

Pengawasan Detak Jantung dan Peringatan Aritmia secara Ubiquitous Menggunakan Protokol MQTT

Tugas Akhir

Kelompok Keahlian: Telematics

**Muhammad Alif Akbar
NIM: 1103132163**



**Program Studi Sarjana Teknik Informatika
Fakultas Informatika
Universitas Telkom
Bandung
2017**

Lembar Pernyataan

Dengan ini Saya menyatakan bahwa Tugas Akhir dengan judul “Pengawasan Detak Jantung dan Peringatan Aritmia secara Ubiquitous Menggunakan Protokol MQTT” beserta seluruh isinya adalah benar-benar karya Saya sendiri dan Saya tidak melakukan penjiplakan atau pengutipan dengan cara-cara yang tidak sesuai dengan etika keilmuan yang berlaku dalam masyarakat keilmuan. Atas pernyataan ini, Saya siap menanggung resiko/sanksi yang dijatuhkan kepada Saya apabila kemudian ditemukan adanya pelanggaran terhadap etika keilmuan dalam karya Saya ini, atau ada klaim dari pihak lain terhadap keaslian karya Saya ini.

Bandung, 1 Agustus 2017

Yang membuat pernyataan,

Muhammad Alif Akbar
NIM: 1103132163

Lembar Pengesahan

Pengawasan Detak Jantung dan Peringatan Aritmia secara
Ubiquitous Menggunakan Protokol MQTT

*Ubiquitous Heart Rate Monitoring and Arrhythmia Alerting
using MQTT protocol*

Muhammad Alif Akbar
NIM: 1103132163

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

Bandung, 1 Agustus 2017

Menyetujui,
Pembimbing

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

NIP: 16730040-1

Mengesahkan,
Kepala Program Studi Teknik Informatika

Ir. Moch. Arif Bijaksana, M.Tech, Ph.D

NIP: 03650312-4

Abstrak

Penyakit jantung (Cardiovascular Diseases, CVDs) merupakan penyakit yang dapat menyerang siapa saja, terjadi kapan saja dan dimana saja. Terdapat banyak produk di pasaran yang dapat melakukan *monitoring* jantung sekaligus merekam aktivitas jantung penggunaannya. Rekam jantung diperlukan oleh dokter jantung untuk melakukan analisis penyakit dan merancang metode pengobatan. Salah satu jenis CVDs yang dapat diidentifikasi dari rekam jantung ialah Aritmia. Aritmia adalah ritme detak jantung yang tidak teratur. Beberapa penelitian sebelumnya telah berhasil mendeteksi Aritmia secara otomatis. Namun produk tersebut tidak dapat melakukan *monitoring* secara terus menerus dan tidak dapat melakukan analisis otomatis. Analisis harus dilakukan secara manual oleh seorang dokter jantung. Di lain pihak, seorang dokter jantung ditargetkan mengawasi sangat banyak pasien. Dalam beberapa kasus, pasien juga memerlukan pengawasan terus menerus. Internet of Things(IoT) telah terbukti handal untuk melakukan monitoring banyak pasien secara terus menerus. Namun implementasi IoT untuk monitoring tersebut masih kurang efisien, seperti tidak adanya notifikasi kepada dokter secara *real time*. Algoritma yang diusulkan pada penelitian sebelumnya pun belum dirancang untuk memaksimalkan kecepatan eksekusi agar dapat memproses banyak pasien secara bersamaan. Oleh karena itu, pada tugas akhir ini dirancang sebuah arsitektur sistem IoT yang menerapkan modifikasi algoritma atas usulan penelitian sebelumnya untuk menyelesaikan masalah diatas. Rancangan memanfaatkan MQTT sebagai protokol komunikasi jaringan agar dapat memproses banyak *sensor* dan banyak *dashboard* sekaligus. Algoritma yang diterapkan pada arsitektur ialah modifikasi algoritma usulan Pan-Tompkin(1985), Tsipouras(2005) dan Kalidas-Tamil(2016). Pan-Tomkins dan Kalidas-Tamil mengusulkan metode untuk melakukan *preprocessing* dan *processing* terhadap hasil baca sensor. Tsipouras mengusulkan metode *rule based classification* untuk mengklasifikasikan aritmia dengan menggunakan fitur *peak* pada rekaman jantung. Berdasarkan hasil pengujian, performa arsitektur sistem yang diusulkan pada tugas akhir ini dinilai baik. Arsitektur dapat melayani maksimum 800 *devices* dengan rata-rata delay 1.72541 ms dan tingkat akurasi 93.11% .

Kata Kunci: Monitoring Jantung, Aritmia, IoT, MQTT.

Abstract

Heart disease (Cardiovascular Diseases, CVD) is a disease that can affect anyone, happens anytime and anywhere. There are many products on the market that can perform cardiac monitoring while recording cardiac activity. Cardiac records are required by cardiologists to perform disease analysis and to design treatment methods. One type of identifiable CVDs of cardiac arrest is arrhythmias. The arrhythmia is an irregular heartbeat rhythm. Several previous studies have successfully detected arrhythmia automatically. However, the product can not monitor continuously and can not perform an automated analysis. The analysis has to be done manually by a cardiologist. On the other hand, a cardiologist is targeted to monitor very many patients. In some cases, patients also require ongoing supervision. Internet of Things (IoT) has proven to be reliable for monitoring many patients on a continuous basis. But the implementation of IoT for monitoring is still less efficient, such as the absence of notification to the doctor in real time. Also, the algorithm proposed in previous studies has not been designed to maximize execution speed in order to process many patients simultaneously. Therefore, in this final project, writer designed an IoT system architecture that applies algorithm modification on the previous research proposal to solve the problem. The architecture design utilizes MQTT as a network communication protocol in order to process multiple sensors and multiple dashboards at once. The algorithm applied to the architecture is the modification of algorithm proposed by Pan and Tompkin (1985), Tsipouras (2005) and Kalidas-Tamil (2016). Pan-Tomkins and Kalidas-Tamil proposed methods for preprocessing and processing of sensor readings. Tsipouras proposed a rule-based classification method to clarify arrhythmias using peak features on heart recordings. Based on the test results, the performance of the proposed system architecture in this final project is considered as good. It can serve a maximum of 800 devices with an average delay of 1.72541 ms and a accuracy rate of 93.11% .

Keywords: Heart Monitoring, Arrhythmia, IoT, MQTT.

Lembar Persembahan

Bismillahirrahmanirrahim, Alhamdulillah, setelah perjalanan sangat panjang penulis dapat menyelesaikan Tugas Akhir ini. Dalam perjalanan panjang ini penulis mendapatkan sangat banyak bantuan dan dukungan dari berbagai pihak. Pada kesempatan ini penulis ingin mengucapkan terima kasih banyak kepada:

1. Allah SWT, yang telah memberikan rahmat dan karunia-Nya kepada penulis, serta kekuatan untuk dapat menyelesaikan Tugas Akhir ini, Alhamdulillah.
2. Keluarga saya, Mama Sophia, Bapak Takdir, dan Adik Nanda, yang tidak hentinya mendukung baik secara materi maupun moral, yang tidak hentinya mendoakan dalam sholat malamnya,
3. Bapak Satria Mandala, ST., MSc., PhD selaku pembimbing yang telah meluangkan banyak waktunya untuk memberikan bimbingan, arahan serta tidak pernah bosan untuk mengingatkan penulis untuk tetap fokus. Semoga Allah selalu memberikan kesehatan dan kemudahan kepada Bapak,
4. Ibu Florita Diana Sari SS, MPd dan Bapak Kiki Maulana Adhinugraha S.Kom., M.T., PhD selaku dosen wali yang telah menyambut penulis di awal perkuliahan, yang selalu memberikan arahan dan dukungan kepada penulis untuk dapat mengarungi dunia perkuliahan dengan baik. Semoga bapak dan ibu diberi kesehatan dan kemudahan selalu dari Allah SWT,
5. Para Staf fakultas dan BK, Pak Said, Pak Bambang, Pak Yasir, yang senantiasa memberikan dukungan dalam penulis berproses di kampus
6. Dosen-dosen yang telah memberikan ilmunya sehingga Penulis mempunyai pengetahuan lebih dibanding sebelumnya.
7. Kawan-kawan satu pembimbing, Shamila, Salim dan lainnya yang telah berbagi derita perjuangan namun tetap saling memotivasi selama Tugas Akhir berlangsung.

8. Kawan-kawan URHUL, Teman tidur, main, makan, main, tidur lagi. Perhatiin alurnya main disebut 2 kali, ya karena serunya kuliah 51% karena main bareng kalian. Fahmi, guru dunia cinta, yang cintanya berakhir di akhir kuliah. Januar yang katanya jago sekali game, baik main dan buatnya. Akbar, contoh terbaik sinetron "Pesantren & Rock n' Roll" dunia nyata. Fachri, yang tidak pernah tidur hanya sedang berfikir. Faruq, contoh ideal cowok metropolitan. Eka, urhul terajin, terserius, terjago tenis. Bragas, sang fotomodel. Zidni dengan style biker kemana saja bahkan ke kelas, Reza yang selalu mengalah disetiap kesempatan, Tesha korban utama penulis latihan sulap plus suhu membaca maksud *chat*, Vira penjual kartu *warewolf* yang bisa memecah sulap terkeren penulis sekali percobaan, Aida yang udah bersedia nama mamanya dipinjam buat penulis ngurus motor, Rafi aa' sunda sejati, Irfan si celana pendek *everywhere*. Terimakasih atas semua kenangan kenangan selama perkuliahan baik yang senang, sedih, dan marah. Semoga untuk kedepannya kalian semua diberi kemudahan dan kesuksesan kedepannya.
9. UKM kedaerahan KBMS dan semua anggotanya, yang menyambut dan membuat penulis selalu merasa memiliki keluarga dan rumah untuk kembali selama jauh dari kampung.
10. Kawan-kawan di Lab Motion, Deas, Devy, Sarah, Adi, Dani, Rizky, Riza, Hanum dan semua anggota yang terlalu banyak untuk disebutkan. Semoga Lab Motion dapat terus berjaya.
11. Kawan-kawan dan Senior di Proclub, Nanda, David, Dean, Caca, Danning, Bagus, Luke, Musa, Ipat, Kak Aul, Kak Wahyu, Kak Byan, Kak Arif, dan Kak Dody yang dengan dorongan mereka penulis dapat mencapai berbagai mimpi selama kuliah
12. Tim BuahBatu, Januar, Kak Taufik, Dede, Pume, Irfa, Fadlu, Ibu Cut, dan Pak Tauhid yang menjadi tim terbaik yang penulis pernah miliki, yang menggapai sangat banyak mimpi bersama, semoga kita semua dapat meraih kesuksesan.
13. Tim Jantung, Shamila, Faida, Qhansa, Mena, Lusi, Husna, dan Hasbi, yang bersama sama penulis menghabiskan uang kampus diakhir perkuliahan.
14. Kawan-kawan liqo Pak Kurniawan, yang bersama mereka penulis dapat meluruskan kembali pikiran, hati dan niat selama berkuliah.
15. Teman-teman kelas IF 37 08, GOIs, Deadliners, dll, yang sudah mengisi hari-hari di masa awal perkuliahan.

16. Semua teman-teman Telkom University dan semua pihak yang tidak dapat disebutkan namanya satupersatu.

Kata Pengantar

Puji syukur atas segala rahmat dan karunia Allah SWT yang telah memberikan kesempatan penulis untuk melaksanakan dan menyelesaikan Tugas Akhir dengan judul “Pengawasan Detak Jantung dan Peringatan Aritmia secara Ubiquitous Menggunakan Protokol MQTT” ini, shalawat serta salam senantiasa dihanturkan kepada junjungan Nabi Muhammad SAW. Tugas Akhir ini disusun sebagai salah satu syarat sidang Tugas Akhir pada program studi Teknik Informatika Fakultas Teknik Informatika Telkom University.

Penulis menyadari masih ada banyak kekurangan pada Tugas Akhir ini, karena itu kritik dan saran sangat diharapkan dan diterima dengan senang hati agar Tugas Akhir ini menjadi lebih baik lagi. Semoga Tugas Akhir ini memberikan manfaat bagi siapapun yang membaca atau turut meneliti apa yang ada di dalam Tugas Akhir ini.

Akhir kata, penulis meminta maaf sebesar besarnya kepada semua pihak apabila ada kesalahan yang penulis lakukan selama proses pembuatan Tugas Akhir ini baik disengaja atau tidak.

Bandung, 1 Agustus 2017

Penulis,

Muhammad Alif Akbar

NIM: 1103132163

Daftar Isi

| | |
|--|------------|
| Abstrak | i |
| Abstract | ii |
| Lembar Persembahan | iii |
| Kata Pengantar | vi |
| Daftar Isi | vii |
| Daftar Gambar | x |
| Daftar Tabel | xii |
| I Pendahuluan | 1 |
| 1.1 Latar Belakang | 1 |
| 1.2 Pernyataan Masalah | 2 |
| 1.3 Perumusan Masalah | 2 |
| 1.4 Batasan Masalah | 2 |
| 1.5 Tujuan | 3 |
| 1.6 Hipotesis | 3 |
| 1.7 Sistematika Penulisan | 4 |
| II Kajian Pustaka | 5 |
| 2.1 Penelitian terkait monitoring aritmia | 5 |
| 2.1.1 Produk Monitoring Jantung di Pasaran | 5 |
| 2.1.2 Riset Terkait | 8 |
| 2.2 ECG dan PPG | 13 |
| 2.2.1 Lokasi Penempatan Sensor | 13 |
| 2.2.2 Titik Fiducial | 15 |
| 2.2.3 Bentuk Sinyal | 16 |
| 2.3 Aritmia | 17 |
| 2.4 Internet of Things | 18 |
| 2.4.1 ESP-12 | 19 |

| | | |
|---|--|-----------|
| 2.4.2 | Node.js | 20 |
| 2.4.3 | MongoDB | 20 |
| 2.5 | Protokol MQTT | 20 |
| 2.6 | Deteksi Puncak R oleh Pan and Tompkins | 22 |
| 2.7 | Deteksi Aritmia oleh Tsipouras | 23 |
| III Metodologi dan Desain Sistem | | 27 |
| 3.1 | Metodologi Penelitian | 27 |
| 3.2 | Gambaran Umum Sistem | 28 |
| 3.3 | Rancangan Perangkat Keras | 29 |
| 3.3.1 | <i>Data Acquisition Unit</i> (DAU) | 29 |
| 3.3.2 | <i>Server Processing Unit</i> (SPU) | 31 |
| 3.3.3 | <i>Client Monitoring Unit</i> (CMU) | 32 |
| 3.4 | Rancangan Proses Pemantauan | 33 |
| 3.5 | Rancangan Proses Deteksi | 33 |
| 3.5.1 | Pengambilan dan Pengiriman Sinyal | 34 |
| 3.5.2 | Preprocessing dan Perekaman | 35 |
| 3.5.3 | Algoritma Deteksi Puncak | 37 |
| 3.5.4 | Algoritma Deteksi Aritmia | 39 |
| 3.5.5 | Visualisasi Gelombang Listrik Jantung | 40 |
| 3.5.6 | Integrasi Algoritma pada Perangkat Keras | 41 |
| 3.6 | Skenario Pengujian | 41 |
| 3.6.1 | Parameter Pengujian | 41 |
| 3.6.2 | Konfigurasi Perangkat Keras | 42 |
| 3.6.3 | Pengujian Algoritma Pemantauan | 44 |
| 3.6.4 | Pengujian Algoritma Deteksi | 45 |
| IV Hasil dan Pembahasan | | 49 |
| 4.1 | Hasil Pengujian | 49 |
| 4.1.1 | Jumlah Fitur Sistem | 49 |
| 4.1.2 | Delay | 50 |
| 4.1.3 | Execution Time | 50 |
| 4.1.4 | Akurasi | 51 |
| 4.2 | Pembahasan | 57 |
| V Kesimpulan dan Saran | | 58 |
| 5.1 | Kesimpulan | 58 |
| 5.2 | Saran | 58 |
| Daftar Pustaka | | 60 |
| 5.3 | Koefisien Proses Filter | 64 |
| 5.4 | Hasil Lengkap Pengukuran Delay | 65 |
| 5.5 | Hasil Lengkap Pengukuran Waktu Eksekusi | 65 |

Daftar Gambar

| | | |
|------|---|----|
| 2.1 | a) Samsung Gear S3; b) Holter Monitor; c) Fingertip Pulse Oximeter; d) Heart Rate App | 7 |
| 2.2 | a) Sensor ECG dengan 3 <i>leads</i> ; b) Sensor PPG ujung jari; c) Sensor PPG di pergelangan tangan | 13 |
| 2.3 | a) Penempatan 3 Elektroda di Dada, b) Penempatan 3 Elektroda di Tangan | 15 |
| 2.4 | Lokasi Penempatan Lebih 3 titik | 15 |
| 2.5 | Sinyal ECG berdasarkan titik fiducial | 16 |
| 2.6 | Sinyal PPG berdasarkan titik fiducial | 16 |
| 2.7 | Perbandingan sinyal ideal PPG dan ECG | 17 |
| 2.8 | Sinkronisasi antara ECG dan PPG | 17 |
| 2.9 | Struktur jantung sederhana | 18 |
| 2.10 | a) Sinyal VF; b) Sinyal PAC; c) Sinyal PVC | 18 |
| 2.11 | Koin Rp200 - ESP-12E - Pulse Sensor | 19 |
| 2.12 | a) Node JS; b) Mongo DB; | 20 |
| 2.13 | MQTT QOS 0 | 21 |
| 2.14 | MQTT QOS 1 | 21 |
| 2.15 | MQTT QOS 2 | 21 |
| 2.16 | Cara kerja MQTT | 22 |
| 2.17 | Preprocessing: Filtering | 22 |
| 2.18 | Processing: Original R Peak Detection | 23 |
| 2.19 | Flowchart of Beat Aritmia Classification | 25 |
| 2.20 | Pseudo code algoritma yang diajukan Tsipouras | 26 |
| 3.1 | Flowchart Metodologi | 27 |
| 3.2 | Gambaran Umum Sistem | 29 |
| 3.3 | Skema Elektronik DAU | 30 |
| 3.4 | Baterai LiPo 3.7v 180mAh | 31 |
| 3.5 | a) Tampilan Web Monitoring b) Tampilan Aplikasi Monitoring . | 32 |
| 3.6 | Flowchart Rancangan Algoritma Pemantauan | 33 |
| 3.7 | Flowchart Rancangan Algoritma Deteksi | 34 |
| 3.8 | Flowchart Pengambilan dan Pengiriman Sinyal | 35 |
| 3.9 | Flowchart Rancangan Algoritma Preprocessing | 37 |

| | | |
|------|--|----|
| 3.10 | Processing: Original R Peak Detection | 38 |
| 3.11 | Flowchart Rancangan Algoritma Preprocessing | 40 |
| 3.12 | Diagram Tahap (<i>Sequence</i>) Algoritma Deteksi | 41 |
| 3.13 | Konfigurasi Perangkat Keras | 44 |
| 4.1 | Hasil Pengukuran Delay 100 sample | 50 |
| 4.2 | Hasil Pengukuran Waktu Eksekusi Pada DAU | 51 |
| 4.3 | Hasil Pengukuran Waktu Eksekusi Pada SPU | 51 |
| 4.4 | Perbandingan Performa Deteksi Data Series 100 | 53 |
| 4.5 | Perbandingan Performa Deteksi Data Series 200 | 53 |
| 4.6 | Perbandingan Waktu Akurasi | 54 |
| 4.7 | Perbandingan akurasi deteksi Aritmia dengan input berbeda . . | 55 |
| 4.8 | Perbandingan sensitivitas deteksi Aritmia dengan input berbeda | 56 |
| 4.9 | Perbandingan spesifisitas deteksi Aritmia dengan input berbeda | 56 |

Daftar Tabel

| | | |
|-----|---|----|
| 2.1 | Perbandingan Produk Berupa Alat | 7 |
| 2.2 | Perbandingan 10 Aplikasi Monitoring Jantung di Play Store . . | 8 |
| 2.3 | Perbandingan Riset Metode Monitoring | 11 |
| 2.4 | Ringkasan riset mengenai klasifikasi aritmia otomasis | 14 |
| 2.5 | Pengelompokan Tipe beat | 24 |
| 3.1 | Tabel Spesifikasi Perangkat Keras | 43 |
| 3.2 | Tabel Rakapitulasi Jumlah Detak | 46 |
| 3.3 | Tabel Rakpitulasi Aritmia Series 100 dengan Kategori | 47 |
| 3.4 | Tabel Rakpitulasi Aritmia Series 200 dengan Kategori | 47 |
| 4.1 | Variabel eksperiment | 52 |
| 4.2 | Hasil Pengujian Algoritma Deteksi Detak Jantung pada Python | 52 |
| 4.3 | Tabel Performa deteksi Aritmia dengan input berbeda | 55 |
| 5.1 | Daftar Koefisien Band Pass Filter | 64 |
| 5.2 | Daftar Koefisien Derivative Filter | 65 |
| 5.3 | Delay Pada Jaringan | 65 |
| 5.4 | Waktu Eksekusi DAU | 65 |
| 5.5 | Waktu Eksekusi SPU | 66 |

Bab I

Pendahuluan

1.1 Latar Belakang

Penyakit jantung (Cardiovascular Diseases, CVDs) merupakan penyakit yang dapat menyerang siapa saja, terjadi kapan saja dan dimana saja. WHO mencatat terjadi sekitar 17,7 juta kematian diakibatkan oleh CVDs di seluruh dunia pada tahun 2015[1]. Dilain pihak menurut Dinas Kesehatan Republik Indonesia, lebih dari 3 juta kematian akibat CVDs terjadi sebelum usia 60 tahun[2]. CVDs juga umumnya tidak memiliki gejala sebelum penyakit menyerang[1]. Oleh karena itu, diperlukan *monitoring* (pemantauan) jantung yang dapat dilakukan secara terus-menerus.

Telah banyak dikembangkan produk untuk melakukan monitoring baik menggunakan *Electrocardiogram* (ECG) [3] dan *Photoplethysmogram* (PPG)[4], [5]. Beberapa produk tersebut bahkan dapat memberitahu jumlah *Beat Per Minute* (BPM, detak jantung tiap menit) dan melakukan perekaman aktivitas jantung. Rekam jantung bermanfaat bagi dokter jantung untuk melakukan analisis penyakit dan merancang metode pengobatan. Salah satu jenis CVDs yang dapat diidentifikasi dari rekam jantung ialah Aritmia.

Menurut literatur, Aritmia adalah ritme detak jantung yang tidak teratur[6]. Aritmia tergolong mudah untuk diidentifikasi berdasarkan gambar rekam jantung seseorang. Sebagian besar jenis aritmia tergolong tidak berbahaya (serangan kecil)[7]. Walaupun tidak berbahaya, jika aritmia tersebut sering diabaikan potensi serangan aritmia yang lebih besar meningkat, contohnya ketika *Premature Ventricular Contraction* (PVC) terjadi berulang kali dapat menandakan terjadinya serangan besar seperti *Ventricular Tachycardia* (VT) dan *Ventricular Fibrillation* (VF) [8]. Berdasarkan studi pustaka yang penulis lakukan, beberapa penelitian sebelumnya telah berhasil membuat algoritma untuk mendeteksi aritmia secara otomatis [8]–[11].

Monitoring jantung yang terus-menerus akan mudah dilakukan kepada pasien yang menjalani perawatan intensif di sebuah rumah sakit. Namun tentunya hal tersebut sulit dilakukan kepada pasien yang menjalani rawat jalan. Akibatnya, aritmia pada pasien yang pernah terkena CVD dan sedang menjalani rawat jalan sulit untuk dipantau.

Produk *monitoring* yang telah ada tidak dapat melakukan *monitoring* secara terus menerus dan tidak dapat melakukan analisis otomatis. Analisis harus dilakukan secara manual oleh seorang dokter jantung. Hal ini tidak menjadi masalah jika jumlah pasien sedikit, namun ternyata jumlah pasien jantung di Indonesia terus meningkat. Di lain pihak, menurut PERKI, Indonesia masih kekurangan jumlah dokter jantung [12]. Bahkan secara ideal pun seorang dokter jantung masih ditargetkan untuk mengawasi sangat banyak pasien (sekitar 100 ribu orang)[12]. Dalam beberapa kasus, pasien juga memerlukan pengawasan terus menerus. Hal ini mengakibatkan seorang dokter tidak mungkin melakukan *monitoring* secara maksimal terhadap semua pasiennya secara terus menerus.

1.2 Pernyataan Masalah

Berdasarkan latar belakang diatas, sub bab 1.1, dapat disimpulkan terdapat permasalahan pada sistem yang sudah ada, yaitu:

1. Sistem yang ada tidak dapat bekerja secara *Ubiquitous* dan terus menerus,
2. Analisis sistem tersebut dilakukan manual oleh dokter jantung sehingga membutuhkan waktu lama,
3. Seorang dokter jantung harus menangani banyak pasien sekaligus.

1.3 Perumusan Masalah

Berdasarkan fakta dan permasalahan yang ditemukan, pada tugas akhir ini penulis merancang sebuah solusi berupa arsitektur sistem monitoring. Yang menjadi rumusan masalah untuk perancangan arsitektur ini ialah sebagai berikut:

1. Bagaimana merancang arsitektur sistem monitoring jantung yang bersifat ubiquitous dan terus menerus?
2. Bagaimana mengembangkan metode *monitoring* dan peringatan yang dapat mengawasi banyak pasien sekaligus?
3. Bagaimana melakukan analisis performansi terhadap arsitektur dan metode yang dikembangkan?

1.4 Batasan Masalah

Untuk membatasi perancangan sistem tugas akhir ini menetapkan batasan sebagai berikut:

1. Sensor dibangun menggunakan PPG dengan kontroller ESP12E;

2. Server dibangun dengan spesifikasi server Processor Intel-i3 (2.3GHz), RAM 6 GB, Storage Samsung SSD EVO 750;
3. Sistem bekerja pada *foreground* atau thread utama
4. Iterkoneksi jaringan menggunakan WiFi;
5. Tidak terdapat hambatan atau masalah komunikasi antara Sensor dan Server;
6. Metode klasifikasi yang diuji ialah metode yang diusulkan oleh Pan-Tomkins dan Tsipouras-Fotiadis
7. Nilai Pulse Arrival Time (PAT) dianggap konstan.

1.5 Tujuan

Berikut adalah tujuan yang ingin dicapai pada penulisan Tugas Akhir yaitu:

1. Merancang arsitektur sistem monitoring jantung yang bersifat ubiquitous dan terus menerus,
2. Mengembangkan metode *monitoring* dan peringatan otomatis yang dapat mengawasi banyak pasien sekaligus,
3. Menganalisis performansi terhadap arsitektur dan metode yang dikembangkan.

1.6 Hipotesis

Hipotesis dari tulisan ini adalah

1. Dengan memindahkan proses perhitungan dan penyimpanan ke server dapat memungkinkan sistem untuk melakukan pemrosesan lebih besar dan cepat,
2. Terdapat fitur yang dimiliki baik oleh ECG maupun PPG sehingga memungkinkan kedua jenis sensor digunakan pada sistem tanpa merubah algoritma klasifikasi,
3. Dengan komunikasi menggunakan MQTT akan memungkinkan banyak pihak untuk mendapat peringatan dari sistem ketika terdeteksi Aritmia,
4. Dengan melakukan pengujian dapat mengetahui performansi dari sistem yang dirancang,

1.7 Sistematika Penulisan

Tugas Akhir ini disusun dengan sistematika penulisan sebagai berikut :

Bab 1 Pendahuluan

Bab ini membahas mengenai latar belakang, rumusan masalah, dan tujuan pengerjaan Tugas Akhir ini. Sekaligus memuat pernyataan mengenai batasan masalah, hipotesis dan sistematika penulisan.

Bab 2 Kajian Pustaka

Bab ini membahas fakta dan teori yang berkaitan dengan perancangan sistem untuk mendirikan landasan berfikir. Dengan menggunakan fakta dan teori yang dikemukakan pada bab ini penulis menganalisis kebutuhan akan rancangan arsitektur sistem yang dibangun.

Bab 3 Metodologi dan Desain Sistem

Bab ini membahas secara rinci tentang rancangan arsitektur dan metode yang dibangun oleh penulis. Sekaligus metode pengujian performansi terhadap arsitektur sistem yang dirancang.

Bab 4 Hasil dan Pembahasan

Pada bab ini membahas tentang hasil pengujian dari implementasi arsitektur sistem. Pengujian dilakukan sesuai skenario yang telah dibuat pada bab sebelumnya.

Bab 5 Kesimpulan dan Saran

Bab ini berisi kesimpulan dari penelitian tugas akhir yang telah dilakukan dan saran yang di perlukan untuk penelitian selanjutnya.

Bab II

Kajian Pustaka

Untuk mendirikan landasan berfikir, bab ini akan membahas fakta dan teori yang berkaitan dengan perancangan sistem tersebut. Dimulai dengan membahas beberapa riset terkait, dilanjutkan dengan penjelasan akan istilah istilah, dan algoritma yang digunakan pada penerapan tugas akhir pada bab 3.

2.1 Penelitian terkait monitoring aritmia

Monitoring jantung bukanlah hal yang baru. Hal ini ditandai dengan banyaknya produk dan riset mengenai *monitoring* jantung. Sebelum berdiskusi lebih jauh tentang riset terkait aritmia ada baiknya penulis mereview beberapa produk terkait monitoring jantung yang ada dipasaran.

2.1.1 Produk Monitoring Jantung di Pasaran

Meningkatnya kesadaran masyarakat akan CVDs mendorong banyak perusahaan untuk membuat produk *monitoring* jantung *portable*. Perusahaan seakan berlomba memproduksi alat monitoring baik yang berstandar medis untuk penggunaan rawat intensif maupun yang tidak berstandar medis untuk penggunaan sehari hari. Selain produk berbentuk alat (*hardware*), produk berbentuk program (*software*) yang hanya memanfaatkan *LED flash* di kamera *smartphone* sebagai sensor PPG juga banyak ditemukan [13].

Salah satu perusahaan yang ikut memproduksi alat *monitoring* ialah perusahaan yang terkenal dengan jam tangan pintarnya yaitu Fitbit. Fitbit mengeluarkan “Fitbit Blaze” pada tahun 2016 [4]. Fitbit Blaze dilengkapi dengan PPG pada bagian belakangnya. Fitbit Blaze juga dilengkapi aplikasi Android untuk melihat hasil *monitoring* secara lengkap. Fitbit Blaze terhubung kepada ponsel Android menggunakan Bluetooth. Fitbit Blaze dirancang sebagai pelengkap *lifestyle* agar penggunaanya dapat menggunakannya sehari hari. Hasil *monitoring* dari Fitbit dapat dibagikan kepada siapapun setelah sebuah *record* selesai direkam.

Sama seperti Fitbit, perusahaan raksasa dari Korea, Samsung, juga mengeluarkan produk *lifestyle* berupa sebuah jam tangan pintar yang dapat melakukan *monitoring* jantung. Produk milik Samsung diberi nama “Gear S3”

(gambar 2.1 a) dan diluncurkan pada tahun 2017 [5]. Gear S3 dilengkapi sensor PPG di bagian belakangnya. Gear S3 dapat terhubung ke jaringan secara *wireless* dengan menggunakan koneksi *Bluetooth* dan *Wi-Fi*. Hasil *monitoring* dari Gear S3 hanya dapat dilihat pada layar yang melengkapinya.

PT. Endo Indonesia, sebuah perusahaan peralatan medis asal Indonesia, juga membuat alat *monitoring* jantung. Tidak seperti Fitbit dan Samsung, produk besutan Endo ditujukan untuk pemakaian medis. Endo memproduksi Holter monitor “EDAN SE-2003” (gambar 2.1 b) dan Pulse Oximeter “EDAN H-10” (gambar 2.1 c). “SE-2003” menggunakan sensor berjenis ECG dan mampu melakukan *monitoring* pada 3 *channel* sekaligus [3]. Berbeda dengan “SE-2003”, “H-10” menggunakan PPG [14]. Pulse Oximeter “H-10”, sesuai namanya, selain mampu *monitoring* jantung (*Pulse*) juga mampu mengukur kadar oksigen dalam darah (*Oximeter*). “H-10” merupakan pulse oximeter yang diletakkan pada ujung jari (*fingertip*). Hasil *monitoring* holter monitor dan pulse oximeter Endo dapat dilihat pada layar produk tersebut dan aplikasi desktop yang menyertainya.

Produk berupa aplikasi/*software* (gambar 2.1 d) dapat ditemukan dengan mudah pada toko aplikasi *virtual* (*playstore*, *app store*, dll). Penulis melakukan studi perbandingan terhadap 10 aplikasi teratas (versi Juli 2017) pada daftar hasil pencarian di *Play Store*, dengan kata kunci “Heart Rate” [13]. Setelah membandingkan aplikasi tersebut, tidak ditemukan aplikasi yang memiliki fitur *monitoring* lengkap untuk menyelesaikan permasalahan yang diajukan pada bab sebelumnya. Hasil perbandingan terhadap 10 aplikasi ini dicantumkan pada tabel 2.2

Semua produk yang disebutkan sebelumnya tidaklah *Open Source*. Sehingga tidak dimungkinkan untuk merubah spesifikasi kinerja produk seperti menaikkan *sample rate* atau mengganti algoritma penghitung detak jantung. Walaupun tidak *Open Source*, baik Fitbit dan Samsung memberikan *Application Programming Interface* (*API*) untuk memudahkan pengembang perangkat lunak membuat sistem yang menggunakan produk mereka. *API* tersebut memungkinkan pengembang mengambil semua data sensor yang tertanam pada produk. Untuk kasus *monitoring* jantung, pengembang dapat mengambil bentuk gelombang jantung dan BPM dari produk. Sedangkan, Endo tidak menyediakan *API* sama sekali. Sehingga, untuk mengakses pengukuran dari produk Endo akan sangat sulit untuk dilakukan (mungkin dilakukan dengan *Reverse Engineering*). Rangkuman perbedaan produk berupa alat dapat dilihat pada tabel 2.1



Gambar 2.1: a) Samsung Gear S3; b) Holter Monitor; c) Fingertip Pulse Oximeter; d) Heart Rate App

| No | Produk | Sensor | Medis | Open | Fitur | | | | | | | |
|----|-----------------|--------|-------|------|-------|---|---|---|---|---|---|---|
| | | | | | A | B | C | D | E | F | G | H |
| 1 | Fitbit Blaze | PPG | N | N | Y | Y | Y | Y | N | Y | N | Y |
| 2 | Samsung Gear S3 | PPG | N | N | Y | Y | Y | N | N | Y | N | Y |
| 3 | ENDO SE-2003 | ECG | Y | N | Y | Y | Y | Y | N | Y | N | N |
| 4 | ENDO H-10 | PPG | Y | N | Y | Y | Y | Y | N | Y | N | N |

Tabel 2.1: Perbandingan Produk Berupa Alat

| No | Aplikasi | Sensor | Fitur | | | | | | | |
|----|---|------------|-------|---|---|---|---|---|---|---|
| | | | A | B | C | D | E | F | G | H |
| 1 | Instant Heart Rate : Heart Rate & Pulse Monitor | PPG | Y | N | Y | Y | N | Y | N | Y |
| 2 | iCare Health Monitor (BP & HR) | PPG | Y | N | Y | N | N | Y | N | Y |
| 3 | Heart Rate Monitor(REPS) | PPG | Y | N | Y | Y | N | Y | N | Y |
| 4 | Runtastic Heart Rate Monitor & Pulse Checker | PPG | Y | N | Y | N | N | Y | N | N |
| 5 | Cardiograph - Heart Rate Meter | PPG | Y | N | Y | Y | N | Y | N | Y |
| 6 | ASUS Heart Rate | PPG | N | N | N | N | N | Y | N | N |
| 7 | Samsung Health | PPG | Y | Y | N | N | N | Y | N | Y |
| 8 | Heart Rate Monitor(Meet Your Need Production) | PPG | N | N | N | N | N | Y | N | N |
| 9 | MobECG | ECG | N | Y | Y | Y | N | Y | N | N |
| 10 | CMS50Dplus | ECG | N | Y | Y | Y | N | Y | N | N |

Tabel 2.2: Perbandingan 10 Aplikasi Monitoring Jantung di Play Store

Ket:

A = Identitas User

B = Real Time Monitoring

C = Melihat Gelombang Jantung

D = Merekam Gelombang Jantung

E = Multiuser Monitoring

F = Deteksi BPM

G = Aritmia Alert

H = Share Result via Network

2.1.2 Riset Terkait

Arsitektur sistem yang dirancang pada tugas akhir ini menerapkan konsep Internet of Things (IoT) dan diharapkan mampu melakukan analisis otomatis, khususnya aritmia. Penerapan IoT kedalam *monitoring* jantung sudah pernah dilakukan sebelumnya, begitupun dengan analisis aritmia secara otomatis. Untuk mengoptimalkan metode yang diajukan pada tugas akhir ini, diperlukan pembelajaran terhadap riset terkait atas kedua hal tersebut.

Pada tahun 2013, Daniel Barata dan rekannya dari Portugal merancang metode *monitoring* yang dapat bekerja secara *portable* [15]. Sensor yang terhubung dengan sebuah tablet android akan berkomunikasi menggunakan protokol MQTT. Penggunaan MQTT memungkinkan *throughput* data yang besar dan banyak sensor yang terhubung secara bersamaan. Metode ini menggunakan sensor ECG dan PPG untuk mengambil data tubuh. Pemilihan MQTT untuk meningkatkan jumlah sensor yang dapat terhubung juga dilakukan oleh Karan Motwani pada penelitiannya di tahun 2016 [16]. Selain digunakan untuk *monitoring* MQTT juga digunakan pada *facebook messenger* [17].

Pada tahun 2014, Paola Pierleoni dan rekannya merancang metode *monitoring* yang melibatkan ponsel dan sensor ECG milik produk “Zephyr HxM-BT” [18]. Zephyr akan mengirimkan sinyal ECG melalui bluetooth menuju ponsel untuk di proses. Metode *monitoring* yang dirancang oleh Pierleoni menambahkan analisis aritmia kedalamnya. Aritmia dianalisis menggunakan *rule based classification* yang diusulkan oleh Tsipouras(2005). Aturan yang diajukan oleh Tsipouras memerhatikan fitur RR interval dari gelombang jantung [8]. Hasil analisis yang dihasilkan dapat dilihat secara *real time* pada layar ponsel.

Pada tahun 2015, M. Surya dan rekannya asal India merancang arsitektur *monitoring* jantung yang dibangun menggunakan Raspberry Pi [19]. Sebuah ECG akan memberikan inputan analog kepada pin GPIO yang dimiliki oleh Raspi. Raspi bertugas untuk mendeteksi detak jantung dan mengirimkan perhitungan BPM ke sebuah *dashboard* melalui jaringan GSM. Setiap data gelombang jantung yang diterima *dashboard* akan disimpan pada database MySQL.

Masih di tahun 2015, Naresh Vemishetty dan rekannya merancang metode *monitoring* yang ditujukan agar memiliki kompleksitas pemrosesan yang rendah [20]. Dengan kompleksitas yang rendah diharapkan metode yang mereka usulkan ini dapat dijalankan pada device yang memiliki kemampuan komputasi yang rendah. Sistem mereka dapat menerima input berupa sinyal ECG dan EEG.

Pada tahun 2016, Vasu Jindal (seorang peneliti dari Universitas Texas) merancang metode *monitoring* yang melibatkan penggunaan ponsel dan *cloud*[21]. Ponsel akan membaca gelombang jantung dari sensor PPG milik produk sejenis Fitbit, kemudian mengirimkannya ke *cloud* untuk di proses. *Cloud* akan memprosesnya menggunakan *deep belief network* dan mengembalikan hasil pengukuran berupa BPM. Setiap data gelombang jantung yang diterima *cloud* akan disimpan sebagai bahan prediksi berikutnya.

Di tempat lain pada tahun 2016, Mamidi Manisha dan rekannya dari India merancang metode *monitoring* yang mirip dengan metode Jindal. Metode Mamidi juga melibatkan ponsel, produk jam tangan pintar dan *cloud processing*[22]. Namun fokus utama dalam sistem yang mereka rancang ialah tentang bagaimana cloud dapat menangani pertukaran data yang besar dan cepat. Sa-

lah satu kunci keberhasilan sistem mereka ialah penerapan *database* No-SQL yang berhasil memaksimalkan kecepatan penulisan data. Selain komunikasi yang cepat, sistem mereka dirancang agar memiliki mekanisme pemberitaan darurat kepada rumah sakit.

Dari keempat riset diatas, masing masing memiliki kelebihan dan kekurangan. Metode Barata dan Motwani dapat menghubungkan banyak sensor sekaligus. Metode Pierloeni dapat medeteksi aritmia tapi hanya pasien yang bisa mengetahuinya. Metode Jindal memiliki sistem analisis terpusat sehingga algoritma yang diterapkan dapat terus berkembang semakin akurat namun hanya bisa menganalisis BPM. Metode Manisha memiliki kehandalan dalam menangani banyak pasien dalam sistem terpusat dan mampu memberikan pemberitaan darurat namun analisis masih harus dilakukan manual. Ringkasan mengenai kelebihan dan kekurangan ini dapat dilihat pada tabel 2.3.

| Judul | Penulis | Tahun | Fitur | | | | |
|--|---|-------|-------|---|---|---|---|
| | | | A | B | C | D | E |
| System of acquisition, transmission, storage and visualization of Pulse Oximeter and ECG data using Android and MQTT | Daniel Barata, Goncalo Louzada et al. | 2013 | N | N | N | Y | N |
| An Android-Based Heart Monitoring System for the Elderly and for Patients with Heart Disease | Paola Pierleoni, Luca Pernini, et al. | 2014 | Y | Y | N | N | Y |
| Healthcare based on IoT using Raspberry Pi | M. Surya Deekshith Gupta, Vamsikrishna Patchava, et al. | 2015 | Y | N | N | N | Y |
| Affordable Low Complexity Heart/Brain Monitoring Methodology for Remote Health Care | Naresh Vemishetty, Pranit Jadhav, et al. | 2015 | Y | N | N | N | N |
| Integrating Mobile and Cloud for PPG Signal Selection to Monitor Heart Rate During Intensive Physical Exercise | Vasu Jindal | 2016 | Y | N | Y | N | N |
| IoT on Heart Attack Detection and Heart Rate Monitoring | Mamidi Manisha, Katakam Neeraja, et al. | 2016 | N | N | Y | Y | Y |

Tabel 2.3: Perbandingan Riset Metode Monitoring

Ket:

A = Deteksi BPM

B = Deteksi Artimia

C = Sistem Deteksi Cloud

D = Optimasi banyak pengguna

E = Sistem Peringatan

Analisis terhadap rekam jantung umumnya dilakukan secara manual oleh dokter jantung. Salah satu bentuk analisis yang dapat dilakukan ialah mengklasifikasi kemunculan aritmia. Untuk mengklasifikasikan aritmia seorang

dokter perlu melihat hasil rekam jantung seorang pasien (gambar 2.10), baik hasil rekam ECG maupun PPG. Namun, seperti yang ditunjukkan oleh Pierleoni (2014) pada penelitiannya dimungkinkan untuk melakukan analisis otomatis, khususnya Aritmia [18]. Selain Pierleoni, banyak penelitian yang telah dilakukan untuk mengklasifikasi aritmia secara otomatis. Bahkan, kini penelitian tersebut telah memiliki keakuratan yang cukup baik, mencapai lebih dari 90%, dengan berbagai macam metode dan ekstraksi fitur.

Pada tahun 1985, Jiapu Pan dan Willis J. Tompkins mengusulkan sebuah metode deteksi QRS secara *Real Time* [23]. Metode ini kemudian menginspirasi peneliti lainnya untuk menerapkannya ke dalam berbagai penelitian mengenai *monitoring* jantung [24], [25]. Salah satu peneliti yang menggunakan metode ini ialah Christos Pavlatos dan rekannya pada tahun 2003 [24]. Christo Pavlatos menerapkan metode Pan dan Tompkins kedalam *embedded system*. Hal ini dimungkinkan karena algoritma pan tompkins cukup hemat dalam penggunaan memori, panjang buffer yg digunakan yaitu 150ms. Algoritma Pan dan Tompkins berhasil mendeteksi detak jantung dengan tingkat sensitivitas 99.62% [23].

Pada tahun 2004, Tsipouras dan rekannya mengajukan algoritma klasifikasi aritmia berdasarkan RR-interval [8]. Klasifikasi yang diajukan berupa *rule-based* yang dirancang bersama dokter ahli jantung. Algoritma yang dianggap simpel dibandingkan penggunaan algoritma *machine learning* seperti SVM [9], ANN [11], dan Regressor. Algoritma ini menghasilkan akurasi 98%.

Pada tahun 2008, Babak Mohammadzadeh dan rekannya mengajukan algoritma yang dapat mendeteksi aritmia menggunakan *Generalized discriminant analysis* (GDA) dan *Support Vector Machine* (SVM) [9]. Algoritma yang diusulkan Mohammadzadeh mampu mengklasifikasikan NSR, PVC, AF, VF and 2° Heart Block pada sinyal ECG. Algoritma yang diusulkan Mohammadzadeh menghasilkan akurasi 99.1%.

Pada tahun 2013, Milan S. Shet dan rekannya mengajukan metode klasifikasi aritmia dengan input ECG [10]. Algoritma mereka melibatkan optimasi *Binary Particle Swarm* dan *Absolute Euclidean Classifier*. Metode Shet menghasilkan akurasi 97.2%.

Pada tahun 2014, Andrius Solosenko dan Vaidotas Marozas mengajukan metode yang dapat mendeteksi PVC (tipe aritmia) yang menggunakan inputan PPG [11]. Algoritma deteksi yang diterapkan menggunakan klasifikasi *artificial neural network* (ANN) dengan 6 fitur pada 12 detik *window*. Algoritma ini menghasilkan spesitivitas 99.85%.

Pada tahun 2015, Luisa F. Polania dan rekannya mengajukan metode klasifikasi aritmia dengan menerima input PPG [26]. Tipe aritmia yang dapat diklasifikasi ialah NSRs, *paced* SVPCs, *paced* VPC dan *paced* VT. SVM digunakan sebagai klasifikasi atas 3 fitur. Algoritma yang mereka ajukan memiliki akurasi 96%.

Pada tahun 2016, Kalidas dan Tamil mengajukan algoritma klasifikasi aritmia yang dapat memproses 3 jenis input, ECG, PPG, dan *Arterial Blood Pressure* (ABP) [27]. Sinyal ECG diproses menggunakan algoritma Pan-Tompkin dan PPG diproses menggunakan "first-order derivative" filter. Klasifikasi yang digunakan kombinasi dari *adaptive threshold* dan SVM. Algoritma yang mereka ajukan menghasilkan spesitivitas 96%

Ringkasan dari hasil keenam riset diatas dapat dilihat pada tabel 2.4.

2.2 ECG dan PPG

Terdapat 2 jenis sensor yang umum digunakan untuk melakukan *monitoring* jantung, yaitu *Electrocardiogram* (ECG) dan *Photoplethysmogram* (PPG) seperti yang terlihat pada gambar 2.2. Kedua jenis sensor ini menjadi pilihan utama dalam *monitoring* jantung karena keduanya mengusung konsep *non-invasive*. Sensor non-invasive memungkinkan melakukan pengambilan data tubuh tanpa perlu melukai/menusuk bagian tubuh tertentu. Secara umum ECG akan menghasilkan pengukuran lebih akurat dari pada PPG. Namun PPG lebih nyaman digunakan dalam jangka panjang dari pada ECG.



Gambar 2.2: a) Sensor ECG dengan 3 *leads*; b) Sensor PPG ujung jari; c) Sensor PPG di pergelangan tangan

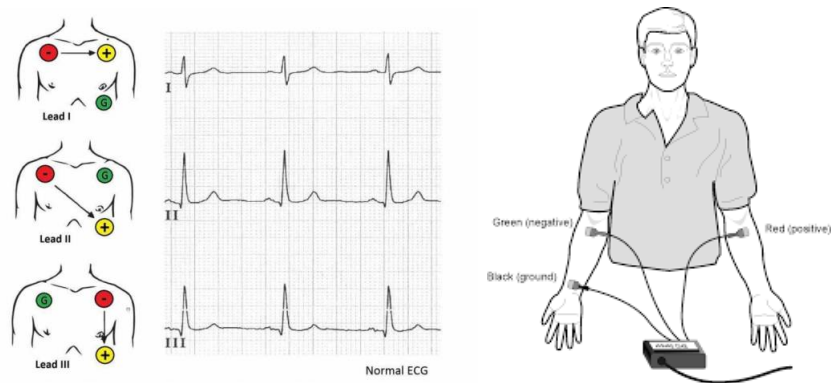
2.2.1 Lokasi Penempatan Sensor

a. ECG

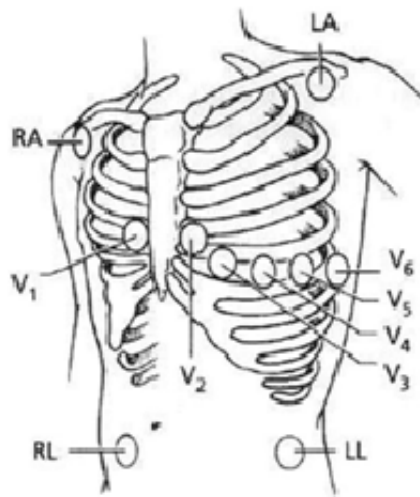
ECG perlu melakukan pengukuran minimal di 3 lokasi. Hal ini disebabkan karena ECG secara langsung mengukur perubahan nilai kelistrikan yang dihasilkan tubuh. Untuk mengukur kelistrikan tubuh ECG memerlukan minimal 3 elektroda yaitu + (positif), - (negatif) dan N (netral), terlihat pada gambar 2.3. Untuk pengukuran lebih dari 3 titik, elektroda ECG dapat diletakkan pada posisi yang ditunjukkan gambar 2.4.

| Judul | Penulis | Ta- hun | Hasil | Sen- sor |
|---|---|------------|---|-------------|
| An arrhythmia classification system based on the RR-interval signal | M.G. Tsipouras, D.I. Fotiadis, <i>et al</i> | 2004 | Algoritma klasifikasi aritmia dengan fitur RR interval dan rule-base decision | ECG |
| Support vector machine-based arrhythmia classification using reduced features of heart rate variability signal | Babak Mohammadzadeh Asl, Seyed Kamaledin Setarehdan, <i>et al</i> | 2008 | Model klasifikasi aritmia berdasar GDA dan SVM | ECG |
| ECG Arrhythmia Classification Using R-Peak Based Segmentation, Binary Particle Swarm Optimization and Absolute Euclidean Classifier | Milan S. Shet, Minal Patel, <i>et al</i> | 2013 | Model particle swarm yang mengklasifikasi segmen R peak | ECG |
| Automatic Premature Ventricular Contraction Detection in Photoplethysmographic Signals | Andrius Solosenko and Vaidotas Marozas | 2014 | Algoritma deteksi PVC pada sinyal PPG | PPG |
| Method for Classifying Cardiac Arrhythmias using Photoplethysmography | Luisa F. Polania, Lalit K. Mestha, <i>et al</i> | 2015 | Algoritma deteksi aritmia berdasarkan 3 fitur PPG | PPG |
| Cardiac arrhythmia classification using multi-modal signal analysis | V Kalidas and L S Tamil | 2016 | Metode klasifikasi aritmia dengan kombinasi input ECG dan PPG | ECG, PPG |

Tabel 2.4: Ringkasan riset mengenai klasifikasi aritmia otomatis



Gambar 2.3: a) Penempatan 3 Elektroda di Dada, b) Penempatan 3 Elektroda di Tangan



Gambar 2.4: Lokasi Penempatan Lebih 3 titik

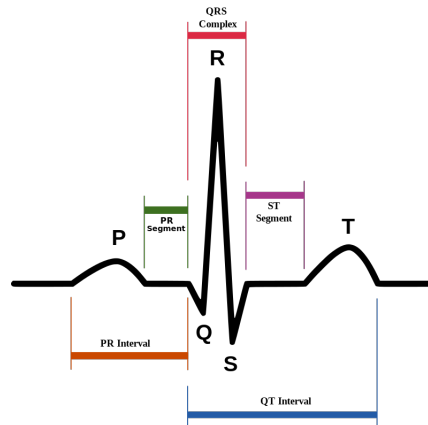
b. PPG

PPG dapat melakukan pengukuran hanya di satu lokasi. Hal ini disebabkan karena PPG mengukur tingkat penyerapan cahaya oleh darah, sedangkan pembuluh darah menjalar ke seluruh tubuh. Namun, untuk menghasilkan pengukuran maksimal PPG perlu diletakkan di lokasi tubuh yang pembuluh darah dekat dengan permukaan kulit [28][29]. Beberapa lokasi tubuh yang sesuai kriteria tersebut terdapat pada ujung jari, pergelangan tangan, lengan atas, leher, dan daun kuping.

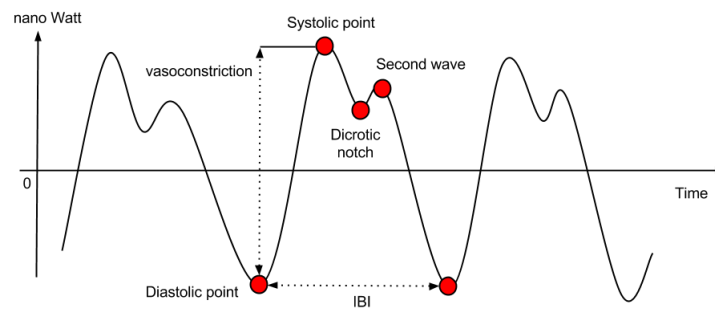
2.2.2 Titik Fiducial

Untuk mengenali sebuah detak jantung pada rekam ECG maupun PPG diperlukan untuk mencari titik titik *fiducial* (pembanding). Kemunculan titik

fiducial menandakan adanya siklus *beat* (detak) pada waktu kemunculan titik tersebut. Sebuah siklus sinyal ECG dapat dilihat dari beberapa titik fiducial yaitu P-QRS-T, seperti terlihat pada gambar 2.5. Sedangkan siklus sinyal PPG dilihat dari siklus Diastolic-Systolic-Dicrotic seperti terlihat pada gambar 2.6



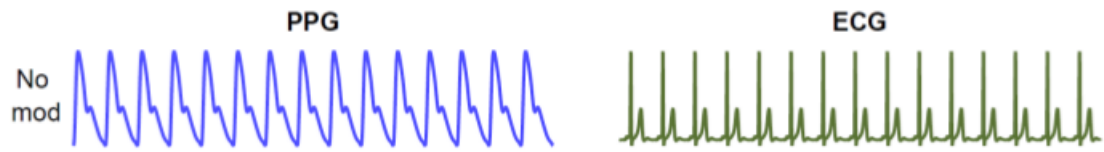
Gambar 2.5: Sinyal ECG berdasarkan titik fiducial



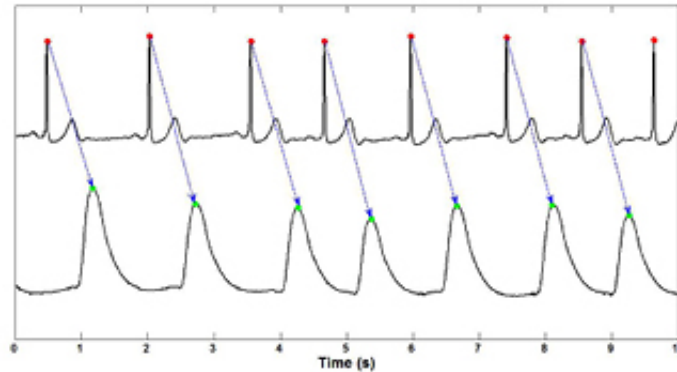
Gambar 2.6: Sinyal PPG berdasarkan titik fiducial

2.2.3 Bentuk Sinyal

Dapat dilihat, pada gambar 2.7, dengan mudah bahwa sinyal bentukan dari PPG dengan ECG berbeda secara morfologi (bentuk)[30]. Namun, karena sumber sinyal yang sama (dari jantung) siklus PPG dan ECG dapat disinkronisasi (saling dipetakan) berdasarkan titik R pada ECG dan puncak sistolik pada PPG seperti gambar 2.8 [27]. Perbedaan waktu kemunculan R dan Sistolik dikenal sebagai *Pulse Arrival Time* (PAT). PAT dapat digunakan sebagai parameter mengukur tekanan darah, yang mana tidak dicakup pada tugas akhir ini.



Gambar 2.7: Perbandingan sinyal ideal PPG dan ECG



Gambar 2.8: Sinkronisasi antara ECG dan PPG

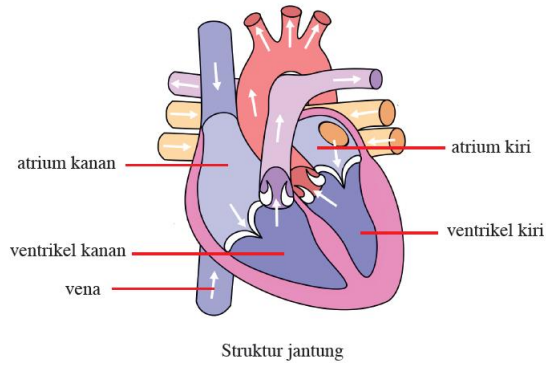
2.3 Aritmia

Aritmia adalah kategori gangguan jantung yang berupa tidak normalnya irama jantung. Beberapa penyakit jantung yang tergolong aritmia antara lain:

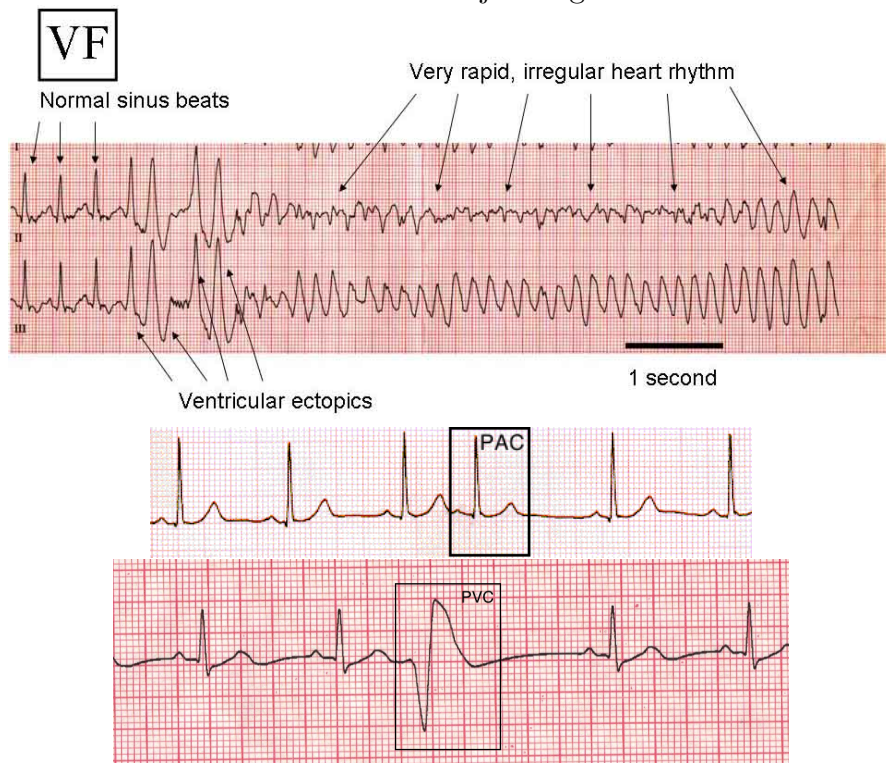
- *Tachycardia* (detak lebih cepat dari normal),
- *Bradycardia* (detak lebih lambat dari normal),
- *Premature Atrial Contraction* (PAC), Kontraksi Atrial Prematur (gambar 2.10 b);
- *Premature Ventricular Contraction* (PVC), Kontraksi Ventrikular Prematur (gambar 2.10 c);
- *Ventricular Tachycardia* (VT) (detak ventrikel sangat cepat), dan
- *Ventricular Fibrillation* (VF), detak ventrikel tidak beraturan (gambar 2.10 a).

Premature Contraction berarti siklus detak yang *premature* (tidak pada waktunya) disebabkan oleh kontraksi pada atrium (bilik) atau ventrikel (serambi) terjadi lebih cepat atau lebih lambat dari seharusnya. *Premature Contraction* tergolong serangan kecil (tidak berbahaya)[xx], sedangkan VT dan

VF tergolong serangan besar (berbahaya). *Premature contraction* yang terjadi berulang kali dan cepat merupakan awal dari peristiwa VT maupun VF. Contoh kemunculan PAC, PVC dan VF dapat dilihat pada gambar 2.10.



Gambar 2.9: Struktur jantung sederhana



Gambar 2.10: a) Sinyal VF; b) Sinyal PAC; c) Sinyal PVC

2.4 Internet of Things

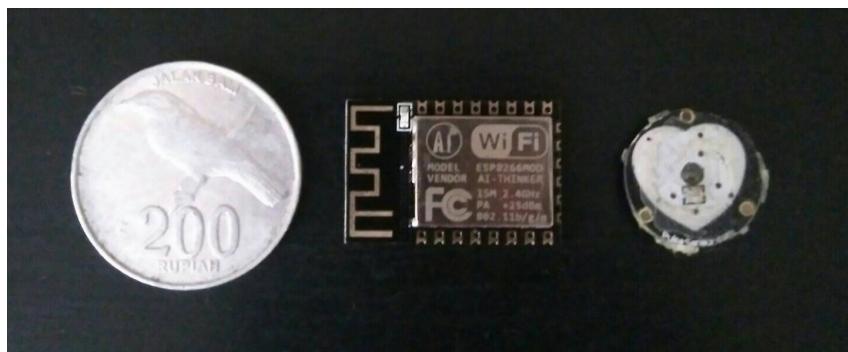
Internet of Things (IoT) ialah konsep dimana objek objek (Things) dapat saling berinteraksi pada jaringan Internet tanpa membutuhkan manusia.

Pada konsep IoT, terdapat sedikitnya 3 komponen dalam arsitekturnya yaitu *sensor*, *server* dan *actuator*. *Sensor* sebagai pengubah data dari dunia nyata ke bentuk digital agar dipahami oleh mesin. *Server* sebagai pemroses data tersebut. *Actuator* sebagai pelaksana keputusan sensor di dunia nyata. IoT memungkinkan komunikasi *Machine to Machine* (M2M) terjadi dalam jaringan Internet. Hal ini berarti *sensor*, *server* dan *receptor* dapat berkomunikasi dimana saja selama dapat terhubung dengan Internet. Selain itu, IoT memodelkan jaringan agar banyak objek (*Things*) dapat saling berkomunikasi. Sehingga, sebuah sistem yang dirancang untuk menangani banyak objek saling berkomunikasi dimana saja, perlu menerapkan IoT.

Sebuah sistem monitoring yang dapat berjalan secara Ubiquitous haruslah dibangun dengan konsep *Internet of Things* (IoT). Sensor yang berfungsi sebagai pengambil data dikendalikan oleh sebuah *controller*. *Controller* adalah bagian yang mengatur jadwal pengambilan data sensor dan pengirimannya ke server. Salah satu jenis controller yang populer ialah ESP-12. Server memerlukan sebuah platform pemrograman yang dapat mengatur pemrosesan data dan komunikasi benda di jaringan. Platform server yang populer untuk penerapan IoT adalah Node.JS. Selain melayani komunikasi, server juga bertugas untuk menyimpan data. Mongo.Db adalah salah satu *database* NoSQL yang populer untuk digunakan dalam sistem yang menyimpan *big data*.

2.4.1 ESP-12

ESP-12 (subversi ESP8266) adalah salah satu tipe *System on Module* (SoC) yang diproduksi oleh Espressif dari China. SoC berarti papan sirkuit yang telah terintegrasi oleh sistem tertentu. Kelebihan utama ESP ialah ukurannya yang kecil (16x24x3 mm) tapi dapat berfungsi sebagai *controller* dan telah dilengkapi modul Wi-Fi. Hal ini memungkinkan komunikasi sensor-server melalui jaringan WiFi tanpa perlu menambah modul jaringan lagi.



Gambar 2.11: Koin Rp200 - ESP-12E - Pulse Sensor

2.4.2 Node.Js

Node.Js adalah teknologi Javascript (Js) *Runtime* yang dibangun diatas Chrome V8 JS Engine. Node.Js memungkinkan bahasa pemrograman Js menjalankan web service. Node.Js dirancang menggunakan skema *event-driven* dan *non-blocking IO*, sangat sesuai untuk aplikasi *data-intensive real-time* [31]. Node.JS juga telah terbukti secara performansi lebih cepat dari bahasa scripting lain seperti PHP, Python, dan Ruby bahkan tidak jauh lambat dibanding bahasa *ter-compile* seperti JAVA, C, dan C++ [32].

2.4.3 MongoDB

MongoDB adalah salah satu jenis program penyimpanan data yang bersifat NoSQL. MongoDB menyimpan data dengan bentuk dokumen dan format JSON. MongoDB dirancang untuk kasus penggunaan yang [33]:

1. Membutuhkan beban penulisan data yang tinggi,
2. Skema data yang tidak stabil,
3. Ukuran data akan menjadi sangat besar,
4. Tidak memiliki seorang *administrator*



Gambar 2.12: a) Node JS; b) Mongo DB;

2.5 Protokol MQTT

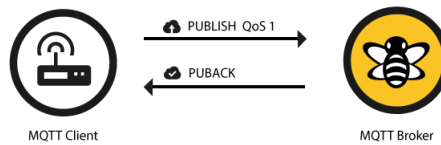
Message Queuing Telemetry Transport (MQTT) adalah protokol transport dengan skema komunikasi publish dan subscribe (gambar 2.16). MQTT dirancang menjadi protokol yang ringan, terbuka dan sederhana. Karakteristik ini membuat MQTT sangat tepat untuk digunakan sebagai protokol komunikasi machine-to-machine (M2M) dan Internet of Things (IoT). Protokol ini menggunakan TCP/IP pada layer transport. Terdapat tiga level Qualities of Service (QoS) dalam penyampaian pesan yaitu:

1. QOS 0 atau “At most once” (gambar 2.13), dimana pesan dikirim dengan skema *fire-and-forget* yang berarti tidak ada upaya menjamin pesan yang dikirim dapat sampai ke tujuan.



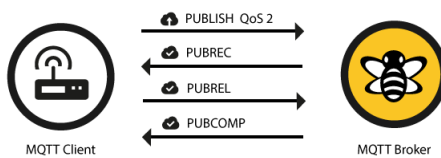
Gambar 2.13: MQTT QOS 0

2. QOS 1 atau “At least once” (gambar 2.14), dimana pesan dikirim dengan jaminan setidaknya pesan sampai sekali ke tujuan. Sehingga memungkinkan terjadinya duplikasi pesan di tujuan akibat pesan yang dikirim ulang dari pengirim.

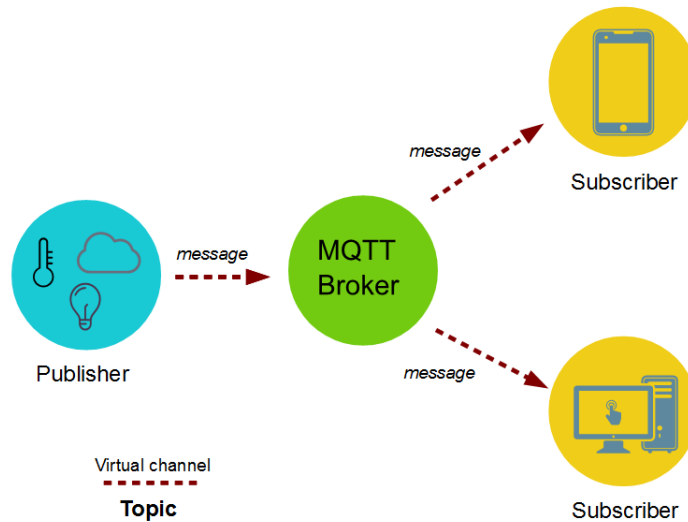


Gambar 2.14: MQTT QOS 1

3. QOS 2 atau “Exactly once” (gambar 2.15), dimana pesan dikirim dengan jaminan diterima tepat sekali ke tujuan. Sehingga tidak ada pesan yang terduplikasi di tujuan.



Gambar 2.15: MQTT QOS 2



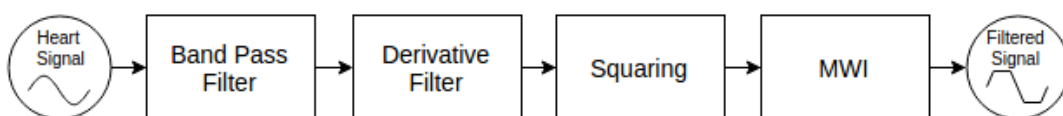
Gambar 2.16: Cara kerja MQTT

2.6 Deteksi Puncak R oleh Pan and Tompkins

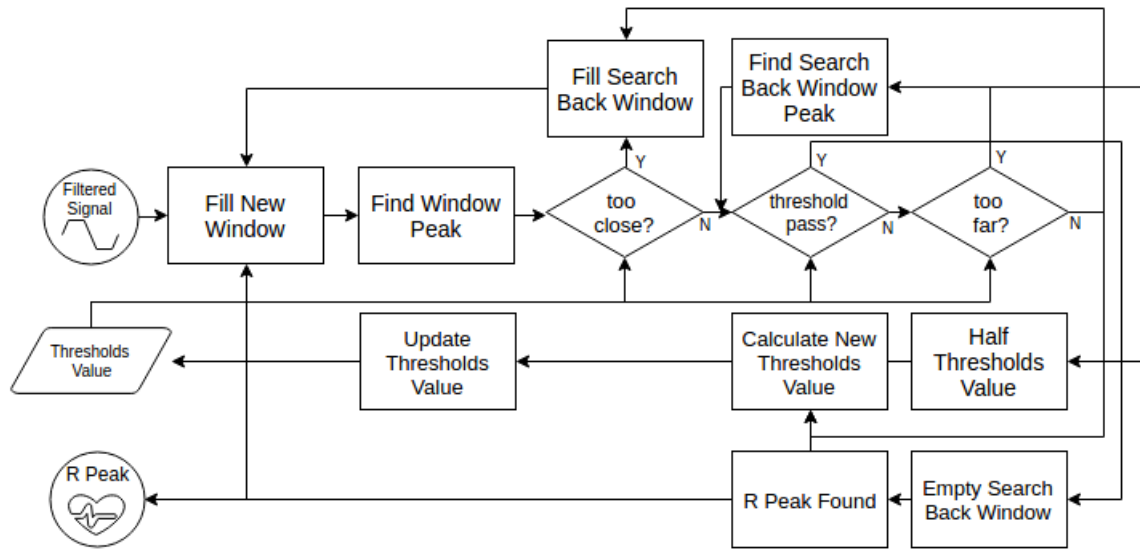
Pan dan Tompkins mengusulkan sebuah algoritma pencarian QRS pada sinyal ECG [23]. Metode ini menjadi populer karena dapat bekerja secara *real-time*. Metode PanTompkins dimulai dari proses processing berupa filtering sinyal. Kemudian dilanjutkan dengan pencarian puncak R menggunakan *adaptive thresholding*.

Pada tahap *preprocessing*, Pan-Tompkins menerapkan 4 proses filtering yaitu Band Pass filtering, Derived filter, Squaring, dan Moving Window Integrator 2.17. Tahapan tersebut dilakukan untuk menghilangkan *noise* pada sinyal dan memudahkan proses pencarian puncak.

Pada tahap processing, Pan-Tompkins menerapkan *adaptive thresholding* terhadap sebuah window sinyal yang berukuran kecil (0.3s) [23]. Pada window tersebut, akan dicari peak yang melewati threshold yang telah ditentukan, kemudian nilai peak akan menjadi input untuk mengupdate nilai threshold. Ketika sekian waktu terlampaui tanpa ditemukannya peak maka harus dilakukan search back hingga posisi R terakhir yang ditemukan (gambar 2.18).



Gambar 2.17: Preprocessing: Filtering



Gambar 2.18: Processing: Original R Peak Detection

2.7 Deteksi Aritmia oleh Tsipouras

Tsipouras et al. mengusulkan algoritma deteksi setelah melakukan percobaan bersama dokter spesialis jantung [8]. Pada penelitiannya, ia membangun *rule-based detector* berdasarkan masukan tim dokter jantung yang terlibat. *Rule* ini dengan menggunakan sebuah (i) *window RR-interval*, yang berisi 3 RR-interval, $RR1_i, RR2_i, RR3_i$. Jenis aritmia yang dapat dideteksinya dibagi menjadi 4 jenis detak aritmia dan 7 jenis ritme aritmia. Namun untuk rancangan pada paper ini aritmia yang dapat dideteksi hanyalah jenis aritmia detak dan dikelompokkan kedalam 3 kategori. Kategori yang dipilih ialah 1) Normal, 2) Premature Contraction, dan 3) Ventricular Flutter. Secara detail pengelompokan ini dapat dilihat pada tabel 2.5.

| MIT-BIH Beat Symbol | Symbol Meaning | Category |
|---------------------------|--|----------|
| N | Normal | 1 |
| . | Normal | |
| / | Paced Beat | |
| f | Fusion of paced and normal beat | |
| L | Left bundle branch block | |
| R | Right bundle branch block | |
| x | Non-conducted P-wave (blocked Atrial Premature Contraction) | |
| Q | Unclassified | 2 |
| V | Premature Ventricular Contraction | |
| A | Premature Atrial Contraction | |
| a | Aberrated Premature Atrial Contraction | |
| J | Nodal (junctional) premature beat | |
| S | Supraventricular premature or ectopic beat (atrial or nodal) | |
| F | Fusion of ventricular and normal beat | 3 |
| ! | Ventricular Flutter | |

Tabel 2.5: Pengelompokan Tipe beat

Algoritma Tsipouras melakukan klasifikasi berdasarkan sebuah window RR interval yaitu $RR1_i$, $RR2_i$, dan $RR3_i$ yang kemudian seterusnya bergeser satu beat. RR interval merupakan jarak waktu antar puncak R dalam satu detik. Tiap beat mulanya dikategorikan sebagai kelas 1 dan kemudian mendapatkan kategori baru sesuai *Rule* yang telah ditetapkan (gambar 2.19).

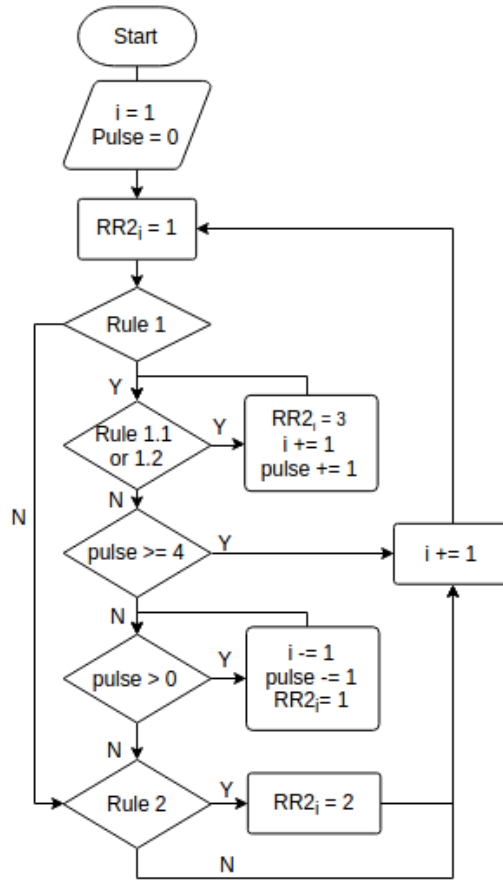
Rule 1. VF: Dimulai ketika $RR1_i > 1.8RR2_i$ dan durasi $RR2_i$ lebih kecil daripada 0.6 s. Maka $RR2_i$ dianggap sebagai awal mula terjandinya VF dan window berikutnya akan dites dengan kedua kondisi berikut:

1. durasi setiap RR interval dalam satu window lebih kecil dari 0.7s
2. jumlah durasi setiap RR interval dalam satu window lebih kecil dari 1.7s

Jika salah satu kondisi tambahan pada rule 1 terpenuhi minimal 4 window berurutan maka RR2 pada tiap window tersebut diklasifikasikan sebagai beat kategori 3. Jika tidak algoritma berlanjut dan window kembali ke posisi awal ditemukannya VF.

Rule 2. PVC: Detak dikategorikan sebagai PVC jika salah satu kondisi berikut terpenuhi:

1. $RR1_i > 1.15RR2_i$ dan $RR3_i > 1.15RR2_i$
2. $|RR1_i - RR2_i| < 0.3s$ dan $RR1_i < 0.8s$ dan $RR2_i < 0.8s$ dan $1.2(RR1_i + RR2_i)/2 < RR3_i$
3. $|RR2_i - RR3_i| < 0.3s$ dan $RR2_i < 0.8s$ dan $RR3_i < 0.8s$ dan $1.2(RR2_i + RR3_i)/2 < RR1_i$



Gambar 2.19: Flowchart of Beat Aritmia Classification

Pseudocode untuk flowchart deteksi aritmia (gambar 2.19) dapat dilihat pada gambar 2.20.


```

1. Initialization
   RR2i from window i is classified as normal (category 1)

2. Rule 1 - Ventricular flutter/fibrillation beat classification
   a. If RR2i < 0.6 sec and 1.8*RR2i < RR1i then
       i. RR2i is classified in category 3.
       ii. The RR2k of all windows k = i+1, i+2, ... i+n with
           (RR1k < 0.7 and RR2k < 0.7 and RR3k < 0.7) or
           (RR1k + RR2k + RR3k < 1.7) are classified in category
           3.
   b. If the number of intervals that are sequentially classified
       in category 3 is less than 4 then they all are classified in
       category 1 and the algorithm returns to window i.

3. Rule 2 - Premature ventricular contractions
   If ((1.15*RR2i < RR1i) and (1.15*RR2i < RR3i)) or
       ((|RR1i - RR2i| < 0.3) and ((RR1i < 0.8) and (RR2i < 0.8)) and
       (RR3i > 1.2*mean(RR1i, RR2i)) or
       ((|RR2i - RR3i| < 0.3) and ((RR2i < 0.8) and (RR3i < 0.8)) and
       (RR1i > 1.2*mean(RR2i, RR3i))
   then RR2i is classified in category 2.

4. Update Window
   i = i + 1

```

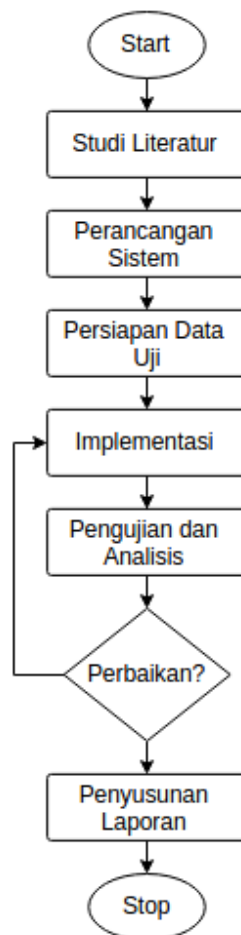
Gambar 2.20: Pseudo code algoritma yang diajukan Tsipouras

Bab III

Metodologi dan Desain Sistem

3.1 Metodologi Penelitian

Diagram alir metodologi yang digunakan untuk menyelesaikan tugas akhir ditampilkan pada gambar 3.1. Berikut penjelasan tiap tahap pada diagram alir tersebut:



Gambar 3.1: Flowchart Metodologi

1. **Studi literatur**

Pada tahap ini penulis mengumpulkan literatur seperti buku, artikel dan *paper* yang berguna menjadi landasan informasi pada penelitian. Hasil tahap ini ialah fakta dan teori serta masalah yang dihadapi.

2. **Perancangan Sistem**

Pada tahap ini penulis memilah masalah yang dapat diselesaikan berdasarkan fakta dan teori yang telah dikumpulkan. Hasil tahap ini ialah rancangan sistem yang diajukan sebagai solusi.

3. **Persiapan Data Uji**

Pada tahap ini penulis mempersiapkan data yang telah tervalidasi kebenarannya untuk dijadikan input pengujian. Hasil tahap ini ialah dataset yang telah dianotasi.

4. **Implementasi**

Pada tahap ini penulis menerapkan rancangan sistem baik yang berupa *software* maupun *hardware*. Hasil tahap ini ialah *software* dan *hardware* yang dapat berjalan tanpa masalah.

5. **Pengujian dan Analisis**

Pada tahap ini penulis melakukan pengujian terhadap sistem yang dibangun menggunakan data uji dan parameter pengujian. Jika ditemukan ada masalah teknis ataupun kemungkinan melakukan peningkatan performansi maka penulis akan kembali ke tahap implementasi. Hasil tahap ini ialah *software* dan *hardware* dengan konfigurasi terbaik yang ditemukan.

6. **Penyusunan Laporan**

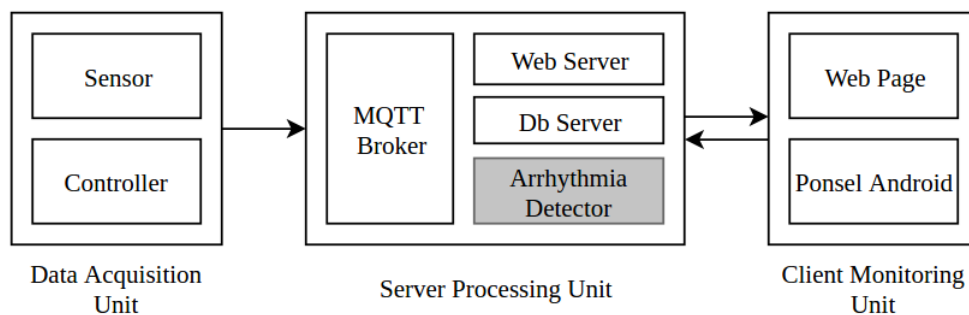
Pada tahap ini penulis melakukan penulisan laporan hasil akhir dari tugas akhir. Hasil dari tahap ini berupa buku tugas akhir dan jurnal penelitian.

3.2 Gambaran Umum Sistem

Untuk menyelesaikan masalah yang ditemukan, penulis merancang sebuah solusi sistem untuk pemantauan jantung. Sistem dirancang untuk bisa dipantau di halaman *web* dan ponsel android. Hasil kalkulasi sistem bukanlah analisa medis, tetapi hanya berupa peringatan. Analisis dokter masih diperlukan untuk mengambil keputusan medis terhadap peringatan yang diberikan oleh sistem. Pengguna sistem ialah dokter, pasien (pengguna yang memakai sensor), dan keluarga pasien. Sistem ditujukan untuk penggunaan non-medis atau sehari-hari yang berfungsi sebagai peringatan dini. Tujuan dari peringatan ini ialah:

1. bagi pasien atau keluarga pasien agar mereka dapat menghubungi dokter untuk melakukan pengecekan lebih lanjut.
2. bagi dokter agar dia dapat merancang pengobatan sesuai analisis dokter tersebut.

Secara umum sistem bekerja dimulai dari pengambilan data jantung menggunakan *Data Acquisition Unit*(DAU) yang diletakkan pada pergelangan tangan. DAU kemudian secara periodik melakukan sampel dan mengirimkan sampel tersebut ke SPU untuk diproses lebih lanjut. Pengguna sistem dapat kapan saja melihat data aktivitas jantung melalui *Client Monitoring Unit*(CMU) berupa halaman *web* atau aplikasi pada ponsel android. Ketika *Server Processing Unit*(SPU) mendeteksi kemunculan aritmia, SPU akan secara otomatis mengirimkan pesan peringatan kepada CMU di pengguna sistem. Arsitektur sistem secara umum digambarkan pada gambar 3.2.



Gambar 3.2: Gambaran Umum Sistem

3.3 Rancangan Perangkat Keras

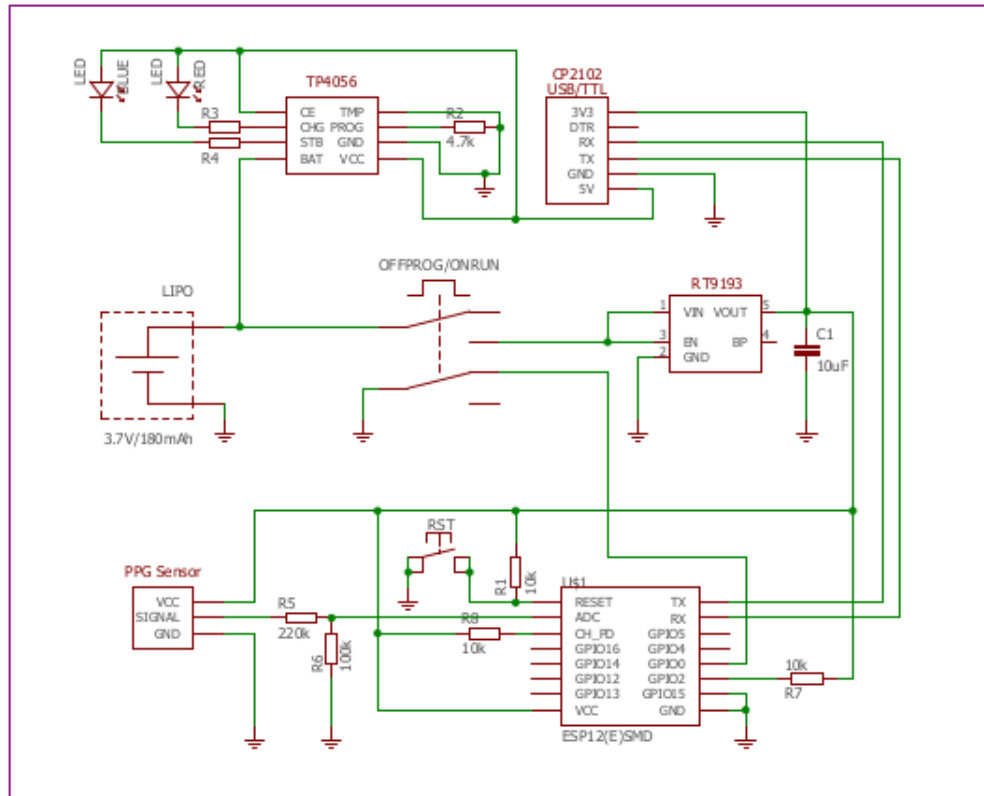
Sistem yang dirancang haruslah diimplementasikan untuk diuji coba. Oleh karena itu perlu dilakukan pemilihan perangkat keras. Perangkat keras dipilih berdasarkan pada kebutuhan rancangan sistem. Perangkat keras dibagi menjadi 3 bagian yaitu *Data Acquisition Unit*(DAU), *Server Processing Unit*(SPU), dan *Client Monitoring Unit*(CMU).

Setelah perangkat keras ditentukan, algoritma yang sesuai untuk diterapkan harus dirancang. Rancangan algoritma terbagi menjadi 2 alur yaitu alur deteksi dan alur pemantauan. Rancangan proses mengenai algoritma dijelaskan lebih lengkap pada sub bab 3.4 dan sub bab 3.5.

3.3.1 *Data Acquisition Unit* (DAU)

Data Acquisition Unit (DAU) berfungsi untuk mengambil data aktivitas jantung seorang pasien. Sistem yang dibangun tidak dapat menggunakan produk *monitoring* yang sudah ada karena sistem tersebut tidak bersifat *Open*

Source. Hal ini mengakibatkan penulis tidak bisa melakukan konfigurasi terhadap *sensor* dan *controller*-nya. Konfigurasi yang dimaksud ialah menaikkan atau menurunkan frekuensi sampel dan transmit. Oleh karena itu penulis merancang DAU khusus untuk penelitian tugas akhir ini. DAU dibangun dengan 3 komponen utama yaitu *Sensor*, *Controller*, dan baterai. Secara lengkap skema elektronik pada DAU dapat dilihat pada gambar 3.3.



Gambar 3.3: Skema Elektronik DAU

a. Sensor

Sistem dirancang untuk mengembangkan produk pemantauan jantung yang sudah ada di pasaran. Berdasarkan pengetahuan yang telah dibangun pada bab kajian pustaka, terdapat 2 jenis sensor yang umum digunakan yaitu ECG dan PPG. Berdasarkan rancangan proses deteksi pada sub bab 3.5 fitur yang dipilih dapat dihasilkan baik oleh ECG maupun PPG. Dengan demikian ECG dan PPG dapat digunakan dalam sistem.

Dalam tugas akhir ini, penulis memilih menggunakan PPG. Sensor PPG yang digunakan merupakan produksi Pulse Sensor yang dirancang oleh Joel dan Yury [34], terlihat pada gambar 2.11. Alasan penulis memilih PPG ialah karena:

1. harganya yang murah,
2. PPG hanya menempel di satu bagian tubuh,
3. PPG berukuran relatif kecil,
4. kekurangan PPG yaitu kurang akurat dibanding ECG, tidak menyalahi tujuan sistem sebagai peringatan dini bukan medis.

b. Controller

Sistem dirancang untuk monitoring terus menerus dan *Ubiquitous*. Maka DAU haruslah cukup kecil untuk dibawa kemana saja dan menggunakan media komunikasi *wireless* (tanpa kabel) untuk berinteraksi dengan SPU. Terdapat banyak jenis media komunikasi *wireless* seperti GSM/CDMA, WiFi, Bluetooth, Infra Red, Zigbee, dll. WiFi dipilih sebagai media, pada sistem, karena jarak cakup yang cukup besar dan mudah untuk dikonfigurasi. Berdasarkan pengetahuan yang telah dibangun pada bab kajian pustaka, terdapat sebuah SoC yang telah memiliki kemampuan *controller* dan memiliki modul WiFi serta berukuran kecil yaitu ESP-12. Oleh karena itu DAU dirancang menggunakan ESP-12.

c. Baterai

Untuk memungkinkan DAU dibawa kemana saja dan dikenakan terus menerus, diperlukan baterai sebagai sumber catuan. Pada tugas akhir ini penulis menggunakan baterai *Li-Polymer* (LiPo) *protected* berkapasitas 180mAh dan tegangan 3.7V. Baterai ini juga berukuran kecil yaitu 25x18x8 mm dan memiliki bobot 5.3 gr, terlihat pada gambar 3.4.



Gambar 3.4: Baterai LiPo 3.7v 180mAh

3.3.2 Server Processing Unit (SPU)

Untuk mengimplemantasikan konsep IoT, sebuah *Server Processing Unit* (SPU) dirancang agar bisa melayani banyak DAU dan CMU. Oleh karena

itu SPU harus mampu melayani komunikasi data dengan arus yang tinggi. Alasan ini mendorong penulis memilih menggunakan protokol MQTT sebagai protokol komunikasi, NodeJs sebagai *runtime* dan MongoDB sebagai pengelola penyimpanan data. SPU dirancang agar bisa berjalan pada satu perangkat. Hal ini berarti MQTT broker, *Web Server*, *Database Server*, dan *Detector* berjalan pada satu alamat IP yang sama.

3.3.3 *Client Monitoring Unit (CMU)*

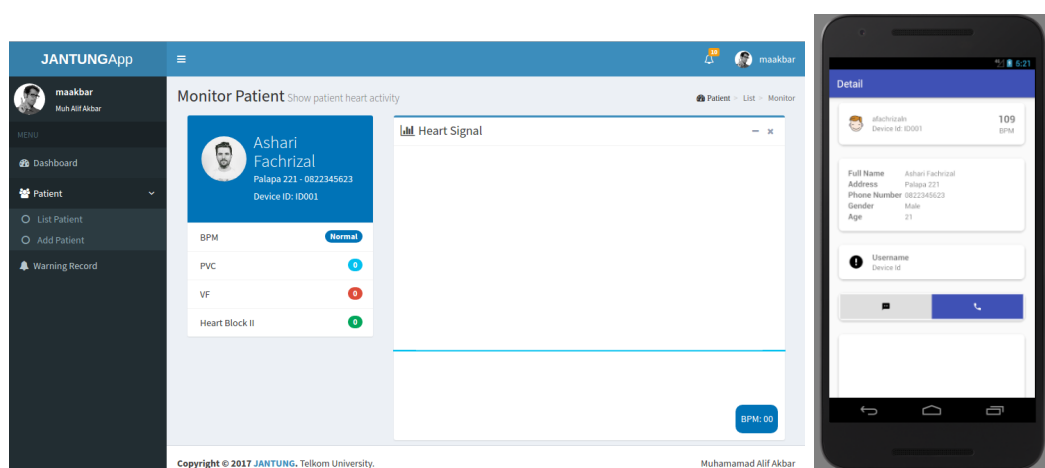
Sistem dirancang memiliki 2 jeni CMU yaitu halaman *web* dan aplikasi pada ponsel Android. Kedua saluran ini dapat melakukan pemantauan selama berada dalam jaringan yang sama dengan SPU. Penulis memilih Android karena memiliki jumlah pengguna terbesar di dunia. Sehingga bisa diasumsikan sistem yang dirancang bisa digunakan oleh banyak orang.

a. Halaman Web

Halaman web dibangun menggunakan *framework* Express.js. Pada halaman web terjadinya aritmia ditandai dengan bunyi dan bertambahnya angka hitungan aritmia yang terdeteksi. Tampilan halaman web dapat dilihat pada gambar 3.5.

b. Aplikasi Ponsel Android

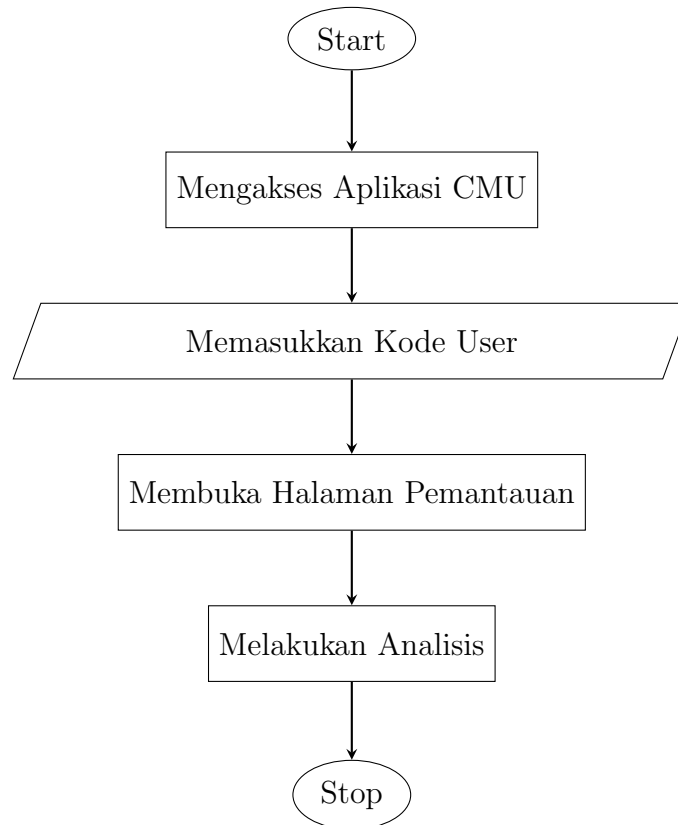
Aplikasi ponsel android dibangun untuk dapat berjalan pada ponsel android ber-OS (*operating system*) minimal Jelly Bean (Android v4.1). Pada aplikasi ini terjadinya aritmia ditandai dengan bunyi atau berubahnya status deteksi dan kode warna ikon seru. Kode warna merah berarti terdeteksi aritmia berbahaya, kuning terdeteksi aritmia tidak berbahaya, dan hijau berarti kondisi normal. Tampilan aplikasi android dapat dilihat pada gambar 3.5.



Gambar 3.5: a) Tampilan Web Monitoring b) Tampilan Aplikasi Monitoring

3.4 Rancangan Proses Pemantauan

Fungsi utama dari sistem yang dibuat ialah melakukan pemantauan. Untuk melakukan pemantauan sistem perlu mengaplikasikan algoritma pemantauan. Algoritma pemantauan hanya berjalan pada perangkat CMU yaitu halaman *web* atau aplikasi ponsel android. Diagram alir algoritma pemantauan dapat dilihat pada gambar 3.6.



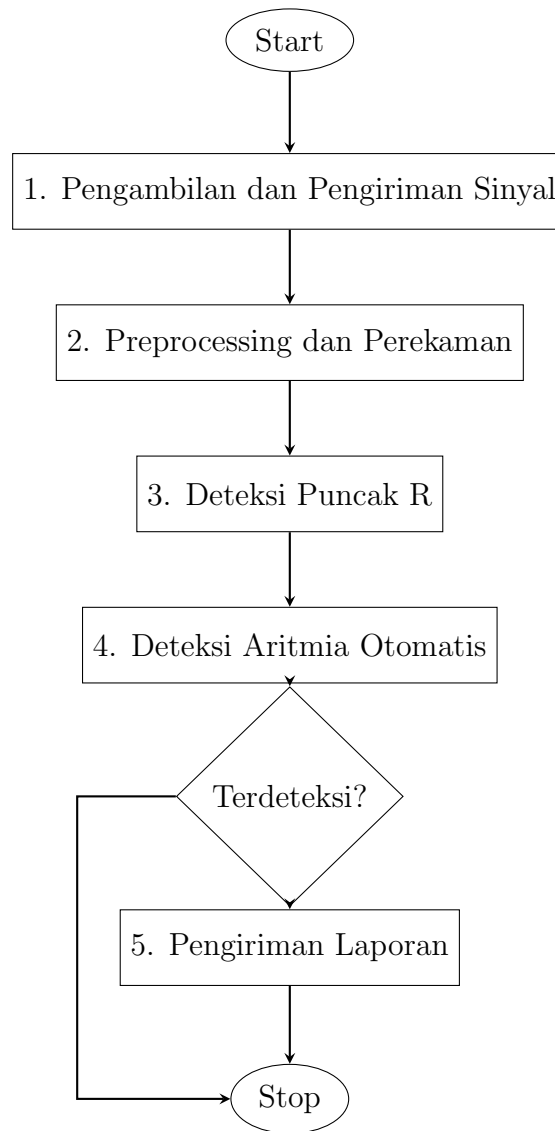
Gambar 3.6: Flowchart Rancangan Algoritma Pemantauan

Flowchart diatas (gambar 3.6) dimulai dengan seorang pengguna baik pasien, keluarga pasien, maupun dokter perlu membuka sebuah perangkat CMU. Setelah aplikasi terbuka, baik web maupun aplikasi ponsel, user perlu memasukkan kode sensor atau user yang ingin dipantau. Setelah kode pantau dimasukkan aplikasi akan membuka halaman pemantauan. Setelah grafik pemantauan mulai berjalan pengguna bisa melakukan analisis.

3.5 Rancangan Proses Deteksi

Fungsi berikutnya yang akan diterapkan dalam sistem ialah dapat melakukan pendeteksian aritmia otomatis. Untuk itu sistem perlu menerapkan algoritma deteksi. Algoritma deteksi yang diterapkan pada tugas akhir ini

terbagi menjadi 5 tahap yaitu Pengambilan dan Pengiriman Sinyal, Preprocessing dan Perekaman, Deteksi Detak Otomatis, Deteksi Aritmia Otomatis dan Pengiriman Laporan. Diagram alir tahap algoritma deteksi digambarkan pada gambar 3.7.

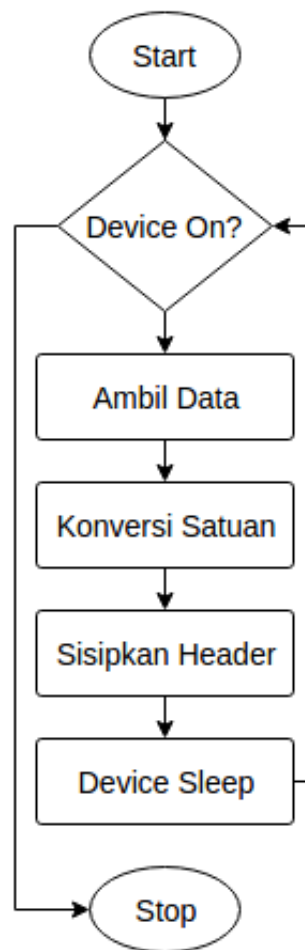


Gambar 3.7: Flowchart Rancangan Algoritma Deteksi

3.5.1 Pengambilan dan Pengiriman Sinyal

Pengambilan dan Pengiriman sinyal dilakukan di DAU. Langkah pertama ialah *controller* mengambil nilai pada pin analognya. Nilai pada pin analog lalu dimasukkan ke dalam sebuah buffer sampel. Selama *buffer* masih berisi maka DAU akan mencoba mengirimkan sampel yang paling pertama masuk (skema

First In First Out). Sebelum dikirim, sampel telah disisipkan header untuk kemudian dikirim menggunakan protokol MQTT dengan QoS 0 kepada SPU. Sampel akan diambil lagi setiap 5ms yang dikendalikan oleh sebuah *thread interrupt*. Ketika tidak ada *interrupt* pengambilan sampel dan *buffer* telah kosong maka DAU akan masuk ke mode sleep. Header berisi kode sensor dan angka index untuk menandakan urutan hasil bacaan. Index tersebut direset setiap angka 1000. Diagram alir untuk memperjelas algoritma bagian ini dapat dilihat pada gambar 3.8.



Gambar 3.8: Flowchart Pengambilan dan Pengiriman Sinyal

3.5.2 Preprocessing dan Perekaman

Setelah data diterima, SPU kemudian melakukan *preprocessing* pada data. *Preprocessing* yang dilakukan terbagi menjadi *data correction*, *filtering*, *squaring*, *moving window integration*(MWI) dan *Adaptive Thresholding*. Setelah *filtering* dilakukan nilai dianggap telah bersih *noise* maka nilai ini di

simpan/rekam di *database*. Preprocessing berlanjut ke *squaring* dan MWI. Terakhir ekstraksi fitur *peak* (R pada ECG dan Systolic pada PPG) dilakukan dengan *Adaptive Thresholding*. Diagram alir algoritma *preprocessing* dapat dilihat pada gambar 3.11. Rangkaian *preprocessing* ini merupakan modifikasi pada algoritma yang diusulkan oleh Pan-Tomkin (1985) (sub bab 2.6) dan Kalidas-Tamil (2016) agar kedua algoritma ini bisa bekerja untuk ECG dan PPG.

a. Data Correction

Data Correction berfungsi ialah algoritma untuk menangani hilangnya data selama pengiriman. Hal ini mungkin terjadi karena data dikirimkan dari DAU menggunakan QoS 0. Pertama, data dipisahkan antara nilai pembacaan sensor dan header. Header kemudian digunakan untuk memisahkan proses perhitungan, setiap kode sensor akan memiliki proses sendiri. Jika terdapat locatan index pada header (index data yg diterima bukan bertambah 1 dari index data sebelumnya) maka proses akan menambah data buffer sebanyak jumlah index yang terlompati dengan nilai berdasarkan proyeksi garis lurus dari nilai terakhir ke nilai terbaru mengikuti persamaan 3.1. Jika tidak ada nilai yang hilang maka nilai akan langsung dimasukkan ke buffer.

$$y(n) = \frac{n(y_2 - y_1)}{d} + v_1 \quad (3.1)$$

v_1 adalah nilai terakhir yang diterima, v_2 adalah nilai terbaru yang diterima, n adalah jarak dari index terakhir, d adalah jarak index terbaru ke terakhir. y adalah nilai index n yang hilang

b. Filtering

Filtering berfungsi untuk menghilangkan noise yang mempengaruhi sinyal. *Noise* yang umum terdapat ialah *muscle noise* dan *baseline wander*. Filtering yang diterapkan terbagi menjadi dua tahap yaitu *Band Pass Finite Impulse Response(FIR) Filter* kemudian *First Order Derivation Filter*. Tahap ini sesuai dengan algoritma yang diusulkan oleh Pan-Tomkins (1985)[23] dan Kalidas-Tamil (2016) [27]. Penerapan algoritma Kalidas-Tamil dilakukan pada desain *Band Pass Filter* yang digunakan. Band Pass Filter ini dirancang untuk memiliki frekuensi response 5-15Hz sehingga bisa memproses sinyal PPG. Daftar koefisien lengkap untuk *Band Pass Filter* tercantum pada tabel 5.3 di bab lampiran. Sedangkan untuk *Derivation Filter* mengikuti persamaan 3.2.

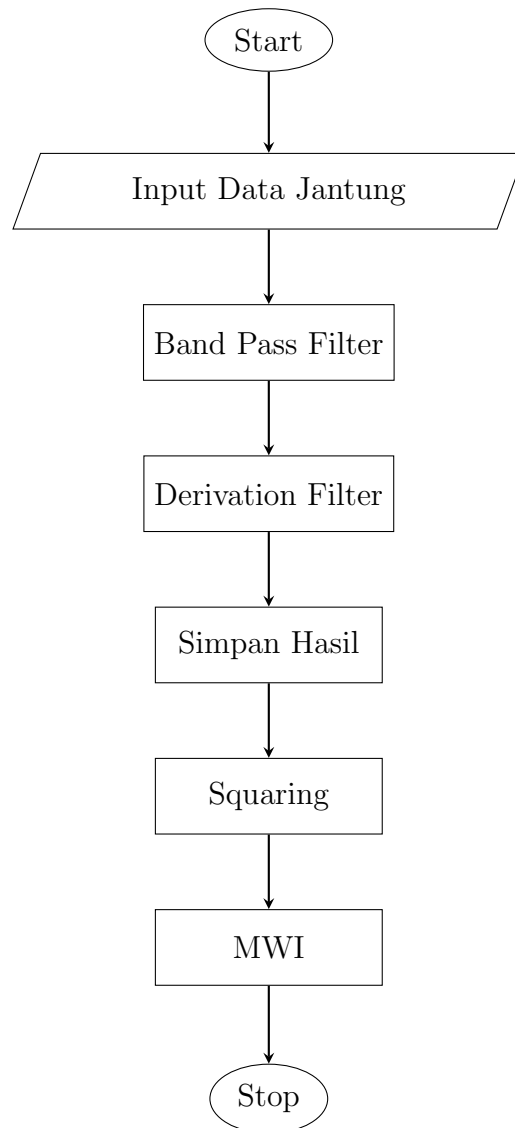
$$y(n) = 2v_n + v_{n-1} - v_{n-3} - 2v_{n-4} \quad (3.2)$$

c. Squaring dan Moving Window Integration (MWI)

Sesuai namanya *Squaring* melakukan penguadratan (persamaan 3.3) terhadap data. Sedangkan MWI melakukan penghalusan data berdasarkan n

data sebelum. MWI dilakukan dengan persamaan menjumlahkan nilai posisi i hingga posisi $i-47$.

$$y = x^2 \quad (3.3)$$



Gambar 3.9: Flowchart Rancangan Algoritma Preprocessing

3.5.3 Algoritma Deteksi Puncak

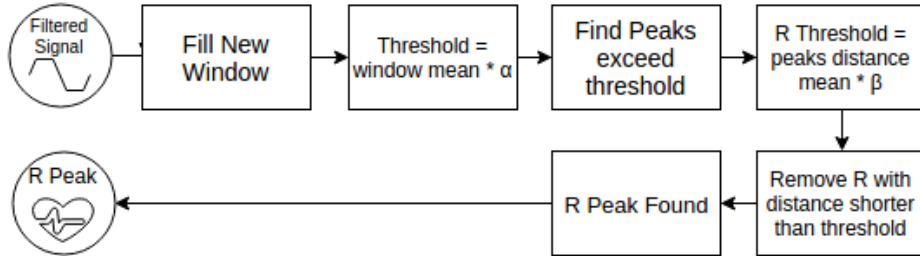
Hal penting untuk dideteksi pada aktivitas jantung ialah detaknya. Tanpa mendeteksi detak deteksi lainnya tidak dapat dilakukan. Deteksi detak dapat dilakukan dengan mencari titik *fiducial*. Titik *fiducial* yang paling menonjol

ialah puncak tertinggi dari siklus detak yaitu R (pada ECG) dan Systolic (pada PPG). Untuk mendeteksi puncak dapat dilakukan dengan *Adaptive Thresholding*. Algoritma yang diterapkan merupakan modifikasi dari yang diusulkan Pan-Tompkins(1985). Pan-Tompkins(1985) mengusulkan teknik *adaptive* dengan merubah nilai *threshold* berdasarkan nilai puncak pada satu detak (sub bab 2.6).

Penulis melihat algoritma Pan-Tompkins dapat dioptimasi sehingga dapat bekerja lebih cepat. Optimasi dilakukan dengan mempebesar window sehingga mengurangi jumlah eksekusi yang perlu dilakukan dalam waktu yang sama, membuat threshold berdasarkan nilai rata-rata window tersebut (persamaan 3.4), menghapus false beat dengan menolak R yang terlalu dekat berdasarkan rata rata jarak R 3.5). Secara detil proses deteksi hasil modifikasi dapat dilihat pada gambar 3.10. Pelebaran window berakibat pada meningkatnya delay atas munculnya hasil deteksi sejak diterimanya sampel namun dapat meningkatkan akurasi deteksi. Pseudocode algoritma deteksi puncak dan penghilang puncak palsu dapat dilihat pada kolom algoritma 1 dan 2.

$$threshold = \alpha \left(\sum_{n=1}^d \frac{v_n}{d} \right) \quad (3.4)$$

$$rThreshold = \beta \left(\sum_{i=1}^j \frac{r_i}{j} \right) \quad (3.5)$$



Gambar 3.10: Processing: Original R Peak Detection

Ket: α merupakan variabel konstan untuk menaikkan threshold. n menunjukkan posisi sampel. d merupakan durasi window, n hingga jumlah sampel pada durasi tersebut. v_n merupakan nilai bacaan pada sample ke- n .

β merupakan variabel konstan untuk menaikkan threshold jarak RR. i menunjukkan index peak. j total peak pada window. r_n merupakan nilai jarak RR ke- n .

Algorithm 1 Fungsi Penentuan Peak

```
1: function PEAKFINDER(window,  $\alpha$ , d)
2:   threshold :=  $\alpha$  * mean(window, d)
3:   for all sample  $\in$  window do
4:     if sample > threshold then
5:       if isPeakArea  $\neq$  true then
6:         isPeakArea := true
7:         tempPeak := sample
8:         peakCounter += 1
9:         peaksArea[peakCounter] := tempPeak
10:      else if sample > tempPeak then
11:        tempPeak := sample
12:        peaksArea[peakCounter] := tempPeak
13:      end if
14:    else
15:      isPeakArea := false
16:    end if
17:  end for
18:  return peaksArea
19: end function
```

Algorithm 2 Prosedur Filter False Peak

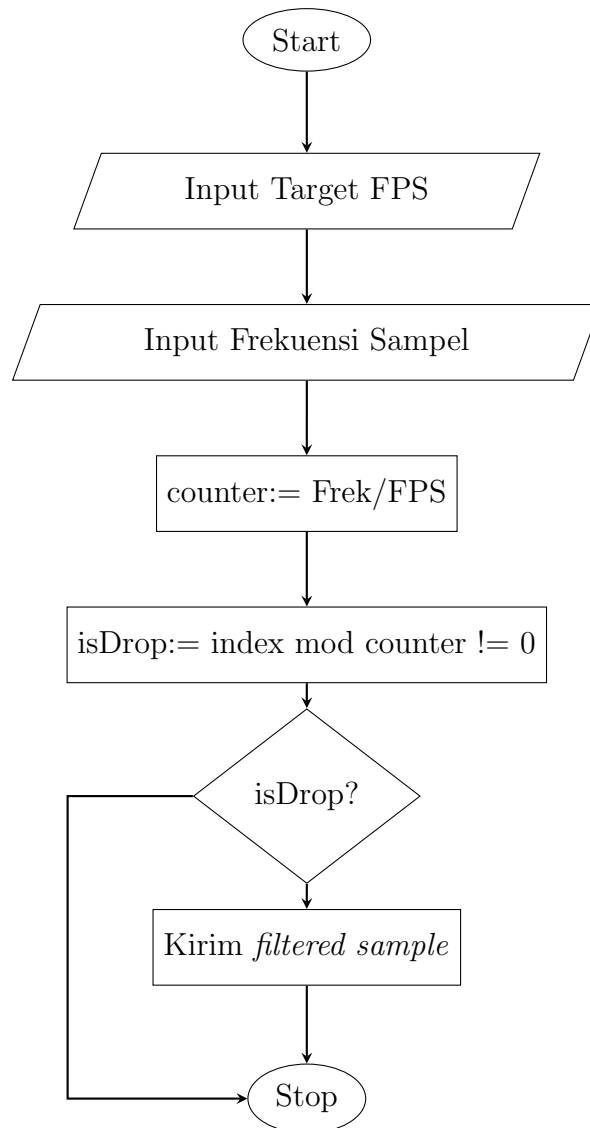
```
1: function FALSEPEAKREMOVAL(peaksArea)
2:   rThreshold :=  $\beta$  * mean(distance(peaksArea))
3:   for all peaki  $\in$  peaksArea do
4:     if distance(peaki, peaki-1) < rThreshold then ▷ Calculate
       distance from peaki to peaki-1
5:       remove(peak, peaksArea) ▷ Remove peak from peak area
6:     end if
7:   end for
8:   return peaksArea
9: end function
```

3.5.4 Algoritma Deteksi Aritmia

Deteksi Aritmia merupakan salah satu tujuan dari tugas akhir ini. Untuk melakukan deteksi aritmia pada tugas akhir ini menerapkan algoritma usulan Tsipouras (sub bab 2.7). Algoritma ini dipilih karena kelebihanannya yaitu dapat melakukan deteksi hanya menggunakan jarak antar titik R pada data ECG. Karena hanya menggunakan titik R, maka algoritma ini juga dapat secara langsung diterapkan pada data PPG.

3.5.5 Visualisasi Gelombang Listrik Jantung

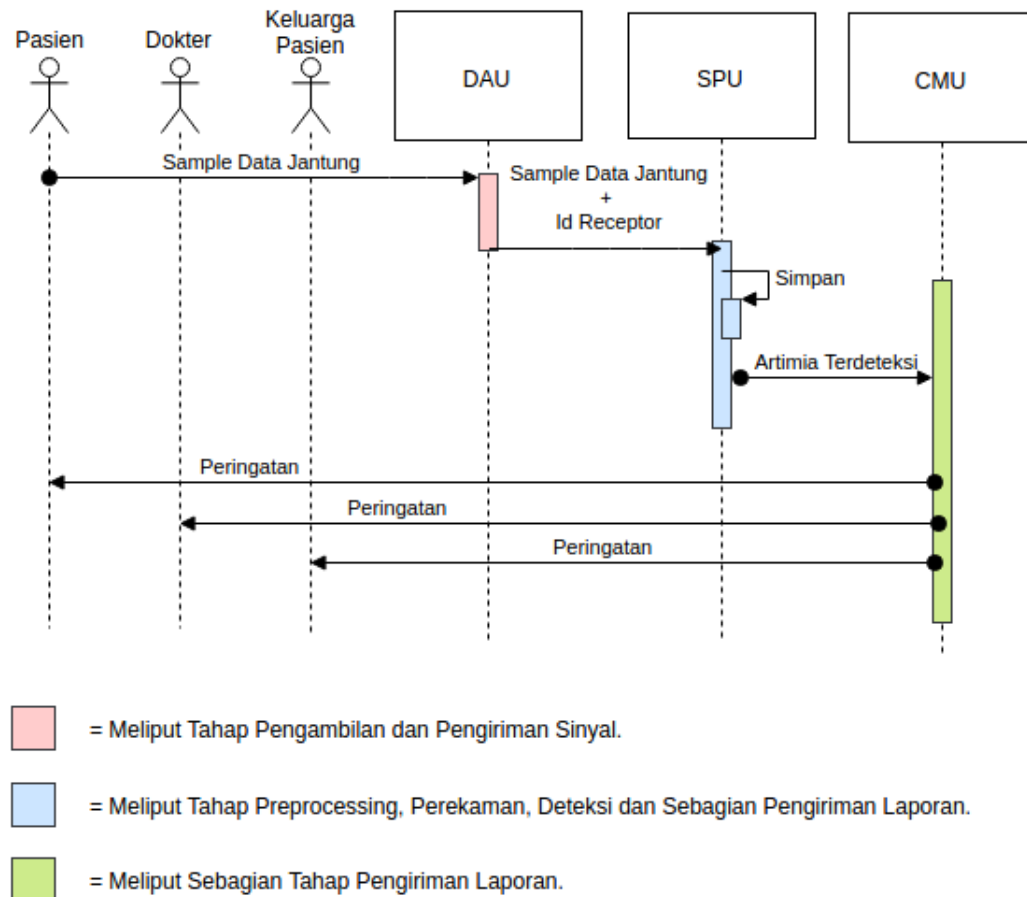
Pengguna dapat menerima laporan atau melakukan pemantauan pada *CMU* mereka. Namun untuk menjaga perangkat *CMU* dapat me-render tampilan dengan baik perlu dilakukan penurunan kecepatan sample (lihat gambar 3.11). Selain itu *CMU* harus terus menerus terkoneksi dengan *SPU* untuk bisa mendapatkan data *real-time*. Sehingga kapanpun *SPU* mendeteksi aritmia pengguna dapat melihat peringatan pada device mereka.



Gambar 3.11: Flowchart Rancangan Algoritma Preprocessing

3.5.6 Integrasi Algoritma pada Perangkat Keras

Tahapan hubungan antara algoritma deteksi dengan perangkat keras digambarkan oleh gambar 3.12. Seorang pasien yang mengenakan *DAU* akan diambil data jantungnya kemudian dikirim ke *SPU*. Ketika terdeteksi aritmia, *SPU* akan mengirim *flag* menandakan aritmia terdeteksi ke *CMU* yang kemudian dilihat oleh pasien, dokter dan keluarga pasien.



Gambar 3.12: Diagram Tahap (*Sequence*) Algoritma Deteksi

3.6 Skenario Pengujian

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

3.6.1 Parameter Pengujian

Berikut hubungan antara parameter pengujian dengan tujuan tugas akhir:

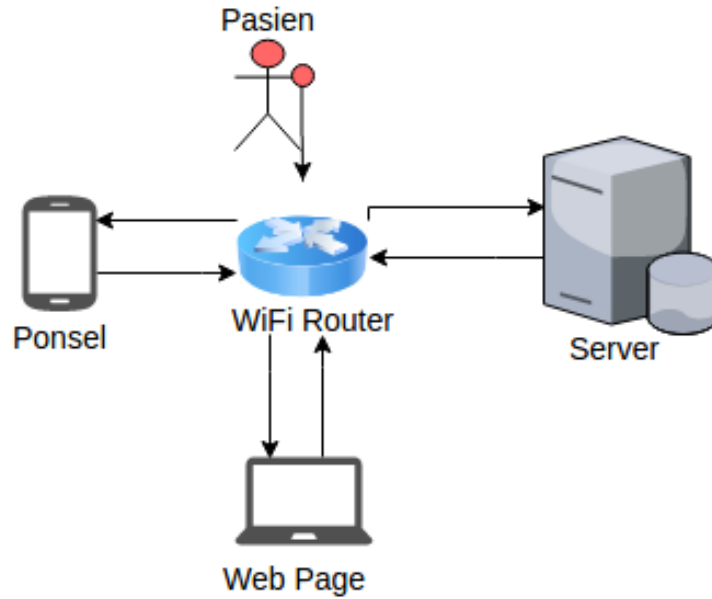
| No | Parameter | Tujuan yg Dicakup | Alasan |
|----|---------------------|-------------------|--|
| 1 | Jumlah Fitur Sistem | 1, 2 | Mengetahui apakah fitur yang direncanakan bisa berjalan. |
| 2 | Delay | 1 | Dengan mengukur <i>delay</i> dapat diketahui berapa lama proses pengiriman sehingga dapat diketahui seberapa <i>Real Time</i> sistem dapat berjalan. |
| 3 | Execution Time | 1, 2 | Dengan mengukur execution time dapat diketahui seberapa banyak sensor yang dapat ditangani oleh spesifikasi SPU tertentu |
| 4 | Akurasi | 3 | Dengan mengukur akurasi dapat diketahui seberapa akurat algoritma deteksi yang diterapkan |

3.6.2 Konfigurasi Perangkat Keras

Spesifikasi perangkat keras yang digunakan selama pengujian dapat dilihat pada tabel 3.1. Pengujian dilakukan dengan menghubungkan sebuah *receiver*, kepada sebuah *SPU* dan dipantau oleh 2 *CMU* (web dan ponsel). Hubungan jaringan perangkat keras digambarkan pada gambar 3.13.

| No | Perangkat | Spesifikasi |
|----|--------------|---|
| 1 | DAU | <ul style="list-style-type: none"> • ESP-12F • Sensor Pulse • Modul USB TTL CP2102 • IC TP4056 LiPo charger • IC LDO regulator RT9193 • Battery LiPo protected 180mAh |
| 2 | SPU | <ul style="list-style-type: none"> • Laptop ASUS A43SD • Intel i3-2350M CPU @ 2.30GHz • RAM DDR3 8GB |
| 3 | CMU Web | <ul style="list-style-type: none"> • Laptop Lenovo G40S • AMD A8 5550M CPU @ 2.1GHz • RAM DDR3 8GB |
| 4 | CMU Android | <ul style="list-style-type: none"> • ZTE A711, Android Lollipop • Snapdragon 615 CPU @ 1.7 GHz • RAM 2GB |
| 5 | WiFi Routers | <ul style="list-style-type: none"> • Jetfi G2 • WiFi IEEE 802.11b/g/n |

Tabel 3.1: Tabel Spesifikasi Perangkat Keras



Gambar 3.13: Konfigurasi Perangkat Keras

3.6.3 Pengujian Algoritma Pemantauan

Sistem tidak bisa dites secara *black box* (memberikan masukan ke sistem dan melihat hasil) karena pengerjaan tugas akhir tidak didampingi oleh dokter ahli jantung untuk melakukan pengecekan atas hasil deteksi. Sehingga pengujian algoritma dipisah menjadi 2 tahap yaitu, algoritma pemantauan dan algoritma deteksi.

Pengujian algoritma pemantauan dilakukan dengan cara seseorang menggunakan *receiver* dan dilihat keberhasilan pemantauan dari *CMU*. Pengujian ini ditujukan untuk menguji semua parameter selain parameter 5 (Akurasi).

a. Delay

Delay yang diukur ialah waktu tempuh sejak dikirimnya data oleh *DAU* hingga diterima oleh *SPU*. *Delay* dihitung dengan mengukur rata-rata selisih waktu (t) diterimanya data oleh *SPU* dikurang dengan waktu kerja sensor (v) (persamaan 3.6).

$$Delay = \alpha \left(\sum_{i=2}^d - \frac{(t_i - v_i)}{d} \right) \quad (3.6)$$

b. Execution Time

Execution Time (Waktu eksekusi) diukur pada *DAU* dan *SPU*. Pada *DAU* *execution time* ialah waktu sejak sampel diambil hingga selesai dikirim ditambah waktu tidur antar sampel, waktu ini disebut sebagai waktu kerja sensor.

Pada *SPU execution time* ialah waktu untuk memproses sebuah sampel hingga dimunculkannya sebuah deteksi. Karena pemrosesan dilakukan setiap sebuah *window* terisi, maka *execution time* yang dihitung dengan membagi durasi *window* terhadap dengan jumlah sampel yang diproses (persamaan 3.7).

$$ET = (\frac{t_d}{d}) \quad (3.7)$$

3.6.4 Pengujian Algoritma Deteksi

Pengujian algoritma deteksi dilakukan dengan memasukkan dataset ECG dari MIT-BIH Arrhythmia Database [35], [36] ke dalam sistem. Untuk mempermudah visualisasi dan analisis data, penulis menjalankan algoritma deteksi pada bahasa pemograman python untuk selanjutnya diimplementasikan ke Node.js. Pengujian ini ditujukan untuk menguji parameter akurasi.

a. Dataset

Dataset dari MIT-BIH [35], [36] terbagi menjadi 48 *records*. Masing masing *record* memiliki panjang 30 menit. Dengan jumlah detak yang berbeda-beda. Rekapitulasi detak yang terekam pada dataset dapat dilihat pada tabel 3.2. Setiap *record* telah dianotasi (ditandai) oleh dokter ahli jantung[36]. Rekapitulasi anotasi pada dataset dapat dilihat pada tabel 3.3 dan 3.4.

| Series 100 | | | Series 200 | | |
|------------|---------------|-------------|------------|---------------|-------------|
| Data Set | Record Length | Number Beat | Data Set | Record Length | Number Beat |
| 100 | 650000 | 2273 | 200 | 650000 | 2601 |
| 101 | 650000 | 1865 | 201 | 650000 | 2000 |
| 102 | 650000 | 2187 | 202 | 650000 | 2136 |
| 103 | 650000 | 2084 | 203 | 650000 | 2980 |
| 104 | 650000 | 2229 | 205 | 650000 | 2656 |
| 105 | 650000 | 2572 | 207 | 650000 | 2332 |
| 106 | 650000 | 2027 | 208 | 650000 | 2955 |
| 107 | 650000 | 2137 | 209 | 650000 | 3005 |
| 108 | 650000 | 1774 | 210 | 650000 | 2650 |
| 109 | 650000 | 2532 | 212 | 650000 | 2748 |
| 111 | 650000 | 2124 | 213 | 650000 | 3251 |
| 112 | 650000 | 2539 | 214 | 650000 | 2262 |
| 113 | 650000 | 1795 | 215 | 650000 | 3363 |
| 114 | 650000 | 1879 | 217 | 650000 | 2208 |
| 115 | 650000 | 1953 | 219 | 650000 | 2287 |
| 116 | 650000 | 2412 | 220 | 650000 | 2048 |
| 117 | 650000 | 1535 | 221 | 650000 | 2427 |
| 118 | 650000 | 2288 | 222 | 650000 | 2483 |
| 119 | 650000 | 1987 | 223 | 650000 | 2605 |
| 121 | 650000 | 1863 | 228 | 650000 | 2053 |
| 122 | 650000 | 2476 | 230 | 650000 | 2256 |
| 123 | 650000 | 1518 | 231 | 650000 | 1573 |
| 124 | 650000 | 1619 | 232 | 650000 | 1780 |
| | | | 233 | 650000 | 3079 |
| | | | 234 | 650000 | 2753 |
| Total Set | Total Length | Total Beat | Total Set | Total Length | Total Beat |
| 23 | 14950000 | 47668 | 25 | 16250000 | 5409 |

Tabel 3.2: Tabel Rakapitulasi Jumlah Detak

| Series 100 | | | | | | | | | | | | | | | | | |
|------------|-------|------|------|------|----|-----|---|---|---|----|------|-----|---|----|---|----|---|
| Data Set | N | / | L | R | Q | f | j | E | e | x | V | A | a | J | S | F | ! |
| 100 | 2239 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 33 | 0 | 0 | 0 | 0 | 0 |
| 101 | 1860 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 0 |
| 102 | 99 | 2028 | 0 | 0 | 0 | 56 | 0 | 0 | 0 | 0 | 4 | 0 | 0 | 0 | 0 | 0 | 0 |
| 103 | 2082 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 |
| 104 | 163 | 1380 | 0 | 0 | 18 | 666 | 0 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 |
| 105 | 2526 | 0 | 0 | 0 | 5 | 0 | 0 | 0 | 0 | 0 | 41 | 0 | 0 | 0 | 0 | 0 | 0 |
| 106 | 1507 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 520 | 0 | 0 | 0 | 0 | 0 | 0 |
| 107 | 0 | 2078 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 59 | 0 | 0 | 0 | 0 | 0 | 0 |
| 108 | 1739 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 11 | 17 | 4 | 0 | 0 | 0 | 2 | 0 |
| 109 | 0 | 0 | 2492 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 38 | 0 | 0 | 0 | 0 | 2 | 0 |
| 111 | 0 | 0 | 2123 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 112 | 2537 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 |
| 113 | 1789 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 6 | 0 | 0 | 0 | 0 |
| 114 | 1820 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 43 | 10 | 0 | 2 | 0 | 4 | 0 |
| 115 | 1953 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 116 | 2302 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 109 | 1 | 0 | 0 | 0 | 0 | 0 |
| 117 | 1534 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 118 | 0 | 0 | 0 | 2166 | 0 | 0 | 0 | 0 | 0 | 10 | 16 | 96 | 0 | 0 | 0 | 0 | 0 |
| 119 | 1543 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 444 | 0 | 0 | 0 | 0 | 0 | 0 |
| 121 | 1861 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| 122 | 2476 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 123 | 1515 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 0 |
| 124 | 0 | 0 | 0 | 1531 | 0 | 0 | 5 | 0 | 0 | 0 | 47 | 2 | 0 | 29 | 0 | 5 | 0 |
| Total | 31545 | 5486 | 4615 | 3697 | 25 | 722 | 6 | 0 | 0 | 21 | 1346 | 155 | 6 | 31 | 0 | 13 | 0 |
| Category | 1 | | | | | | | | | | 2 | | | | | | 3 |
| Total | 46117 | | | | | | | | | | 1551 | | | | | | 0 |

Tabel 3.3: Tabel Rakpitulasi Aritmia Series 100 dengan Kategori

| Series 200 | | | | | | | | | | | | | | | | | |
|------------|-------|------|------|------|---|-----|-----|-----|----|-----|------|------|-----|----|---|-----|-----|
| Data Set | N | / | L | R | Q | f | j | E | e | x | V | A | a | J | S | F | ! |
| 200 | 1743 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 826 | 30 | 0 | 0 | 0 | 2 | 0 |
| 201 | 1625 | 0 | 0 | 0 | 0 | 0 | 10 | 0 | 0 | 37 | 198 | 30 | 97 | 1 | 0 | 2 | 0 |
| 202 | 2061 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 19 | 36 | 19 | 0 | 0 | 1 | 0 |
| 203 | 2529 | 0 | 0 | 0 | 4 | 0 | 0 | 0 | 0 | 0 | 444 | 0 | 2 | 0 | 0 | 1 | 0 |
| 205 | 2571 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 71 | 3 | 0 | 0 | 0 | 11 | 0 |
| 207 | 0 | 0 | 1457 | 86 | 0 | 0 | 0 | 105 | 0 | 0 | 105 | 107 | 0 | 0 | 0 | 0 | 472 |
| 208 | 1586 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 992 | 0 | 0 | 0 | 2 | 373 | 0 |
| 209 | 2621 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 383 | 0 | 0 | 0 | 0 | 0 |
| 210 | 2423 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 194 | 0 | 22 | 0 | 0 | 10 | 0 |
| 212 | 923 | 0 | 0 | 1825 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 213 | 2641 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 220 | 25 | 3 | 0 | 0 | 362 | 0 |
| 214 | 0 | 0 | 2003 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 256 | 0 | 0 | 0 | 0 | 1 | 0 |
| 215 | 3195 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 164 | 3 | 0 | 0 | 0 | 1 | 0 |
| 217 | 244 | 1542 | 0 | 0 | 0 | 260 | 0 | 0 | 0 | 0 | 162 | 0 | 0 | 0 | 0 | 0 | 0 |
| 219 | 2082 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 133 | 64 | 7 | 0 | 0 | 0 | 1 | 0 |
| 220 | 1954 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 94 | 0 | 0 | 0 | 0 | 0 |
| 221 | 2031 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 396 | 0 | 0 | 0 | 0 | 0 | 0 |
| 222 | 2062 | 0 | 0 | 0 | 0 | 0 | 212 | 0 | 0 | 0 | 0 | 208 | 0 | 1 | 0 | 0 | 0 |
| 223 | 2029 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 16 | 0 | 473 | 72 | 1 | 0 | 0 | 14 | 0 |
| 228 | 1688 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 362 | 3 | 0 | 0 | 0 | 0 | 0 |
| 230 | 2255 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 231 | 314 | 0 | 0 | 1254 | 0 | 0 | 0 | 0 | 0 | 2 | 2 | 1 | 0 | 0 | 0 | 0 | 0 |
| 232 | 0 | 0 | 0 | 397 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1382 | 0 | 0 | 0 | 0 | 0 |
| 233 | 2230 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 831 | 7 | 0 | 0 | 0 | 11 | 0 |
| 234 | 2700 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 0 | 0 | 50 | 0 | 0 | 0 |
| Total | 43507 | 1542 | 3460 | 3562 | 8 | 260 | 223 | 106 | 16 | 172 | 5784 | 2391 | 144 | 52 | 2 | 790 | 472 |
| Category | 1 | | | | | | | | | | 2 | | | | | | 3 |
| Total | 52856 | | | | | | | | | | 9163 | | | | | | 472 |

Tabel 3.4: Tabel Rakpitulasi Aritmia Series 200 dengan Kategori

b. Akurasi

Pengukuran akurasi terbagi menjadi 2 yaitu akurasi deteksi detak dan akurasi deteksi aritmia. Untuk mengukur akurasi jumlah kelas (detak dan aritmia) digunakan persamaan 3.8, 3.9 dan 3.10.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (3.8)$$

$$Specificity = \frac{TN}{TN + FP} \quad (3.9)$$

$$Sensitivity = \frac{TP}{FN + TP} \quad (3.10)$$

| | Predicted: No | Predicted Yes |
|-------------|---------------|---------------|
| Actual: No | TN | FP |
| Actual: Yes | FN | TP |

Bab IV

Hasil dan Pembahasan

4.1 Hasil Pengujian

Setelah melaksanakan pengujian sistem seperti yang telah dibahas pada bab sebelumnya sub bab ini akan memaparkan hasil dari percobaan.

4.1.1 Jumlah Fitur Sistem

Pemantauan (aktivitas jantung) dan Deteksi (detak dan aritmia) berhasil dilakukan secara realtime (parameter lainnya, dibahas pada sub bab berikutnya). Berikut perbandingan mode *monitoring* pada sistem di Tugas Akhir ini dengan sistem sejenis lainnya yang berada pada puncak 10 Android Playstore (kata pencarian heart rate) [13].

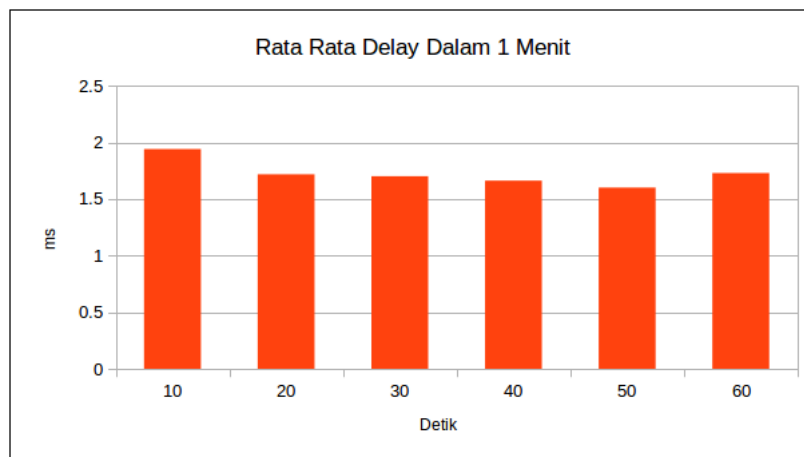
| No | Produk | Sensor | Fitur | | | | | | | |
|----|---|------------|----------|----------|----------|----------|----------|----------|----------|----------|
| | | | A | B | C | D | E | F | G | H |
| 1 | Instant Heart Rate : Heart Rate & Pulse Monitor | PPG | Y | N | Y | Y | N | Y | N | Y |
| 2 | iCare Health Monitor (BP & HR) | PPG | Y | N | Y | N | N | Y | N | Y |
| 3 | Heart Rate Monitor(REPS) | PPG | Y | N | Y | Y | N | Y | N | Y |
| 4 | Runtastic Heart Rate Monitor & Pulse Checker | PPG | Y | N | Y | N | N | Y | N | N |
| 5 | Cardiograph - Heart Rate Meter | PPG | Y | N | Y | Y | N | Y | N | Y |
| 6 | ASUS Heart Rate | PPG | N | N | N | N | N | Y | N | N |
| 7 | Samsung Health | PPG | Y | Y | N | N | N | Y | N | Y |
| 8 | Heart Rate Monitor(Meet Your Need Production) | PPG | N | N | N | N | N | Y | N | N |
| 9 | MobECG | ECG | N | Y | Y | Y | N | Y | N | N |
| 10 | CMS50Dplus | ECG | N | Y | Y | Y | N | Y | N | N |
| * | Tugas Akhir | PPG | Y | Y | Y | Y | Y | Y | Y | Y |

Ket:

- A = Identitas User
- B = Real Time Monitoring
- C = Melihat Gelombang Jantung
- D = Merekam Gelombang Jantung
- E = Multiuser Monitoring
- F = Deteksi BPM
- G = Aritmia Alert
- H = Share Result via Network

4.1.2 Delay

Pengujian dilakukan oleh pengguna yang bergerak secara bebas dalam wilayah cakupan router (*DAU*, *router* dan *SPU* masih dalam satu wilayah) sehingga tidak ada proses routing antara router. Setelah melakukan pengujian sebanyak 10 kali didapatkan rata-rata delay sekitar 1.72541 ms. Hasil rata rata pengukuran delay tiap 10 detik dalam 1 menit tertera pada gambar 4.1. Tabel pengukuran delay yang lengkap terlampir pada lampiran 5.4.



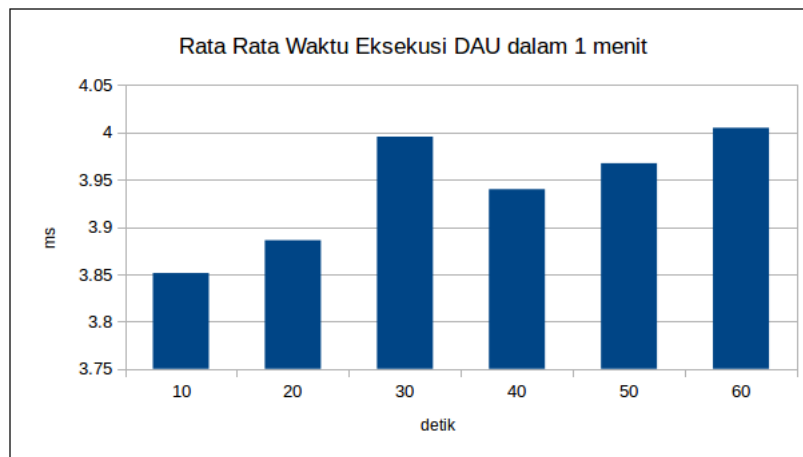
Gambar 4.1: Hasil Pengukuran Delay 100 sample

4.1.3 Execution Time

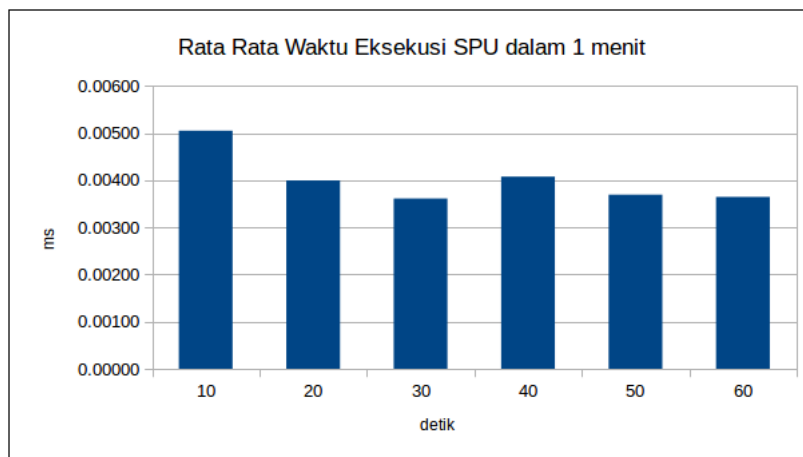
Execution Time perlu diukur pada 2 lokasi yaitu *DAU* dan *SPU*. Hal ini ditujukan untuk mengetahui *maximum sampling speed* yang mungkin dilakukan pada *DAU* sebelum terjadi *bottleneck*.

Pengujian pada *DAU* dilakukan sebanyak 10 kali. Berdasarkan pengujian, didapatkan rata-rata waktu eksekusi ialah 3.94 ms ms per sampel. Hasil rata rata pengukuran waktu eksekusi tiap 10 detik dalam 1 menit tertera pada gambar 4.2. Tabel pengukuran waktu eksekusi pada *DAU* yang lengkap terlampir pada lampiran 5.5.

Pengujian pada *SPU* dilakukan sebanyak 10 kali. Berdasarkan pengujian, didapatkan rata-rata waktu eksekusi ialah 0.00401 ms ms per sampel. Hasil rata rata pengukuran waktu eksekusi tiap 10 detik dalam 1 menit tertera pada gambar 4.3. Tabel pengukuran waktu eksekusi pada SPU yang lengkap terlampir pada lampiran 5.5.



Gambar 4.2: Hasil Pengukuran Waktu Eksekusi Pada DAU



Gambar 4.3: Hasil Pengukuran Waktu Eksekusi Pada SPU

4.1.4 Akurasi

Setelah melakukan pengujian didapatkan hasil akurasi 99% untuk deteksi detak dan 93.11% untuk deteksi aritmia.

a. Akurasi Deteksi Puncak

Dilakukan percobaan untuk menemukan konfigurasi konstanta deteksi d (durasi window), α (*peak threshold, val by mean*), dan β (*min distance threshold, idx by R*). Skenario percobaan variabel dapat dilihat pada tabel 4.1. Hasil percobaan tersebut dapat dilihat pada tabel 4.2. Sebagai perbandingan kolom skenario *original* pada tabel 4.2 merupakan hasil percobaan menggunakan algoritma Pan-Tompkins yang asli.

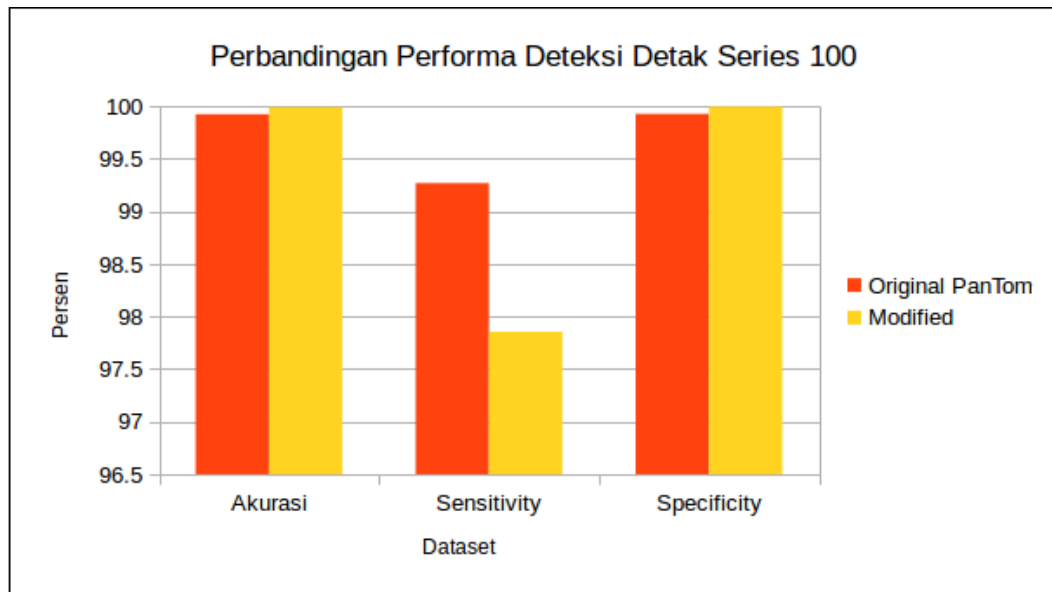
| Variables | Scenario | | | | |
|-----------|----------|-----|-----|------|------|
| | 1 | 2 | 3 | 4 | 5 |
| d (s) | 8 | 7 | 6.5 | 6.5 | 6.5 |
| α | 1.1 | 1.1 | 1.1 | 0.65 | 0.65 |
| β | 0.8 | 0.8 | 0.8 | 0.87 | 0.93 |

Tabel 4.1: Variabel eksperiment

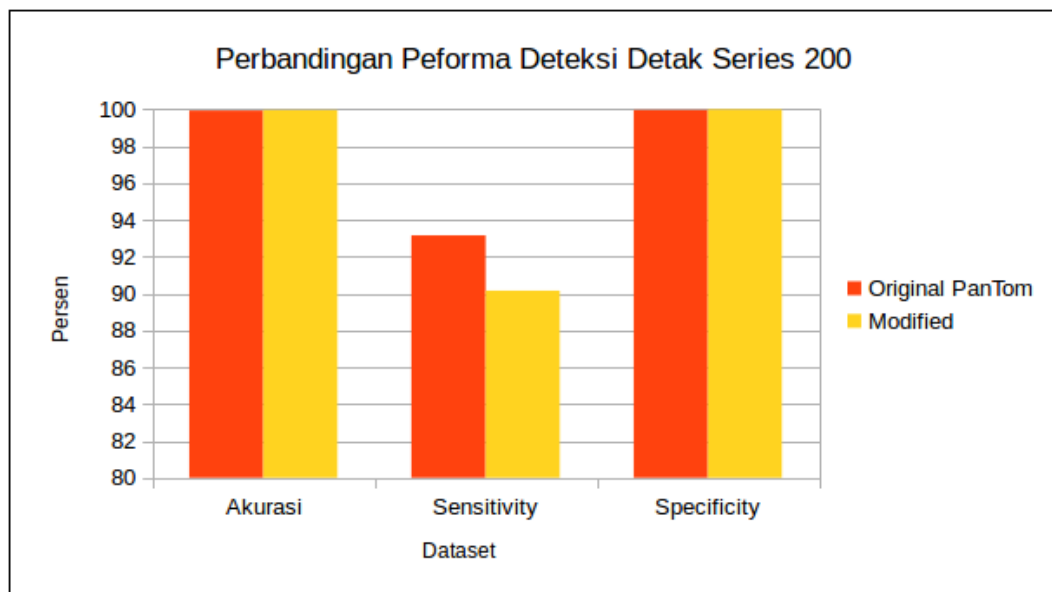
| Dataset | Metric | Scenario | | | | | |
|------------|-----------------|----------|----------|----------|----------|----------|----------|
| | | Original | 1 | 2 | 3 | 4 | 5 |
| Series 100 | Accuracy (%) | 99.92585 | 99.99153 | 99.99140 | 99.99094 | 99.99033 | 99.98884 |
| | Sensitivity (%) | 99.27048 | 97.83673 | 97.87055 | 97.85481 | 98.27833 | 96.77770 |
| | Specificity (%) | 99.92848 | 99.99854 | 99.99828 | 99.99789 | 99.99789 | 99.99932 |
| | Avg Time (ms) | 0.00957 | 0.00790 | 0.00602 | 0.00732 | 0.00782 | 0.00836 |
| Series 200 | Accuracy (%) | 99.93059 | 99.96025 | 99.96060 | 99.96060 | 99.96571 | 99.95492 |
| | Sensitivity (%) | 93.15539 | 90.02946 | 90.14932 | 90.15590 | 91.79111 | 88.63900 |
| | Specificity (%) | 99.96130 | 99.99967 | 99.99962 | 99.99956 | 99.99803 | 99.99948 |
| | Avg Time (ms) | 0.01026 | 0.00803 | 0.00621 | 0.00767 | 0.00771 | 0.00781 |

Tabel 4.2: Hasil Pengujian Algoritma Deteksi Detak Jantung pada Python

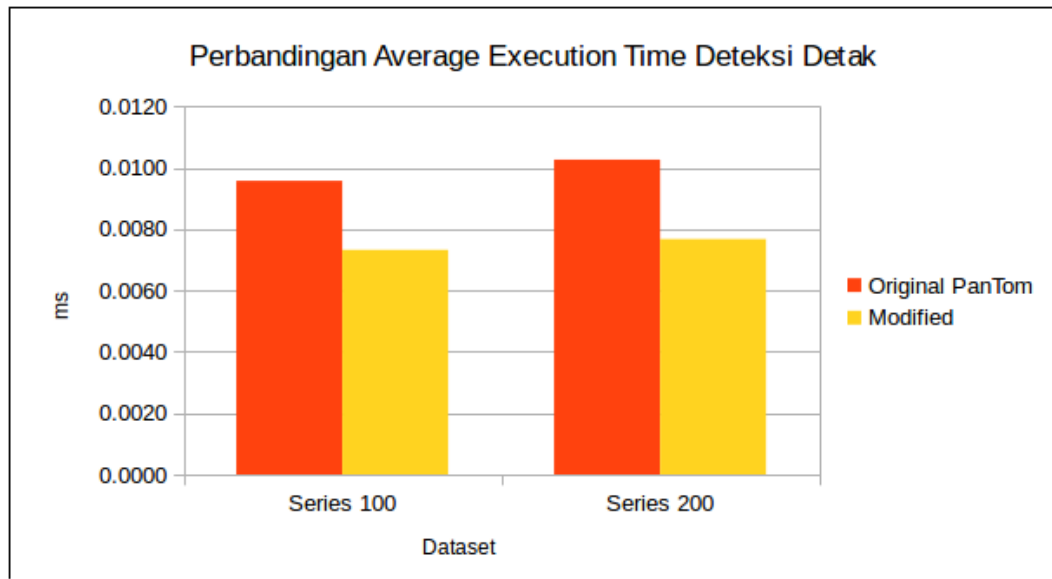
Perbandingan antara algoritma modifikasi dengan algoritma pantom yang asli dapat dilihat pada diagram batang 4.4, 4.5 dan 4.6.



Gambar 4.4: Perbandingan Performa Deteksi Data Series 100



Gambar 4.5: Perbandingan Performa Deteksi Data Series 200



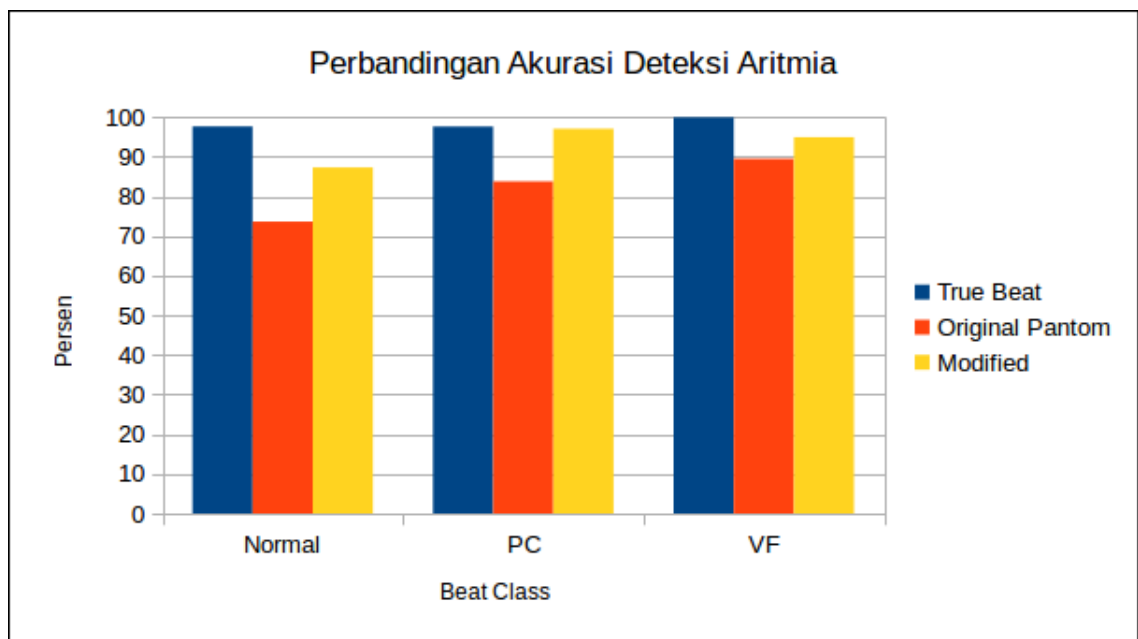
Gambar 4.6: Perbandingan Waktu Akurasi

b. Akurasi Deteksi Aritmia

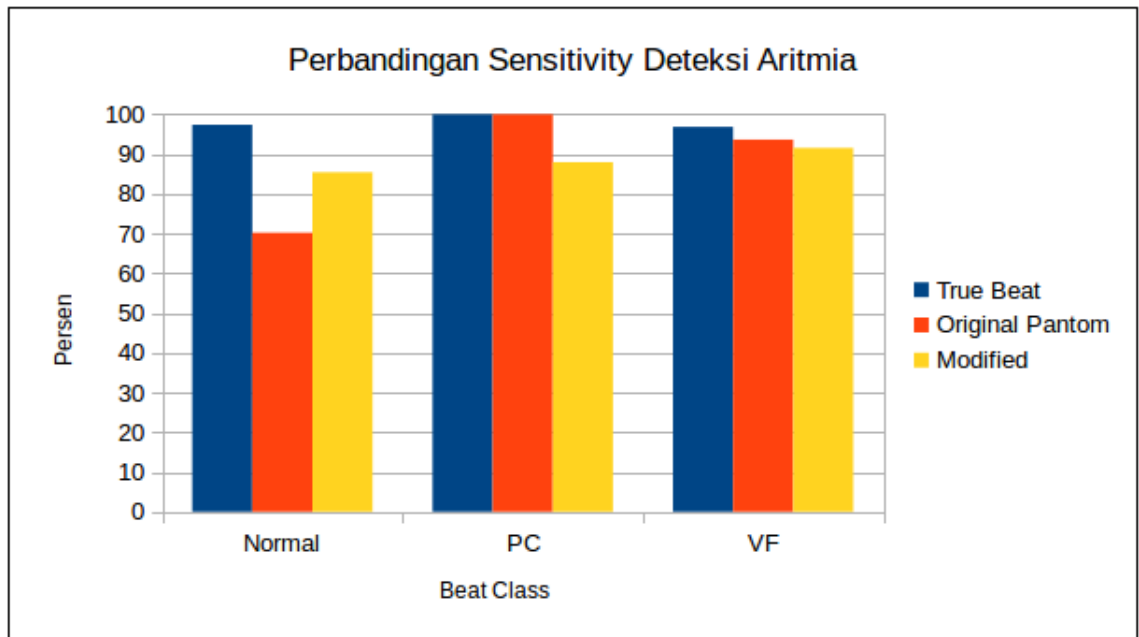
Pengujian akurasi deteksi aritmia terbagi menjadi 3 bagian. Pengujian pertama, sistem deteksi menerima inputan detak jantung asli sesuai data dari MIT BIH. Pengujian kedua, sistem deteksi menerima inputan detak jantung hasil deteksi tahap sebelumnya dengan algoritma asli dari PanTomkins. Pengujian ketiga, sistem deteksi menerima inputan detak jantung hasil deteksi tahap sebelumnya dengan algoritma modifikasi dari PanTomkins. Hasil pengujian tertera pada tabel 4.3 merupakan hasil percobaan menggunakan algoritma Pan-Tompkins yang asli. Untuk mempermudah pemahaman hasil pengujian juga dapat dilihat dalam diagram batang pada gambar 4.7, 4.8, dan 4.9.

| Input Beat | Series | Class | Metric | | |
|------------------|------------|--------|-----------|-----------|-----------|
| | | | Acc (%) | Sp (%) | Se (%) |
| True Beat | Series 100 | Normal | 98.29666 | 100.00000 | 98.23850 |
| | | PC | 98.29666 | 98.23854 | 100.00000 |
| | | VF | 100.00000 | 100.00000 | 100.00000 |
| | Series 200 | Normal | 97.03332 | 100.00000 | 96.48097 |
| | | PC | 96.98531 | 96.45585 | 100.00000 |
| | | VF | 99.95200 | 100.00000 | 93.65751 |
| Original -PanTom | Series 100 | Normal | 80.53953 | 100.00000 | 79.87505 |
| | | PC | 74.41212 | 73.53898 | 100.00000 |
| | | VF | 94.49141 | 94.49129 | 100.00000 |
| | Series 200 | Normal | 66.70400 | 100.00000 | 60.50489 |
| | | PC | 93.15454 | 91.95229 | 100.00000 |
| | | VF | 84.46091 | 84.34240 | 100.00000 |
| Modified -Pantom | Series 100 | Normal | 93.83483 | 100.00000 | 93.62431 |
| | | PC | 97.76384 | 97.68754 | 100.00000 |
| | | VF | 99.14833 | 99.14831 | 100.00000 |
| | Series 200 | Normal | 80.76135 | 100.00000 | 77.17946 |
| | | PC | 96.39645 | 100.00000 | 75.87832 |
| | | VF | 90.74631 | 90.67574 | 100.00000 |

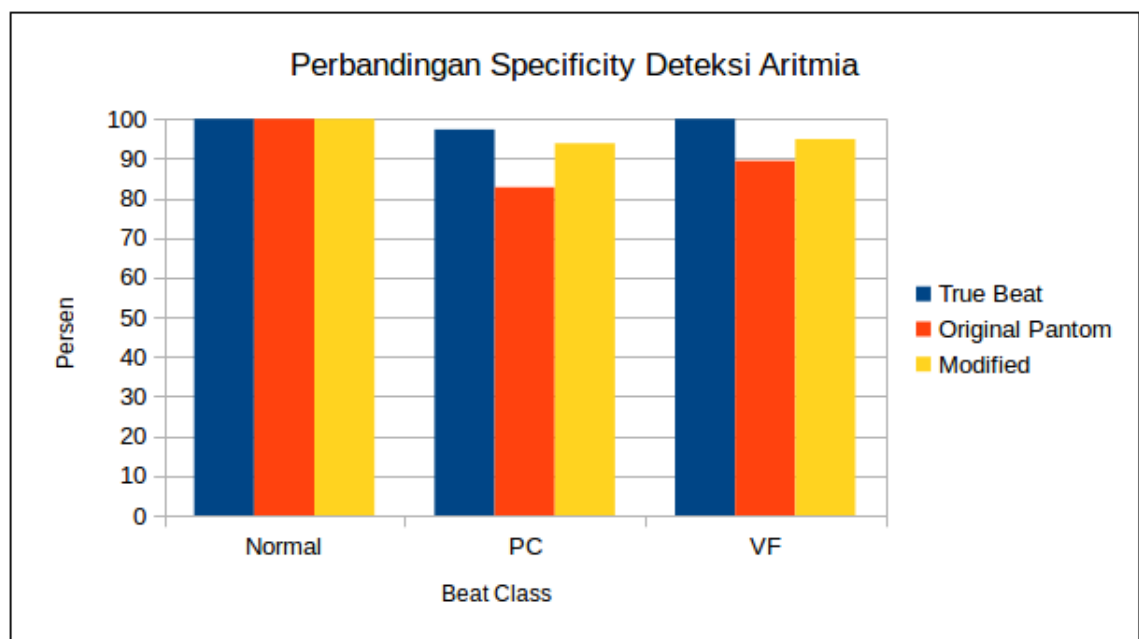
Tabel 4.3: Tabel Performa deteksi Aritmia dengan input berbeda



Gambar 4.7: Perbandingan akurasi deteksi Aritmia dengan input berbeda



Gambar 4.8: Perbandingan sensitivitas deteksi Aritmia dengan input berbeda



Gambar 4.9: Perbandingan spesifisitas deteksi Aritmia dengan input berbeda

4.2 Pembahasan

Dengan pengimplementasian rancangan arsitektur, sistem yang dibangun dapat memiliki semua fitur yang dimiliki sistem sejenis (4.1.1). Delay yang terjadi selama pengiriman sampel selalu berubah ubah namun memiliki rata rata 1.72541 ms . Perubahan nilai delay terjadi karena pengaruh kehandalan jaringan (WiFi) dan subjek yang bebas bergerak.

Waktu eksekusi diukur pada *DAU* dan *SPU*. Pada *DAU* terlihat waktu eksekusi yang berubah ubah (lihat gambar 4.2), ini dipengaruhi oleh adanya *interrupt* pengambilan sampel, pembuatan dan pengiriman paket, dan *buffer memory management*. Sebagai perbandingan, tanpa melakukan pengiriman waktu eksekusi hanya selama 0.001 ms (waktu untuk mengambil sampel). Karena rata rata waktu eksekusi pada *DAU* ialah 3.94 ms maka maksimum frekuensi sampel ialah 200Hz, atau setara dengan 1 sampel/5ms.

Sedangkan pada *SPU*, terlihat waktu eksekusi sangat tinggi pada permulaan kemudian sangat rendah lalu berfluktuasi (lihat gambar 4.3), hal ini disebabkan oleh *SPU* yang membutuhkan waktu lebih lama untuk memulai *subprocess* untuk tiap *DAU* yang terhubung. Kemudian, tiap proses mengisi buffer hingga terisi untuk dilakukan operasi deteksi. Penggunaan Node.JS sebagai *SPU* juga memberikan dampak dalam kecepatan. Dibandingkan dengan Python, Node.JS dapat bekerja hampir 3 kali lebih cepat (gambar 4.2). Karena maksimum frekuensi sampel yang dapat diterapkan ialah 1 sampel tiap 5ms dan waktu eksekusi 1 sampel ialah 0.00401 ms maka secara matematis *SPU* dapat memproses sekitar 800 device secara bersamaan (maksimum total sensor dan *CMU* yang terhubung, 5ms / 0.00401 ms).

Pada segi performansi, algoritma deteksi puncak R terlihat hasil modifikasi menghasilkan akurasi, *specificity* dan *sensitivity* yang hampir menyamai algoritma pantomkins yang asli bahkan cenderung lebih baik (kecuali *sensitivity*) (lihat gambar 4.4 dan 4.5). Hal ini secara langsung berdampak pada performa algoritma deteksi aritmia yang diterapkan sehingga menghasilkan tren performa yang sama (lebih rendah pada paramater *sensitivity*) (lihat gambar 4.7, 4.8, dan 4.9).

Bab V

Kesimpulan dan Saran

5.1 Kesimpulan

Berdasarkan hasil pengujian dapat diambil beberapa kesimpulan sebagai berikut:

1. Arsitektur *monitoring* jantung yang bersifat ubiquitous dan terus menerus berhasil dirancang dan diimplementasikan. Hal ini dapat dilihat dari berhasilnya dilakukan pengujian terhadap arsitektur yang dibahas pada bab IV.
2. Mode monitoring yang dirancang dapat mengawasi banyak pasien sekaligus serta memberikan peringatan otomatis ketika terjadi aritmia. Pernyataan ini didukung oleh hasil pembuktian matematis dan performa akurasi deteksi yang dibahas pada sub bab 4.2.
3. Arsitektur yang dirancang memiliki delay pengiriman sebesar 1.72541 ms , waktu eksekusi pada DAU sebesar 3.94 ms , waktu eksekusi pada SPU sebesar 0.00401 ms , dan performa akurasi sebesar 93.11% . Receptor memiliki maksimum frekuensi sampel sebesar 200Hz dan secara matematis server bisa menangani hingga 800 *devices* secara bersamaan.

5.2 Saran

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

1. Bekerjasama dengan dokter ahli jantung untuk melakukan pengujian nyata
2. Memilih fitur dan klasifikasi lain untuk meningkatkan kehandalan akurasi deteksi
3. Melakukan simulasi jaringan *unreliable* dengan menggunakan WANem. Hal ini ditujukan agar dapat menguji kehandalan sistem jika diterapkan di dunia nyata.

4. Merancang *receptor* yang lebih hemat daya,
5. Merancang *Device Interface* dan *Application Programming Interface* (API) sehingga sistem dapat menerima input dari perangkat yang telah tersedia dipasaran.

Daftar Pustaka

- [1] WHO. (2017). World health organization - fact sheet: Cardiovascular diseases (cvds), [Online]. Available: <http://www.who.int/mediacentre/factsheets/fs317/en/>.
- [2] D. Kesehatan, “Situasi kesehatan jantung”, in *Publikasi Data dan Informasi*, 2014. [Online]. Available: <http://www.depkes.go.id/folder/view/01/structure-publikasi-pusdatin-info-datin.html>.
- [3] E. Indonesia. (2017). Holter system: Edan se-2003, [Online]. Available: <https://endo.id/catalog/product/edan-holter-ecg-3-channel>.
- [4] G. Beavis. (Jul. 2017). Fitbit blaze review, [Online]. Available: <http://www.techradar.com/reviews/wearables/samsung-gear-s3-classic-1327492/review>.
- [5] J. Peckham. (Feb. 2017). Samsung gear s3 review, [Online]. Available: <http://www.techradar.com/reviews/wearables/fitbit-blaze-1312660/review>.
- [6] A. H. Association. (2017). What is cardiovascular disease?, [Online]. Available: http://www.heart.org/HEARTORG/Conditions/What-is-Cardiovascular-Disease_UCM_301852_Article.jsp#.WWzTmnaGPeQ.
- [7] N. H. Lung and B. Institute. (2017). What is an arrhythmia?, [Online]. Available: <https://www.nhlbi.nih.gov/health/health-topics/topics/arr>.
- [8] M. Tsipouras, D. Fotiadis, and D. Sideris, “An arrhythmia classification system based on the rr-interval signal”, *Artificial Intelligence in Medicine*, 2005.
- [9] B. M. Asl, S. K. Setarehdan, and M. Mohebbi, “Support vector machine-based arrhythmia classification using reduced features of heart rate variability signal”, *Artificial Intelligence in Medicine*, 2008. DOI: 10.1016/j.artmed.2008.04.007.

- [10] M. S. Shet, M. Patel, A. Rao, C. Kantharaj, and S. K.V., “Ecg arrhythmia classification using r-peak based segmentation, binary particle swarm optimization and absolute euclidean classifier”, in *Proceedings of International Conference on Advances in Computing*, 2013. DOI: 10.1007/978-81-322-0740-5_37.
- [11] A. Solosenko and V. Marozas, “Automatic premature ventricular contraction detection in photoplethysmographic signals”, in *IEEE Biomedical Circuits and Systems Conference*, 2014. DOI: 10.1109/BioCAS.2014.6981642.
- [12] A. E. Prawira. (2016). 1.500 dokter jantung harus tersedia pada 2019, [Online]. Available: <http://health.liputan6.com/read/2484942/1500-dokter-jantung-harus-tersedia-pada-2019>.
- [13] A. P. Store. (2017). Apps, [Online]. Available: <https://play.google.com/store/search?q=heart%20rate&c=apps&hl=en>.
- [14] E. Indonesia. (2017). Fingertip pulse oximeter, edan h-10, [Online]. Available: <https://endo.id/catalog/product/edan-h-10>.
- [15] D. Barata, G. Louzada, A. Carreiro, and A. Damasceno, “System of acquisition, transmission, storage and visualization of pulse oximeter and ecg data using android and mqtt”, *International Conference on Health and Social Care Information Systems and Technologies*, 2013.
- [16] K. Motwani, D. Mirchandani, YogeetaRohra, H. Tarachandani, and P. A. Yeole, “Smart nursing home patient monitoring system”, *Imperial Journal of Interdisciplinary Research*, 2016.
- [17] A. Piper. (2013). Notable uses of mqtt, [Online]. Available: <https://github.com/mqtt/mqtt.github.io/wiki/use-cases>.
- [18] P. Pierleoni, L. Pernini, A. Belli, and L. Palma, “An android-based heart monitoring system for the elderly and for patients with heart disease”, *International Journal of Telemedicine and Applications*, 2014. DOI: 10.1155/2014/625156.
- [19] M. S. D. Gupta, V. Patchava, and V. irginia Menezes, “Healthcare based on iot using raspberry pi”, in *2015 International Conference on Green Computing and Internet of Things (ICGCIoT)*, 2015. DOI: 10.1109/ICGCIoT.2015.7380571.
- [20] N. Vemishetty, P. Jadhav, B. Adapa, A. Acharyya, R. Pachamuthu, and G. R. Naik, “2015 37th annual international conference of the ieee engineering in medicine and biology society (embc)”, 2014. DOI: 10.1109/EMBC.2015.7319534.

- [21] V. Jindal, “Integrating mobile and cloud for ppg signal selection to monitor heart rate during intensive physical exercise”, in *IEEE/ACM International Conference on Mobile Software Engineering and Systems*, 2016. DOI: 10.1145/2897073.2897132.
- [22] M. Manisha, K. Neeraja, V. Sindhura, and P. Ramya, “Iot on heart attack detection and heart rate monitoring”, *International Journal of Innovations in Engineering and Technology*, 2016.
- [23] J. Pan and W. J. Tompkins, “A real-time qrs detection algorithm”, *IEEE Transactions on Biomedical Engineering*, vol. BME-32, no. 3, 1985.
- [24] C. Pavlatos, A. Dimopoulos, and et al., “Hardware implementation of pan and tompkins qrs detection algorithm”, in *EMBECC05 Conference*, 2005.
- [25] M. Darweesh, H. Saleh, B. Mohammad, and M. Ismail, “Novel logarithmic ecg feature extraction algorithm based on pan and tompkins”, in *IEEE 59th International Midwest Symposium on Circuits and Systems*, 2016.
- [26] L. F. Polania, L. K. Mestha, D. T. Huang, and J.-P. Couderc, “Method for classifying cardiac arrhythmias using photoplethysmography”, in *37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 2015. DOI: 10.1109/EMBC.2015.7319899.
- [27] V. Kalidas and L. S. Tamil, “Cardiac arrhythmia classification using multi-modal signal analysis”, *Physiological Measurement*, vol. July 2016, no. 7, 2016.
- [28] S. Lee, H. Shin, and C. Y. Hahm, “Effective ppg sensor placement for reflected red and green light, and infrared wristband-type photoplethysmography”, in *18th International Conference on Advanced Communication Technology*, 2016. DOI: 10.1109/ICACT.2016.7423469.
- [29] T. Tamura, Y. Maeda, M. Sekine, and M. Yoshida, “Wearable photoplethysmographic sensors—past and present”, *Electronics*, vol. 3, no. 2, pp. 282–302, 2014, ISSN: 2079-9292. DOI: 10.3390/electronics3020282.
- [30] M. Pimentel and et al., “Probabilistic estimation of respiratory rate from wearable sensors”, *Springer International Publishing*, no. 15, 241-262, 2015. DOI: 10.1007/978-3-319-18191-2_10.
- [31] Node.js. (2017). Node js, [Online]. Available: <https://nodejs.org/en/>.
- [32] I. Gouy. (2017). The computer language benchmarks game, [Online]. Available: <http://benchmarksgame.alioth.debian.org/>.
- [33] DA-14. (2017). Mongodb vs mysql comparison: Which database is better?, [Online]. Available: <http://benchmarksgame.alioth.debian.org/>.

- [34] J. Murphy and Y. Gitman. (2017). Pulse sensor amped, [Online]. Available: <https://pulsesensor.com>.
- [35] M. GB and M. RG, “The impact of the mit-bih arrhythmia database”, *IEEE Eng in Med and Biol*, 2001.
- [36] G. AL, A. LAN, G. L, H. JM, I. PCh, M. RG, M. JE, M. GB, P. C-K, and S. HE. (Jun. 2000). Physiobank, physiotoolkit, and physionet: Components of a new research resource for complex physiologic signals, [Online]. Available: <http://circ.ahajournals.org/content/101/23/e215.full>.

Lampiran

5.3 Koefisien Proses Filter

| Sample | Koef | | |
|--------------|------------------------|----|------------------------|
| 1 (terbaru) | 0.00000335962691819121 | 29 | 0.01142549902791990000 |
| 2 | 0.00001174538201448820 | 30 | 0.01161563130763460000 |
| 3 | 0.00002751129270325190 | 31 | 0.01167964404894330000 |
| 4 | 0.00005408436863137360 | 32 | 0.01161563130763460000 |
| 5 | 0.00009565996022636050 | 33 | 0.01142549902791990000 |
| 6 | 0.00015717362674254100 | 34 | 0.01111487912356260000 |
| 7 | 0.00024421527302049600 | 35 | 0.01069287666627230000 |
| 8 | 0.00036288000194863700 | 36 | 0.01017166468126630000 |
| 9 | 0.00051955401554523300 | 37 | 0.00956594962500232000 |
| 10 | 0.00072063848325571600 | 38 | 0.00889233764515457000 |
| 11 | 0.00097221928971587000 | 39 | 0.00816863674493544000 |
| 12 | 0.00127969561740799000 | 40 | 0.00741313269012500000 |
| 13 | 0.00164738501239717000 | 41 | 0.00664387677594401000 |
| 14 | 0.00207812651333650000 | 42 | 0.00587802145832024000 |
| 15 | 0.00257290620892021000 | 43 | 0.00513123556668870000 |
| 16 | 0.00313053089976755000 | 44 | 0.00441722471938748000 |
| 17 | 0.00374737514156545000 | 45 | 0.00374737514156545000 |
| 18 | 0.00441722471938748000 | 46 | 0.00313053089976755000 |
| 19 | 0.00513123556668870000 | 47 | 0.00257290620892021000 |
| 20 | 0.00587802145832024000 | 48 | 0.00207812651333650000 |
| 21 | 0.00664387677594401000 | 49 | 0.00164738501239717000 |
| 22 | 0.00741313269012500000 | 50 | 0.00127969561740799000 |
| 23 | 0.00816863674493544000 | 51 | 0.00097221928971587000 |
| 24 | 0.00889233764515457000 | 52 | 0.00072063848325571600 |
| 25 | 0.00956594962500232000 | 53 | 0.00051955401554523300 |
| 26 | 0.01017166468126630000 | 54 | 0.00036288000194863700 |
| 27 | 0.01069287666627230000 | 55 | 0.00024421527302049600 |
| 28 | 0.01111487912356260000 | 56 | 0.00015717362674254100 |
| 57 | 0.00009565996022636050 | | |
| 58 | 0.00005408436863137360 | | |
| 59 | 0.00002751129270325190 | | |
| 60 | 0.00001174538201448820 | | |
| 61 (terlama) | 0.00000335962691819121 | | |

Tabel 5.1: Daftar Koefisien Band Pass Filter

| Sample | Koef |
|-------------|------|
| 1 (terbaru) | 2 |
| 2 | 1 |
| 3 | 0 |
| 4 | -1 |
| 5 (terlama) | -2 |

Tabel 5.2: Daftar Koefisien Derivative Filter

5.4 Hasil Lengkap Pengukuran Delay

| Experi ment | Detik | | | | | |
|----------------|----------------|----------------|----------------|----------------|----------------|----------------|
| | 10 | 20 | 30 | 40 | 50 | 60 |
| 1 | 0.96781 | 0.97748 | 0.70120 | 1.00834 | 0.99337 | 0.78680 |
| 2 | 2.04501 | 1.57936 | 3.10104 | 1.50462 | 2.05914 | 1.41137 |
| 3 | 2.95064 | 1.87745 | 1.55256 | 1.68320 | 1.99337 | 1.75872 |
| 4 | 1.90500 | 2.81460 | 1.45587 | 1.74969 | 1.79524 | 1.78970 |
| 5 | 1.34296 | 1.70416 | 1.55502 | 1.49499 | 1.32837 | 1.86993 |
| 6 | 1.79524 | 2.35499 | 2.00991 | 1.63211 | 1.63934 | 1.91232 |
| 7 | 2.98713 | 1.76267 | 2.13135 | 2.62583 | 1.42691 | 2.15037 |
| 8 | 1.92480 | 1.40257 | 1.71335 | 1.47729 | 1.74376 | 1.64422 |
| 9 | 1.75514 | 1.33225 | 1.48840 | 1.57669 | 1.57118 | 1.73444 |
| 10 | 1.73430 | 1.37618 | 1.30703 | 1.86579 | 1.45731 | 2.23460 |
| Avg | 1.94080 | 1.71817 | 1.70157 | 1.66185 | 1.60080 | 1.72925 |

Tabel 5.3: Delay Pada Jaringan

5.5 Hasil Lengkap Pengukuran Waktu Eksekusi

| Experi ment | Detik | | | | | |
|----------------|----------------|----------------|----------------|----------------|----------------|----------------|
| | 10 | 20 | 30 | 40 | 50 | 60 |
| 1 | 3.91158 | 3.57066 | 3.91097 | 3.77816 | 3.79250 | 4.51817 |
| 2 | 4.11731 | 3.87191 | 4.62059 | 3.55539 | 3.92082 | 3.51719 |
| 3 | 3.61455 | 4.05920 | 4.66070 | 4.28257 | 4.55494 | 4.48154 |
| 4 | 3.70313 | 3.85844 | 3.66662 | 4.54609 | 3.87143 | 3.61102 |
| 5 | 3.99576 | 4.29974 | 3.89293 | 3.88644 | 3.47110 | 3.78994 |
| 6 | 4.12148 | 3.85044 | 3.83510 | 3.88253 | 4.12112 | 4.03644 |
| 7 | 4.03073 | 3.95178 | 4.05500 | 4.32149 | 3.75316 | 4.80987 |
| 8 | 4.00440 | 4.14767 | 3.87206 | 4.15978 | 3.72070 | 3.84310 |
| 9 | 3.52069 | 3.71721 | 3.61846 | 3.50409 | 3.93295 | 3.63816 |
| 10 | 3.49329 | 3.53279 | 3.82239 | 3.48210 | 4.53517 | 3.80298 |
| Avg | 3.85129 | 3.88598 | 3.99548 | 3.93986 | 3.96739 | 4.00484 |

Tabel 5.4: Waktu Eksekusi DAU

| Experiment | Detik | | | | | |
|------------|----------------|----------------|----------------|----------------|----------------|----------------|
| | 10 | 20 | 30 | 40 | 50 | 60 |
| 1 | 0.00585 | 0.00369 | 0.00343 | 0.00373 | 0.00408 | 0.00357 |
| 2 | 0.00503 | 0.00393 | 0.00334 | 0.00680 | 0.00440 | 0.00369 |
| 3 | 0.00546 | 0.00454 | 0.00424 | 0.00362 | 0.00374 | 0.00393 |
| 4 | 0.00528 | 0.00580 | 0.00325 | 0.00332 | 0.00306 | 0.00345 |
| 5 | 0.00468 | 0.00337 | 0.00351 | 0.00356 | 0.00351 | 0.00386 |
| 6 | 0.00467 | 0.00359 | 0.00337 | 0.00433 | 0.00382 | 0.00350 |
| 7 | 0.00455 | 0.00348 | 0.00338 | 0.00341 | 0.00348 | 0.00363 |
| 8 | 0.00456 | 0.00346 | 0.00346 | 0.00305 | 0.00275 | 0.00296 |
| 9 | 0.00591 | 0.00451 | 0.00457 | 0.00534 | 0.00441 | 0.00452 |
| 10 | 0.00450 | 0.00355 | 0.00356 | 0.00358 | 0.00367 | 0.00334 |
| Avg | 0.00505 | 0.00399 | 0.00361 | 0.00407 | 0.00369 | 0.00364 |

Tabel 5.5: Waktu Eksekusi SPU