelas *Normal Beat* dengan benar sebanyak 712 sampel (99.85%), kelas RBBB sebanyak 709 sampel (99.44%), kelas LBBB sebanyak 709 sampel (99.44%), kelas FVN sebanyak 713 sampel (100%), dan kelas PVC sebanyak 702 sampel (98.46%). Gambar 4.9 merupakan *confusion matrix* dan Tabel 4.3 merupakan *classification report* dari hasil pengujian.



Gambar 4.9: *Confusion Matrix* Hasil Pengujian Dengan Menggunakan Variasi Pembagian *Dataset* dan *Filtering Dataset*

Tabel 4.3: *Classification Report* Hasil Pengujian Menggunakan Variasi Pembagian *Dataset* dan *Filtering Dataset*

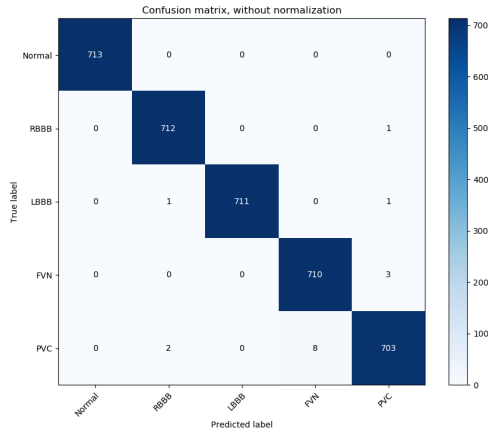
Kelas	<i>Precision</i>	<i>Recall</i>	<i>F1 Score</i>
NOR	1.00	1.00	1.00
RBBB	1.00	0.99	1.00
LBBB	1.00	0.99	1.00
FVN	0.98	1.00	0.99
PVC	0.99	0.98	0.99
<i>Accuracy</i>			0.99

4.3.3 Pengujian *Testing Model* Menggunakan Penambahan *Dataset*

Pada pengujian sebelumnya dilakukan *testing* dengan model hasil *training* menggunakan *dataset* yaitu dengan jumlah sampel *training* sebanyak 5703, sampel *validation* sebanyak 713, dan sampel *testing* sebanyak 713 dengan melakukan pembagian *dataset* menggunakan variasi pada setiap *subset training*, *validation* dan *testing* namun tanpa menggunakan proses *filtering* pada *dataset* namun tanpa menggunakan tambahan *dataset*.

Pada pengujian ini dilakukan *testing* dengan menggunakan *dataset* yaitu dengan jumlah sampel *training* sebanyak 5703, sampel *validation* sebanyak 713, dan sampel *testing* sebanyak 713 dengan melakukan pembagian *dataset* menggunakan variasi pada setiap *subset training*, *validation* dan *testing* serta menggunakan proses *filtering* dan menambah *dataset*. Didapatkan hasil akurasi sebesar 99.55% dengan rincian deteksi kelas *Normal Beat* dengan benar sebanyak 713 sampel (100%), kelas RBBB sebanyak 712 sampel (99.86%), kelas LBBB sebanyak 711 sampel (99.72%), kelas FVN sebanyak 710 sampel (99.58%), dan kelas PVC sebanyak 703 sampel (98.60%). Gambar 4.10 merupakan *confusion matrix* dan Tabel 4.4 merupakan hasil klasifikasi.

Pada Tabel 4.5 menunjukkan perbandingan hasil pengujian *testing* model. Pada baris pertama menunjukkan akurasi model dari penelitian yang dilakukan oleh Firdaus Nanda Pradanggapas-ti. Baris kedua menunjukkan akurasi model menggunakan variasi



Gambar 4.10: *Confusion Matrix* Hasil Pengujian Dengan Menggunakan Variasi Pembagian *Dataset*, *Filtering Dataset* dan Penambahan *Dataset*

Tabel 4.4: *Classification Report* Hasil Pengujian Menggunakan Variasi Pembagian *Dataset*, *Filtering Dataset* dan Penambahan *Dataset*

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

pembagian *dataset*. Baris ketiga menunjukkan akurasi model menggunakan variasi pembagian *dataset* dan *filter* pada *dataset*. Baris keempat menunjukkan akurasi model menggunakan variasi pembagian *dataset*, *filter* pada *dataset* dan penambahan *dataset*. Pengujian yang memiliki akurasi hasil *tesing* paling tinggi yaitu Pengujian menggunakan variasi *dataset*, *filter dataset* dan penambahan *dataset* yaitu 99.55%.

Tabel 4.5: Perbandingan Hasil Pengujian *Testing* Model

Pengujian	Akurasi
Pengujian Firdaus Nanda Pradanggapasti	98.60%
Pengujian menggunakan pembagian variasi <i>dataset</i>	98.93%
Pengujian menggunakan pembagian variasi <i>dataset</i> dan <i>filtering dataset</i>	99.44%
Pengujian menggunakan pembagian variasi <i>dataset</i> , <i>filtering dataset</i> dan penambahan <i>dataset</i>	99.55%

4.4 Pengujian *R Peak*

Hasil berupa *r peak* yang akan digunakan sebagai titik anotasi pada pemotongan data. Uji coba dilakukan dengan mendeteksi *r peak* pada *dataset* MIT-BIH *Arrhythmia Database* pada *record* ke-100, 101, 103, 105, 106, 109, 118, 124, 208 dan 213 yang dipotong selama 3 menit sebanyak 64801 data pada setiap rekaman. Pada Tabel 4.6 menunjukkan *confusion matrix* dari hasil deteksi *r peak*. Nilai *accuracy*, *precision*, *recall* dan *F1 score* ditunjukkan pada Tabel 4.7. Nilai *accuracy* rata-rata adalah 99.98%, *precision* rata-rata adalah 96.46%, *recall* rata-rata adalah 97.52% dan *F1 score* rata-rata adalah 96.91%.

Tabel 4.6: *Confusion Matrix* dari Deteksi *R Peak*

No.	Record	TP	FP	TN	FN	Total
1	100	223	1	64575	2	64801
2	101	212	17	64565	7	64801
3	103	212	1	64586	2	64801
4	105	250	5	64542	4	64801
5	106	212	55	64521	13	64801
6	109	263	3	64534	1	64801
7	118	217	1	64579	4	64801
8	124	148	1	64650	2	64801
9	208	298	6	64472	25	64801
10	213	311	1	64465	4	64801

Tabel 4.7: Nilai *Accuracy*, *Precision*, *Recall*, *F1-Score* dari Deteksi *R Peak*

No.	<i>Record</i>	<i>Accuracy</i>	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>
1	100	100%	99.55%	99.11%	99.33%
2	101	99.96%	92.58%	96.80%	94.69%
3	103	100%	99.53%	99.07%	99.30%
4	105	99.99%	98.04%	98.43%	98.23%
5	106	99.90%	79.40%	94.22%	86.18%
6	109	99.99%	98.87%	99.69%	99.25%
7	118	99.99%	99.54%	98.19%	98.19%
6	109	99.99%	98.87%	99.69%	99.25%
7	118	99.99%	99.54%	98.19%	98.19%
8	124	99.95%	98.03%	92.26%	95.06%
9	208	100%	99.33%	98.67%	99.00%
10	213	99.99%	99.70%	98.81%	99.25%

4.5 Pengujian Sistem Menggunakan *Dataset*

Pengujian dilakukan secara keseluruhan dari proses deteksi *r peak*, konversi nilai *range*, *filtering*, pemotongan sinyal, transformasi spektogram hingga *testing* menggunakan model yang menghasilkan klasifikasi pada setiap titik anotasinya. Dengan menggunakan *dataset* MIT-BIH *Arrhythmia Database* pada *record* ke-100, 101, 103, 105, 106, 109, 118, 124, 208 dan 213 yang diambil sebagian selama 3 menit sebagai data *testing* menghasilkan prediksi seperti Tabel 4.8. Pengambilan *dataset record* ke-100, 101, 103, 105 memiliki diagnosis klasifikasi terbanyak adalah normal, *record* ke-106 memiliki variasi diganosa klasifikasi normal dan PVC, *record* ke-109 memiliki diagnosis klasifikasi terbanyak adalah LBBB, *record* ke-118 dan 124 memiliki klasifikasi terbanyak adalah RBBB sedangkan *record* ke-208, 213 memiliki variasi klasifikasi normal, FVN dan PVC.

Kelas	<i>Record-100</i>	
	<i>Dataset</i>	Prediksi
NOR	223	231
RBBB	0	0
LBBB	0	0
FVN	0	0
PVC	0	0

Kelas	<i>Record-101</i>	
	<i>Dataset</i>	Prediksi
NOR	212	201
RBBB	0	0
LBBB	0	0
FVN	0	0
PVC	0	0

Kelas	<i>Record-103</i>	
	<i>Dataset</i>	Prediksi
NOR	212	211
RBBB	0	0
LBBB	0	0
FVN	0	0
PVC	0	0

Kelas	<i>Record-105</i>	
	<i>Dataset</i>	Prediksi
NOR	242	239
RBBB	0	0
LBBB	0	0
FVN	0	0
PVC	8	11

Kelas	<i>Record-106</i>	
	<i>Dataset</i>	Prediksi
NOR	153	98
RBBB	0	0
LBBB	0	0
FVN	0	38
PVC	59	56

Kelas	<i>Record-109</i>	
	<i>Dataset</i>	Prediksi
NOR	0	1
RBBB	0	0
LBBB	256	257
FVN	2	2
PVC	4	14

Kelas	<i>Record-118</i>	
	<i>Dataset</i>	Prediksi
NOR	0	0
RBBB	212	212
LBBB	0	0
FVN	0	0
PVC	2	2

Kelas	<i>Record-124</i>	
	<i>Dataset</i>	Prediksi
NOR	0	7
RBBB	148	127
LBBB	0	0
FVN	0	5
PVC	0	7

Tabel 4.8: Hasil Pengujian Sistem Menggunakan *Dataset*

Kelas	<i>Record-208</i>	
	<i>Dataset</i>	Prediksi
NOR	163	142
RBBB	0	0
LBBB	0	0
FVN	39	41
PVC	105	109

Kelas	<i>Record-213</i>	
	<i>Dataset</i>	Prediksi
NOR	281	265
RBBB	0	0
LBBB	0	0
FVN	40	49
PVC	9	5

4.6 Pengujian Sistem Menggunakan Data Akuisisi

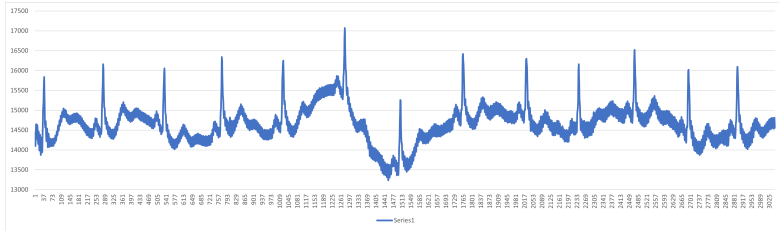
Pengujian dilakukan secara keseluruhan dari proses deteksi *r peak*, konversi nilai *range*, *filtering*, pemotongan sinyal, transformasi spektogram hingga *testing* menggunakan model yang menghasilkan klasifikasi aritmia. Akuisisi data dilakukan menggunakan Raspberry Pi 3 Model B+, Intel Neural Compute Stick 2, ADS1115, Sensor AD8232 mendapatkan frekuensi data 300 - 307 Hz. Dilakukan konversi nilai sinyal dengan *range* -0.5 hingga 0.5 dan dideteksi *r peak* untuk menentukan titik anotasinya. Proses *filtering* dilakukan sebelum proses pemotongan data dan transformasi spektogram. Pengujian ini dilakukan akuisisi data pada pasien dan simulator.

1. Pengujian Pada Pasien
2. Pengujian Pada Simulator

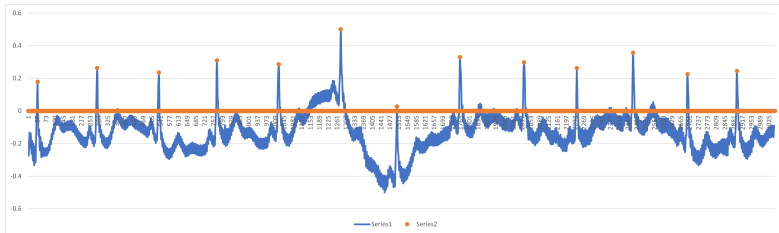
4.6.1 Pengujian Pada Pasien

Akuisisi data dilakukan pada pasien yang belum dipastikan memiliki diagnosis penyakit aritmia. Hasil prediksi dari akuisisi data tersebut adalah terdiagnosis normal. Pada Gambar 4.11 adalah grafik akuisisi data pasien. Data akuisisi yang telah disesuaikan *range* nilainya akan dideteksi *r peak* seperti pada Gambar 4.12. Proses *Filtering* dilakukan sebelum proses pemotongan data dan transformasi spektogram seperti Gambar 4.13. Pemotongan data dilakukan sepanjang 300 data dengan poros pada titik anotasi, kemudian dilakukan konversi frekuensi. Terdapat 60 data baru hasil dari proses konversi frekuensi jadi dalam satu pemotongan data yaitu sepan-

jang 360 data. Gambar 4.14 menunjukkan hasil konversi frekuensi.



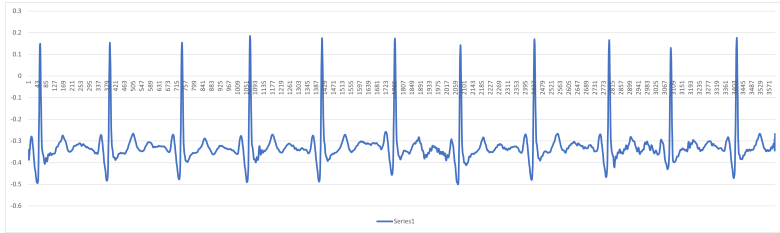
Gambar 4.11: Hasil Proses Akuisisi Data Pasien selama 10 Detik



Gambar 4.12: Hasil Proses Konversi Nilai *Range* dan Deteksi *R Peak*



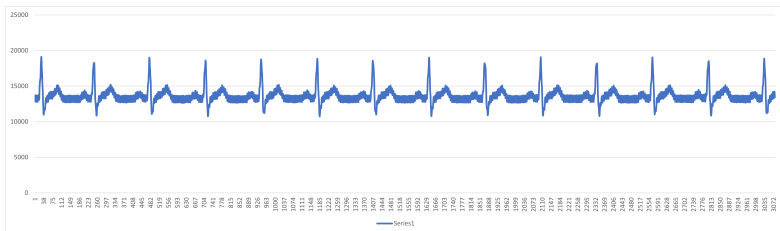
Gambar 4.13: Hasil *Filtering*



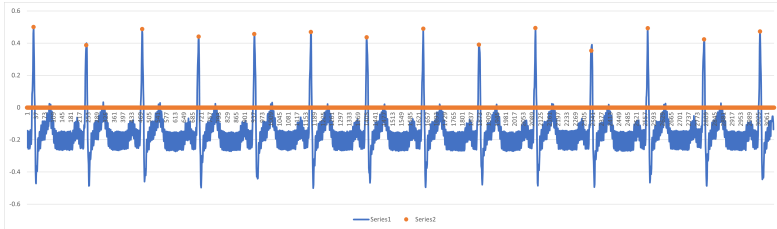
Gambar 4.14: Hasil Proses Konversi Frekuensi

4.6.2 Pengujian Pada Simulator

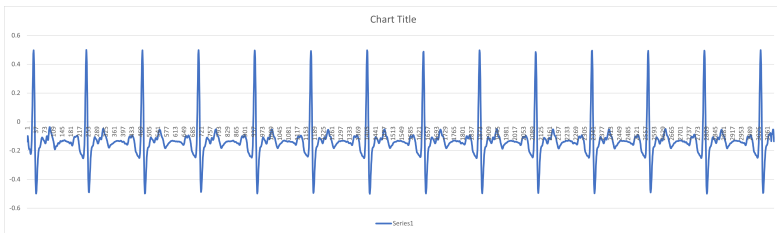
Akuisisi data dilakukan dengan menggunakan simulator yang sudah dikonfigurasi dalam kondisi normal dan *heart rate* 60 bpm. Sistem dapat mengklasifikasikan seluruh anotasi dari simulator tersebut adalah normal. Pada Gambar 4.15 adalah grafik akuisisi data simulator. Data akuisisi yang telah disesuaikan *range* nilainya akan dideteksi *r peak* seperti pada Gambar 4.16. Proses *Filtering* dilakukan sebelum dilakukan proses pemotongan data dan transformasi spektrogram seperti Gambar 4.17. Pemotongan data dilakukan sepanjang 300 data dengan poros pada titik anotasi, kemudian dilakukan konversi frekuensi. Terdapat 60 data baru hasil dari proses konversi frekuensi jadi dalam satu pemotongan data yaitu sepanjang 360 data. Gambar 4.18 menunjukkan hasil konversi frekuensi.



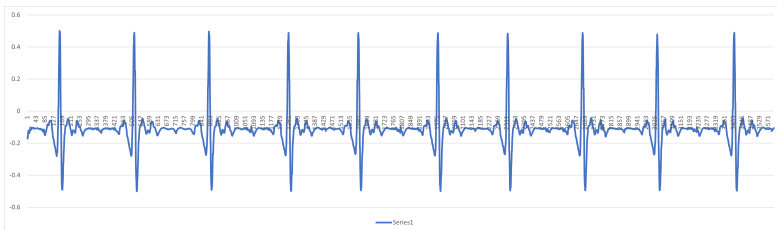
Gambar 4.15: Hasil Proses Akuisisi Data Simulator selama 10 Detik



Gambar 4.16: Hasil Proses Konversi Nilai *Range* dan Deteksi *R Peak*



Gambar 4.17: Hasil Proses *Filtering*



Gambar 4.18: Hasil Proses Konversi Frekuensi

4.7 Pengujian *Execution Time* Analisis

Pengujian *execution time* dari analisis sinyal ECG dilakukan dengan menjalankan program *python* dari secara keseluruhan mulai dari proses deteksi *r peak*, konversi nilai *range*, *filtering*, pemotongan sinyal, transformasi spektogram hingga *testing* menggunakan model yang menghasilkan klasifikasi pada setiap titik anotasinya. Pengujian ini menggunakan *Personal Computer* dengan spesifikasi pada Tabel 4.9 dan Raspberry Pi 3 Model B+ dan Intel Compute Stick

dengan spesifikasi pada Tabel 4.10.

Tabel 4.9: Spesifikasi *Personal Computer*

Processor	Intel(R) Core(TM) i7-7700HQ CPU @ 2.40GHz
RAM	16 GB SODIMM DDR4 2400 MHz Dual Channel
Storage	SSD M.2 NVMe 128GB
Graphic Card	NVIDIA GeForce GTX 1050 Ti 2GB
Operating System	Windows 10 Home 64-bit

Tabel 4.10: Spesifikasi Raspberry Pi 3 Model B+

Processor	Quad-core ARM Cortex-A53 CPU @ @ 1.40GHz
RAM	1 GB LPDDR SDRAM
Storage	microSDHC V-Gen 16 GB
Graphic Card	—————Tidak <i>Support</i> —————
Operating System	Raspbian 32-bit

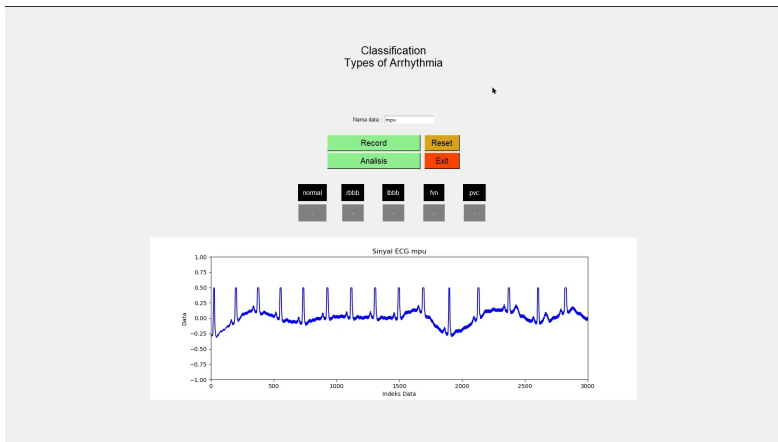
Hasil rata-rata perbandingan menggunakan *personal computer* dan raspberry pi 3 model B+ adalah 1 : 3. Pada Tabel 4.11 menunjukkan perbandingan *execution time*.

Tabel 4.11: Hasil *Execution Time Personal Computer* dan Raspberry Pi 3 Model B+

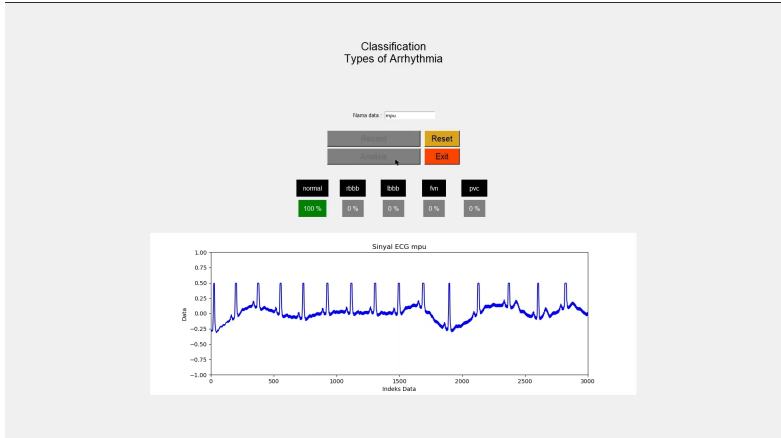
Jumlah Data	<i>Execution time</i> (detik)	
	<i>Personal Computer</i>	Raspberry Pi
1080	6.346	17.2889
1440	7.8120	22.1165
1800	9.6496	27.3665
3600	20.1307	63.5902

4.8 Pengujian Menggunakan *User Interface*

Pengujian sistem dilakukan menggunakan *user interface* yang dibuat menggunakan *package* TKinter. *User interface* dijalankan dengan memberikan nama rekaman pada *text box* nama data. Tombol *record* digunakan untuk melakukan akuisisi data selama 10 detik yang disimpan dalam format *file* csv. Tombol analisis digunakan untuk melakukan analisa sinyal ECG agar dapat mengklasifikasikan aritmia. Tombol *reset* digunakan untuk menghapus grafik dan *cache* dari rekaman sebelumnya. Tombol *exit* digunakan untuk keluar dari aplikasi. Gambar 4.19 menampilkan hasil *record* selama 10 detik dan Gambar 4.20 menampilkan hasil analisis yang menunjukkan prediksi 100% kelas normal serta tidak ada indikasi kelas aritmia lain.



Gambar 4.19: Tampilan Hasil *Record*



Gambar 4.20: Tampilan Hasil Analisis

Halaman ini sengaja dikosongkan

BAB 5

PENUTUP

5.1 Kesimpulan

Berdasarkan hasil pengujian dapat disimpulkan bahwa sistem dapat bekerja dengan cukup baik sesuai dengan rancangan. Penulis mengimplementasikan model hasil *training* dengan perangkat ECG menggunakan Raspberry Pi 3 Model B+ dan Intel Compute Stick sebagai perangkat akuisisi data dan prediksi klasifikasi jenis aritmia dengan baik. Kesimpulan lainnya, sebagai berikut:

1. Berdasarkan hasil pengujian deteksi *r peak* dengan menguji data hasil dengan data *record* dari dataset anotasi MIT-BIH *Arrhythmia Database* menunjukkan nilai *accuracy* rata-rata adalah 99.98%, *precision* rata-rata adalah 96.46%, *recall* rata-rata adalah 97.52% dan *F1 score* rata-rata adalah 96.91%.
2. Berdasarkan hasil *testing*, klasifikasi aritmia pada sinyal ECG dengan menggunakan metode CNN sudah dapat dilakukan dengan baik, dengan tingkat hasil akurasi sebesar 99.55% dengan rincian deteksi kelas *Normal Beat* dengan benar sebanyak 713 sampel (100%), kelas RBBB sebanyak 712 sampel (99.86%), kelas LBBB sebanyak 711 sampel (99.72%), kelas FVN sebanyak 710 sampel (99.58%), dan kelas PVC sebanyak 703 sampel (98.60%).
3. Pengujian keseluruhan sistem dilakukan dari deteksi *r peak*, pemotongan sinyal, transformasi menjadi spektogram dan tahap *testing* menggunakan Raspberry Pi 3 B+. Pengujian dilakukan dengan menggunakan dataset MIT-BIH *Arrhythmia Database* pada *record* ke-100, 101, 103, 105, 106, 109, 118, 124, 208 dan 213 yang diambil sebagian selama 3 menit sebagai data *testing* menghasilkan prediksi dengan akurasi 91,44%.
4. Pengujian keseluruhan sistem menggunakan data akuisisi diambil menggunakan simulator dan subjek pasien selama 10 detik. Simulator yang sudah dikonfigurasi diagnosa normal dapat diklasifikasikan seluruh anotasinya oleh sistem dengan diagnosa normal. Pengujian data akuisisi oleh subjek pasien dilakukan oleh subjek penulis sendiri yang diklasifikasikan

oleh sistem adalah normal.

5. Pengujian *execution time* menggunakan *personal computer* dengan Raspberry Pi 3 Model B+ memiliki perbandingan performa 1:3. Raspberry Pi 3 Model B+ dapat melakukan komputasi sistem keseluruhan dengan 3600 data (akuisisi data selama 10 detik) selama 63.5902 detik.

5.2 Saran

Untuk pengembangan penelitian selanjutnya terdapat beberapa saran sebagai berikut :

1. Menambah jumlah sampel pada kelas aritmia yang masih memiliki jumlah sampel yang sedikit yang akan digunakan pada klasifikasi aritmia.
2. Sistem akan lebih baik jika menggunakan 12 *lead* untuk akuisisi data agar dapat mengklasifikasikan kelas aritmia lainnya.
3. Mengoptimasi metode agar klasifikasi aritmia dapat dijalankan secara *realtime*.

DAFTAR PUSTAKA

- [1] A. Isin and S. Ozdalili, "Cardiac arrhythmia detection using deep learning," Procedia computer science, vol. 120, pp. 268–275, 2017. (Dikutip pada halaman 1).
- [2] 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 2020-11-30. (Dikutip pada halaman 1).
- [3] MedlinePlus, "Arrhythmia," 2016. <https://medlineplus.gov/arrhythmia.html>, Last accessed on 2020-11-30. (Dikutip pada halaman 1).
- [4] MedlinePlus, "Holter monitor (24h)," 2018. <https://medlineplus.gov/ency/article/003877.htm>, Last accessed on 2020-11-30. (Dikutip pada halaman 1).
- [5] MathWorks, "What is deep learning?," <https://ww2.mathworks.cn/en/discovery/deep-learning.html>, Last accessed on 2020-12-14. (Dikutip pada halaman 2).
- [6] Kartik Chaudhary, "Understanding audio data, fourier transform, fft and spectrogram features for a speech recognition system," 2019. dropsosai.com. (Dikutip pada halaman 2).
- [7] I. K. E. P. Arief Kurniawan, Eko Mulyanto Y., QVAT QRS Complex Detection based on Variance Analysis and Adaptive Threshold for Electrocardiogram Signal. Surabaya, Jawa Timur, Indonesia: Institut Teknologi Sepuluh Nopember dan Universitas Airlangga, 2020. (Dikutip pada halaman 2, 9, 18, 30).
- [8] R. D. K. Akash Chatlani, Shradha Kabu, IOT Based Detection of Cardiac Arrhythmias Using Raspberry PI. (Dikutip pada halaman 7).
- [9] X. G. Tancheng Lu, Peng Liu and Q. Lu, A Portable ECG Monitor with Low Power Consumption and Small Size Based on AD8232 Chip. (Dikutip pada halaman 8).

- [10] A. Isin and S. Ozdalili, Cardiac arrhythmia detection using deep learning, vol. 120. Elsevier, 2017. (Dikutip pada halaman 8).
- [11] Ö. Yıldırım, P. Plawiak, R.-S. Tan, and U. R. Acharya, Arrhythmia detection using deep convolutional neural network with long duration ECG signals, vol. 102. Elsevier, 2018. (Dikutip pada halaman 8).
- [12] 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, vol. 89. Elsevier, 2017. (Dikutip pada halaman 8).
- [13] U. Hasanah, L. R. Mayangsari, A. Pratama, and I. Cholissodin, Perbandingan Metode Svm, Fuzzy-Knn, Dan Bdt-Svm Untuk Klasifikasi Detak Jantung Hasil Elektrokardiografi, vol. 3. 2016. (Dikutip pada halaman 9).
- [14] I. Masruroh, Pendeteksi Penyakit Aritmia berbasis Prediksi Fitur QRS pada Sinyal ECG dengan Metode Feed Forward Backpropagation menggunakan Sistem Tertanam. 2019. (Dikutip pada halaman 9).
- [15] F. N. Pradangapasti, Klasifikasi Aritmia pada Sinyal ECG Berbasis Deep Learning. 2019. (Dikutip pada halaman 9).
- [16] Mohammed Sunasra, “Performance metrics for classification problems in machine learning,” 2017. medium.com. (Dikutip pada halaman 10, 16).
- [17] A. Isin and S. Ozdalili, “Cardiac arrhythmia detection using deep learning,” Procedia computer science, vol. 120, pp. 268–275, 2017. (Dikutip pada halaman 11).
- [18] L. M. Azizah, S. F. Umayah, S. Riyadi, C. Damarjati, and N. A. Utama, “Deep learning implementation using convolutional neural network in mangosteen surface defect detection,” in 2017 7th IEEE International Conference on Control System, Computing and Engineering (ICCSCE), pp. 242–246, 2017. (Dikutip pada halaman 13).

- [19] N. Jmour, S. Zayen, and A. Abdelkrim, “Convolutional neural networks for image classification,” in 2018 International Conference on Advanced Systems and Electric Technologies, pp. 397–402, 2018. (Dikutip pada halaman 13).
- [20] R. P. Foundation, Raspberry Pi 3 Model B+. <https://static.raspberrypi.org>, 2017. (Dikutip pada halaman 19).
- [21] O. T. Way, Single-Lead, Heart Rate Monitor Front End AD8232. Analog Devices, Inc, 2012-2018. (Dikutip pada halaman 20).
- [22] T. Instruments, Ultra Small, Low Power, 16 Bit. Analog to Digital Converter with Internal Reference. <https://www.ti.com>, 2009. (Dikutip pada halaman 21).

Halaman ini sengaja dikosongkan

BIOGRAFI PENULIS



Mpu Hambyah Syah Bagaskara Aji, atau biasa dipanggil Mpu, lahir di Klaten, Jawa Tengah pada tanggal 1 November 1998. Merupakan anak ketiga dari berlima bersaudara. Penulis lulus dari SMP Negeri 1 Delanggu dan melanjutkan ke SMA Negeri 1 Karanganyar. Penulis melanjutkan ke jenjang strata satu di Departemen Teknik Komputer Fakultas Teknologi Elektro dan Informatika Cerdas ITS. Dalam masa kuliah, penulis tertarik dengan pengembangan Robotika, *Web Apps* dan *Internet of Things* (IoT). Penulis pernah aktif menjadi salah satu anggota dari Tim Barunastra ITS dan Laboratorium B401, serta menjadi Koordinator Dana Usaha MAGE 5. Bagi pembaca yang memiliki kritik, saran, atau pertanyaan mengenai tugas akhir ini dapat menghubungi penulis melalui email mpu.hambyah@gmail.com.

Halaman ini sengaja dikosongkan