



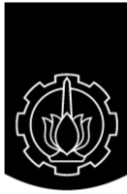
TUGAS AKHIR - EC184801

**PENDETEKSI PENYAKIT ARITMIA BERBASIS PREDIKSI
FITUR QRS PADA SINYAL ECG DENGAN METODE
FEED FORWARD BACKPROPAGATION
MENGUNAKAN SISTEM TERTANAM**

Izzatul Masruroh
NRP 07211640000005

Dosen Pembimbing
Arief Kurniawan, ST., MT.
Dr. Eko Mulyanto Yuniarno, ST., MT.

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



ITS
Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR - EC184801

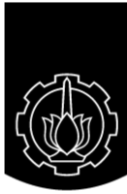
**PENDETEKSI PENYAKIT ARITMIA BERBASIS PREDIKSI
FITUR QRS PADA SINYAL ECG DENGAN METODE
FEED FORWARD BACKPROPAGATION
MENGUNAKAN SISTEM TERTANAM**

Izzatul Masrurroh
NRP 07211640000005

Dosen Pembimbing
Arief Kurniawan, ST., MT.
Dr. Eko Mulyanto Yuniarno, ST., MT.

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

[Halaman ini sengaja dikosongkan].



ITS
Institut
Teknologi
Sepuluh Nopember

FINAL PROJECT - EC184801

**ARRHYTHMIA DETECTOR BASED ON FEATURE OF
QRS IN ECG SIGNAL WITH FEED FORWARD
BACKPROPAGATION METHOD USING EMBEDDED
SYSTEM**

Izzatul Masrurroh
NRP 07211640000005

Advisor
Arief Kurniawan, ST., MT.
Dr. Eko Mulyanto Yuniarno, ST., MT.

DEPARTMENT OF COMPUTER ENGINEERING
Faculty of Intelligent Electrical and Informatics Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2020

[Halaman ini sengaja dikosongkan].

PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul **“Pendeteksi Penyakit Aritmia berbasis Prediksi Fitur QRS pada Sinyal ECG dengan Metode *Feed Forward Backpropagation* menggunakan Sistem Tertanam”** adalah benar-benar hasil karya intelektual sendiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya orang lain yang saya akui sebagai karya sendiri.

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

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

Surabaya, Juli 2020

Izzatul Masruroh
NRP. 07211640000005

LEMBAR PENGESAHAN

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

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

Oleh: Izzatul Masruroh (NRP: 07211640000005)

Tanggal Ujian : 7 Juli 2020

Periode Wisuda : September 2020

Disetujui oleh:

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

(Pembimbing I)

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

(Pembimbing II)

Prof. Dr. Ir. Yoyon Kusnendar Suprpto, M.Sc.
NIP: 19540925 197803 1 001

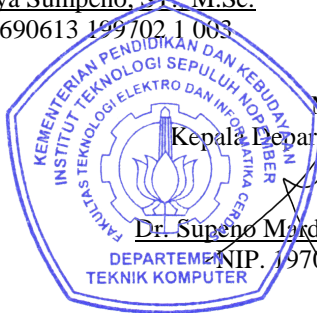
(Penguji I)

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

(Penguji II)

Dr. Surya Sumpeno, ST., M.Sc.
NIP. 19690613 199702 1 003

(Penguji III)



Mengetahui
Kepala Departemen Teknik Komputer

Dr. Supeno Mukdi Susiki Nugroho, ST., MT.
NIP. 19700313 199512 1 001

ABSTRAK

Nama Mahasiswa : Izzatul Masruroh
Judul Tugas Akhir : Pendeteksi Penyakit Aritmia berbasis
Prediksi Fitur QRS pada Sinyal ECG dengan Metode *Feed Forward Backpropagation* menggunakan Sistem Tertanam.
Pembimbing : 1. Arief Kurniawan, ST., MT.
2. Dr. Eko Mulyanto Yuniarno, S.T., M.T.

Aritmia adalah suatu tanda atau gejala dari gangguan detak jantung atau irama jantung yang bisa mengindikasikan pada penyakit kardiovaskular (penyakit yang berkaitan dengan organ jantung dan pembuluh darah). Berdasarkan data dari *American Heart Association* (AHA) penyakit kardiovaskular terdata sebagai penyebab utama kematian, terhitung sebanyak hampir 836.546 kematian di US 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 [1]. Maka dari itu, deteksi aritmia sangat diperlukan dengan menggunakan teknologi pada bidang machine learning yang akhir-akhir ini sangat berkembang dengan pesat, sehingga penyakit aritmia bisa diketahui se-dini mungkin dan dapat segera menerima terapi / pengobatan tertentu untuk mengurangi resiko terkena penyakit kardiovaskular.

Kata kunci : Aritmia, *Electrocardiography*, *Machine Learning Backpropagation*, Sistem Tertanam.

Halaman ini sengaja dikosongkan

ABSTRACT

Name : Izzatul Masruroh
Title : *Arrhythmia Detector Based on Feature of QRS in ECG Signal With Feed Forward Backpropagation Method Using Embedded System.*
Advisors : 1. Arief Kurniawan, ST., MT.
2. Dr. Eko Mulyanto Yuniarno, S.T., M.T.

Arrhythmia is a sign or symptom of a disturbance in heart rate or heart rhythm that can indicate cardiovascular disease (a disease related to the heart organ and blood vessels). Based on data from American Heart Association (AHA) recorded cardiovascular disease as the leading cause of death, accounting for nearly 836,546 deaths in the US and being the cause of death globally, which reported more than 17.9 million deaths each year in 2015, the figure it is expected to grow to more than 23.6 million by 2030 [1]. Therefore, arrhythmia detection is needed by using technology in the field of machine learning which is very rapidly growing lately, so that arrhythmia can be known as early as possible and can immediately receive certain therapies / treatments to reduce the risk of cardiovascular disease.

Keyword : Arrhythmia, Electrocardiography, Machine Learning Backpropagation, Embedded System

Halaman ini sengaja dikosongkan

KATA PENGANTAR

Puji dan syukur kehadirat Allah SWT atas segala limpahan berkah dan hidayah-Nya, penulis dapat menyelesaikan penelitian ini dengan judul "**Pendeteksi Penyakit Aritmia berbasis Prediksi Fitur QRS pada Sinyal ECG dengan Metode *Feed Forward Backpropagation* menggunakan Sistem Tertanam**" untuk mendeteksi denyut jantung yang tidak normal (aritmia) menggunakan sebuah perangkat. Penelitian ini disusun dalam rangka pemenuhan bidang riset di Departemen Teknik Komputer, FTEIC-ITS serta digunakan sebagai persyaratan menyelesaikan pendidikan S1. Penelitian ini dapat terselesaikan tidak lepas dari bantuan berbagai pihak. Oleh karena itu, penulis mengucapkan terima kasih kepada:

1. Keluarga, Ibu, Bapak 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 Teknologi Elektro dan Informatika Cerdas, Institut Teknologi Sepuluh Nopember.
3. Secara khusus penulis mengucapkan terima kasih yang sebesar - besarnya kepada Bapak Arief Kurniawan, ST., MT. dan Bapak Dr. Eko Mulyanto Yuniarno, ST., MT. atas bimbingan selama pengerjaan tugas akhir ini.
4. Bapak-ibu dosen pengajar Departemen Teknik Komputer, atas pengajaran, bimbingan, serta perhatian yang diberikan kepada penulis selama ini.
5. Teman-teman Teknik Komputer Angkatan 2016 serta Asisten Laboratorium Telematika B401 yang tidak dapat saya sebutkan satu-persatu.

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 2020

Penulis

Halaman ini sengaja dikosongkan

DAFTAR ISI

ABSTRAK	i
ABSTRACT	iii
KATA PENGANTAR	v
DAFTAR ISI	vii
DAFTAR GAMBAR	xi
DAFTAR TABEL	xv
DAFTAR ALGORITMA	xvii
NUMENKLATUR	xix
1 PENDAHULUAN	1
1.1 Latar belakang	1
1.2 Permasalahan	2
1.3 Tujuan	2
1.4 Batasan masalah	2
1.5 Sistematika Penulisan	2
1.6 Relevansi	3
2 TINJAUAN PUSTAKA	5
2.1 Dasar Teori	5
2.1.1 Segmen Sinyal Jantung	5
2.1.2 Aritmia	7
2.1.3 <i>Machine Learning</i>	7
2.1.3.1 Algoritma <i>Machine Learning</i> Berda- sarkan Metode Pembelajaran	8
2.1.3.2 Dataset	9
2.1.3.3 Arsitektur <i>Neural Network</i>	10
2.1.3.4 <i>Weight</i> dan Bias	11
2.1.3.5 <i>Activation Function</i>	11
2.1.3.6 <i>Loss Function</i>	12

2.1.3.7	<i>Optimizers</i>	13
2.1.4	Raspberry Pi 3 Model B+	14
2.1.5	Sparkfun AD8232 <i>Single Lead Heart Monitor</i>	15
2.1.6	Intel <i>Neural Compute Stick 2</i>	16
2.1.7	ADS1115	17
2.2	Penelitian Terkait	18
2.2.1	<i>IOT Based Detection of Cardiac Arrhythmias Using Raspberry PI</i>	18
2.2.2	Perangkat <i>Treadmill</i> untuk Mendeteksi Ke- lainan Detak Jantung Menggunakan Sensor ECG Berbasis <i>Single Board Computer</i>	18
2.2.3	<i>ECG Signal Analysis Using Wavelet Transforms</i>	20
2.2.4	<i>An Approach to Detect QRS Complex Using Backpropagation Neural Network</i>	23
2.2.5	<i>The ECG Signal Prediction By Using Neural Network</i>	23
2.2.6	<i>Backpropagation and His Application in ECG Classification</i>	24
2.2.7	<i>QVAT: QRS Complex Detection based on Va- riance Analysis and Adaptive Threshold for Electrocardiogram Signal</i>	25
3	DESAIN DAN IMPLEMENTASI SISTEM	27
3.1	Desain Sistem	27
3.2	Desain Perangkat ECG	27
3.3	Perancangan Perangkat ECG	30
3.4	Desain <i>User Interface</i>	32
3.5	Akuisisi Data ECG	32
3.6	Normalisasi	33
3.7	Deteksi QRS Menggunakan <i>Feed Forward Backpro- pagation</i>	34
3.7.1	Data ECG	35
3.7.2	<i>Set-Up Training Set</i>	35
3.7.3	<i>Forward Propagation</i>	39
3.7.4	Hasil Prediksi	39
3.7.5	<i>Backward Propagation</i>	39
3.8	Deteksi Aritmia	40
3.8.1	Deteksi Puncak R	40

3.8.2	Deteksi <i>RR-Interval</i>	41
3.8.3	<i>Decision</i>	42
3.9	Visualisasi	42
4	PENGUJIAN DAN ANALISA	45
4.1	Pengujian Akuisisi Data ECG	45
4.2	Pengujian Deteksi QRS	45
4.3	Pengujian Deteksi <i>R-Peak</i>	45
4.4	Pengujian Deteksi <i>RR-Interval</i>	46
4.5	Pengujian Deteksi Aritmia	47
5	PENUTUP	49
5.1	Kesimpulan	49
5.2	Saran	49
	DAFTAR PUSTAKA	51
	LAMPIRAN	55
	BIOGRAFI PENULIS	91

Halaman ini sengaja dikosongkan

DAFTAR GAMBAR

2.1	Segmen Sinyal Jantung [2]	5
2.2	Arsitektur <i>Neural Network</i>	10
2.3	Arsitektur <i>Neural Network</i> dan <i>Weight</i> [3]	11
2.4	Raspberry Pi 3 Model B+ [4]	14
2.5	Modul Sparkfun ECG AD8232 [5]	15
2.6	<i>Intel Nerual Compute Stick 2</i> [6]	17
2.7	ADS1115	17
2.8	Blok Diagram Perangkat <i>Treadmill</i> [7]	19
2.9	Sinyal ECG Normal	21
2.10	Sinyal ECG Tachycardia	22
2.11	Sinyal ECG Bradycardia	22
2.12	Perbedaan Sinyal Estimasi menggunakan <i>Backpropagation</i> dengan Sinyal ECG Asli [8]	24
2.13	Blok Diagram QVAT [9]	25
3.1	Blok Diagram Kerja Sistem	27
3.2	Desain Rangkaian Perangkat ECG	28
3.3	Desain Tiap Komponen Secara Terpisah	29
3.4	Desain Tiap Komponen Secara Tersambung	29
3.5	Desain Tiap Komponen Secara Tersambung Tampak Atas	29
3.6	Desain Perangkat Dalam <i>Package</i>	29
3.7	Desain <i>Package</i> Dalam Kondisi Tertutup	29
3.8	Blok Diagram Hubungan Tiap Komponen	30
3.9	Perangkat Tampak Atas	31
3.10	Perangkat Tampak Atas Tanpa Tutup	31
3.11	Perangkat Tampak Samping Atas	32
3.12	Perangkat Tampak Samping	32
3.13	Desain <i>User Interface</i>	32
3.14	Penempatan Posisi Elektroda ECG	33
3.15	Blok Diagram <i>Training Data</i>	34
3.16	Arsitektur <i>Neural Network</i>	35
3.17	Ilustrasi <i>Windowing Data</i>	37
3.18	Ilustrasi Proses Pemberian Label	38
3.19	Blok Diagram Deteksi Aritmia	40

3.20	Visualisasi Hasil Deteksi	43
1	Akuisisi Data a1	55
2	Akuisisi Data a2	55
3	Akuisisi Data a3	56
4	Akuisisi Data a4	56
5	Akuisisi Data a5	57
6	Akuisisi Data a6	57
7	Akuisisi Data a7	58
8	Akuisisi Data a8	58
9	Deteksi QRS Pada Data a1	59
10	Deteksi QRS Pada Data a2	59
11	Deteksi QRS Pada Data a3	60
12	Deteksi QRS Pada Data a4	60
13	Deteksi QRS Pada Data a5	61
14	Deteksi QRS Pada Data a6	61
15	Deteksi QRS Pada Data a7	62
16	Deteksi QRS Pada Data a8	62
17	Deteksi Puncak R Data a1	63
18	Deteksi Puncak R Data a2	63
19	Deteksi Puncak R Data a3	64
20	Deteksi Puncak R Data a4	64
21	Deteksi Puncak R Data a5	65
22	Deteksi Puncak R Data a6	65
23	Deteksi Puncak R Data a7	66
24	Deteksi Puncak R Data a8	66
25	Deteksi <i>RR-Interval</i> Data a1	67
26	Deteksi <i>RR-Interval</i> Data a2	67
27	Deteksi <i>RR-Interval</i> Data a3	68
28	Deteksi <i>RR-Interval</i> Data a4	68
29	Deteksi <i>RR-Interval</i> Data a5	69
30	Deteksi <i>RR-Interval</i> Data a6	69
31	Deteksi <i>RR-Interval</i> Data a7	70
32	Deteksi <i>RR-Interval</i> Data a8	70
33	Deteksi Aritmia Pada Data a1	71
34	Deteksi Aritmia Pada Data a2	71
35	Deteksi Aritmia Pada Data a3	72
36	Deteksi Aritmia Pada Data a4	72

37	Deteksi Aritmia Pada Data a5	73
38	Deteksi Aritmia Pada Data a6	73
39	Deteksi Aritmia Pada Data a7	74
40	Deteksi Aritmia Pada Data a8	74

Halaman ini sengaja dikosongkan

DAFTAR TABEL

2.1	Kondisi Segmen Jantung Normal [10]	20
2.2	Karakter Beberapa Kelainan Jantung [10]	21
3.1	Konfigurasi Perangkat ECG	28
4.1	Deteksi Puncak R	46
4.2	Deteksi RR Interval	46
4.3	Deteksi Aritmia	48

Halaman ini sengaja dikosongkan

DAFTAR ALGORITMA

1	Algoritma <i>Training</i>	85
2	Algoritma Akuisisi Data	86
3	Algoritma Deteksi QRS	87
4	Algoritma Deteksi Puncak R	88
5	Algoritma Deteksi <i>RR-Interval</i>	89
6	Algoritma Deteksi Aritmia	90

Halaman ini sengaja dikosongkan

NOMENKLATUR

<i>rentang</i>	= Jangkauan dari data ECG
<i>data[i]</i>	= <i>Array</i> penyimpanan data ECG pada indeks ke <i>i</i>
<i>min(data)</i>	= Nilai minimal dari data ECG
<i>max(data)</i>	= Nilai maksimal dari data ECG
<i>i</i>	= Indeks ke <i>i</i> pada suatu <i>array</i>
<i>j</i>	= Indeks ke <i>j</i> pada suatu <i>array</i>
<i>norm[i]</i>	= <i>Array</i> penyimpanan data normalisasi pada indeks ke <i>i</i>
<i>QRS[i]</i>	= <i>Array</i> penyimpanan data QRS pada indeks ke <i>i</i>
<i>predict</i>	= Nilai prediksi (probabilitas) pada proses deteksi QRS
<i>Hz</i>	= Hertz (satuan dari frekuensi)
<i>I</i>	= <i>Node</i> pada <i>input layer</i>
<i>u</i>	= Banyaknya <i>node</i> pada <i>input layer</i>
<i>H</i>	= <i>Node</i> pada <i>hidden layer</i>
<i>v</i>	= Banyaknya <i>node</i> pada <i>hidden layer</i>
<i>O</i>	= <i>Node</i> pada <i>output layer</i>
<i>lab_{qrs}[i]</i>	= <i>Array</i> penyimpanan identifikasi QRS pada indeks ke <i>i</i>
<i>label</i>	= <i>Array</i> penyimpanan label QRS
<i>w</i>	= Panjang data sebagai hasil dari proses pemecahan data
<i>W[i]</i>	= <i>Array</i> penyimpanan <i>window</i> pada indeks ke <i>i</i>
<i>QRS_{real}</i>	= Kumpulan data dalam satu kompleks QRS
<i>freq</i>	= Frekuensi
<i>n_{label}</i>	= Jumlah data label
<i>n_{data}</i>	= Jumlah data ECG
<i>r_{temp}</i>	= Penyimpanan sementara nilai puncak R
<i>std</i>	= Standar deviasi dari data normalisasi
<i>r</i>	= <i>Array</i> penyimpanan indeks yang merupakan puncak R
<i>waktu_{ecg}</i>	= <i>Array</i> penyimpanan waktu saat akuisisi data ECG
<i>rr</i>	= <i>Array</i> penyimpanan <i>RR interval</i>
<i>Aritmia</i>	= <i>Array</i> penyimpanan data aritmia

Halaman ini sengaja dikosongkan

BAB 1

PENDAHULUAN

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

1.1 Latar belakang

Berdasarkan data dari *American Heart Association* penyakit kardiovaskular terdata sebagai penyebab utama kematian, terhitung sebanyak hampir 836.546 kematian di US 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 [1]. Salah satu cara untuk mendiagnosis penyakit kardiovaskular yaitu dengan menganalisa sinyal eeg untuk melihat aktifitas kelistrikan yang dihasilkan oleh jantung.

Jumlah dokter dengan pasien kardiovaskular tidak sebanding diantaranya memiliki rasio perbandingan mencapai 1:250.000 [11] sehingga pasien akan masuk *waiting list* untuk menerima pemeriksaan/penanganan medis oleh dokter spesialis jantung.

Teknologi *machine learning* merupakan salah satu cabang dari kecerdasan buatan, adalah disiplin ilmu yang mencakup perancangan dan pengembangan algoritma yang memungkinkan komputer untuk mengembangkan perilaku yang didasarkan pada data empiris, seperti dari sensor data basis data. Fokus besar penelitian *machine learning* adalah bagaimana mengenali secara otomatis pola kompleks dan membuat keputusan cerdas berdasarkan data.

Maka dari itu, deteksi aritmia sangat diperlukan dengan menggunakan teknologi pada bidang *machine learning* untuk mengenali pola kompleks yang ada pada sinyal ECG. Sistem ini akan mengambil sinyal jantung selama periode waktu tertentu untuk menjadi masukan dalam proses *learning* yang kemudian hasilnya akan ditampilkan pada monitor / *mobile application* / *website* / *lcd i2c*.

1.2 Permasalahan

Rasio jumlah dokter jantung dengan jumlah pasien penyakit jantung di Indonesia mencapai 1:250.000 [11] sehingga pasien akan masuk *waiting list* untuk menerima pemeriksaan/penanganan medis sehingga perlunya sistem untuk diagnosa awal mengenai kondisi jantung secara mandiri.

1.3 Tujuan

Membuat perangkat untuk membantu memonitor dan mendiagnosa awal kondisi jantung berdasarkan sinyal ECG secara mandiri dan *portable*.

1.4 Batasan masalah

Batasan masalah pada penelitian Tugas Akhir ini adalah:

1. Penyakit aritmia yang dideteksi yaitu jenis *tachycardia* dan *bradycardia* dengan mengamati jarak R-R pada sinyal ECG.
2. Subjek dalam kondisi relaks (tidak setelah berolahraga atau tidak sedang tidur) 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 uraian secara sistematis teori / buku / *paper* yang berhubungan dengan permasalahan pada penelitian ini. Teori / buku / *paper* ini digunakan sebagai dasar dalam penelitian, yaitu pola sinyal ECG, aritmia, *machine learning* dan *datasheet* setiap komponen penyusunan perangkat.
3. BAB III Perancangan Sistem dan Implementasi Bab ini berisi tentang penjelasan-penjelasan terkait eksperimen yang akan dilakukan dan langkah-langkah data diolah hingga mengha-

silkan visualisasi. Guna mendukung eksperimen pada penelitian ini, digunakanlah blok diagram atau *work flow* agar sistem yang akan dibuat dapat terlihat dan mudah dibaca untuk implementasi pada pelaksanaan tugas akhir.

4. BAB IV Pengujian dan Analisa Bab ini menjelaskan tentang pengujian eksperimen yang dilakukan terhadap data dan analisisnya. Beberapa teknik visualisasi akan ditunjukkan hasilnya pada bab ini dan dilakukan analisa terhadap hasil visualisasi dan informasi yang didapat dari hasil mengamati visualisasi yang tersaji
5. BAB V Penutup
Bab ini merupakan penutup yang berisi kesimpulan yang diambil dari penelitian dan pengujian yang telah dilakukan. Saran dan kritik yang membangun untuk pengembangan lebih lanjut juga dituliskan pada bab ini.

1.6 Relevansi

Penelitian mengenai Aritmia, *Electrocardiography*, *Machine Learning Backpropagation*, Sistem Tertanam.

Halaman ini sengaja dikosongkan

BAB 2

TINJAUAN PUSTAKA

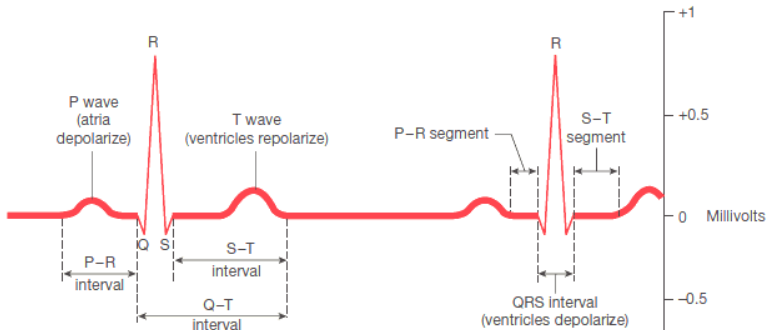
Demi mendukung penelitian ini, dibutuhkan beberapa teori penunjang sebagai bahan acuan dan referensi. Dengan demikian penelitian ini menjadi lebih terarah.

2.1 Dasar Teori

Pada sub bab ini merupakan dasaran teori yang digunakan untuk referensi dalam penelitian diantaranya segmen sinyal jantung, aritmia, *machine learning*, raspberry pi 3 model b+, sparkfun AD8232 *single lead heart monitor*, Intel *Neural Compute Stick 2*, dan modul *analog to digital converter ADS1115*.

2.1.1 Segmen Sinyal Jantung

Segmen sinyal jantung dapat disadap dengan menggunakan elektroda dan digunakan untuk analisa bentuk maupun ritme jantung menggunakan elektrokardiograf. Segmen sinyal jantung dapat diamati pada Gambar 2.1.



Gambar 2.1: Segmen Sinyal Jantung [2]

Sinyal jantung yang terbentuk memiliki 5 segmen yaitu segment P, Q, R, S dan T yang masing - masing unsur memiliki ka-

rakteristik diantaranya:

1. Gelombang P :
 - (a) Bentuk melengkung kecil ke atas
 - (b) Menunjukkan depolarisasi atrium
 - (c) Kelainan Gelombang P = kelainan atrium
2. Interval PR :
 - (a) Jarak antara gelombang P dan permulaan kompleks QRS
 - (b) Untuk mengukur waktu perjalanan depolarisasi dari atrium ke ventrikel
3. Interval QRS :
 - (a) Mengindikasikan depolarisasi (dan kontraksi) ventrikel
 - (b) Gel Q : defleksi negatif pertama setelah P
 - (c) Gel R : defleksi positif pertama setelah P
 - (d) Gel S : defleksi negatif pertama setelah R
4. Segmen ST :
 - (a) Jarak antara gelombang S dan permulaan gelombang T
 - (b) Menunjukkan repolarisasi ventrikel
5. Gelombang T :
 - (a) Gelombang Lengkungan ke atas yang mengikuti QRS
 - (b) Menunjukkan repolarisasi ventrikel
6. Interval QT :
 - (a) Permulaan QRS hingga akhir T
 - (b) Menunjukkan aktivitas ventrikel total
7. Gelombang U :
 - (a) Mudah terlihat pada denyut jantung yang pelan
 - (b) Beberapa referensi menyatakan bahwa gelombang ini menunjukkan repolarisasi serat purkinje

Gelombang sinyal P diawal, bersamaan dengan depolarisasi dari atrium. Atrium mulai berkontraksi sekitar 25 milidetik setelah gelombang P dimulai. Interval QRS *complex*, 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, mengindikasikan 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 [12].

2.1.2 Aritmia

Aritmia adalah kelainan denyut jantung dan irama (kadang-kadang terasa seperti jantung berdebar). Aritmia dapat dibagi menjadi dua kategori besar yaitu denyut jantung cepat dan lambat. Beberapa menyebabkan beberapa atau sedikit gejala. Yang lain menghasilkan gejala yang lebih serius seperti sakit kepala ringan, pusing dan pingsan.

Beberapa pasien dengan jantung yang normal dapat memiliki jalur listrik abnormal jantung yang menyebabkan aritmia. Pasien dengan masalah mendasar pada fungsi dan struktur jantung lebih rentan terhadap masalah irama jantung. Sebagai pasien yang telah berhasil menjalani operasi jantung dapat hidup lebih lama, dokter mendiagnosis lebih banyak kelainan irama jantung

Kelainan irama jantung dievaluasi dengan cara yang sama seperti yang digunakan untuk mengevaluasi masalah kesehatan lainnya. Sejarah gejala Anda, termasuk sensasi jantung berdetak cepat, pusing dan pingsan sangat penting.

Dokter Anda dapat menggunakan beberapa tes untuk mendiagnosis penyakit aritmia. Yang pertama biasanya sebuah elektrokardiogram (EKG atau EKG). Mesin ECG merekam aktivitas listrik jantung Anda. Penelusuran dapat direkam di atas kertas atau penyimpanan komputer. Namun, EKG hanyalah potret singkat irama jantung Anda dan mungkin tidak mendeteksi aritmia yang sebenarnya. Tes yang digunakan untuk mendiagnosis aritmia meliputi pengujian olahraga, pemantauan Holter, pencatat kejadian, dan studi elektrofisiologi [13].

2.1.3 *Machine Learning*

Machine Learning yaitu teknik untuk melakukan inferensi terhadap data dengan pendekatan matematis. Inti machine learning adalah untuk membuat model (matematis) yang merefleksikan pola-pola data.

Persoalan pada *machine learning* digunakan untuk mencari tahu populasi dengan cara menyelidiki fitur (*features* atau sifat-sifat) yang dimiliki sampel. Kemudian, menginferensi aksi yang harus dilakukan terhadap *unobserved* data berdasarkan kecocokan fitur-fitur

unobserved data dengan model/aturan yang sudah ada.

Secara intuitif, *machine learning* mirip seperti saat manusia belajar. Awalnya membuat banyak kesalahan, tetapi mengetahui/diberi tahu mana yang benar sehingga akan menyesuaikan diri secara perlahan agar menjadi benar (iteratif). Inilah yang juga dilakukan *machine learning*, yaitu mengubah-ubah parameter w (*weight*) untuk mengoptimalkan suatu fungsi tujuan [14].

Sub bab *machine learning* memiliki banyak sekali topik bahasan, oleh karena itu bahasan tersebut akan dijelaskan satu persatu mulai dari klasifikasi algoritma *machine learning* berdasarkan metode pembelajaran, klasifikasi dataset, *weight* dan bias, *activation function*, arsitektur *neural network*, *loss function*, sampai dengan bahasan *optimizers*.

2.1.3.1 Algoritma *Machine Learning* Berdasarkan Metode Pembelajaran

Dari sisi metode pembelajaran, algoritma *machine learning* dapat dikategorikan sebagai *supervised learning*, *semi-supervised learning*, *unsupervised learning* [14].

1. *Supervised Learning*

Jika diterjemahkan secara literal, *supervised learning* adalah pembelajaran terarah/terawasi. Artinya, pada pembelajaran ini, ada guru yang mengajar (mengarahkan) dan siswa yang diajar. Tujuan *supervised learning*, secara umum untuk melakukan klasifikasi.

2. *Semi-supervised Learning*

Semi-supervised learning mirip dengan *supervised learning*, bedanya pada proses pelabelan data. Pada *supervised learning*, ada "guru" yang harus membuat "kunci jawaban" *input-output*. Sedangkan pada *semi-supervised learning* tidak ada "kunci jawaban" eksplisit yang harus dibuat guru. Kunci jawaban ini dapat diperoleh secara otomatis (misal dari hasil *clustering*).

Pada kategori pembelajaran ini, umumnya hanya memiliki sedikit data, kemudian menciptakan data tambahan baik menggunakan *supervised* ataupun *unsupervised learning*, kemudian membuat model belajar dari data tambahan tersebut.

3. *Unsupervised Learning*

Jika pada *supervised learning* ada guru yang mengajar, maka pada *supervised learning* tidak ada guru yang mengajar. Contoh permasalahan *supervised learning* adalah *clustering*.

Clustering adalah membuat kelompok-kelompok berdasarkan karakteristik kue, misal kelompok kue biru, kelompok kue kuning, atau kelompok kue merah. Contoh algoritma *supervised learning* sederhana adalah *K-means*.

2.1.3.2 Dataset

Terdapat dua istilah penting dalam pembangunan model *machine learning* yaitu: *training* dan *testing*. *Training* adalah proses membangun model dan *testing* adalah proses menguji kinerja model pembelajaran. Dataset adalah kumpulan data (sampel dalam statistik). Sampel ini adalah data yang kita gunakan untuk membuat model maupun mengevaluasi model *machine learning*.

Umumnya, dataset dibagi menjadi tiga jenis yang tidak beririsan diantaranya *training set*, *development set* / *validation set*, dan *testing set* [14].

1. *Training set*

Training set adalah himpunan data yang digunakan untuk melatih atau membangun model. Pada buku ini, istilah *training data(set)* mengacu pada *training set*.

2. *Development set* / *Validation set*

Development set atau *validation set* adalah himpunan data yang digunakan untuk mengoptimisasi saat melatih model. Model dilatih menggunakan *training set* dan pada umumnya

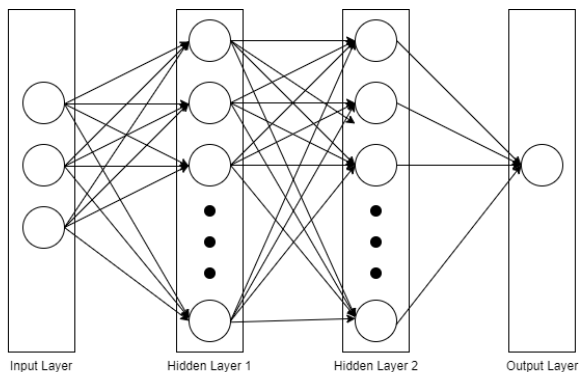
kinerja saat latihan diuji dengan *development set*. Hal ini berguna untuk generalisasi (agar model mampu mengenali pola secara generik).

3. *Testing set*

Testing set adalah himpunan data yang digunakan untuk menguji model setelah proses latihan selesai. Pada buku ini, istilah *testing data(set)* atau *test set* mengacu pada *testing set*. *Testing set* adalah *unseen data*. Artinya, model dan manusia tidak boleh melihat sampel ini saat proses latihan.

2.1.3.3 *Arsitektur Neural Network*

Arsitektur *neural network* adalah kerangka yang tersusun dari beberapa *layer neural network* dimana tiap *layer* memiliki *node* atau *neuron* dan *weight* seperti yang ditunjukkan pada Gambar 2.2.



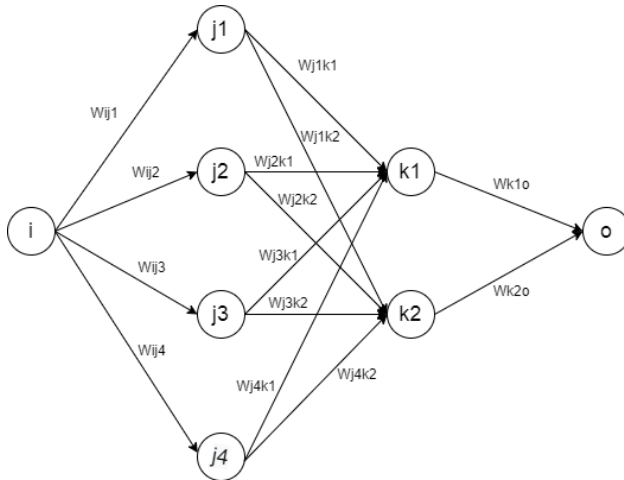
Gambar 2.2: *Arsitektur Neural Network*

Arsitektur *neural network* berdasarkan pada Gambar 2.2 memiliki 3 buah *neuron* pada *input layer*, dan 1 buah *neuron* pada *output layer*. Diantara *input layer* dan *output layer*, terdapat 2 *hidden layer* yang masing-masing memiliki beberapa *neuron*.

2.1.3.4 *Weight* dan Bias

Kinerja *machine learning* berubah-ubah sesuai dengan parameter w (*weight*). Kinerja *machine learning* diukur oleh fungsi tujuan (*utility function/performance measure*), yaitu mengoptimalkan nilai fungsi tertentu [14].

Setiap neuron pada arsitektur saling berhubungan yang ditandai dengan tanda panah pada Gambar 2.3 dimana tiap koneksi memiliki *weight* yang nantinya nilai dari tiap *weight* akan berbeda-beda. *hidden layer* dan *output layer* memiliki tambahan *input* yang biasa disebut dengan bias (Tidak disebutkan pada gambar tersebut) [3].



Gambar 2.3: Arsitektur *Neural Network* dan *Weight* [3]

2.1.3.5 *Activation Function*

Activation function berfungsi untuk menentukan apakah neuron tersebut harus “aktif” atau tidak berdasarkan dari *weighted sum* dari *input* [3]. *Activation function* ada berbagai macam diantaranya [15] :

1. CReLU (*Concatenated Rectified Linear Unit*)
2. ELU (*Exponential Linear Unit*)
3. GELU (*Gaussian Error Linear Unit*)
4. LReLU (*Leaky Rectified Linear Unit*)
5. Maxout
6. PReLU (*Parametric Rectified Linear Unit*)
7. ReLU (*Rectified Linear Unit*)
8. SELU (*Scaled Exponential Linear Unit*)
9. Sigmoid
10. Softmax
11. Softplus
12. Tanh.

2.1.3.6 Loss Function

Loss function merupakan fungsi yang bertugas untuk mengukur tingkat *error* dalam *machine learning*. *Loss function* dapat diklasifikasikan menjadi beberapa bagian berdasarkan kebutuhan *machine learning* diantaranya [16] :

1. *Classification*

Loss function untuk kebutuhan *classification* diantaranya:

- (a) *Cross Entropy Loss*
- (b) *Hinge Loss*
- (c) *Focal Loss*
- (d) *Noise Contrastive Estimation*

2. *Embeddings*

Loss function untuk kebutuhan *embeddings* diantaranya:

- (a) *Contrastive Loss*
- (b) *Negative Sampling*
- (c) *Noise Contrastive Estimation*

3. *Triplet Loss*

4. Regression

Loss function untuk kebutuhan *regression* diantaranya:

- (a) *Huber Loss*
- (b) *Squared Loss*

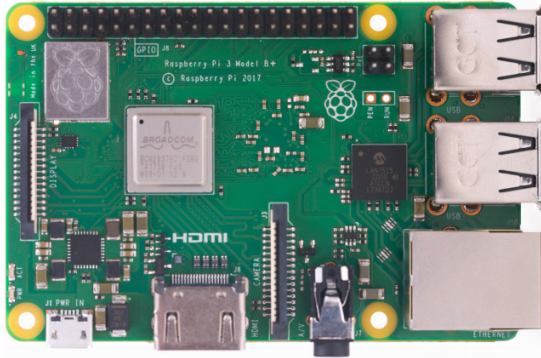
2.1.3.7 Optimizers

Optimizers merupakan fungsi yang bertugas untuk meminimalkan nilai *error* yang telah diukur menggunakan *loss function*. *Optimizers* ada berbagai macam diantaranya :

1. AdaDelta
2. AdaGrad
3. Adam
4. Averaged SGD (ASGD)
5. BFGS
6. *Conjugate Gradient*
7. *Coordinate Descent*
8. *Krylov Subspace Descent*
9. *Natural Gradient*
10. *Newton's Method*
11. *Nesterov's Method*
12. RMSProp
13. *Subgradient Method*

2.1.4 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. Visualisasi dari Raspberry Pi 3 Model B+ ditunjukkan pada Gambar 2.4.



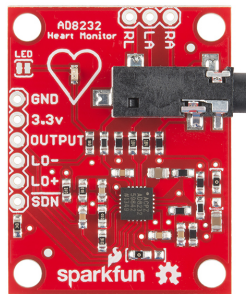
Gambar 2.4: Raspberry Pi 3 Model B+ [4]

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 [4].

2.1.5 Sparkfun AD8232 *Single Lead Heart Monitor*

Modul AD8232 adalah sebuah modul sensor yang digunakan untuk mengakuisisi sinyal ECG dengan *pin input / output* serta catu daya nya tersambung pada *microcontroller* ataupun *single board computer*. Visualisasi dari sparkfun AD8232 *single lead heart monitor* ditunjukkan pada Gambar 2.5.



Gambar 2.5: Modul Sparkfun ECG AD8232 [5]

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 *Analog 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 pemfilteran 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 se-

mua filter untuk disesuaikan dengan berbagai jenis aplikasi. Untuk meningkatkan penolakan mode-umum dari frekuensi saluran dalam sistem dan gangguan lain yang tidak diinginkan, AD8232 menyertakan penguat untuk aplikasi lead yang digerakkan, seperti *right leg drive* (RLD).

AD8232 mencakup fungsi pemulihan cepat yang mengurangi durasi tail settling high-pass yang lama. Setelah perubahan sinyal tiba-tiba yang menghubungkan amplifier (seperti kondisi lead off), AD8232 secara otomatis menyesuaikan ke cutoff filter yang lebih tinggi. Fitur ini memungkinkan AD8232 untuk pulih dengan cepat, dan karenanya, untuk melakukan pengukuran yang valid segera setelah menghubungkan elektroda ke subjek.

AD8232 tersedia dalam ukuran 4 mm × 4 mm, paket LFCSP 20 lead. Kinerja ditentukan dari 0°C hingga 70°C dan beroperasi dari -40°C hingga +85°C [5].

2.1.6 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.

Pada Intel Movidius NCS 2 dapat mengembangkan, menyempurnakan, dan menggunakan *convolutional neural networks* (CNNs) pada aplikasi berdaya rendah yang memerlukan kesimpulan waktu nyata.

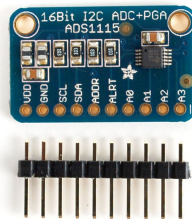
Dapat dilihat melalui Gambar 2.6 bahwa konektivitas nya menggunakan USB 3.0 *Type-A*, dan ukuran dimensi 72.5 mm x 27 mm x 14 mm [6].



Gambar 2.6: Intel Neural Compute Stick 2 [6]

2.1.7 ADS1115

ADS1115 adalah konverter analog ke digital presisi dengan resolusi 16 bit yang ditawarkan dalam paket QFN-10 ultra 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. Visualisasi dari modul ADS1115 ditunjukkan pada Gambar 2.7.



Gambar 2.7: ADS1115

ADS1115 dapat melakukan konversi dengan laju hingga 860 sampel per detik. PGA onboard tersedia pada ADS1115 yang menawarkan rentang input dari pasokan hingga serendah $/pm$ 256mV,

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 mode sekali pakai yang secara otomatis mati setelah konversi dan sangat mengurangi konsumsi saat ini selama periode idle. Suhu pengoperasian ADS1115 dalam rentang nilai -40°C hingga $+125^{\circ}\text{C}$ [17].

2.2 Penelitian Terkait

Pada sub bab ini merupakan beberapa penelitian yang digunakan untuk referensi dalam sistem diantaranya *ECG signal analysis using wavelet transforms*, perangkat *treadmill* untuk mendeteksi kelainan detak jantung menggunakan sensor ECG berbasis *single board computer*, *IOT based detection of cardiac arrhythmias using Raspberry PI*, *the ECG signal prediction by using neural network*, *an approach to detect QRS complex using backpropagation neural network*, dan *backpropagation and his application in ECG classification*.

2.2.1 IOT Based Detection of Cardiac Arrhythmias Using Raspberry PI

Pada penelitian *IOT based detection of cardiac arrhythmias using Raspberry PI* menghasilkan sebuah perangkat yang melakukan pemrosesan bagian QRS dengan bantuan perangkat lunak MATLAB yang kemudian hasil pemrosesan akan disimpan pada database MySQL sehingga dapat diakses dari jarak jauh dan bisa kapan saja [18].

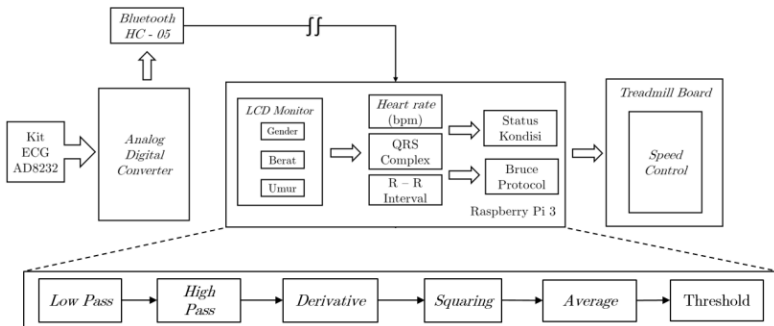
2.2.2 Perangkat *Treadmill* untuk Mendeteksi Kelainan Detak Jantung Menggunakan Sensor ECG Berbasis *Single Board Computer*

Pada penelitian perangkat *treadmill* untuk mendeteksi kelainan detak jantung menggunakan sensor ECG berbasis *single board computer* menghasilkan sebuah sistem perangkat *treadmill* yang bertujuan untuk memantau kondisi kelainan jantung khususnya *tachycardia* dan *bradycardia* serta dapat mendeteksi adanya estima-

si intensitas *Volume Maximal Oxygen* pada saat berolahraga diatas *treadmill*, sehingga perangkat ini dapat digunakan pasien untuk memantau dan melakukan terapi secara mandiri bagi penderita kelainan jantung. Protokol Bruce di implementasikan pada perangkat untuk fungsi tambahan perangkat *treadmill* sebagai parameter mengendalikan kecepatan dan waktu berolah raga pada *treadmill*.

Alur kerja dari perangkat ini mulai dari akuisisi sinyal ECG menggunakan modul ECG AD8232, kemudian sinyal tersebut akan dikirimkan melalui Bluetooth ke Raspberry Pi untuk mendapatkan *heart rate* berdasarkan pola *QRS Complex* dan *R - R Interval* yang dikomputasi sesuai dengan algoritma Pan Tompkins (algoritma yang terdiri dari *filter low pass filter*, *high pass*, *derivative*, *squaring*, *average*, dan *threshold*).

Pada Gambar 2.8 menunjukkan blok diagram dari perangkat *treadmill* dan pada blok diagram tersebut menunjukkan beberapa komponen yang digunakan yaitu *treadmill board*, Raspberry Pi 3, modul Bluetooth HC-05, LCD *monitor* dan modul sensor ECG AD8232 [7].



Gambar 2.8: Blok Diagram Perangkat *Treadmill* [7]

2.2.3 ECG Signal Analysis Using Wavelet Transforms

Pada penelitian *ECG signal analysis using wavelet transforms* bahwa irama sinus normal (keadaan jantung normal) interval P-R berada dalam kisaran 0,12 hingga 0,2 detik. Interval QRS adalah dari 0,04 hingga 0,12 detik. Interval Q-T kurang dari 0,42 detik dan denyut jantung normal dari 60 hingga 100 denyut per menit

Elektrokardiogram adalah perekaman grafik atau tampilan tegangan varian waktu yang dihasilkan oleh miokardium selama siklus jantung. Gelombang P, QRS, dan T merefleksikan depolarisasi listrik ritmik dan repolarisasi miokardium yang terkait dengan kontraksi atrium dan ventrikel. ECG ini digunakan secara klinis dalam mendiagnosis berbagai kelainan dan kondisi yang terkait dengan jantung. Tabel 2.1 merupakan data kondisi mengenai segmen jantung yang normal.

Tabel 2.1: Kondisi Segmen Jantung Normal [10]

Amplitudo	Gelombang P	0.25 mV
	Gelombang R	1.60 mV
	Gelombang Q	25% R wave
	Gelombang T	0.1 to 0.5 mV
Durasi	P-R interval	0.12 to 0.20 s
	Q-T interval	0.35 to 0.44 s
	S-T interval	0.05 to 0.15 s
	Gelombang P interval	0.11 s
	QRS interval	0.09 s

Nilai normal dari detak jantung terletak pada kisaran 60 hingga 100 detak / menit. Tingkat yang lebih lambat dari ini disebut bradikardia (jantung lambat) dan tingkat yang lebih tinggi disebut takikardia (jantung cepat). Jika siklusnya tidak merata, aritmia dapat diindikasikan. Jika interval P-R lebih besar dari 0,2 detik, mungkin menyarankan penyumbatan node AV.

Kelainan ritme jantung memiliki karakteristik masing-masing yang berkaitan dengan segmen sinyal jantung tersebut seperti yang ditunjukkan pada Tabel 2.2.

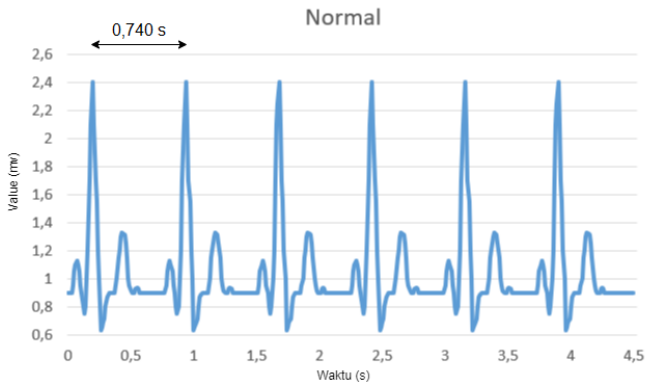
Tabel 2.2: Karakter Beberapa Kelainan Jantung [10]

No	Name of Abnormality	Characteristic Features
1	Dextrocardia	Inverted P-wave
2	Tachycardia	R-R interval <0.6 s
3	Bradycardia	R-R interval >1 s
4	Hyperkalemia	Tall T-wave and absence of P-wave
5	Myocardial Ischaemia	Inverted T-wave
6	Hypercalcaemia	QRS interval <0.1 s

Pada penelitian *ECG signal analysis using wavelet transforms* menyatakan bahwa irama sinus normal (keadaan jantung normal dalam keadaan relaks) berdasarkan *RR-interval* yang memiliki nilai dalam *range* waktu 600 *milisecond* sampai dengan 1000 *milisecond*.

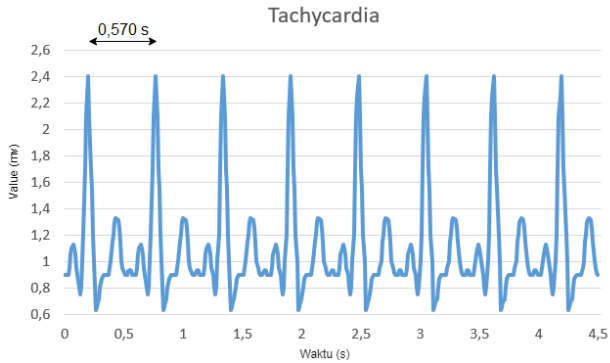
Manfaat transformasi wavelet terletak pada kapasitasnya untuk menyoroti detail sinyal ECG dengan resolusi frekuensi waktu yang optimal.

Pada Gambar 2.9 merupakan contoh sinyal ECG yang masuk dalam kategori normal berdasarkan *RR interval* dimana *RR interval* yang dimiliki masih dalam *range* normal.



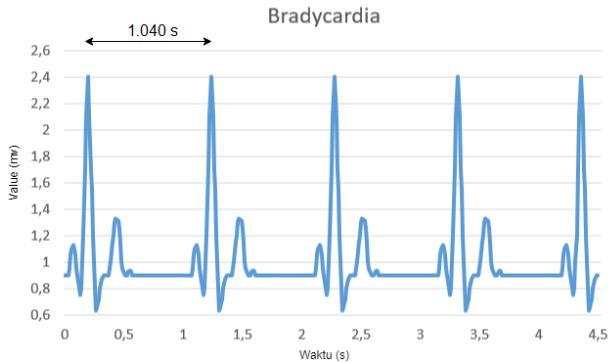
Gambar 2.9: Sinyal ECG Normal

Pada Gambar 2.10 merupakan contoh sinyal ECG tachycardia dimana RR interval bernilai kurang dari 600 ms.



Gambar 2.10: Sinyal ECG Tachycardia

Pada Gambar 2.11 merupakan contoh sinyal ECG bradycardia dimana RR interval bernilai lebih dari 1000 ms [10].



Gambar 2.11: Sinyal ECG Bradycardia

2.2.4 An Approach to Detect QRS Complex Using Backpropagation Neural Network

Pada penelitian *an approach to detect QRS complex using backpropagation neural network* mendapatkan hasil akurasi untuk mendeteksi puncak positif yang benar sebesar 91,16% .

Neural network yang digunakan memiliki tiga *layer* diantaranya terdapat satu *input layer*, satu *hidden layer* dan satu *output layer*.

Pada *input layer* memiliki enam *neuron* diantaranya berisi mengenai amplitudo, nilai diferensiasi, durasi melebihi ambang batas, *RR-interval*, *flag zero-crossing* dan *flag* elemen pertama untuk setiap titik yang perlu dinilai jika itu adalah puncak R.

Pada *hidden layer* memiliki tiga belas buah *neuron*, hal ini dipilih berdasarkan pada teorema Kolmogorov. Namun teorema Kolmogorov tidak menentukan apakah jaringan ini merupakan solusi optimal untuk *mapping*. Penentuan jumlah *neuron* di *hidden layer* jika jumlah terlalu besar, jaringan membutuhkan lebih banyak penyimpanan dan pelatihan lebih kompleks. Sedangkan jika jumlahnya terlalu kecil, meskipun jaringan masih bisa melakukan *mapping* yang tepat, mungkin ada yang *over fitting*. *Over fitting* berarti bahwa jaringan tidak dapat membuat generalisasi ketika disajikan dengan *input* yang sedikit berbeda.

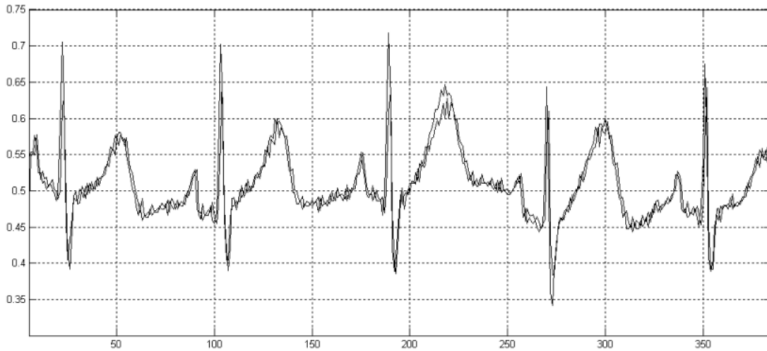
Pada *output layer* memiliki satu *neuron* yang berisi dua varian nilai saja diantaranya bernilai 1 untuk puncak R dan bernilai 0 untuk non puncak R [19].

2.2.5 The ECG Signal Prediction By Using Neural Network

Pada penelitian *the ECG signal prediction by using neural network* menyatakan bahwa rekaman ECG digital berikutnya dapat diprediksi ketika sampel sebelumnya diketahui menggunakan *neural network* .

Pada penelitian ini menggunakan *neural network backpropagation* dengan 3 *layer* diantaranya ada tiga *neuron* pada *input layer*, dua *neuron* pada *hidden layer*, dan satu *neuron* pada *output la-*

yer. Dalam semua *layer* menggunakan fungsi aktivasi log-sigmoid (logsig), serta *learning rate* bernilai 0,05 dan toleransi kesalahan 5×10^{-4} . Dalam hal ini *neural network* mencapai ketepatan yang lebih baik. Hasil yang didapatkan pada penelitian ini ditunjukkan pada Gambar 2.12 [8].



Gambar 2.12: Perbedaan Sinyal Estimasi menggunakan *Backpropagation* dengan Sinyal ECG Asli [8]

2.2.6 *Backpropagation and His Application in ECG Classification*

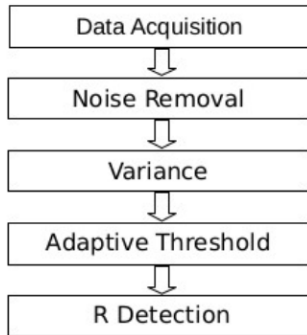
Pada penelitian *backpropagation and his application in ECG classification* bertujuan untuk mengklasifikasikan ECG ke kelas ECG normal dan ECG abnormal dengan menggunakan tiga buah *layer* diantaranya *input layer*, *hidden layer*, dan *output layer*. Fungsi aktivasi yang digunakan diantaranya fungsi aktivasi sigmoid dan fungsi aktivasi logistic. Pada *input layer* terdapat 100 *neuron*, pada *output layer* terdapat satu *neuron*, dan beberapa varian jumlah *neuron* pada *hidden layer* diantaranya 25, 50, 75, dan 100 buah *neuron*. Dari 100 iterasi jaringan diadaptasi dan diklasifikasikan dengan benar dan kesalahan jaringan relatif kecil [20].

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

Penelitian *QVAT: QRS complex detection based on variance analysis and adaptive threshold for electrocardiogram signal* merupakan sebuah algoritma untuk mendeteksi kompleks QRS dalam sinyal EKG

Penelitian ini menguji kekokohan metode yang diusulkan menggunakan 20 catatan saluran MLII dari Database Aritmia MITBIH. Hasil percobaan eksperimen deteksi kompleks QRS, metode yang diusulkan memiliki sensitivitas Se 99,79%

QVAT memiliki beberapa langkah dalam mendeteksi kompleks QRS diantaranya *data acquisition, noise removal, variance, adaptive threshold, R detection* seperti yang ditunjukkan pada Gambar 2.13 [9].



Gambar 2.13: Blok Diagram QVAT [9]

Halaman ini sengaja dikosongkan

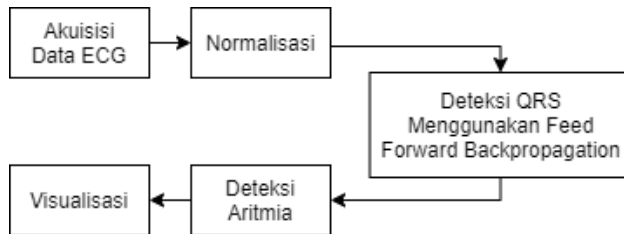
BAB 3

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 data deteksi aritmia diantaranya akuisisi data ECG, normalisasi, deteksi QRS, deteksi aritmia, sampai dengan visualisasi seperti yang ditunjukkan pada Gambar 3.1 sebagai blok diagram kerja sistem yang akan digunakan pada penelitian.



Gambar 3.1: Blok Diagram Kerja Sistem

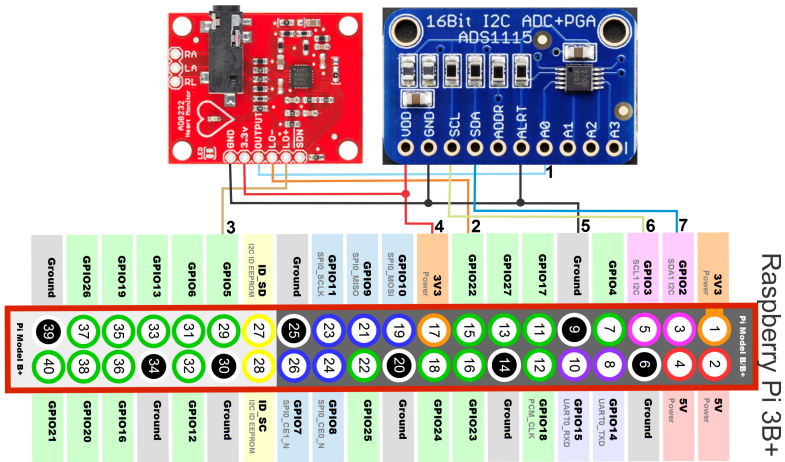
3.2 Desain Perangkat ECG

Perangkat ECG terdiri dari modul sparkfun AD8232, *analog to digital converter* ADS1115 dan Raspberry yang dikonfigurasi sesuai dengan Tabel 3.1.

Tabel 3.1: Konfigurasi Perangkat ECG

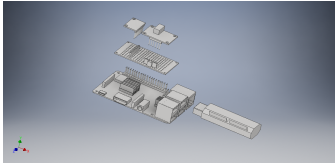
No	AD8232	ADS1115	Raspberry
1	OUTPUT	A0	-
2	LO-	-	GPIO22
3	LO+	-	GPIO5
4	3,3v	VDD	3,3v
5	GND	GND, ALRT	GND
6	-	SCL	SCL
7	-	SDA	SDA

Desain rangkaian perangkat ECG ditunjukkan pada Gambar 3.2 dengan keterangan nomor menyesuaikan dengan tabel konfigurasi tersebut.

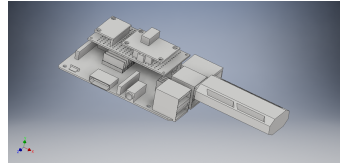


Gambar 3.2: Desain Rangkaian Perangkat ECG

Desain 3 dimensi dari tiap komponen pada perangkat ECG secara terpisah yang digunakan oleh sistem ditunjukkan pada Gambar 3.3 dan desain 3 dimensi saat tiap komponen saat disambungkan ditunjukkan pada Gambar 3.4.

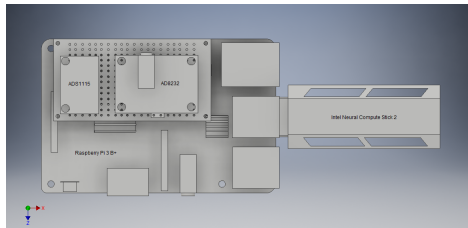


Gambar 3.3: Desain Tiap Komponen Secara Terpisah



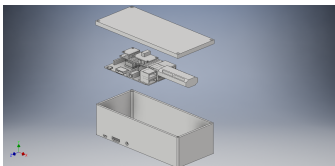
Gambar 3.4: Desain Tiap Komponen Secara Tersambung

Desain 3 dimensi tiap komponen saat disambungkan dari sudut pandang atas ditunjukkan pada Gambar 3.5.

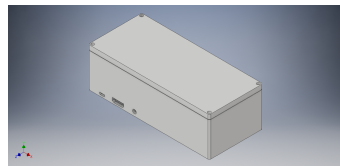


Gambar 3.5: Desain Tiap Komponen Secara Tersambung Tampak Atas

Desain 3 dimensi saat perangkat yang tersambung dimasukkan kedalam *package* ditunjukkan pada Gambar 3.6 dan desain 3 dimensi *package* perangkat dalam kondisi tertutup ditunjukkan pada Gambar 3.7.



Gambar 3.6: Desain Perangkat Dalam *Package*

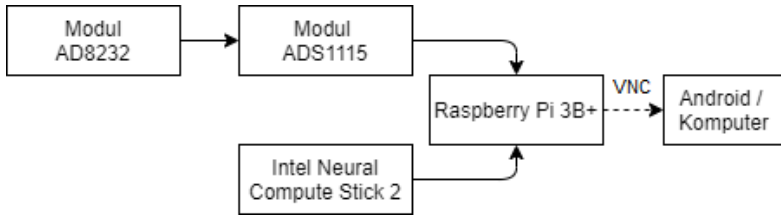


Gambar 3.7: Desain *Package* Dalam Kondisi Tertutup

3.3 Perancangan Perangkat ECG

Dalam perancangan perangkat ECG diperlukan beberapa komponen diantaranya modul sparkfun AD8232 yang saling terkoneksi dengan beberapa *pin* pada ADS1115 dan atau *pin* pada Raspberry, komponen intel *neural compute stick 2* yang tersambung dengan Raspberry melalui *port* USB dan *device* android / komputer yang sudah terdapat aplikasi *VNC Viewer*.

Komponen - komponen tersebut terhubung antara satu sama lain yang ditunjukkan pada Gambar 3.8.



Gambar 3.8: Blok Diagram Hubungan Tiap Komponen

Berikut merupakan fungsi dari masing - masing komponen :

1. Modul AD8232

Komponen modul ad8232 berfungsi untuk akuisisi data ECG pada subjek.

2. Modul ADS1115

Komponen modul ADS1115 berfungsi untuk mengkonversi dari nilai *analog* yang didapat oleh modul AD8232 menjadi nilai *digital*.

3. Intel Neural *Compute Stick 2*

Komponen intel neural *compute stick 2* (INCS 2) berfungsi untuk mengakses *library Keras* yang digunakan pada saat proses deteksi QRS. Tahapan konfigurasi perangkat INCS 2 dapat dilihat pada Lampiran 10.

4. Raspberry Pi 3B+

Komponen Raspberry Pi 3B+ memiliki beberapa fungsi, diantaranya :

- (a) Beberapa *pin* pada Raspberry akan digunakan untuk modul AD8232 dan modul ADS1115
- (b) Menerima dan menyimpan data dari hasil konversi oleh ADS1115
- (c) Mengolah data untuk proses normalisasi
- (d) Mengolah data untuk proses deteksi QRS dengan *library Keras* yang di *support* oleh komponen NCS
- (e) Mengolah data untuk proses deteksi aritmia
- (f) Membuat *user interface* untuk menampilkan hasil deteksi
- (g) Sebagai VNC *server* agar tampilan pada Raspberry dapat dilihat melalui android atau komputer

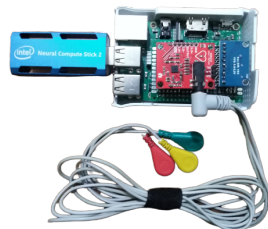
5. Android / Komputer

Device ini digunakan untuk monitor dari tampilan *user interface* pada Raspberry Pi menggunakan VNC *viewer*.

Perancangan perangkat ECG ditunjukkan pada Gambar 3.9, Gambar 3.10, Gambar 3.11, dan Gambar 3.12.



Gambar 3.9: Perangkat Tampak Atas



Gambar 3.10: Perangkat Tampak Atas Tanpa Tutup



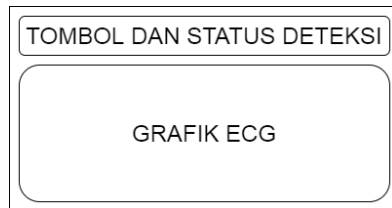
Gambar 3.11: Perangkat Tampak Samping Atas



Gambar 3.12: Perangkat Tampak Samping

3.4 Desain *User Interface*

User interface digunakan untuk menampilkan data ECG dalam bentuk grafik, serta status dan hasil deteksi dari perangkat. Desain *user interface* yang digunakan pada sistem ditunjukkan pada Gambar 3.13.

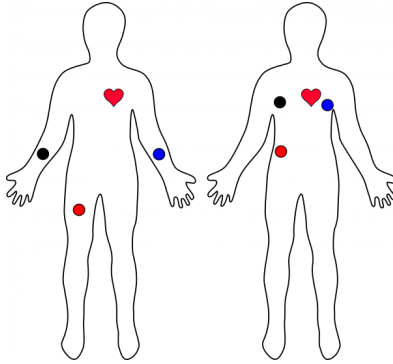


Gambar 3.13: Desain *User Interface*

3.5 Akuisisi Data ECG

Akuisisi data ECG dilakukan dengan frekuensi 100Hz, *baud rate* 9600 dan menggunakan catu daya dari *power bank*. Modul sparkfun AD8232 menggunakan 3 elektroda ECG yang ditempelkan ke badan seseorang.

Untuk penempatan elektroda ECG tidak bisa pada semua bagian badan, penempatan elektroda ECG memiliki standar diantaranya seperti yang ditunjukkan pada Gambar 3.14. Pada gambar tersebut terdapat 2 pilihan untuk penempatan elektroda ECG.



Gambar 3.14: Penempatan Posisi Elektroda ECG

3.6 Normalisasi

Normalisasi digunakan untuk 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) dan Persamaan (3.2) agar hasil normalisasi dapat menjadi data *input* untuk *neural network* saat proses deteksi QRS.

$$rentang = |(min(data) - max(data))| \quad (3.1)$$

Pada Persamaan (3.1) didapatkan nilai *rentang* yang merupakan jangkauan data ECG dengan melalui komputasi antara $min(data)$ yang merupakan nilai terkecil dari data ECG dan $max(data)$ yang merupakan nilai terbesar dari data ECG.

$$norm[i] = (data[i]/rentang) - (max(data)/rentang) + 0.5 \quad (3.2)$$

Pada Persamaan (3.2) didapatkan *array norm[i]* yang merupakan *array* untuk menyimpan data normalisasi pada indeks ke i .

3.7 Deteksi QRS Menggunakan *Feed Forward Backpropagation*

Deteksi QRS menggunakan *feed forward backpropagation* membutuhkan beberapa data diantaranya data arsitektur *neural network*, model serta *weight* yang didapatkan dari *training* data menggunakan algoritma *backpropagation*, dan data ECG yang berasal dari tahap normalisasi.

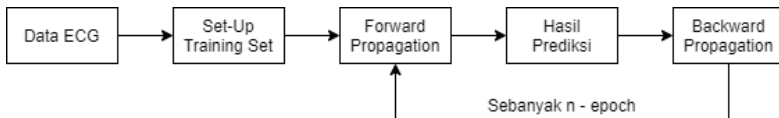
Proses ini membutuhkan *library Keras* untuk melakukan *machine learning* baik saat proses *training* maupun saat menggunakan algoritma *feed forward backpropagation*. Raspberry Pi 3B+ tidak *support* untuk mengakses *library* tersebut, sehingga membutuhkan komponen tambahan yaitu intel *neural compute stick 2* yang dihubungkan dengan Raspberry Pi 3B+ melalui *port* USB.

Output dari proses ini akan dijadikan sebagai syarat dan dilakukan komputasi sesuai dengan Persamaan (3.3).

$$\begin{cases} QRS[i] = norm[i] & , if \text{ predict } \geq 0.8 \\ QRS[i] = 0 & , otherwise \end{cases} \quad (3.3)$$

Pada Persamaan (3.3) didapatkan *array QRS[i]* yang merupakan *array* untuk menyimpan data QRS pada indeks ke *i* dan bergantung dengan nilai *predict* yang merupakan nilai prediksi QRS dari hasil *training* (yang bernilai antara 0 sampai dengan 1).

Training data dilakukan dalam beberapa tahapan yang ditunjukkan pada Gambar 3.15 diantaranya menyiapkan data ECG, *set-up training set*, *forward propagation*, hasil prediksi, dan *backward propagation*.



Gambar 3.15: Blok Diagram *Training* Data

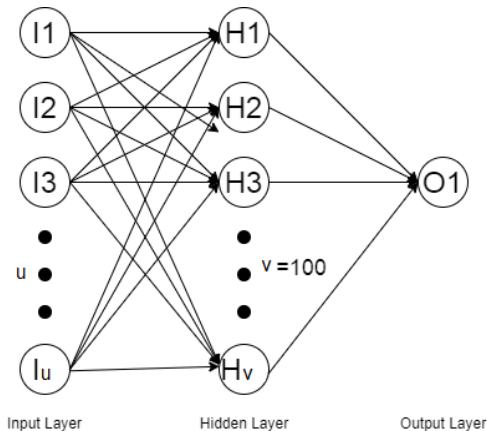
3.7.1 Data ECG

Data ECG yang digunakan untuk *training* didapatkan dari akuisisi data ECG oleh perangkat dengan subjek berisi sebanyak 3600 data ECG dengan frekuensi akuisisi 100Hz.

3.7.2 Set-Up Training Set

Set-up training set merupakan tahap untuk mempersiapkan data-data yang dibutuhkan dalam *training* data seperti arsitektur *neural network*, model, dan data label.

Pada Gambar 3.16 menunjukkan arsitektur *neural network* yang akan digunakan pada proses *training*, dimana I merupakan *neuron* atau *node* yang berada pada *input layer*, u merupakan banyaknya *neuron* atau *node* yang ada di *input layer* atau yang selanjutnya disebut dengan *window*, H merupakan *neuron* atau *node* yang berada di *hidden layer*, v merupakan banyaknya *neuron* atau *node* yang ada di *hidden layer*, dan O merupakan *neuron* atau *node* yang berada di *output layer*.



Gambar 3.16: Arsitektur *Neural Network*

Jumlah *node* pada *input layer* yang digunakan pada sistem

yaitu sepanjang *window* dari tahap *windowing* data (akan dijelaskan pada sub bab berikutnya).

Model yang digunakan pada *training* terdapat beberapa informasi diantaranya *dense* dan parameter. *Dense* adalah kedalaman *layer*, hal ini menyesuaikan dengan banyaknya fungsi aktivasi yang digunakan pada proses *training*. Sedangkan parameter pada model *training* berisi nilai - nilai *weight* yang dihasilkan dari proses *training*.

Data label mewakili apakah himpunan data tertentu termasuk QRS atau bukan, sehingga data label hanya berisi 2 varian angka yaitu angka 0 (bukan QRS) atau angka 1 (QRS).

1. Identifikasi QRS

Identifikasi QRS berfungsi untuk mengidentifikasi data ECG yang dikenali sebagai QRS dengan cara melihat data ECG secara manual dan memberikan identitas (1 atau 0) sesuai dengan Persamaan (3.4).

$$lab_{qrs}[i] = \begin{cases} 1 & , \text{if } data[i] \in QRS_{real} \\ 0 & , \text{Otherwise} \end{cases} \quad (3.4)$$

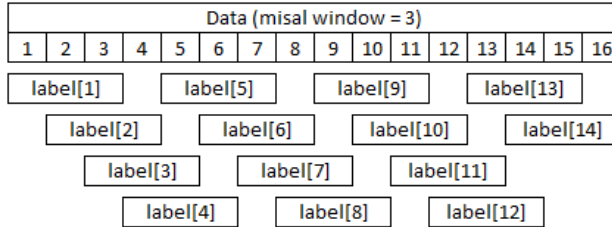
Pada Persamaan (3.4) didapatkan *array* $lab_{qrs}[i]$ yang merupakan *array* untuk menyimpan data identifikasi QRS pada indeks ke i dan QRS_{real} merupakan kumpulan data yang termasuk dalam satu kompleks QRS.

Panjang data dalam 1 kompleks QRS yaitu sebesar 9% dari frekuensi akuisisi. Data ECG yang digunakan berjumlah sangat banyak sehingga perlu adanya cara untuk mengkerucutkan data ECG yang merupakan QRS diantaranya mengetahui indeks data yang merupakan *R-peak*.

2. *Windowing* Data

Hasil yang didapatkan dari tahap sebelumnya akan di pecah-pecah menjadi beberapa bagian dimana setiap bagian akan

terjadi *overlap* / saling irisan dengan bagian yang lain seperti ilustrasi yang ditunjukkan pada Gambar 3.17.



Gambar 3.17: Ilustrasi *Windowing* Data

Panjang *window* yang digunakan pada sistem sebesar setengah dari panjang 1 kompleks QRS atau sebesar 4,5% dari frekuensi sesuai dengan Persamaan (3.5).

$$w = \lceil 0.045 X \text{ freq} \rceil \quad (3.5)$$

Pada Persamaan (3.5) didapatkan nilai w yang merupakan panjang *window* dan bergantung dengan nilai freq yang merupakan frekuensi akuisisi.

Banyaknya bagian yang didapatkan sesuai dengan Persamaan (3.6), hal ini dapat mempengaruhi banyaknya data label yang akan digunakan pada saat *training*.

$$n_{\text{label}} = n_{\text{data}} - w + 1 \quad (3.6)$$

Pada Persamaan (3.6) didapatkan n_{label} yang merupakan banyaknya bagian hasil dari *windowing* yang bergantung dari panjang *window* dan nilai n_{data} yang merupakan banyaknya data ECG.

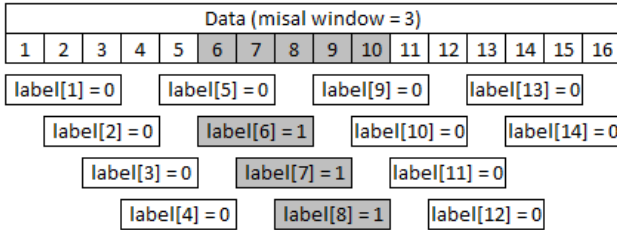
3. Labelling Data

Setiap hasil *windowing* data akan dilakukan *labelling* data atau pemberian label pada data. Konsep yang dijalankan untuk *labelling* data pada sistem ini menggunakan konsep klasifikasi atau hanya memberikan 2 pilihan (misal 1 dengan 0, iya dengan tidak) sesuai dengan Persamaan (3.7).

$$\begin{cases} \text{label}[i] = 1 & , W[i] \subset QRS_{real} \\ \text{label}[i] = 0 & , \text{Otherwise} \end{cases} \quad (3.7)$$

Pada Persamaan (3.7) menunjukkan bahwa varian nilai yang dimiliki saat pemberian label ada 2 varian diantaranya label bernilai 0 untuk bagian yang tidak termasuk dalam kompleks QRS dan label bernilai 1 untuk bagian yang termasuk dalam kompleks QRS yang disimpan pada *array label* dan bergantung dari *array W* yang merupakan *array* penyimpanan *window* dari hasil *windowing* dan QRS_{real} merupakan kumpulan data yang termasuk dalam satu kompleks QRS.

Ilustrasi proses *labelling* data ditunjukkan pada Gambar 3.18.



Gambar 3.18: Ilustrasi Proses Pemberian Label

Berdasarkan ilustrasi Gambar 3.18 menunjukkan bahwa pemberian label ini akan menyebabkan *range* indeks data dari deteksi kompleks QRS lebih sedikit dari kompleks QRS yang sebenarnya, hal ini dikarenakan label akan bernilai 1 jika semua

data dalam *window* tersebut termasuk dalam kompleks QRS dari data.

3.7.3 *Forward Propagation*

Forward propagation atau yang biasa disebut *forward pass* adalah proses komputasi data dari *input layer* yang melewati tiap neuron pada *hidden layer* sampai kepada *output layer* dan mendapatkan hasil prediksi dari proses *training*.

Fungsi aktivasi yang digunakan pada proses *Forward Propagation* diantaranya *rectified linear unit* (ReLU) untuk *hidden layer* dan Sigmoid untuk *output layer* .

3.7.4 Hasil Prediksi

Hasil yang didapatkan pada proses *forward propagation* memiliki nilai dengan range angka antara 0 sampai dengan 1. Hal ini menandakan apabila hasil yang didapat mendekati angka 0 maka data tersebut bukan termasuk dalam wilayah kompleks QRS dan sebaliknya apabila hasil yang didapat dari proses ini mendekati angka 1 maka data tersebut kemungkinan merupakan kompleks QRS.

Prediksi terkadang tidak sesuai dengan target yang diinginkan, hal ini dikarenakan *weight* pada saat *training* memiliki nilai yang tidak optimal. *Weight* dapat diperbaharui dalam tahap *backward propagation*.

3.7.5 *Backward Propagation*

Backward propagation adalah proses untuk memperbaharui nilai *weight* dan akan terus dilakukan sampai berulang-ulang sebanyak *n epoch* atau sampai dengan mendapatkan *error* sekecil mungkin. Perhitungan *error* menggunakan *loss function*.

Epoch yang digunakan pada sistem sebanyak 100 kali. Banyaknya *epoch* ditentukan berdasarkan percobaan dengan mempertimbangkan *loss* bernilai kecil dan *accuracy* bernilai besar.

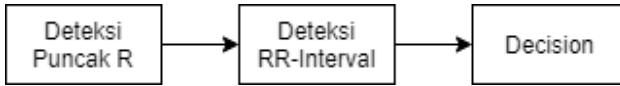
Loss function yang digunakan pada sistem adalah *loss function binary cross-entropy* dimana *loss function* tersebut biasa digunakan untuk klasifikasi 2 kelas.

Nilai *error* yang didapatkan dapat diminimalisir menggunakan *optimizer*. *Optimizer* yang akan digunakan pada sistem adalah

Optimizer Adam.

3.8 Deteksi Aritmia

Deteksi aritmia dilakukan dengan melalui beberapa tahapan yang ditunjukkan pada Gambar 3.19, tahapan tersebut diantaranya deteksi puncak R, deteksi *RR-interval*, dan *Decision*.



Gambar 3.19: Blok Diagram Deteksi Aritmia

3.8.1 Deteksi Puncak R

Deteksi puncak R menggunakan data dari deteksi QRS dengan mencari nilai terbesar dari tiap kompleks QRS yang memenuhi kondisi pada Persamaan (3.8) maka indeks data disimpan pada suatu variabel sesuai dengan Persamaan (3.9).

$$if(max(QRS[i - w : i + w]) == QRS[i]) \quad (3.8)$$

$$then r_{temp} = QRS[i] \quad (3.9)$$

Pada Persamaan (3.8) dan Persamaan (3.9) didapatkan nilai r_{temp} yang merupakan penyimpanan sementara nilai puncak R dimana bergantung dengan nilai $max(QRS[i - w : i + w])$ yang merupakan nilai maksimal lokal dari data QRS mulai dari indeks ke $i - w$ sampai dengan indeks ke $i + w$.

Data yang didapatkan terkadang mendeteksi puncak dengan karakter yang mirip dengan puncak R sehingga perlu dilakukan komputasi lagi yang memenuhi kondisi pada Persamaan (3.10) atau

Persamaan (3.11).

$$(|r_{temp}| - |QRS[i - 1]|) > std \quad (3.10)$$

Pada Persamaan (3.10) merupakan kondisi pertama yang digunakan untuk mendeteksi puncak R dimana nilai std merupakan standar deviasi dari data ECG normalisasi.

$$(|r_{temp}| - |QRS[i - 2]|) > std \quad (3.11)$$

Pada Persamaan (3.11) merupakan kondisi kedua yang digunakan untuk mendeteksi puncak R.

Jika memenuhi salah satu dari kedua kondisi tersebut, maka indeks data akan disimpan pada suatu *array* sesuai dengan Persamaan (3.12).

$$r[j] = i \quad (3.12)$$

Pada Persamaan (3.12) didapatkan *array* $r[j]$ yang merupakan *array* penyimpanan indeks data ECG yang merupakan puncak R pada indeks ke j .

3.8.2 Deteksi *RR-Interval*

RR-Interval adalah waktu yang dibutuhkan dari *R-peak* ke j sampai *R-peak* ke $j + 1$ sesuai dengan Persamaan (3.13).

$$rr[j] = waktu_{ecg}[r[j + 1]] - waktu_{ecg}[r[j]] \quad (3.13)$$

Pada Persamaan (3.13) didapatkan *array* rr yang merupakan *array* penyimpanan *RR interval* dan bergantung pada *array* $waktu_{ecg}$ yang merupakan *array* penyimpanan waktu saat akuisisi data ECG.

3.8.3 Decision

Decision merupakan tahapan untuk menentukan data ECG termasuk aritmia atau bukan. Aritmia yang akan dideteksi dalam penelitian diantaranya jenis aritmia yang disebabkan oleh *RR-interval* yang tidak normal.

Berdasarkan penelitian *ECG Signal Analysis Using Wavelet Transforms* bahwa *RR-interval* bernilai normal jika memiliki nilai antara 600 *milisecond* sampai dengan 1 *second* [10], sehingga secara matematis ditunjukkan pada Persamaan (3.14).

$$\begin{cases} \text{Aritmia}[j] = r[j] & , 600 > rr[j] > 1000 \\ \text{Aritmia}[j] = 0 & , \text{Otherwise} \end{cases} \quad (3.14)$$

Pada Persamaan (3.14) didapatkan *array Aritmia* yang merupakan *array* penyimpanan data aritmia.

3.9 Visualisasi

Visualisasi berfungsi untuk menampilkan hasil deteksi aritmia diantaranya *plot* data ECG dan prosentase *RR-Interval* abnormal. Tampilan tersebut dapat dilihat melalui *device* android atau komputer dengan membutuhkan koneksi dari VNC. .

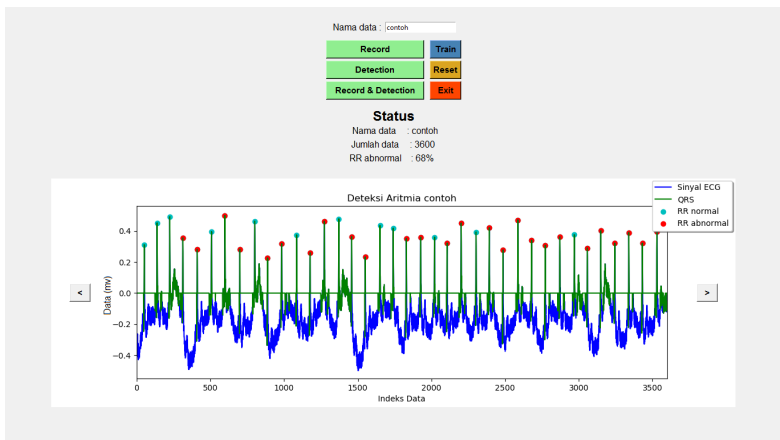
Koneksi VNC adalah perangkat lunak berbagi layar yang memungkinkan terhubung ke komputer jarak jauh, menonton layarnya secara *realtime*, dan mengambil kendali. Koneksi VNC terdiri dari 2 perangkat lunak diantaranya VNC *Server* untuk komputer yang akan dikontrol dan VNC *Viewer* untuk perangkat yang menjadi *remote*.

Koneksi VNC yang digunakan oleh sistem untuk keperluan visualisasi melalui beberapa tahapan diantaranya :

1. Install perangkat lunak VNC *Server* pada SBC (*Single Board Computer*)
2. Install perangkat lunak VNC *Viewer* pada *device* yang akan digunakan

3. Sambungkan SBC dengan jaringan internet yang digunakan oleh *device*
4. Jalankan VNC *Viewer* pada *device*
5. Isi kolom yang tersedia pada VNC *Viewer* dengan IP *address* dari SBC
6. Klik *enter* kemudian tunggu beberapa saat hingga muncul tampilan dari SBC
7. Koneksi VNC pada perangkat sudah terhubung

Tampilan dari proses visualisasi ditunjukkan pada Gambar 3.20.



Gambar 3.20: Visualisasi Hasil Deteksi

Halaman ini sengaja dikosongkan

BAB 4

PENGUJIAN DAN ANALISA

Pada bab ini dipaparkan hasil pengujian serta analisa dari desain sistem dan implementasi. Pengujian dibagi menjadi empat bagian antara lain:

1. Pengujian Akuisisi Data ECG
2. Pengujian Deteksi QRS
3. Pengujian Deteksi *R-Peak*
4. Pengujian Deteksi *RR-Interval*
5. Pengujian Deteksi Aritmia

Dengan dilaksanakannya beberapa pengujian tersebut, sehingga dapat ditarik kesimpulan dari pelaksanaan tugas akhir ini.

4.1 Pengujian Akuisisi Data ECG

Pengujian akuisisi data ECG dilakukan dengan menggunakan perangkat akuisisi dengan frekuensi akuisisi 100Hz sebanyak 3600 data atau sekitar 36 detik. Percobaan dilakukan sebanyak 8 kali dengan nama label a1, a2, a3, a4, a5, a6, a7 dan a8. Hasil akuisisi yang diperoleh oleh sistem dapat dilihat pada Lampiran 1.

4.2 Pengujian Deteksi QRS

Pengujian deteksi QRS dilakukan dengan menggunakan data ECG hasil dari normalisasi akuisisi data ECG. Hasil deteksi QRS yang diperoleh oleh sistem dapat dilihat pada Lampiran 2.

Deteksi QRS yang didapatkan oleh sistem diantaranya data yang menunjukkan lonjakan (sesuai karakter kompleks QRS) sehingga data yang bukan merupakan kompleks QRS namun memiliki nilai lonjakan akan dikenali sebagai kompleks QRS.

4.3 Pengujian Deteksi *R-Peak*

Pengujian deteksi puncak R dilakukan dengan menggunakan data deteksi QRS. Hasil deteksi puncak R yang diperoleh oleh sistem dapat dilihat pada Lampiran 3.

Data rekapan dari deteksi puncak R ditunjukkan pada Tabel

4.1 dan didapatkan rata - rata akurasi sebesar 0,872217.

Tabel 4.1: Deteksi Puncak R

No	Record	TP	FP	TN	FN	Akurasi
1	a1	33	0	5	3562	0,868421
2	a2	38	0	0	3562	1
3	a3	31	0	7	3562	0,815789
4	a4	37	0	2	3561	0,948718
5	a5	30	0	8	3562	0,789474
6	a6	32	0	6	3562	0,842105
7	a7	31	0	7	3562	0,815789
8	a8	35	0	4	3561	0,897436
Rata - Rata						0,872217

4.4 Pengujian Deteksi *RR-Interval*

Pengujian deteksi *RR-interval* dilakukan dengan menggunakan data deteksi R. Hasil deteksi *RR-interval* yang diperoleh oleh sistem dapat dilihat pada Lampiran 4.

Data rekapan dari deteksi *RR-interval* ditunjukkan pada Tabel 4.2 dimana data berwarna merah merupakan *RR-interval* diluar *range* normal (*RR-interval* abnormal), dan data berwarna hitam merupakan *RR-interval* didalam *range* normal.

Tabel 4.2: Deteksi RR Interval

No	RR Interval pada Record Data (ms)							
	a1	a2	a3	a4	a5	a6	a7	a8
1	1002	945	831	1062	1111	1049	1008	1006
2	949	927	791	1064	2228	1035	2167	1029
3	1009	964	1795	1085	1102	1001	997	939
4	1012	1054	1055	2044	1060	1063	1100	1040
5	1992	1061	1037	1081	1126	1063	1121	1059
6	1053	978	2138	999	2138	2027	2089	982
7	870	1106	1072	1043	1115	1097	1088	953
8	934	1106	1080	1099	2144	1026	974	1008

Table 4.2 continued from previous page

9	1706	951	1026	989	1104	1127	1014	1027
10	937	1062	1083	1070	1975	1115	1065	941
11	979	1117	1020	1080	1011	2114	974	1028
12	886	1000	926	1004	927	1126	1021	1023
13	1003	1046	1050	1092	1007	2125	1087	1993
14	1085	1092	1103	1081	1026	1122	2091	1056
15	1075	961	1003	2128	1994	1045	1109	953
16	1090	1012	1094	1035	1110	1117	2065	1072
17	1114	1104	1082	1082	2075	1127	1102	1080
18	1023	995	1094	1114	1133	2132	2066	951
19	1102	983	1135	1001	1059	1096	1101	986
20	1128	1082	2170	1097	1085	2117	2068	1021
21	1071	1038	1131	1116	1115	1110	1100	1934
22	1054	937	2126	1027	2120	2044	1000	1052
23	2226	1051	1137	1097	1103	1085	1021	2054
24	1056	1116	2107	1097	994	994	1057	1125
25	1096	995	1102	966	1066	1079	966	1000
26	1084	1025	2112	1084	1080	1106	1022	1102
27	999	1114	1106	1108	2017	1038	1101	1133
28	1063	1028	1080	1027	1088	1075	2067	1001
29	1080	1028	1128	1149	1015	1101	2133	1071
30	981	1093	2124	1184	1015	983	1087	1078
31	1028	1081	2124	1052	-	1062	1087	2091
32	995	971	-	1142	-	1062	-	1130
33	995	1016	-	1154	-	-	-	1024
34	-	1050	-	1087	-	-	-	1065
35	-	1050	-	1105	-	-	-	1065
36	-	1016	-	1105	-	-	-	-
37	-	1095	-	-	-	-	-	-
38	-	1095	-	-	-	-	-	-

4.5 Pengujian Deteksi Aritmia

Pengujian deteksi aritmia dilakukan dengan menggunakan data deteksi *RR-interval* dimana data akan dikenali *RR-interval* abnormal jika kurang dari 600 *milisecond* atau lebih dari 1000 *milise-*

cond [10]. Hasil deteksi aritmia yang diperoleh oleh sistem dapat dilihat pada Lampiran 5.

Data rekapan dari deteksi aritmia ditunjukkan pada Tabel 4.3.

Tabel 4.3: Deteksi Aritmia

No	Record	RR Interval Normal	RR Interval Abnormal	Total	Aritmia (%)
1	a1	10	23	33	70
2	a2	11	26	37	70
3	a3	2	28	30	93
4	a4	3	33	36	92
5	a5	2	28	30	93
6	a6	2	30	32	94
7	a7	5	26	31	84
8	a8	8	27	35	77

BAB 5

PENUTUP

5.1 Kesimpulan

Berdasarkan hasil pengujian dapat disimpulkan bahwa sistem dapat bekerja dengan cukup baik sesuai dengan rancangan. Penulis mengimplementasikan perangkat ECG sebagai pengakuisisi data dan sistem dijalankan untuk mendeteksi aritmia berdasarkan RR interval. Kemudian untuk lebih detail dapat ditarik kesimpulan sebagai berikut:

1. Perangkat dapat mengakuisisi data ECG pada subjek dengan baik.
2. Sistem dapat berfungsi untuk membantu memonitor kondisi jantung pada subjek.
3. Sistem dapat mendiagnosa awal kondisi jantung mengenai prosentase terjadinya aritmia berdasarkan RR interval yang tidak normal.
4. Berdasarkan hasil pengujian deteksi QRS menggunakan algoritma feed forward backpropagation menunjukkan hasil beberapa anomali deteksi QRS dikarenakan sifat kompleks QRS mirip dengan segmen sinyal ECG yang lain.
5. Berdasarkan hasil pengujian deteksi R-peak mendapatkan rata - rata akurasi sebesar 0,872217.
6. Berdasarkan hasil pengujian deteksi RR-interval menunjukkan hasil yang baik.
7. Berdasarkan hasil pengujian deteksi aritmia menunjukkan hasil yang baik.

5.2 Saran

Untuk pengembangan penelitian selanjutnya terdapat beberapa saran sebagai berikut :

1. Perangkat menggunakan catu daya baterai yang sudah terintegrasi dengan SBC serta sensor.
2. Tambahkan fitur kalkulasi nilai *heart rate* untuk membantu proses *decision* pada sistem.

Halaman ini sengaja dikosongkan

DAFTAR PUSTAKA

- [1] “Heart disease and stroke statistics 2018.” American Heart Association. (Dikutip pada halaman i, iii, 1).
- [2] F. Martini, J. L. Nath, and E. F. Bartholomew, Fundamentals of anatomy & physiology. Pearson, 2015. (Dikutip pada halaman xi, 5).
- [3] S. Sena, Pengenalan Deep Learning Part 1 : Neural Network. <http://www.medium.com>, 2017. (Dikutip pada halaman xi, 11).
- [4] R. P. Foundation, Raspberry Pi 3 Model B+. <https://static.raspberrypi.org>, 2017. (Dikutip pada halaman xi, 14).
- [5] O. T. Way, Single-Lead, Heart Rate Monitor Front End AD8232. Analog Devices, Inc, 2012-2018. (Dikutip pada halaman xi, 15, 16).
- [6] “Intel movidius neural compute stick 2.” <https://software.intel.com/en-us/neural-compute-stick>. Terakhir Diakses pada tanggal 22 maret 2020. (Dikutip pada halaman xi, 16, 17).
- [7] F. A. Darmawan, Perangkat Treadmill Untuk Mendeteksi Kelainan Detak Jantung Menggunakan Sensor ECG Berbasis Single Board Computer. Surabaya: Institut Teknologi Sepuluh Nopember, 2019. (Dikutip pada halaman xi, 19).
- [8] J. M. B. Babuslak, The ECG Signal Prediction By Using Neural Network. Ostrava, Czechia: Technical Universal of Ostrava. (Dikutip pada halaman xi, 24).
- [9] 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. (Dikutip pada halaman xi, 25).

- [10] Y. N. M. C. Saritha, V. Sukanya, ECG Signal Analysis Using Wavelet Transforms. Anantapur - 515 001, Andhrapradesh, India: Department of Physics and Electronics, S.S.B.N. COLLEGE (Autonomous), 2008. (Dikutip pada halaman xv, 20, 21, 22, 42, 48).
- [11] “Dokter jantung di indonesia langka.” <https://sains.kompas.com>. Terakhir Diakses pada tanggal 21 Juli 2020. (Dikutip pada halaman 1, 2).
- [12] A. M. Alim, “Pocket ecg: How to learn ecg from zero,” Jakarta: Intan Cendekia, 2008. (Dikutip pada halaman 6).
- [13] A. H. Association, Arrhythmias. <http://www.heart.org>, 2015. (Dikutip pada halaman 7).
- [14] J. W. G. Putra, Pengenalan Konsep Pembelajaran Mesin dan Deep Learning. <http://www.researchgate.net>, 2019. (Dikutip pada halaman 8, 9, 11).
- [15] M. Compiled, Activation Functions. <http://ml-compiled.readthedocs.io>, 2018. (Dikutip pada halaman 11).
- [16] M. Compiled, Loss Functions. <http://ml-compiled.readthedocs.io>, 2018. (Dikutip pada halaman 12).
- [17] T. Instruments, Ultra Small, Low Power, 16 Bit. Analog to Digital Converter with Internal Reference. <https://www.ti.com>, 2009. (Dikutip pada halaman 18).
- [18] R. D. K. Akash Chatlani, Shradha Kabu, IOT Based Detection of cardiac Arrhythmias Using Raspberry PI. Kolkana, West Bengal, India: MIT College of Engineering, 2017. (Dikutip pada halaman 18).
- [19] R. A. R. Mamun B.I. Reaz, Muhammad I. Ibrahimy, An Approach to Detect QRS Complex Using Backpropagation Neural Network. Cyberjaya and Selangor, Malaysia: Multimedia University and International Islamic University, 2006. (Dikutip pada halaman 23).

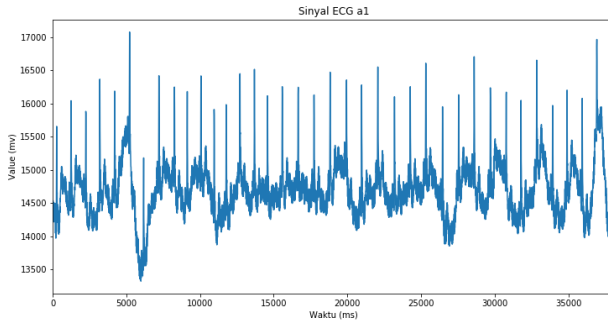
- [20] O. Polakovic, Backpropagation and His Application in ECG Classification. Ostrava, Czechia: University of Ostrava, 2005. (Dikutip pada halaman 24).

Halaman ini sengaja dikosongkan

LAMPIRAN

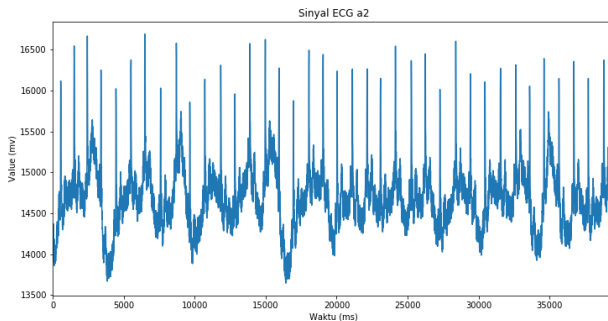
- Lampiran 1 : Akuisisi Data ECG

Pengambilan data pertama saat di *plot* menjadi sebuah grafik ditunjukkan pada Gambar 1.



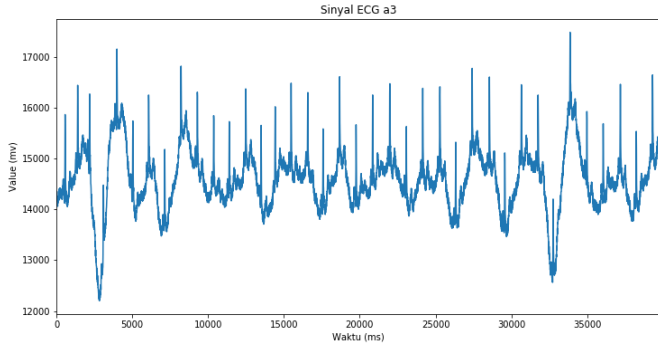
Gambar 1: Akuisisi Data a1

Pengambilan data kedua saat di *plot* menjadi sebuah grafik ditunjukkan pada Gambar 2.



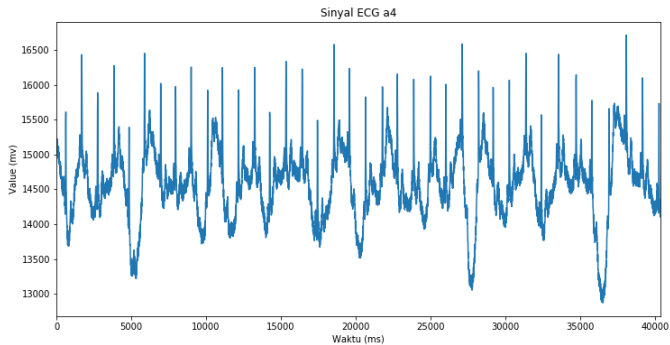
Gambar 2: Akuisisi Data a2

Pengambilan data ketiga saat di *plot* menjadi sebuah grafik ditunjukkan pada Gambar 3.



Gambar 3: Akuisisi Data a3

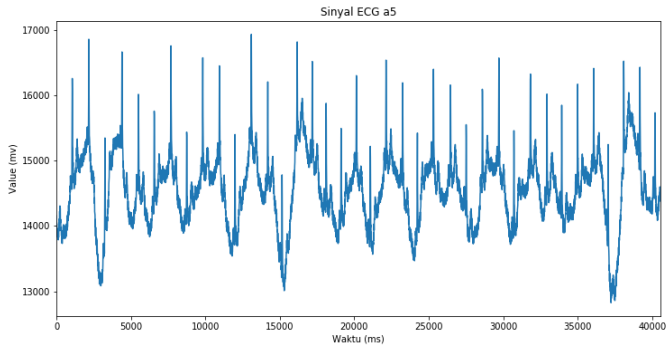
Pengambilan data keempat saat di *plot* menjadi sebuah grafik ditunjukkan pada Gambar 4.



Gambar 4: Akuisisi Data a4

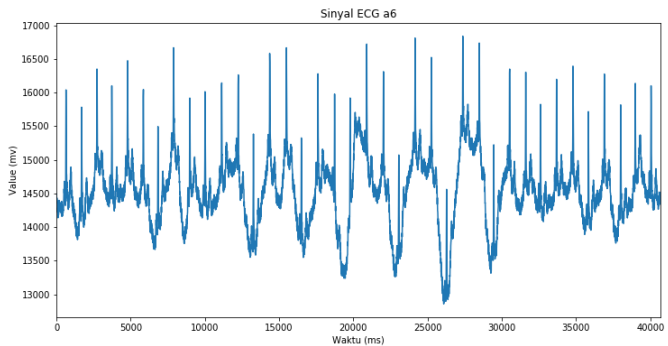
Pengambilan data kelima saat di *plot* menjadi sebuah grafik

ditunjukkan pada Gambar 5.



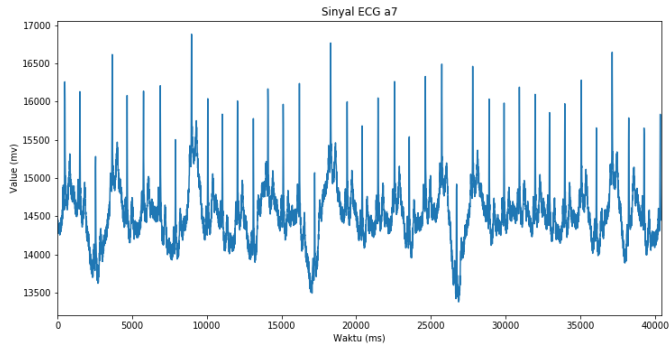
Gambar 5: Akuisisi Data a5

Pengambilan data keenam saat di *plot* menjadi sebuah grafik ditunjukkan pada Gambar 6.



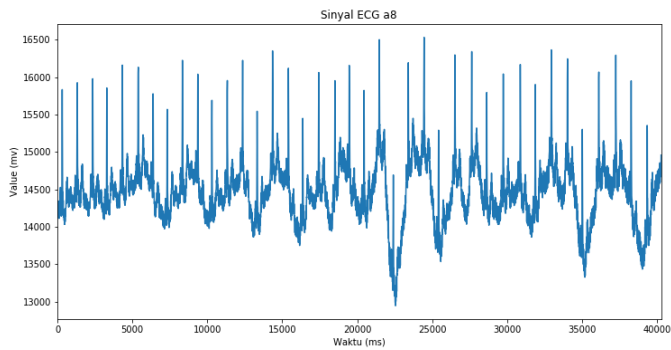
Gambar 6: Akuisisi Data a6

Pengambilan data ketujuh saat di *plot* menjadi sebuah grafik ditunjukkan pada Gambar 7.



Gambar 7: Akuisisi Data a7

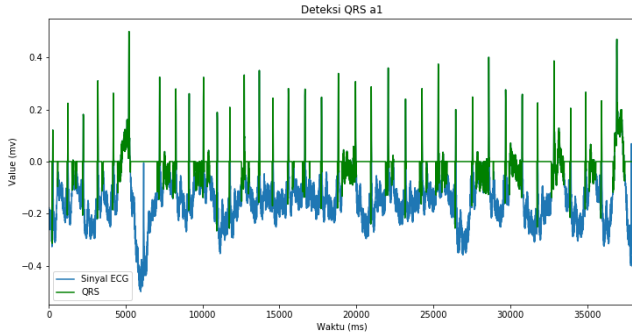
Pengambilan data kedelapan saat di *plot* menjadi sebuah grafik ditunjukkan pada Gambar 8.



Gambar 8: Akuisisi Data a8

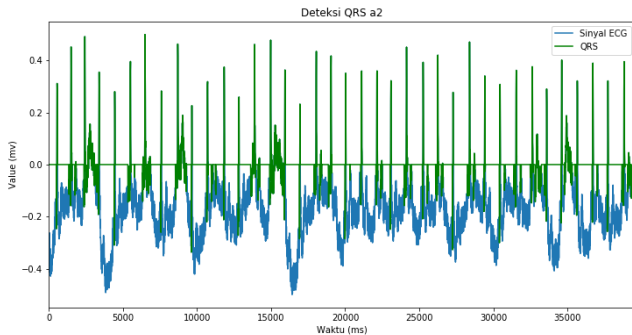
- Lampiran 2 : Deteksi QRS

Hasil deteksi QRS dari data pertama saat di *plot* menjadi sebuah grafik ditunjukkan pada Gambar 9.



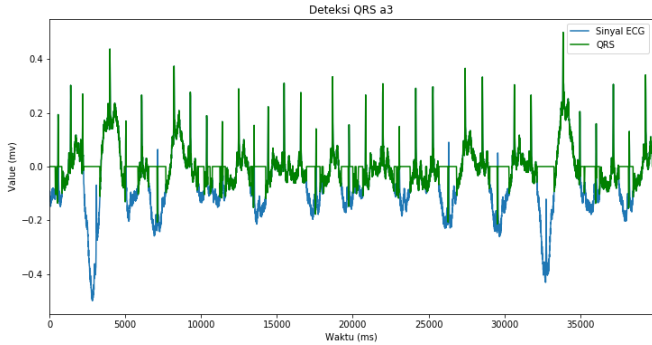
Gambar 9: Deteksi QRS Pada Data a1

Hasil deteksi QRS dari data kedua saat di *plot* menjadi sebuah grafik ditunjukkan pada Gambar 10.



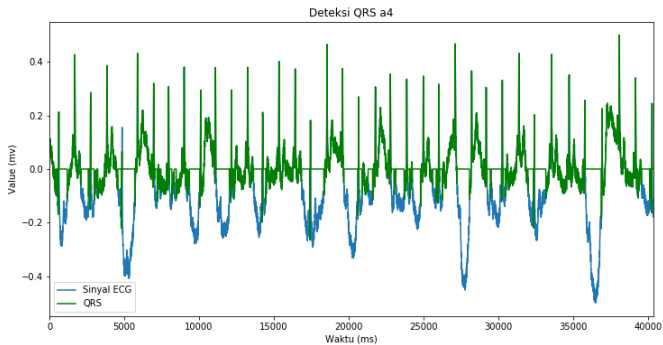
Gambar 10: Deteksi QRS Pada Data a2

Hasil deteksi QRS dari data ketiga saat di *plot* menjadi sebuah grafik ditunjukkan pada Gambar 11.



Gambar 11: Deteksi QRS Pada Data a3

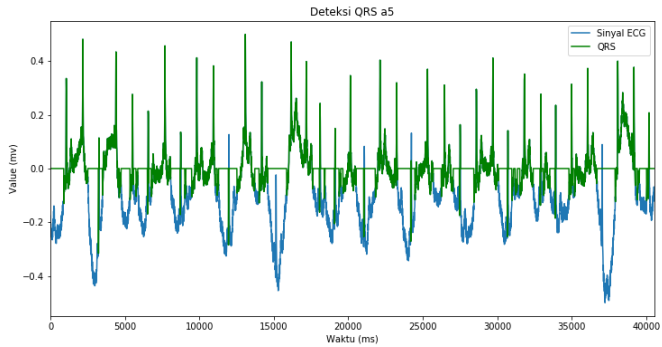
Hasil deteksi QRS dari data keempat saat di *plot* menjadi sebuah grafik ditunjukkan pada Gambar 12.



Gambar 12: Deteksi QRS Pada Data a4

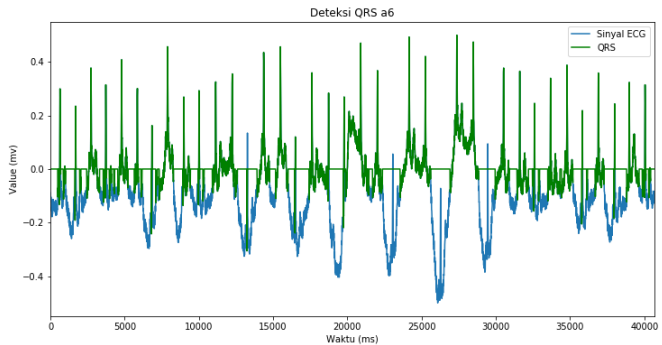
Hasil deteksi QRS dari data kelima saat di *plot* menjadi sebu-

ah grafik ditunjukkan pada Gambar 13.



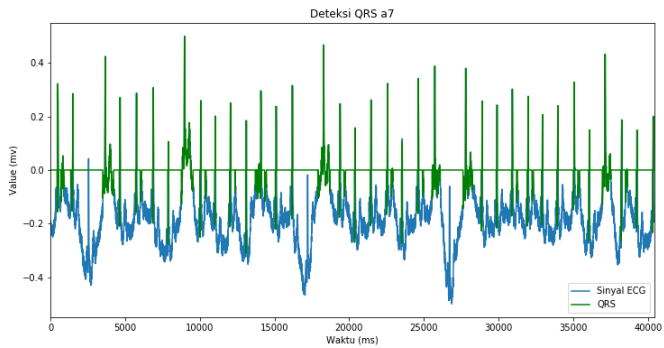
Gambar 13: Deteksi QRS Pada Data a5

Hasil deteksi QRS dari data keenam saat di *plot* menjadi sebuah grafik ditunjukkan pada Gambar 14.



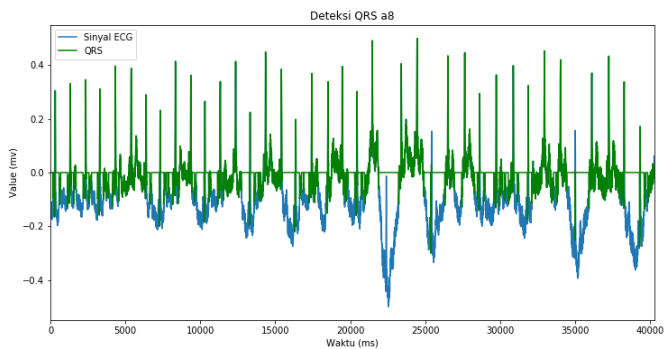
Gambar 14: Deteksi QRS Pada Data a6

Hasil deteksi QRS dari data ketujuh saat di *plot* menjadi sebuah grafik ditunjukkan pada Gambar 15.



Gambar 15: Deteksi QRS Pada Data a7

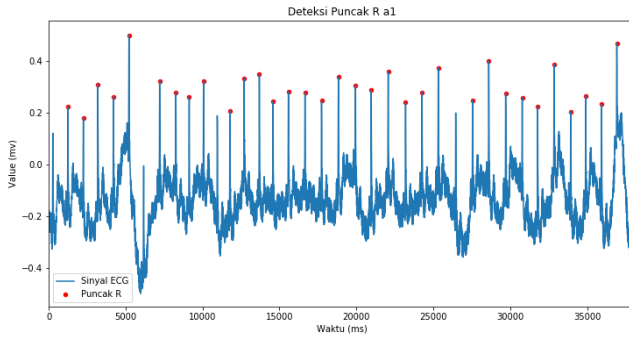
Hasil deteksi QRS dari data kedelapan saat di *plot* menjadi sebuah grafik ditunjukkan pada Gambar 16.



Gambar 16: Deteksi QRS Pada Data a8

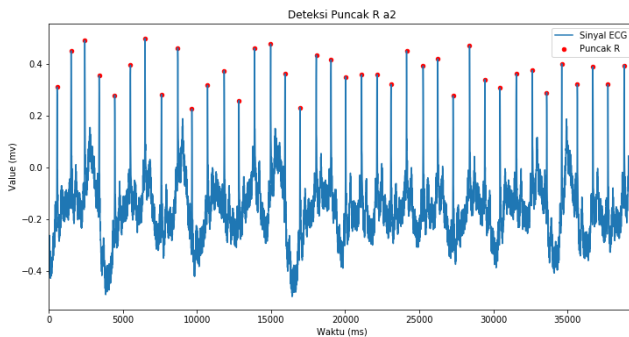
- Lampiran 3 : Deteksi *R-Peak*

Hasil deteksi R dari data pertama saat di *plot* menjadi sebuah grafik ditunjukkan pada Gambar 17.



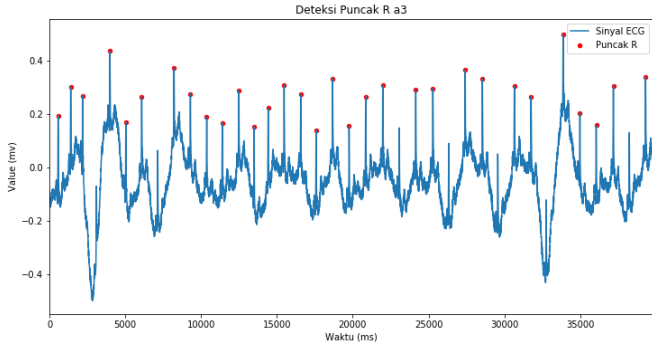
Gambar 17: Deteksi Puncak R Data a1

Hasil deteksi R dari data kedua saat di *plot* menjadi sebuah grafik ditunjukkan pada Gambar 18.



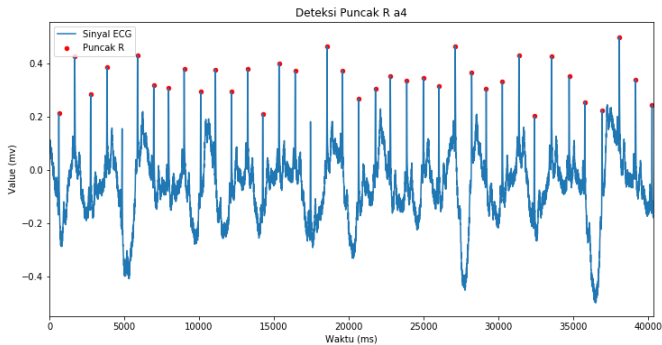
Gambar 18: Deteksi Puncak R Data a2

Hasil deteksi R dari data ketiga saat di *plot* menjadi sebuah grafik ditunjukkan pada Gambar 19.



Gambar 19: Deteksi Puncak R Data a3

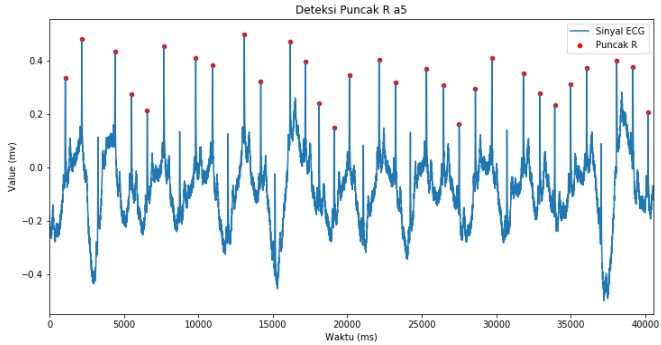
Hasil deteksi R dari data keempat saat di *plot* menjadi sebuah grafik ditunjukkan pada Gambar 20.



Gambar 20: Deteksi Puncak R Data a4

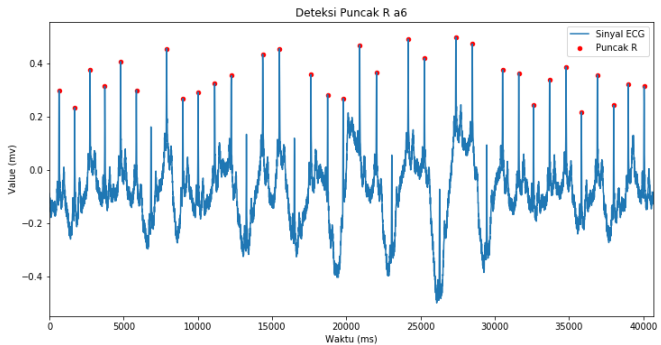
Hasil deteksi R dari data kelima saat di *plot* menjadi sebuah

grafik ditunjukkan pada Gambar 21.



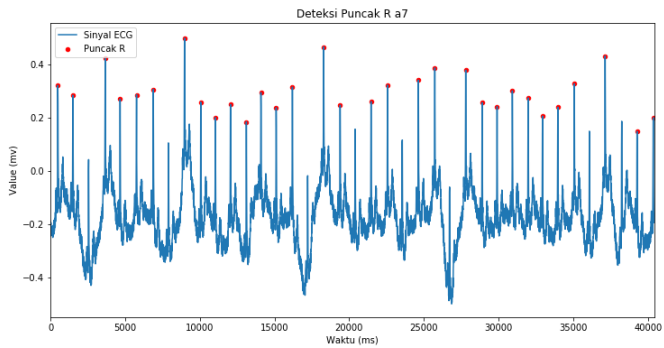
Gambar 21: Deteksi Puncak R Data a5

Hasil deteksi R dari data keenam saat di *plot* menjadi sebuah grafik ditunjukkan pada Gambar 22.



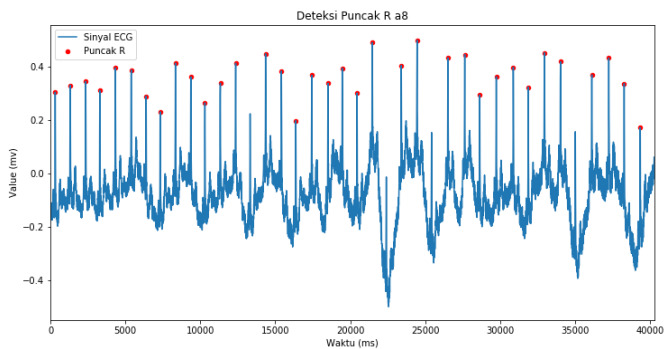
Gambar 22: Deteksi Puncak R Data a6

Hasil deteksi R dari data ketujuh saat di *plot* menjadi sebuah grafik ditunjukkan pada Gambar 23.



Gambar 23: Deteksi Puncak R Data a7

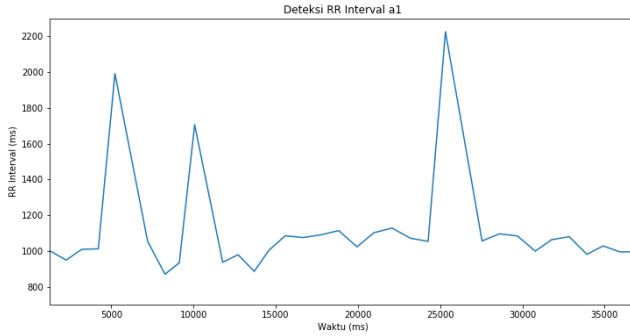
Hasil deteksi R dari data kedelapan saat di *plot* menjadi sebuah grafik ditunjukkan pada Gambar 24.



Gambar 24: Deteksi Puncak R Data a8

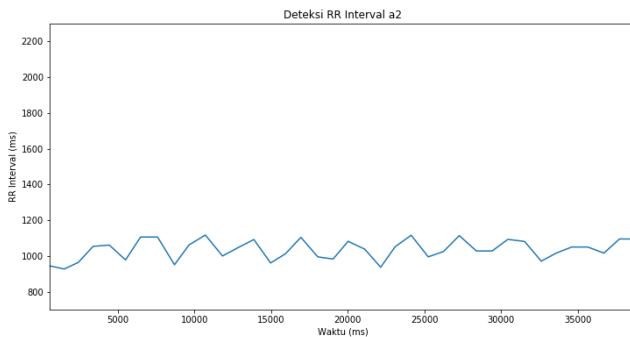
- Lampiran 4 : Deteksi *RR-Interval*

Hasil deteksi *RR-interval* dari data pertama saat di *plot* menjadi sebuah grafik ditunjukkan pada Gambar 25.



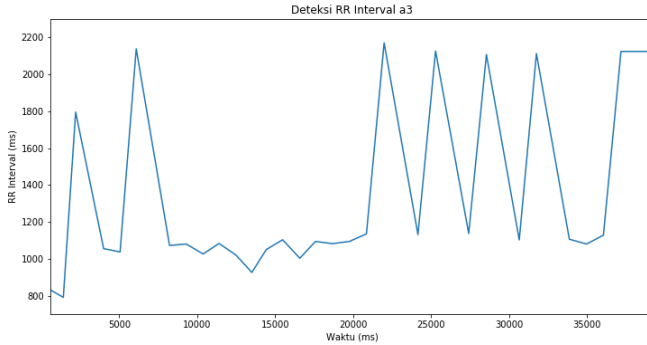
Gambar 25: Deteksi *RR-Interval* Data a1

Hasil deteksi *RR-interval* dari data kedua saat di *plot* menjadi sebuah grafik ditunjukkan pada Gambar 26.



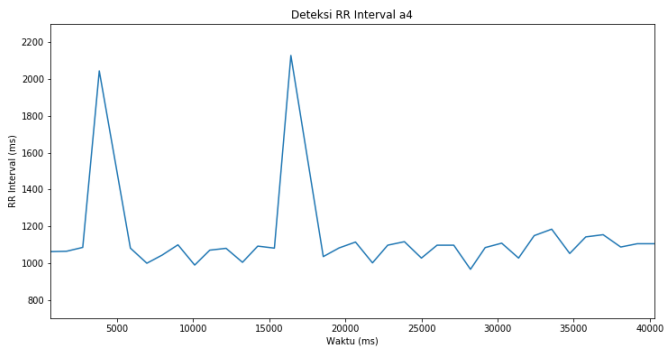
Gambar 26: Deteksi *RR-Interval* Data a2

Hasil deteksi *RR-interval* dari data ketiga saat di *plot* menjadi sebuah grafik ditunjukkan pada Gambar 27.



Gambar 27: Deteksi *RR-Interval* Data a3

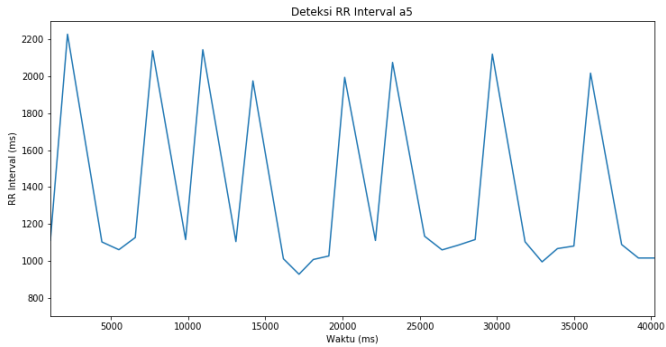
Hasil deteksi *RR-interval* dari data keempat saat di *plot* menjadi sebuah grafik ditunjukkan pada Gambar 28.



Gambar 28: Deteksi *RR-Interval* Data a4

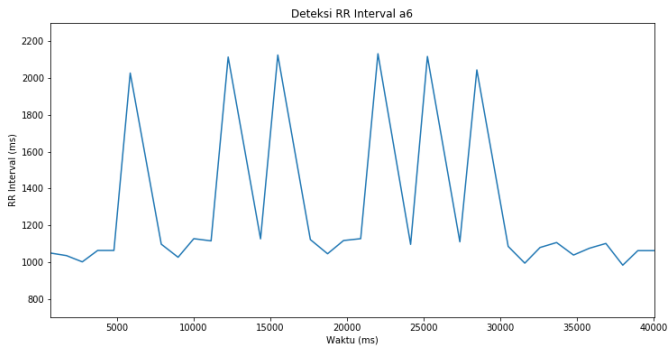
Hasil deteksi *RR-interval* dari data kelima saat di *plot* menjadi

sebuah grafik ditunjukkan pada Gambar 29.



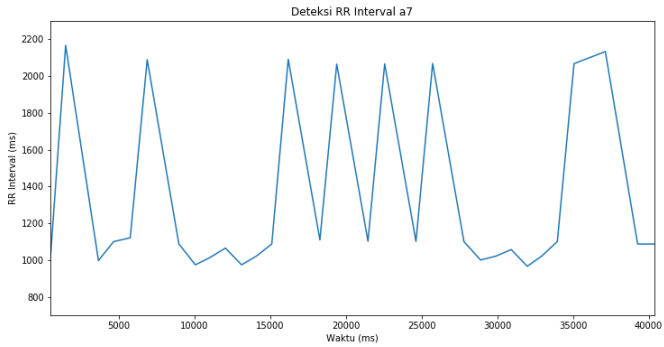
Gambar 29: Deteksi *RR-Interval* Data a5

Hasil deteksi *RR-interval* dari data keenam saat di *plot* menjadi sebuah grafik ditunjukkan pada Gambar 30.



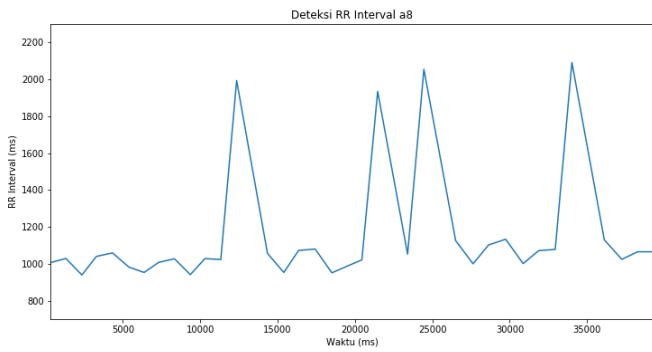
Gambar 30: Deteksi *RR-Interval* Data a6

Hasil deteksi *RR-interval* dari data ketujuh saat di *plot* menjadi sebuah grafik ditunjukkan pada Gambar 31.



Gambar 31: Deteksi *RR-Interval* Data a7

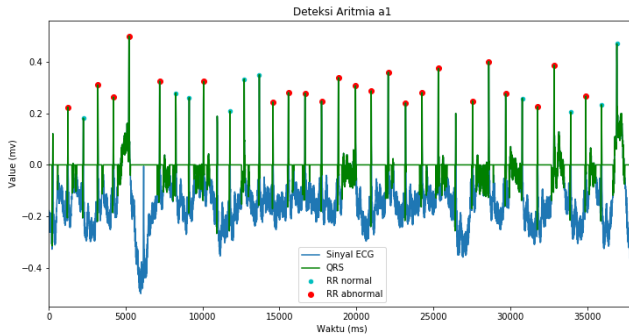
Hasil deteksi *RR-interval* dari data kedelapan saat di *plot* menjadi sebuah grafik ditunjukkan pada Gambar 32.



Gambar 32: Deteksi *RR-Interval* Data a8

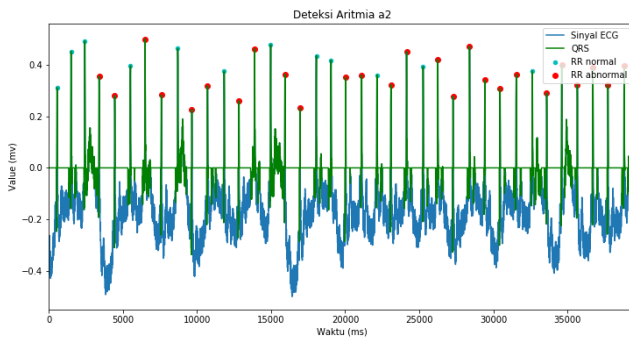
- Lampiran 5 : Deteksi Aritmia

Hasil deteksi aritmia dari data pertama saat di *plot* menjadi sebuah grafik ditunjukkan pada Gambar 33.



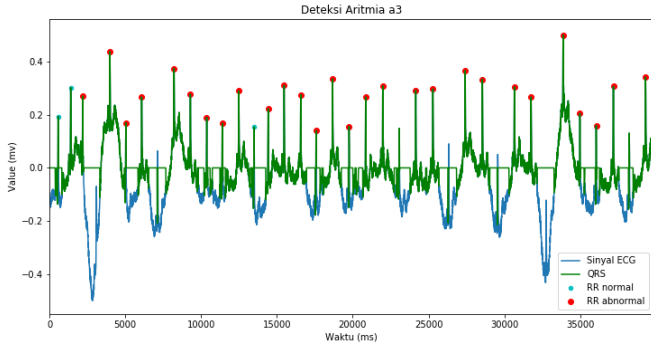
Gambar 33: Deteksi Aritmia Pada Data a1

Hasil deteksi aritmia dari data kedua saat di *plot* menjadi sebuah grafik ditunjukkan pada Gambar 34.



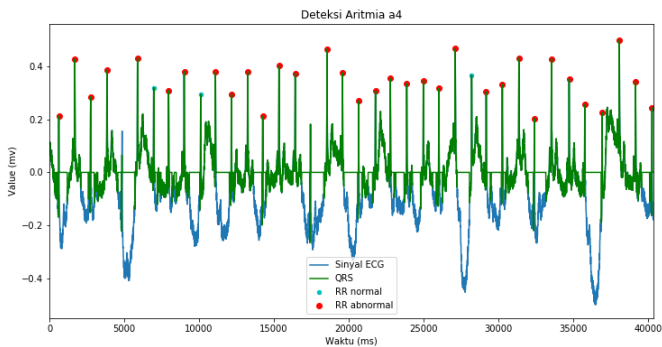
Gambar 34: Deteksi Aritmia Pada Data a2

Hasil deteksi aritmia dari data ketiga saat di *plot* menjadi sebuah grafik ditunjukkan pada Gambar 35.



Gambar 35: Deteksi Aritmia Pada Data a3

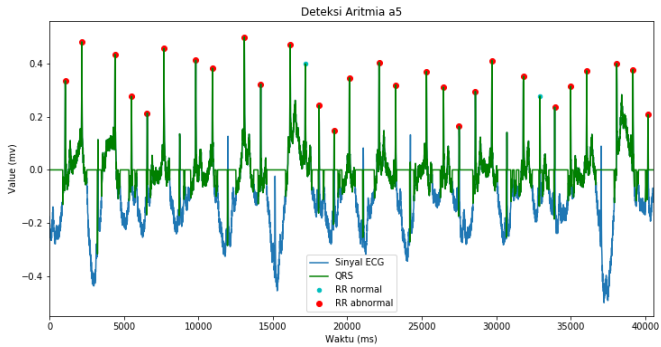
Hasil deteksi aritmia dari data keempat saat di *plot* menjadi sebuah grafik ditunjukkan pada Gambar 36.



Gambar 36: Deteksi Aritmia Pada Data a4

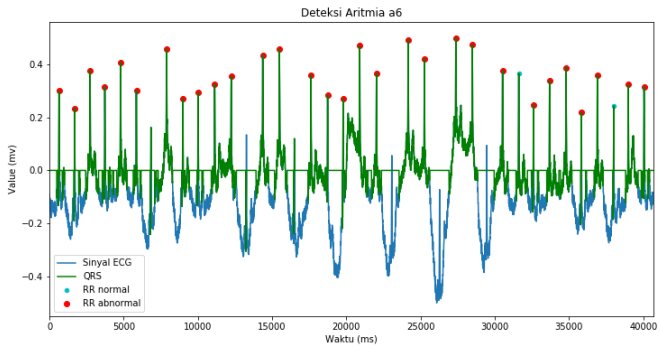
Hasil deteksi aritmia dari data kelima saat di *plot* menjadi

sebuah grafik ditunjukkan pada Gambar 37.



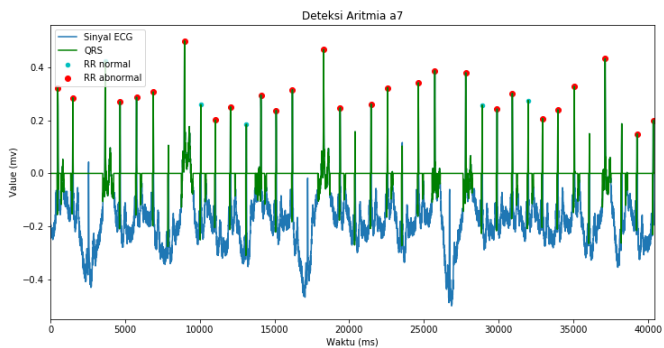
Gambar 37: Deteksi Aritmia Pada Data a5

Hasil deteksi aritmia dari data keenam saat di *plot* menjadi sebuah grafik ditunjukkan pada Gambar 38.



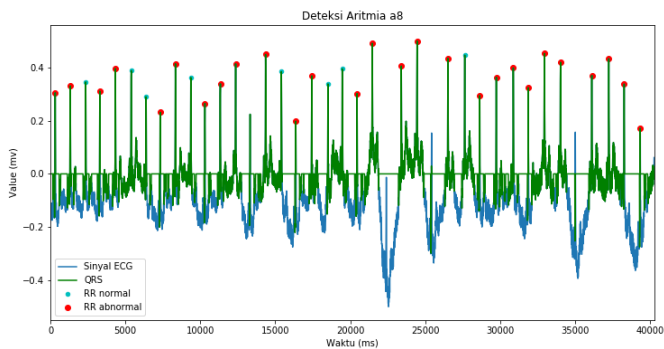
Gambar 38: Deteksi Aritmia Pada Data a6

Hasil deteksi aritmia dari data ketujuh saat di *plot* menjadi sebuah grafik ditunjukkan pada Gambar 39.



Gambar 39: Deteksi Aritmia Pada Data a7

Hasil deteksi aritmia dari data kedelapan saat di *plot* menjadi sebuah grafik ditunjukkan pada Gambar 40.



Gambar 40: Deteksi Aritmia Pada Data a8

- Lampiran 6 : Spesifikasi Teknis Raspberry Pi 3B +

Specifications

Processor:	Broadcom BCM2837B0, Cortex-A53 64-bit SoC @ 1.4GHz
Memory:	1GB LPDDR2 SDRAM
Connectivity:	<ul style="list-style-type: none">■ 2.4GHz and 5 GHz IEEE 802.11.b/g/n/ac wireless LAN, Bluetooth 4.2, BLE■ Gigabit Ethernet over USB 2.0 (maximum throughput 300Mbps)■ 4 × USB 2.0 ports
Access:	Extended 40-pin GPIO header
Video & sound:	<ul style="list-style-type: none">■ 1 × full size HDMI■ MIPI DSI display port■ MIPI CSI camera port■ 4 pole stereo output and composite video port
Multimedia:	H.264, MPEG-4 decode (1080p30); H.264 encode (1080p30); OpenGL ES 1.1, 2.0 graphics
SD card support:	Micro SD format for loading operating system and data storage
Input power:	<ul style="list-style-type: none">■ 5V/2.5A DC via micro USB connector■ 5V DC via GPIO header■ Power over Ethernet (PoE)–enabled (requires separate PoE HAT)
Environment:	Operating temperature, 0–50°C
Compliance:	For a full list of local and regional product approvals, please visit www.raspberrypi.org/products/raspberry-pi-3-model-b+
Production lifetime:	The Raspberry Pi 3 Model B+ will remain in production until at least January 2023.

- Lampiran 7 : Spesifikasi Teknis Modul AD8232

SPECIFICATIONS

$V_S = 3\text{ V}$, $V_{REF} = 1.5\text{ V}$, $V_{CM} = 1.5\text{ V}$, $T_A = 25^\circ\text{C}$, FR=low, SDN=high, AC/DC = low, unless otherwise noted.

Table 1.

Parameter	Symbol	Test Conditions/Comments	Min	Typ	Max	Unit
INSTRUMENTATION AMPLIFIER						
Common-Mode Rejection Ratio, DC to 60 Hz	CMRR	$V_{CM} = 0.35\text{ V to } 2.85\text{ V}$, $V_{OFF} = 0\text{ V}$	80	86		dB
Power Supply Rejection Ratio	PSRR	$V_{CM} = 0.35\text{ V to } 2.85\text{ V}$, $V_{OFF} = \pm 0.3\text{ V}$		80		dB
Offset Voltage (RTI)	V_{OS}	$V_S = 2.0\text{ V to } 3.5\text{ V}$	76	90		dB
Instrumentation Amplifier Inputs				3	8	mV
DC Blocking Input ¹				5	50	μV
Average Offset Drift				10		$\mu\text{V}/^\circ\text{C}$
Instrumentation Amplifier Inputs				0.05		$\mu\text{V}/^\circ\text{C}$
DC Blocking Input ¹				50	200	pA
Input Bias Current	I_B	$T_A = 0^\circ\text{C to } 70^\circ\text{C}$		1		nA
Input Offset Current	I_{OS}	$T_A = 0^\circ\text{C to } 70^\circ\text{C}$		25	100	pA
				1		nA
Input Impedance						
Differential				10 7.5		GΩ pF
Common Mode				5 15		GΩ pF
Input Voltage Noise (RTI)						
Spectral Noise Density		$f = 1\text{ kHz}$		100		nV/ $\sqrt{\text{Hz}}$
Peak-to-Peak Voltage Noise		$f = 0.1\text{ Hz to } 10\text{ Hz}$		12		$\mu\text{V p-p}$
		$f = 0.5\text{ Hz to } 40\text{ Hz}$		14		$\mu\text{V p-p}$
Input Voltage Range		$T_A = 0^\circ\text{C to } 70^\circ\text{C}$			$+V_S$	V
DC Differential Input Range	V_{OFF}		-300		+300	mV
Output						
Output Swing		$R_L = 50\text{ k}\Omega$	0.1		$+V_S - 0.1$	V
Short-Circuit Current	I_{OUT}			6.3		mA
Gain	A_V			100		V/V
Gain Error		$V_{OFF} = 0\text{ V}$		0.4		%
		$V_{OFF} = -300\text{ mV to } +300\text{ mV}$		1	3.5	%
Average Gain Drift		$T_A = 0^\circ\text{C to } 70^\circ\text{C}$		12		ppm/ $^\circ\text{C}$
Bandwidth	BW			2		kHz
RFI Filter Cutoff (Each Input)				1		MHz
OPERATIONAL AMPLIFIER (A1)						
Offset Voltage	V_{OS}	$T_A = 0^\circ\text{C to } 70^\circ\text{C}$		1	5	mV
Average TC				5		$\mu\text{V}/^\circ\text{C}$
Input Bias Current	I_B	$T_A = 0^\circ\text{C to } 70^\circ\text{C}$		100		pA
Input Offset Current	I_{OS}	$T_A = 0^\circ\text{C to } 70^\circ\text{C}$		1		nA
				100		pA
		$T_A = 0^\circ\text{C to } 70^\circ\text{C}$		1		nA
Input Voltage Range			0.1		$+V_S - 0.1$	V
Common-Mode Rejection Ratio	CMRR	$V_{CM} = 0.5\text{ V to } 2.5\text{ V}$		100		dB
Power Supply Rejection Ratio	PSRR			100		dB
Large Signal Voltage Gain	A_{VO}			110		dB
Output Voltage Range		$R_L = 50\text{ k}\Omega$	0.1		$+V_S - 0.1$	V
Short-Circuit Current Limit	I_{OUT}			12		mA
Gain Bandwidth Product	GBP			100		kHz
Slew Rate	SR			0.02		V/ μs
Voltage Noise Density (RTI)	e_n	$f = 1\text{ kHz}$		60		nV/ $\sqrt{\text{Hz}}$
Peak-to-Peak Voltage Noise (RTI)	e_{n-p-p}	$f = 0.1\text{ Hz to } 10\text{ Hz}$		6		$\mu\text{V p-p}$
		$f = 0.5\text{ Hz to } 40\text{ Hz}$		8		$\mu\text{V p-p}$

RIGHT LEG DRIVE AMPLIFIER (A2)						
Output Swing		$R_L = 50\text{ k}\Omega$	0.1		$+V_S - 0.1$	V
Short-Circuit Current	I_{OUT}			11		mA
Integrator Input Resistor			120	150	180	k Ω
Gain Bandwidth Product	GDP				100	kHz
REFERENCE BUFFER (A3)						
Offset Error	V_{OS}	$R_L > 50\text{ k}\Omega$		1		mV
Input Bias Current	I_B			100		pA
Short-Circuit Current Limit	I_{OUT}	$R_L = 50\text{ k}\Omega$		12		mA
Voltage Range			0.1		$+V_S - 0.7$	V
DC LEADS OFF COMPARATORS						
Threshold Voltage					$+V_S - 0.5$	V
Hysteresis				60		mV
Propagation Delay				0.5		μ s
AC LEADS OFF DETECTOR						
Square Wave Frequency	F_{AC}		50	100	175	kHz
Square Wave Amplitude	I_{AC}			200		nA p-p
Impedance Threshold		Between +IN and -IN	10	20		M Ω
Detection Delay				110		μ s
FAST RESTORE CIRCUIT						
Switches		S1 and S2				
On Resistance	R_{ON}		8	10	12	k Ω
Off Leakage				100		pA
Window Comparator						
Threshold Voltage		From either rail		50		mV
Propagation Delay				2		μ s
Switch Timing Characteristics						
Feedback Recovery Switch On Time	t_{SW1}			110		ms
Filter Recovery Switch On Time	t_{SW2}			55		ms
Fast Restore Reset	t_{RES}			2		μ s
LOGIC INTERFACE						
Input Characteristics						
Input Voltage ($\overline{AC/DC}$ and FR)						
Low	V_L			1.24		V
High	V_H			1.35		V
Input Voltage (\overline{SDN})						
Low	V_L			2.1		V
High	V_H			0.5		V
Output Characteristics		LOD+ and LOD- terminals				
Output Voltage						
Low	V_{OL}			0.05		V
High	V_{OH}			2.95		V
SYSTEM SPECIFICATIONS						
Quiescent Supply Current		$T_A = 0^\circ\text{C to }70^\circ\text{C}$		170	230	μ A
				210		μ A
Shutdown Current		$T_A = 0^\circ\text{C to }70^\circ\text{C}$		40	500	nA
				100		nA
Supply Range			2.0		3.5	V
Specified Temperature Range			0		70	$^\circ\text{C}$
Operational Temperature Range			-40		+85	$^\circ\text{C}$

- Lampiran 8 : Spesifikasi Teknis Intel Neural Compute Stick 2

INTEL[®] NEURAL COMPUTE STICK 2

Technical Specifications

Specifications	Intel [®] Neural Compute Stick 2
Vision Processing Unit (VPU)	The Intel [®] Movidius™ Myriad™ X VPU
Software development kit	The Intel [®] Distribution of OpenVINO™ toolkit
Operating Systems support	Ubuntu* 16.04.3 LTS (64 bit), Windows® 10 (64 bit), or CentOS* 7.4 (64 bit)
Supported framework	TensorFlow* and Caffe*
Connectivity	USB 3.1 Type-A, USB 2.0 Type-A
USB stick dimensions	72.5mm X 27mm X 14mm
Operating temperature	0° - 40° C
Material Master Number	964486
MSRP	\$69 USD July 14, 2019

- Lampiran 9 : Spesifikasi Teknis Modul ADS1115

ABSOLUTE MAXIMUM RATINGS ⁽¹⁾

	ADS1113, ADS1114, ADS1115	UNIT
VDD to GND	-0.3 to +5.5	V
Analog input current	100, momentary	mA
Analog input current	10, continuous	mA
Analog input voltage to GND	-0.3 to VDD + 0.3	V
SDA, SCL, ADDR, ALERT/RDY voltage to GND	-0.5 to +5.5	V
Maximum junction temperature	+150	°C
Operating temperature range	-40 to +140	°C
Storage temperature range	-60 to +150	°C

ELECTRICAL CHARACTERISTICS

All specifications at -40°C to +125°C, VDD = 3.3V, and Full-Scale (FS) = ±2.048V, unless otherwise noted. Typical values are at +25°C.

PARAMETER	TEST CONDITIONS	ADS1113, ADS1114, ADS1115			UNIT
		MIN	TYP	MAX	
ANALOG INPUT					
Full-scale input voltage ⁽¹⁾	$V_{IN} = (AIN_P) - (AIN_N)$		±4.096/PGA		V
Analog input voltage	AIN_P or AIN_N to GND	GND		VDD	V
Differential input impedance			See Table 2		
Common-mode input impedance	FS = ±6.144V ⁽¹⁾		10		MΩ
	FS = ±4.096V ⁽¹⁾ , ±2.048V		6		MΩ
	FS = ±1.024V		3		MΩ
	FS = ±0.512V, ±0.256V		100		MΩ
SYSTEM PERFORMANCE					
Resolution	No missing codes	16			Bits
Data rate (DR)			8, 16, 32, 64, 128, 250, 475, 860		SPS
Data rate variation	All data rates	-10		10	%
Output noise		See Typical Characteristics			
Integral nonlinearity	DR = 8SPS, FS = ±2.048V, best fit ⁽²⁾			1	LSB
Offset error	FS = ±2.048V, differential inputs		±1	±3	LSB
	FS = ±2.048V, single-ended inputs		±3		LSB
Offset drift	FS = ±2.048V		0.005		LSB/°C
Offset power-supply rejection	FS = ±2.048V		1		LSB/V
Gain error ⁽³⁾	FS = ±2.048V at 25°C		0.01	0.15	%
	FS = ±0.256V		7		ppm/°C
	FS = ±2.048V		5	40	ppm/°C
Gain drift ⁽³⁾	FS = ±6.144V ⁽¹⁾		5		ppm/°C
			80		ppm/V
PGA gain match ⁽³⁾	Match between any two PGA gains		0.02	0.1	%
Gain match	Match between any two inputs		0.05	0.1	%
Offset match	Match between any two inputs		3		LSB
Common-mode rejection	At dc and FS = ±0.256V		105		dB
	At dc and FS = ±2.048V		100		dB
	At dc and FS = ±6.144V ⁽¹⁾		90		dB
	$f_{CM} = 60\text{Hz}$, DR = 8SPS		105		dB
	$f_{CM} = 50\text{Hz}$, DR = 8SPS		105		dB
DIGITAL INPUT/OUTPUT					
Logic level					
V_{IH}			0.7VDD	5.5	V
V_{IL}			GND - 0.5	0.3VDD	V
V_{OHL}	$I_{OHL} = 3\text{mA}$		GND	0.4	V
Input leakage					
I_H	$V_{IH} = 5.5\text{V}$			10	μA
I_L	$V_{IL} = \text{GND}$		10		μA

(1) This parameter expresses the full-scale range of the ADC scaling. In no event should more than VDD + 0.3V be applied to this device.

(2) 99% of full-scale.

(3) Includes all errors from onboard PGA and reference.

PARAMETER	TEST CONDITIONS	ADS1113, ADS1114, ADS1115			UNIT
		MIN	TYP	MAX	
POWER-SUPPLY REQUIREMENTS					
Power-supply voltage		2		5.5	V
Supply current	Power-down current at 25°C		0.5	2	μA
	Power-down current up to 125°C			5	μA
	Operating current at 25°C		150	200	μA
	Operating current up to 125°C			300	μA
Power dissipation	VDD = 5.0V		0.9		mW
	VDD = 3.3V		0.5		mW
	VDD = 2.0V		0.3		mW
TEMPERATURE					
Storage temperature		-60		+150	°C
Operating temperature		-40		+140	°C
Specified temperature		-40		+125	°C

PIN CONFIGURATIONS



PIN DESCRIPTIONS

PIN #	DEVICE			ANALOG/ DIGITAL INPUT/ OUTPUT	DESCRIPTION
	ADS1113	ADS1114	ADS1115		
1	ADDR	ADDR	ADDR	Digital Input	I ² C slave address select
2	NC (1)	ALERT/RDY	ALERT/RDY	Digital Output	Digital comparator output or conversion ready (NC for ADS1113)
3	GND	GND	GND	Analog	Ground
4	AIN0	AIN0	AIN0	Analog Input	Differential channel 1: Positive Input or single-ended channel 1 Input
5	AIN1	AIN1	AIN1	Analog Input	Differential channel 1: Negative Input or single-ended channel 2 Input
6	NC	NC	AIN2	Analog Input	Differential channel 2: Positive Input or single-ended channel 3 Input (NC for ADS1113/4)
7	NC	NC	AIN3	Analog Input	Differential channel 2: Negative Input or single-ended channel 4 Input (NC for ADS1113/4)
8	VDD	VDD	VDD	Analog	Power supply: 2.0V to 5.5V
9	SDA	SDA	SDA	Digital I/O	Serial data: Transmits and receives data
10	SCL	SCL	SCL	Digital Input	Serial clock Input: Clocks data on SDA

(1) NC pins may be left floating or tied to ground.

- Lampiran 10 : Tahapan instalasi *Library Keras* dan konfigurasi perangkat *Intel Neural Compute Stick 2*

Instalasi *Library Keras* pada *Raspberry Pi 3B+* dan konfigurasi perangkat *Intel Neural Compute Stick 2* melalui beberapa tahapan diantaranya :

A. *Install OpenCV*

Instalasi *opencv* melalui beberapa tahapan diantaranya :

1. *Expand Filesystem*

Expand filesystem dilakukan untuk memperluas *filesystem*. Tahapan nya yaitu jalankan perintah `sudo raspi-config` tunggu beberapa saat maka akan muncul sebuah tampilan kemudian pilih *Advanced Options* -> *Expand Filesystem* pada tampilan tersebut.

2. Hapus aplikasi yang tidak diperlukan

Hal ini berfungsi untuk menghemat kapasitas Kartu SD. Jika tidak ingin menghapus aplikasi yang tidak diperlukan maka dapat mengabaikan langkah ini.

Tahap ini dilakukan dengan cara menjalankan perintah :

```
sudo apt-get purge wolfram-engine -y
sudo apt-get purge libreoffice* -y
sudo apt-get clean
sudo apt-get autoremove -y
```

3. *Install* perangkat lunak yang dibutuhkan

Tahap ini dilakukan dengan cara menjalankan perintah :

```
sudo apt-get update && sudo apt-get upgrade -y
sudo apt-get install build-essential cmake unzip pkg-config -y
sudo apt-get install libjpeg-dev libpng-dev libtiff-dev -y
sudo apt-get install libavcodec-dev libavformat-dev libswscale-dev libv4l-dev -y
sudo apt-get install libxvidcore-dev libx264-dev -y
sudo apt-get install libgtk-3-dev -y
sudo apt-get install libcanberra-gtk* -y
sudo apt-get install libatlas-base-dev gfortran -y
sudo apt-get install python3-dev -y
```

4. *Download OpenCV*

Tahap ini dilakukan dengan cara menjalankan perintah :

```
cd ~
wget -O opencv.zip https://github.com/opencv/opencv/archive/4.1.0.zip
wget -O opencv_contrib.zip
https://github.com/opencv/opencv_contrib/archive/4.1.0.zip

unzip opencv.zip
unzip opencv_contrib.zip
mv opencv-4.1.0 opencv
mv opencv_contrib-4.1.0 opencv_contrib
```

5. Konfigurasi *Virtual Environment* untuk *Python3*

Tahap ini dilakukan dengan cara menjalankan perintah :

```
sudo pip install virtualenv virtualenvwrapper
sudo pip3 install virtualenv virtualenvwrapper
```

```
sudo rm -rf ~/get-pip.py ~/.cache/pip
```

```
nano ~/.profile
```

Kemudian salin kode berikut pada file *profile* tersebut.

```
export WORKON_HOME=$HOME/.virtualenvs
export VIRTUALENVWRAPPER_PYTHON=/usr/bin/python3
source /usr/local/bin/virtualenvwrapper.sh
```

Simpan perubahan pada file *profile*

Jalankan perintah berikut :

```
source ~/.profile
mkdir cv
cd cv
python3 -m venv venv
. venv/bin/activate
```

6. *Intall Numpy*

Tahap ini dilakukan dengan cara menjalankan perintah :

```
pip install numpy
```

7. *Compile OpenCV*

Tahap ini dilakukan dengan cara menjalankan perintah :

```
cd ~/opencv
mkdir build
cd build
```

```
cmake -D CMAKE_BUILD_TYPE=RELEASE \
-D CMAKE_INSTALL_PREFIX=/usr/local \
-D OPENCV_EXTRA_MODULES_PATH=~/.opencv_contrib/modules \
-D ENABLE_NEON=ON \
-D ENABLE_VFPV3=ON \
-D BUILD_TESTS=OFF \
-D OPENCV_ENABLE_NONFREE=ON \
-D INSTALL_PYTHON_EXAMPLES=OFF \
-D BUILD_EXAMPLES=OFF ..
```

```
sudo nano /etc/dphys-swapfile
```

Ubah `CONF_SWAPSIZE` pada file *dphys-swapfile* menjadi:

```
CONF_SWAPSIZE=2048
```

Jalankan perintah berikut :

```
sudo dphys-swapfile setup
sudo dphys-swapfile swapon
free -m
```

```
make -j4
sudo make install
sudo ldconfig
```

```
In -s /usr/local/lib/python3.7/site-packages/cv2/python-3.7/cv2.cpython-37m-
arm-linux-gnueabihf.so ~/cv/venv/lib/python3.7/site-packages/cv2.so
```

B. Install Tensorflow dan Keras

Tahap ini dilakukan dengan cara menjalankan perintah :

```
sudo apt-get install python3-numpy
sudo apt-get install libblas-dev
sudo apt-get install liblapack-dev
sudo apt-get install python3-dev
sudo apt-get install libatlas-base-dev
sudo apt-get install gfortran
sudo apt-get install python3-setuptools
sudo apt-get install python3-scipy
sudo apt-get update
sudo apt-get install python3-h5py
```

```
cd cv
. venv/bin/activate
```

```
pip install --upgrade scipy
pip install --upgrade cython
pip install tensorflow
pip install keras
```

```
sudo apt-get install python3-matplotlib
pip install matplotlib
```

Jika muncul error `libhdfs` maka lakukan perintah berikut :

```
. cv/venv/bin/activate
sudo apt-get install -y libhdfs5-dev libc-ares-dev libeigen3-dev
python3 -m pip install keras_applications==1.0.8 --no-deps
python3 -m pip install keras_preprocessing==1.1.0 --no-deps
python3 -m pip install h5py==2.9.0
sudo apt-get install -y openmpi-bin libopenmpi-dev
sudo apt-get install -y libatlas-base-dev
python3 -m pip install -U six wheel mock
```

```
wget https://github.com/lhelontra/tensorflow-on-arm/releases/download/v2.0.0/tensorflow-2.0.0-cp37-none-linux_armv7l.whl
python3 -m pip uninstall tensorflow -y
python3 -m pip install tensorflow-2.0.0-cp37-none-linux_armv7l.whl
```

Kemudian restart terminal dan jalankan perintah `. cv/venv/bin/activate`

C. Konfigurasi perangkat Intel Neural Compute Stick 2

Tahap ini dilakukan dengan cara menjalankan perintah :

```
sudo mkdir -p /opt/intel/opencvino
cd ~/Downloads
sudo tar -xvf intel_opencvino_toolkit_runtime_raspbian_p_2020.1.023.tgz --strip 1 -C
/opt/intel/opencvino
sudo apt install cmake
source /opt/intel/opencvino/bin/setupvars.sh
echo "source /opt/intel/opencvino/bin/setupvars.sh" >> ~/.bashrc
```



```
sudo usermod -a -G users pi  
sh /opt/intel/opencvino/install_dependencies/install_NCS_udev_rules.sh2
```

Sambungkan perangkat *Intel Neural Compute Stick 2* dengan Raspberry

- Lampiran 11 : Algoritma *training*

Pada Algoritma 1 merupakan algoritma yang digunakan dalam proses *training* dengan data *input* nama_model, nama_data, freq, dan batas serta data *output* model.

Algoritma 1: Algoritma *Training*

Input: nama_model, nama_data, freq, batas
Output: model

```

1 /* Inisialisasi variabel */
2 df = read_csv(nama_data) // baca file csv
3 ecg = df.ecg
4 jml = len(ecg)
5 window = 0.045 x freq
6 jml_label = jml - window + 1
7 rentang = | min(ecg) - max(ecg) |
8 norm = (ecg/rentang) - (max/rentang) + 0,5
9 label_train = df.label[0:jml_label]
10 ecg_train = zeros(jml_label, window)
11 /* Mengisi array data input untuk training */
12 for i in range (jml_label) do
13 | ecg_train[i] = norm[i : i + window]
14 end
15
16 /* Pembuatan arsitektur untuk training */
17 model = Sequential()
18 model.add(Dense(100, input_dim = window, activation =
   'relu'))
19 model.add(Dense(1, activation = 'sigmoid'))
20 model.compile(loss = 'binary_crossentropy', optimizer =
   'adam')
21
22 /* Proses learning */
23 model.fit(ecg_train, label_train, epochs = 100)
24 /* Menyimpan model ke dalam file */
25 model.save(nama_model)

```

- Lampiran 12 : Algoritma sistem keseluruhan

Pada sistem menggunakan beberapa algoritma diantaranya algoritma akuisisi data, algoritma deteksi QRS, algoritma deteksi puncak R, deteksi *RR-interval*, dan deteksi aritmia.

Algoritma yang digunakan untuk akuisisi data ditunjukkan pada Algoritma 2.

Algoritma 2: Algoritma Akuisisi Data

```

Input: nama_data, freq, waktu
Output: df
1 konfigurasi pin
2 /* Inisialisasi variabel */
3 jml = freq x waktu
4 ecg = []
5 waktu_arr = []
6 time_awal = time.time() x 1000
7 no = [0, 1, 2, ... , jml-1]
8 zero = zeros(jml)
9
10 /* Proses akuisisi data */
11 for i in range (jml) do
12     value = input_dari_sensor // Nilai yang didapat oleh
        sensor
13
14     timee = (time.time() x 1000) - time_awal // Selisih
        waktu sekarang dengan waktu saat mulai akuisisi
15
16     ecg.append(value)
17     waktu_arr.append(timee)
18 end
19
20 /* Menyimpan hasil ke file */
21 Data = {'no': no, 'waktu': waktu_arr, 'ecg': ecg, 'norm':
        zero, 'qrs': zero, 'puncakR': zero,}
22 save_to_csv(Data, nama_data)

```

Algoritma yang digunakan untuk deteksi QRS ditunjukkan pada Algoritma 3.

Algoritma 3: Algoritma Deteksi QRS

Input: nama_model, nama_data, freq
Output: qrs

```

1 /* Inisialisasi variabel */
2 df = read_csv(nama_data) // baca file csv
3 ecg = df.ecg
4 jml = len(ecg)
5 window = 0.045 x freq
6 jml_label = jml - window + 1
7 rentang = | min(ecg) - max(ecg) |
8 norm = (ecg/rentang) - (max/rentang) + 0,5
9 model = load_model(nama_model) // load model dari
   hasil training
10 qrs = zeros(jml)
11
12 /* Proses deteksi QRS */
13 for i in range (jml_label) do
14     hasil_test = model.predict(norm[i : i + window])
15     /* thresholding */
16     if hasil_test >= 0.8 then
17         |
18         |     qrs[i] = norm[i] // menyimpan nilai ke dalam arrat
19         |
20     end
21 end
22
23 /* Menyimpan hasil ke file */
24 Data = 'no': df.no, 'waktu': df.waktu, 'ecg': df.ecg, 'norm':
   norm, 'qrs': qrs, 'puncakR': df.puncakR,
25 save_to_csv(Data, nama_data)

```

Algoritma yang digunakan untuk deteksi puncak R ditunjukkan pada Algoritma 4.

Algoritma 4: Algoritma Deteksi Puncak R

Input: nama_data, nama_data_deteksi, freq**Output:** puncakR, r

```
1 /* Inisialisasi variabel */
2 df = read_csv(nama_data) // baca file csv
3 qrs = df.qrs
4 jml = len(qrs)
5 window = 0.045 x freq
6 jml_label = jml - window + 1
7 stdev = std(df.norm)
8 puncakR = zeros(jml)
9 r = []
10 Rpeak = []
11
12 /* Proses deteksi puncak R */
13 for i in range(jml) do
14     if max(qrs[i - window : i + window]) == qrs[i] then
15         delta1 = |qrs[i]| - |qrs[i-1]|
16         delta2 = |qrs[i]| - |qrs[i-2]|
17         if delta1 > stdev OR delta2 > stdev then
18             puncakR[i] = qrs[i]
19             r.append(i)
20             Rpeak.append(qrs[i])
21         end
22     end
23 end
24 no = [1, 2, 3, ... , len(r)] ; zero=zeros(len(r))
25
26 /* Menyimpan hasil ke file */
27 Data = 'no': df.no, 'waktu': df.waktu, 'ecg': df.ecg, 'norm':
    df.norm, 'qrs': df.qrs, 'puncakR': puncakR,
28 save_to_csv(Data, nama_data)
29 /* Menyimpan hasil ke file */
30 Data2 = 'no': no, 'puncakR': Rpeak, 'r': r, 'rr': zero,
    'aritmia': zero
31 save_to_csv(Data2, nama_data_deteksi)
```

Algoritma yang digunakan untuk deteksi *RR-interval* ditunjukkan pada Algoritma 5.

Algoritma 5: Algoritma Deteksi *RR-Interval*

Input: nama_data,nama_data_deteksi
Output: rr

```
1
2 /* Pembacaan data pada file csv */
3 df = read_csv(nama_data)
4 df2 = read_csv(nama_data_deteksi)
5
6 /* Inisialisasi variabel */
7 waktu=df.waktu
8 r=df2.r
9 jml=len(r)
10 rr=zeros(jml)
11
12 /* Proses deteksi RR-interval */
13 for i in range(jml-1) do
14     |
15     | rr_interval = waktu[r[i+1]] - waktu[r[i]]
16     | rr[i] = rr_interval
17     |
18 end
19
20 /* mengisi array rr pada indeks terakhir dengan nilai
    rr_interval */
21 rr[jml-1] = rr_interval
22
23 /* Menyimpan hasil ke file */
24 Data2 = 'no': df2.no, 'puncakR': df2.puncakR, 'r': df2.r,
    'rr': rr, 'aritmia': df2.aritmia
25 save_to_csv(Data2, nama_data_deteksi)
```

Algoritma yang digunakan untuk deteksi aritmia ditunjukkan pada Algoritma 6.

Algoritma 6: Algoritma Deteksi Aritmia

Input: nama_data_deteksi

Output: aritmia

```
1 /* Inisialisasi variabel */
2 df2 = read_csv(nama_data_deteksi) // baca file csv
3 r = df2.r
4 rr = df2.rr
5 jml = len(r)
6 aritmia = zeros(jml)
7
8 /* ada data aritmia */
9 if jml > 0 then
10     for i in range(jml) do
11         if rr[i] >= 600 AND rr[i] <= 1000 then
12             | aritmia[i] = 0 // normal
13         end
14         else
15             | aritmia[i] = r[i] // tidak normal
16         end
17     end
18 end
19
20 /* Menyimpan hasil ke file */
21 Data2 = 'no': df2.no, 'puncakR': df2.puncakR, 'r': df2.r,
        'rr': df2.rr, 'aritmia': aritmia
22 save_to_csv(Data2, nama_data_deteksi)
```

BIOGRAFI PENULIS



Izzatul Masruroh, lahir pada tanggal 22 Desember 1997 di Kota Surabaya, Provinsi Jawa Timur. Penulis merupakan anak keenam dari delapan bersaudara. Saat ini penulis tinggal di Jojoran 1 Perintis 2 Surabaya. Pada tahun 2010 menyelesaikan pendidikan di MI Muhammadiyah 5 Surabaya, tahun 2013 lulus dari SMP Negeri 19 Surabaya, serta pada tahun 2016 lulus dari SMA Negeri 21 Surabaya. Penulis diterima di Program Studi S1 Departemen Teknik Komputer, Fakultas Teknologi Elektro dan Informatika Cerdas ITS. Penulis aktif menjadi Anggota Unit Kegiatan Mahasiswa Korps Sukarela PMI ITS dan salah satu pemain cabang olahraga futsal putri ITS. Saat di kuliah penulis aktif menjadi Asisten laboratorium B401 Komputasi Multimedia. Penulis memiliki hobby *Travelling* dan futsal. Penulis berhasil menyelesaikan tugas akhir dengan judul "Pendeteksi Penyakit Aritmia berbasis Prediksi Fitur QRS pada Sinyal ECG dengan Metode *Feed Forward Backpropagation* menggunakan Sistem Tertanam". Bagi pembaca yang memiliki kritik, saran atau pertanyaan mengenai tugas akhir ini dapat menghubungi penulis melalui email izzatulmasrurog@gmail.com.

Halaman ini sengaja dikosongkan