

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. Menurut literatur, 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 (sekitar 100 ribu orang). Hal ini mengakibatkan seorang dokter tidak mungkin melakukan *monitoring* seorang pasien secara terus menerus. Padahal dengan menerapkan konsep *Internet of Things*(IoT) dan algoritma yang diusulkan pada penelitian sebelumnya, sistem dapat bekerja secara terus menerus dan memberikan peringatan tentang aritmia kepada dokter yang terkoneksi secara *real time*. Oleh karena itu tugas akhir ini merancang arsitektur sistem IoT dan menerapkan algoritma yang diusulkan penelitian sebelumnya untuk menyelesaikan masalah diatas. Rancangan arsitektur memanfaatkan MQTT sebagai protokol komunikasi jaringan yang dirancang agar dapat memproses banyak *sensor* dan banyak *viewer* sekaligus. Algoritma yang diterapkan pada arsitektur ialah modifikasi algoritma usulan Pan-Tompkin(1985), Deshmene(2009) dan Tsipouras(2005). Pan-Tomkins dan Deshmene mengusulkan metode untuk melakukan *preprocessing* dan *processing* terhadap hasil baca sensor. Tsipouras mengusulkan metode *rule based classification* untuk mengklasifikasikan aritmia dengan menggunakan fitur *peak* (titik R atau titik Systolic) pada rekaman jantung. Berdasarkan pengujian, performa arsitektur sistem yang diusulkan pada tugas akhir ini dinilai baik, yaitu dapat melayani maksimum xx sensor dengan rata-rata delay xx ms dan tingkat akurasi xx% untuk deteksi detak walau deteksi aritmia memiliki perfoma yang buruk hanya xx%.

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

Abstract

Heart disease (Cardiovascular Diseases, CVDs) is a disease that can affect anyone, happens anytime and anywhere. There are many products on the market that can perform heart monitoring while recording cardiac (heart) activity. Cardiac records are required by cardiologists to perform disease analysis and to design treatment methods. One type of CVDs that can be identified from the cardiac record is the arrhythmia. According to the literature, 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 (about 100 thousand people). This makes a doctor impossible to monitor anyone continuously. Though by applying the concept of Internet of Things (IoT) and algorithms proposed in previous research, the system can work continuously and provide warnings about arrhythmia to a cardiologist who is connected in real time. Therefore this final project designs the IoT system architecture and applies the proposed algorithm of previous research to solve the above problem. The architecture design utilizes MQTT as a network communication protocol designed to process multiple sensors and multiple viewers at once. Algorithm applied to the architecture is the modification algorithm proposed Pan-Tompkin (1985), Deshmane (2009) and Tsipouras (2005). Pan-Tomkins and Deshmane proposed a method for preprocessing and processing the sensor readings data. Tsipouras proposed a rule-based classification method to classify arrhythmias using the peak feature (point R or Systolic point) of cardiac record. Based on testing, the performance of the proposed system architecture in this final project is considered good, that is, it can serve a maximum of xxx sensor with average delay xxx ms and xx% accuracy level for detection of transient detection although arrhythmia has poor performance with accuracy only xx%.

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 xx, Pak yy, 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, Lusi, dan Husna 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, Danang, 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 | ix |
| Daftar Tabel | x |
| I Pendahuluan | 1 |
| 1.1 Latar Belakang | 1 |
| 1.2 Pernyataan Masalah | 2 |
| 1.3 Perumusan Masalah | 2 |
| 1.4 Batasan Masalah | 3 |
| 1.5 Tujuan | 3 |
| 1.6 Hipotesis | 3 |
| 1.7 Sistematika Penulisan | 4 |
| II Kajian Pustaka | 5 |
| 2.1 ECG dan PPG | 5 |
| 2.1.1 Lokasi Penempatan Sensor | 5 |
| 2.1.2 Titik Fiducial | 7 |
| 2.1.3 Bentuk Sinyal | 7 |
| 2.2 Aritmia | 8 |
| 2.2.1 Riset Klasifikasi Aritmia Otomatis | 9 |
| 2.3 Produk Monitoring Jantung di Pasaran | 10 |
| 2.4 Node.Js dan Mongo.Db | 11 |
| 2.4.1 Node.Js | 11 |
| 2.4.2 MongoDB | 11 |

| | | |
|---|---|-----------|
| 2.5 | ESP-12 | 12 |
| 2.6 | Protokol MQTT | 12 |
| III Metodologi dan Desain Sistem | | 14 |
| 3.1 | Metodologi Penelitian | 14 |
| 3.2 | Gambaran Umum Sistem | 16 |
| 3.3 | Rancangan Perangkat Keras | 16 |
| 3.3.1 | Receptor | 17 |
| 3.3.2 | Server | 19 |
| 3.3.3 | Viewer | 19 |
| 3.4 | Rancangan Algoritma Pemantauan | 20 |
| 3.5 | Rancangan Algoritma Deteksi | 21 |
| 3.5.1 | Pengambilan dan Pengiriman Sinyal | 22 |
| 3.5.2 | Preprocessing dan Perekaman | 23 |
| 3.5.3 | Deteksi Detak Otomatis | 25 |
| 3.5.4 | Deteksi Aritmia Otomatis | 27 |
| 3.5.5 | Pengiriman Laporan | 29 |
| 3.5.6 | Hubungan Algoritma Deteksi Dengan Perangkat Keras . | 30 |
| 3.6 | Skenario Pengujian | 31 |
| 3.6.1 | Parameter Pengujian | 31 |
| 3.6.2 | Konfigurasi Perangkat Keras | 32 |
| 3.6.3 | Pengujian Algoritma Pemantauan | 34 |
| 3.6.4 | Pengujian Algoritma Deteksi | 35 |
| IV Hasil dan Pembahasan | | 39 |
| 4.1 | Hasil Pengujian | 39 |
| 4.1.1 | Eksistensi Fitur | 39 |
| 4.1.2 | Delay | 40 |
| 4.1.3 | Execution Time | 40 |
| 4.1.4 | FPS | 41 |
| 4.1.5 | Akurasi | 41 |
| 4.2 | Pembahasan | 41 |
| V Kesimpulan dan Saran | | 42 |
| 5.1 | Kesimpulan | 42 |
| 5.2 | Saran | 42 |
| Daftar Pustaka | | 43 |
| Lampiran | | 45 |

Daftar Gambar

| | | |
|------|--|----|
| 2.1 | a. Sensor ECG dengan 3 titik timbal; b. Sensor PPG ujung jari; c. Sensor PPG di pergelangan tangan | 5 |
| 2.2 | a. Penempatan 3 Elektroda di Dada, b. Penempatan 3 Elektroda di Tangan | 6 |
| 2.3 | Lokasi Penempatan Lebih 3 titik | 6 |
| 2.4 | Sinyal ECG berdasarkan titik fiducial | 7 |
| 2.5 | Sinyal PPG berdasarkan titik fiducial | 7 |
| 2.6 | Perbandingan sinyal ideal PPG dan ECG | 8 |
| 2.7 | Sinkronisasi antara ECG dan PPG | 8 |
| 2.8 | Struktur jantung sederhana | 9 |
| 2.9 | a. Sinyal VF; b. Sinyal PAC; c. Sinyal PVC | 9 |
| 2.10 | a. Gear S3, Jam pintar produksi Samsung; b. Finger clip PPG; c. Portable ECG; d. Heart Rate App; e. Heart Rate App 2 | 10 |
| 2.11 | a. Node JS; b. Mongo DB; | 11 |
| 2.12 | Koin Rp200 - ESP-12E - Pulse Sensor | 12 |
| 2.13 | Cara kerja MQTT | 13 |
| 3.1 | Flowchart Metodologi | 15 |
| 3.2 | Gambaran Umum Sistem | 16 |
| 3.3 | Skema Elektronik Receptor | 17 |
| 3.4 | Baterai LiPo 3.7v 180mAh | 19 |
| 3.5 | a. Tampilan Web Monitoring b. Tampilan Aplikasi Monitoring . | 20 |
| 3.6 | Flowchart Rancangan Algoritma Pemantauan | 21 |
| 3.7 | Flowchart Rancangan Algoritma Deteksi | 22 |
| 3.8 | Flowchart Pengambilan dan Pengiriman Sinyal | 23 |
| 3.9 | Flowchart Rancangan Algoritma Preprocessing | 25 |
| 3.10 | <i>Rule</i> /Algoritma yang diajukan Tsipouras (2005) | 29 |
| 3.11 | Flowchart Rancangan Algoritma Preprocessing | 30 |
| 3.12 | Diagram Tahap Algoritma Deteksi | 31 |
| 3.13 | Konfigurasi Perangkat Keras | 34 |
| 4.1 | Hasil Pengukuran Delay 100 sample | 40 |

Daftar Tabel

| | | |
|-----|--|----|
| 2.1 | Perbandingan riset mengenai klasifikasi aritmia otomasis | 10 |
| 3.1 | Pengelompokan Tipe beat | 28 |
| 3.2 | Tabel Spesifikasi Perangkat Keras | 33 |
| 3.3 | Tabel Rakpitulasi Dataset | 36 |
| 3.4 | Tabel Rakpitulasi Aritmia series 100 | 37 |
| 3.5 | Tabel Rakpitulasi Aritmia series 200 | 37 |

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 penyakit jantung (Cardiovascular Diseases, 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) dan *Photoplethysmogram* (PPG)[xx, xx]. 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.

Aritmia merupakan salah satu jenis dari CVDs yang merupakan kemunculan pola tidak beratur pada detak jantung[3]. Aritmia tergolong mudah untuk diidentifikasi berdasarkan gambar rekam jantung seseorang. Sebagian besar jenis aritmia tergolong tidak berbahaya (serangan kecil). Walaupun tidak berbahaya, ketika aritmia serangan kecil sering terjadi dapat menandakan kemunculan serangan besar, contohnya ketika *Premature Ventricular Contraction* (PVC) terjadi berulang kali dapat menandakan terjadinya serangan besar seperti *Ventricular Tachycardia* (VT) dan *Ventricular Fibrillation* (VF) [xx]. Berdasarkan studi pustaka yang penulis lakukan, beberapa penelitian sebelumnya telah berhasil membuat algoritma untuk mendeteksi aritmia secara otomatis[xx-xx].

Monitoring 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. Padahal seseorang yang pernah terkena penyakit jantung akan rawan mengalami se-

rangan baik kecil (tidak berbahaya) maupun besar (berbahaya) dimana saja dan kapan saja.

Rekam jantung yang dihasilkan oleh produk yang telah ada harus diberikan di lain hari kepada dokter. Hal ini tentunya akan merepotkan seorang pasien. Bukan hanya menyulitkan pasien, bagi seorang dokter untuk memeriksa rekaman jantung sepanjang 24 jam saja akan membutuhkan waktu yang lama. Terlebih menurut sekretaris PERKI, Isman Firdaus, Indonesia masih kekurangan jumlah dokter [4]. Bahkan secara ideal pun seorang dokter jantung masih ditargetkan untuk melayani sangat banyak pasien yaitu sekitar 100,000 pasien[4].

Berdasarkan fakta diatas penulis melihat adanya kebutuhan akan pengembangan terhadap sistem monitoring yang telah ada. Oleh karena itu pada tugas akhir ini penulis merancang arsitektur sistem monitoring dimana sistem dapat melakukan *monitoring* jantung secara ubiquitous dan terus menerus, dapat terkoneksi secara *real time* kepada dokter, mampu melayani banyak pasien sekaligus dan dapat memberikan peringatan ketika aritmia terjadi.

1.2 Pernyataan Masalah

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

1. Sistem tidak dapat berjalan secara *Ubiquitous* dan terus menerus,
2. Analisis sistem dilakukan manual 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. Jenis aritmia yang dapat dideteksi ialah PAC, PVC, *Tachycardia*, dan *Bradycardia*
7. Metode klasifikasi yang diuji ialah metode yang diusulkan oleh Pan-Tomkins dan Tsipouras-Fotiadis

1.5 Tujuan

Berikut adalah tujuan yang ingin dicapai pada penulisan proposal/TA.

1. Untuk merancang arsitektur sistem monitoring detak jantung yang bersifat ubiquitous dan terus menerus,
2. Untuk mengembangkan metode *monitoring* dan peringatan otomatis yang dapat mengawasi banyak pasien sekaligus,
3. Untuk 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 re mengenai teori penunjang seperti skema IBE, skema ECC dan RSA, fungsi Hashing, metode El-Gamal serta metode brute force dan parameter-parameter pengujian seperti running time, memory usage dan avalanche affect.

Bab 3 Metodologi dan Desain Sistem

Pada bab ini membahas tentang alur pengerjaan tugas akhir yang meliputi desain IBE-ECC dan skenario pengujian yang akan dilakukan pada bab selanjutnya.

Bab 4 Hasil dan Pembahasan

Pada bab ini membahas tentang pengujian dari hasil implementasi. Pengujian dilakukan dengan skenario yang telah dibuat pada bab sebelumnya untuk menguji dan menganalisis metode sesuai dengan permasalahan yang sudah di definisikan pada bab pendahuluan

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

Tugas akhir ini membahas perancangan sistem pengawasan jantung. Untuk mendirikan landasan berfikir, bab ini akan membahas teori dan fakta yang berkaitan dengan perancangan sistem tersebut.

2.1 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 2.1. 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.



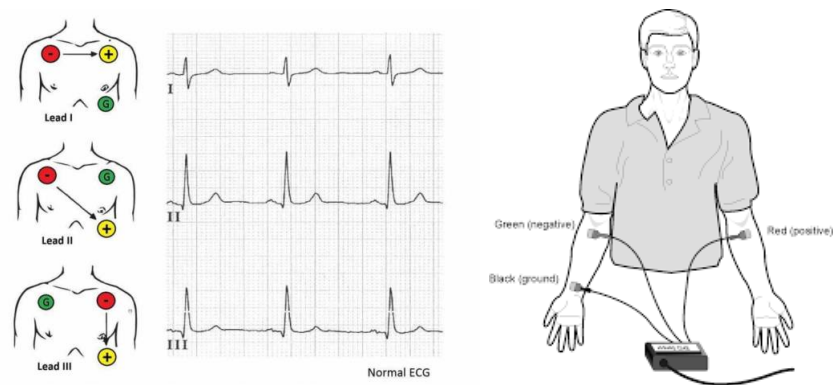
Gambar 2.1: a. Sensor ECG dengan 3 titik timbal; b. Sensor PPG ujung jari; c. Sensor PPG di pergelangan tangan

2.1.1 Lokasi Penempatan Sensor

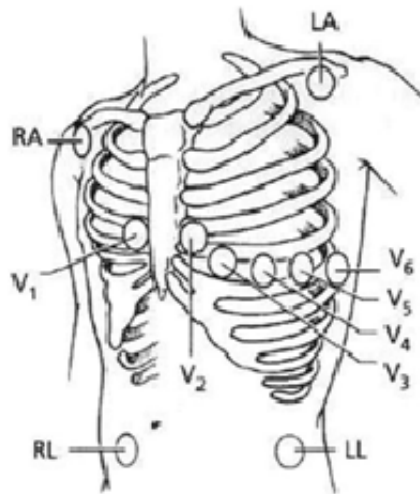
ECG

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

2.2. Untuk pengukuran lebih dari 3 titik, elektroda ECG dapat diletakkan pada posisi yang ditunjukkan gambar 2.3.



Gambar 2.2: a. Penempatan 3 Elektroda di Dada, b. Penempatan 3 Elektroda di Tangan



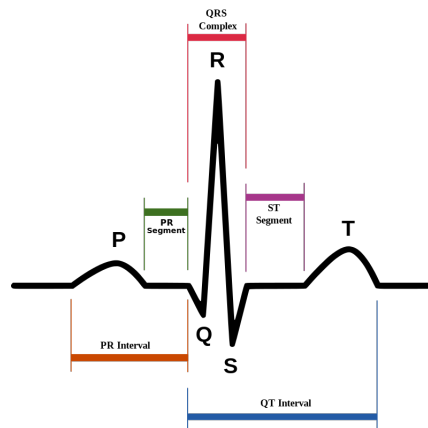
Gambar 2.3: Lokasi Penempatan Lebih 3 titik

PPG

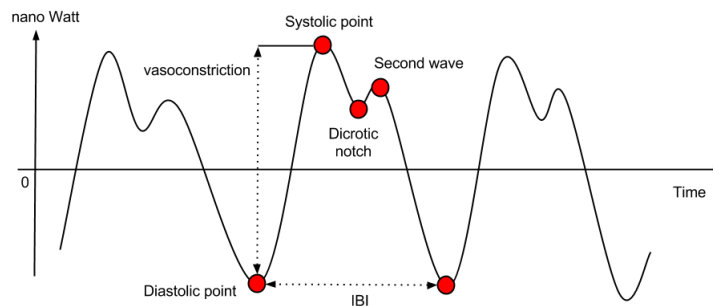
PPG dapat melakukan pengukuran hanya di satu lokasi. Hal ini disebabkan karena PPG mengukur tingkat penyerapan cahaya oleh darah, sedangkan pembuluh darah menjalar ke seluruh tubuh. Namun, untuk menghasilkan pengukuran maksimal PPG perlu diletakkan di lokasi tubuh yang pembuluh darah dekat dengan permukaan kulit [5][6]. Beberapa lokasi tubuh yang sesuai kriteria tersebut terdapat pada ujung jari, pergelangan tangan, lengan atas, leher, dan daun kuping.

2.1.2 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 2.4. Sedangkan siklus sinyal PPG dilihat dari siklus Diastolic-Systolic-Dicrotic seperti terlihat pada gambar 2.5



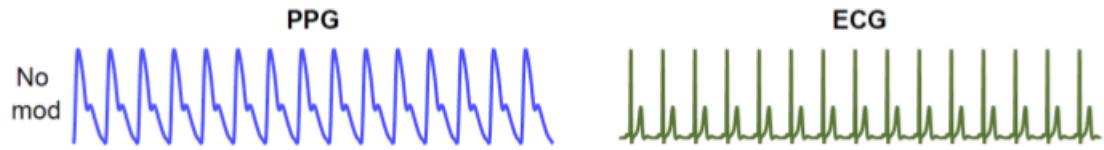
Gambar 2.4: Sinyal ECG berdasarkan titik fiducial



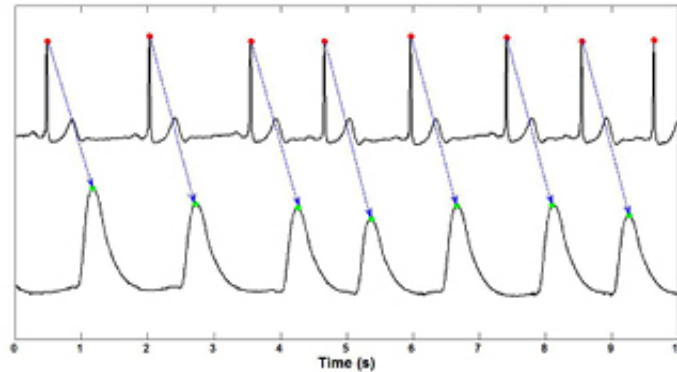
Gambar 2.5: Sinyal PPG berdasarkan titik fiducial

2.1.3 Bentuk Sinyal

Dapat dilihat, pada gambar 2.6, dengan mudah bahwa sinyal bentukan dari PPG dengan ECG berbeda secara morfologi (bentuk)[7]. 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 2.7 [8]. 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 2.6: Perbandingan sinyal ideal PPG dan ECG



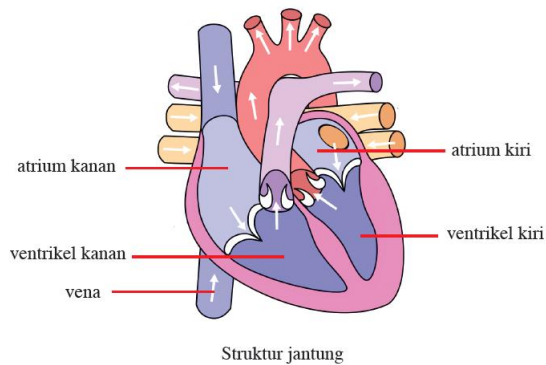
Gambar 2.7: Sinkronisasi antara ECG dan PPG

2.2 Aritmia

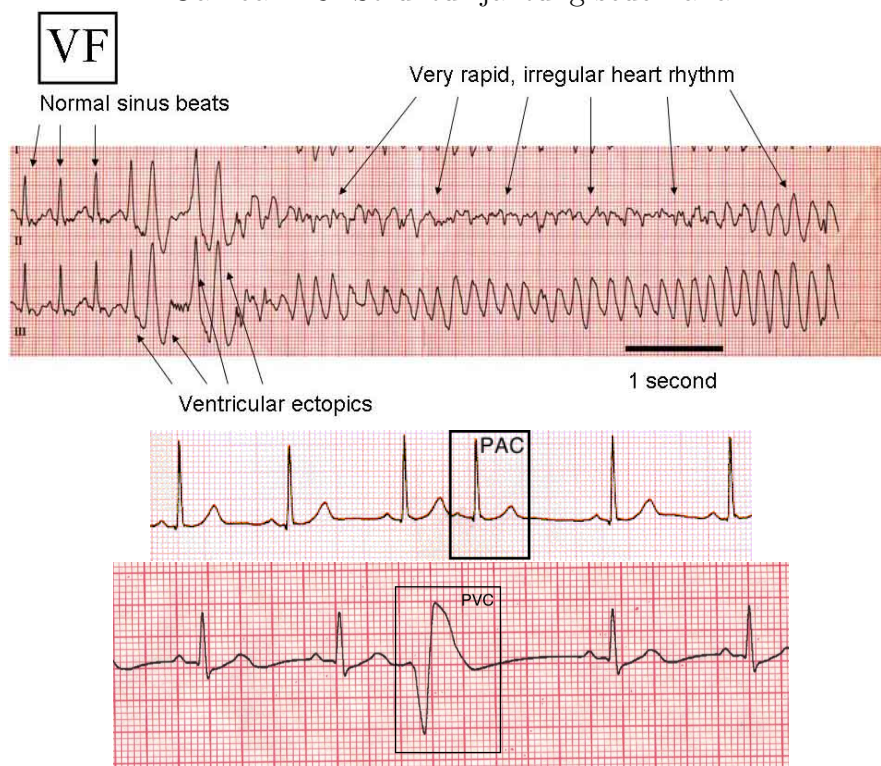
Aritmia adalah kategori gangguan jantung yang berupa tidak normalnya irama jantung. Beberapa penyakit jantung yang tergolong aritmia antara lain:

- *Tachycardia* (detak lebih cepat dari normal),
- *Bradycardia* (detak lebih lambat dari normal),
- *Premature Atrial Contraction* (PAC),
- *Premature Ventricular Contraction* (PVC),
- *Ventricular Tachycardia* (VT) (detak ventrikel sangat cepat), dan
- *Ventricular Fibrillation* (VF) (detak ventrikel tidak beraturan).

Premature Contraction berarti siklus detak yang *premature* (tidak pada waktunya) disebabkan oleh kontraksi pada atrium (bilik) atau ventrikel (se-rambi) 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 2.9.



Gambar 2.8: Struktur jantung sederhana



Gambar 2.9: a. Sinyal VF; b. Sinyal PAC; c. Sinyal PVC

2.2.1 Riset Klasifikasi Aritmia Otomatis

Untuk mengklasifikasi aritmia seorang dokter perlu melihat hasil rekam jantung seorang pasien seperti gambar 2.9, baik hasil rekam ECG maupun PPG. Akan sangat melelahkan jika seorang dokter secara terus menerus memeriksa rekam jantung seorang pasien. Oleh karena itu dibutuhkan sebuah algoritma yang dapat melakukan klasifikasi secara otomatis.

Telah banyak penelitian yang dilakukan yang melakukan otomatisasi klasifikasi

aritmia[xx]. Kini penelitian tersebut telah memiliki keakuratan yang cukup baik, mencapai 90%, dengan berbagai macam metode dan ekstraksi fitur.

Pada tahun xxx, asme asme asem
 Pada tahun xxx, asme asme asem
 Pada tahun xxx, asme asme asem
 Pada tahun xxx, asme asme asem
 Pada tahun xxx, asme asme asem
 Pada tahun xxx, asme asme asem

| Judul | Penulis | Fitur | Metode | Hasil |
|--------------------------|------------|---------|---------|-------|
| Penemuan bla bla adu adu | Alif Akbar | Titik R | Decisin | 90% |
| Penemuan bla bla adu adu | Alif Akbar | Titik R | Decisin | 90% |
| Penemuan bla bla adu adu | Alif Akbar | Titik R | Decisin | 90% |

Tabel 2.1: Perbandingan riset mengenai klasifikasi aritmia otomasis

2.3 Produk Monitoring Jantung di Pasaran

Meningkatnya kesadaran masyarakat akan penyakit jantung mendorong banyak perusahaan untuk membuat produk *monitoring* jantung. Perusahaan seakan berlomba memproduksi alat monitoring baik yang berstandar medis untuk penggunaan rawat intesif maupun yang tidak berstandar medis untuk penggunaan sehari hari. Salah satu perusahaan yang ikut memproduksi alat *monitoring* ialah perusahaan raksasa dari Korea, Samsung, yang mengeluarkan "Gear S3" pada tahun 2017 [xx]. Selain produk berbentuk alat (*hardware*), produk berbentuk program (*software*) yang hanya memanfaatkan *LED flash* di kamera *smartphone* sebagai sensor PPG juga banyak ditemukan[xx-xx-xx].



Gambar 2.10: a. Gear S3, Jam pintar produksi Samsung; b. Finger clip PPG; c. Portable ECG; d. Heart Rate App 1; e. Heart Rate App 2

2.4 Node.Js dan Mongo.Db

Sebuah sistem monitoring yang dapat berjalan secara Ubiquitous haruslah dibangun dengan konsep *Internet of Things* (IoT). IoT ialah konsep dimana objek objek (Things) dapat saling berinteraksi pada jaringan Internet tanpa membutuhkan manusia. Pada konsep IoT diperlukan setidaknya 3 komponen yaitu Sensor, Server dan Actuator. Sensor berfungsi sebagai pengambil data. Server yang menjalankan *web service* (layanan web, contoh: http server, mqtt broker dan db server) berfungsi sebagai pengolah data. Actuator berfungsi sebagai pelaksana perintah dari server, seperti mengeluarkan suara dan membelokkan/memutus arus listrik. Node.Js dan Mongo.Db, keduanya dibutuhkan untuk membangun sebuah web service pada server.

2.4.1 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* [9]. 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++ [10].

2.4.2 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 [11]:

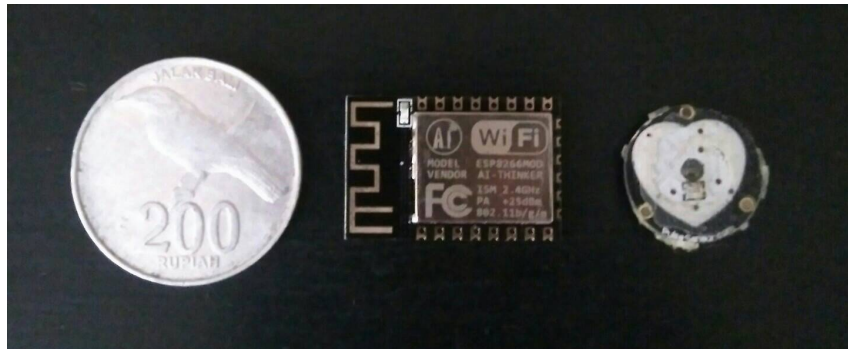
1. Membutuhkan beban penulisan data yang tinggi,
2. Skema data yang tidak stabil,
3. Ukuran data akan menjadi sangat besar,
4. Tidak memiliki seorang *administrator*



Gambar 2.11: a. Node JS; b. Mongo DB;

2.5 ESP-12

ESP-12 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.

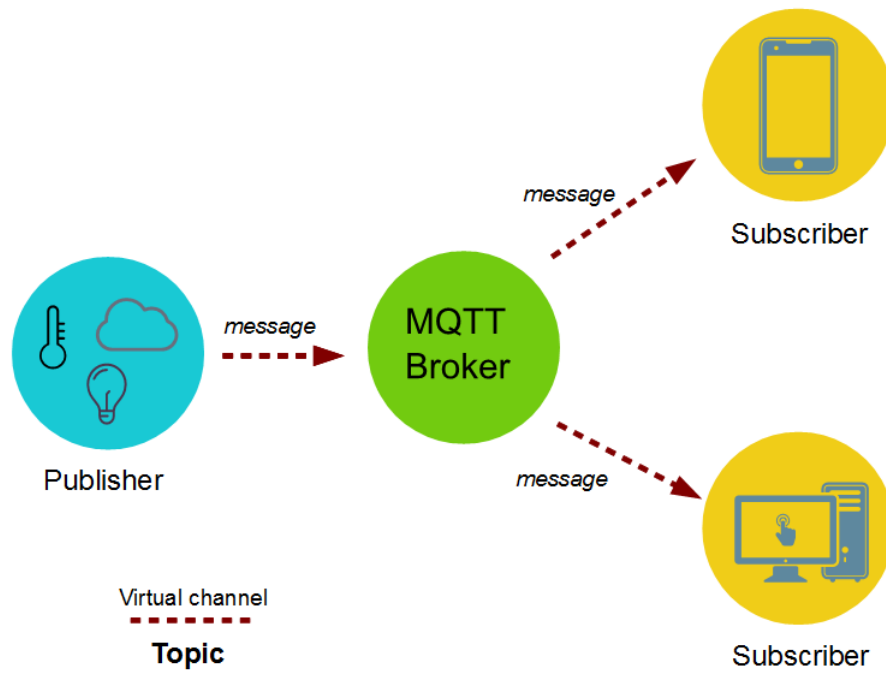


Gambar 2.12: Koin Rp200 - ESP-12E - Pulse Sensor

2.6 Protokol MQTT

Message Queuing Telemetry Transport (MQTT) adalah protokol transport dengan skema komunikasi publish dan subscribe. 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”, dimana pesan dikirim dengan skema *fire-and-forget* yang berarti tidak ada upaya menjamin pesan yang dikirim dapat sampai ke tujuan.
2. QOS 1 atau “At least once”, 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.
3. QOS 2 atau “Exactly once”, dimana pesan dikirim dengan jaminan diterima tepat sekali ke tujuan. Sehingga tidak ada pesan yang terduplikasi di tujuan.



Gambar 2.13: Cara kerja MQTT

Bab III

Metodologi dan Desain Sistem

3.1 Metodologi Penelitian

Metodologi yang digunakan menyelesaikan tugas akhir, ditunjukkan pada gambar diagram alir 3.1. Berikut penjelasan tiap tahap pada diagram alir:

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**

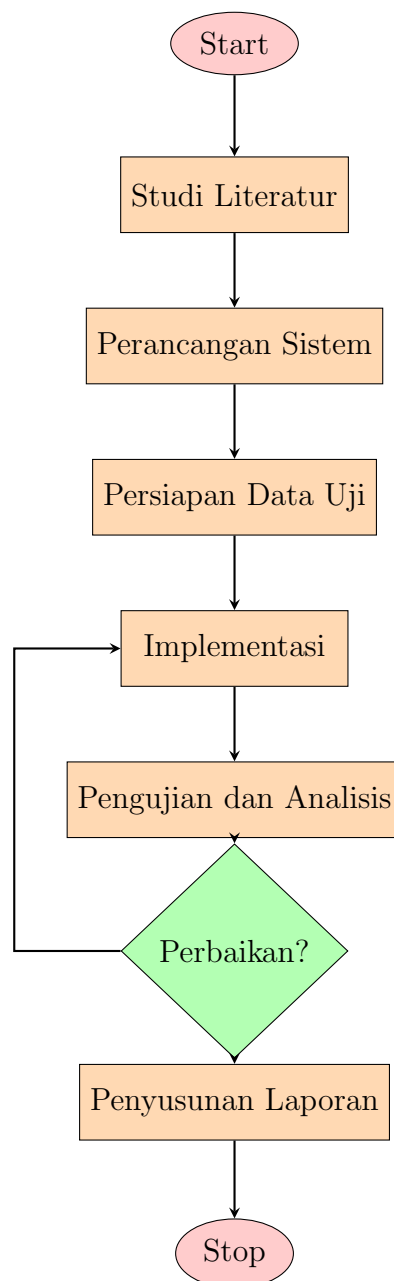
Pada tahap ini penulis menerapkan rancangan sistem baik yang berupa *software* maupun *hardware*. Hasil tahap ini ialah *software* dan *hardware* yang dapat berjalan tanpa masalah.

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.



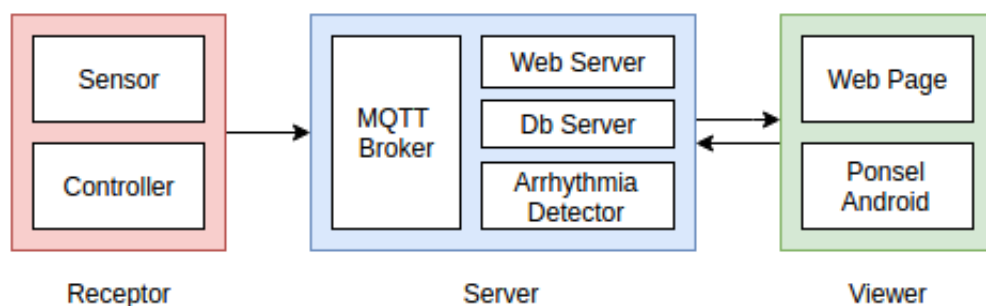
Gambar 3.1: Flowchart Metodologi

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, diperlukan dokter untuk mengambil kebijakan/keputusan medis. 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 *Receptor* yang diletakkan pada pergelangan tangan. Receptor kemudian secara periodik melakukan sampel dan mengirimkan sampel tersebut ke server untuk diproses lebih lanjut. Pengguna sistem dapat kapan saja melihat data aktivitas jantung melalui *Viewer* berupa halaman *web* atau ponsel android. Ketika server mendeteksi kemunculan aritmia, server akan secara otomatis mengirimkan pesan peringatan kepada *viewer* 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 *Receptor*, *Server*, dan *Viewer*.

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 algoritma pada sub bab ?? 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 [12], terlihat pada gambar 2.12. Alasan penulis milihan 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.

Controller

Sistem dirancang untuk monitoring terus menerus dan *Ubiquitous*. Maka receptor haruslah cukup kecil untuk dibawa kemana saja dan menggunakan media komunikasi *wireless* (tanpa kabel) untuk beriteraksi dengan server. 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 dengan ukuran yang kecil yaitu ESP-12. Oleh karena itu receptor dirancang menggunakan ESP-12.

Baterai

Untuk memungkinkan receptor dibawa kemana saja dan dikenakan terus menerus diperlukan baterai sebagai 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

Untuk mengimplementasikan konsep IoT server dirancang agar bisa melayani banyak *receiver* dan *viewer*. Oleh karena itu server harus melayani komunikasi dengan arus data yang tinggi. Alasan ini mendorong penulis memilih menggunakan protokol MQTT sebagai protokol komunikasi, NodeJs sebagai runtime dan MongoDB sebagai penyimpanan data. Server dirancang agar bisa berjalan pada satu perangkat. Hal ini berarti MQTT broker, Web server, DB server, dan Algoritma Detector berjalan pada satu alamat IP yang sama.

3.3.3 Viewer

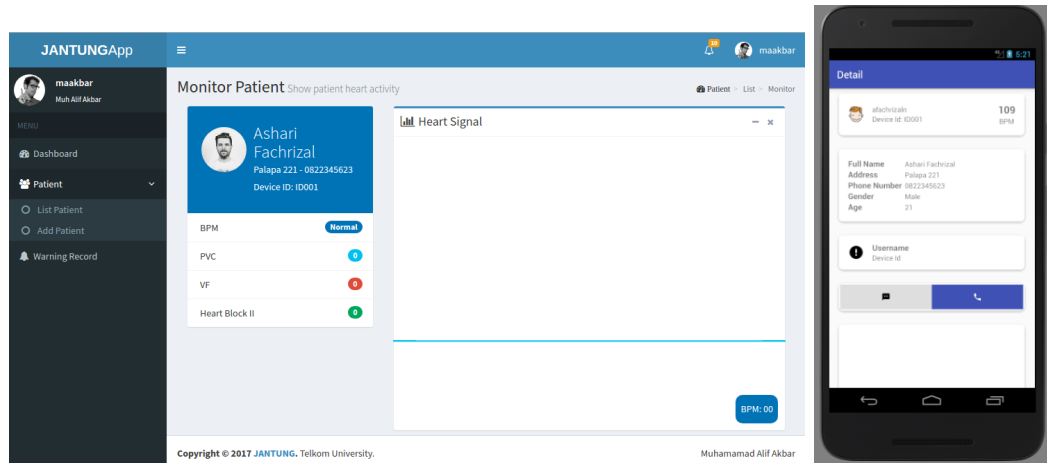
Sistem dirancang memiliki 2 saluran pemantauan yaitu halaman *web* dan aplikasi pada ponsel Android. Kedua saluran ini dapat melakukan pemantauan selama berada dalam jaringan yang sama dengan server. Penulis memilih Android karena memiliki jumlah pengguna terbesar didunia[xx] sehingga bisa diasumsikan sistem yang dirancang bisa digunakan oleh banyak orang.

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.

Aplikasi Ponsel Android

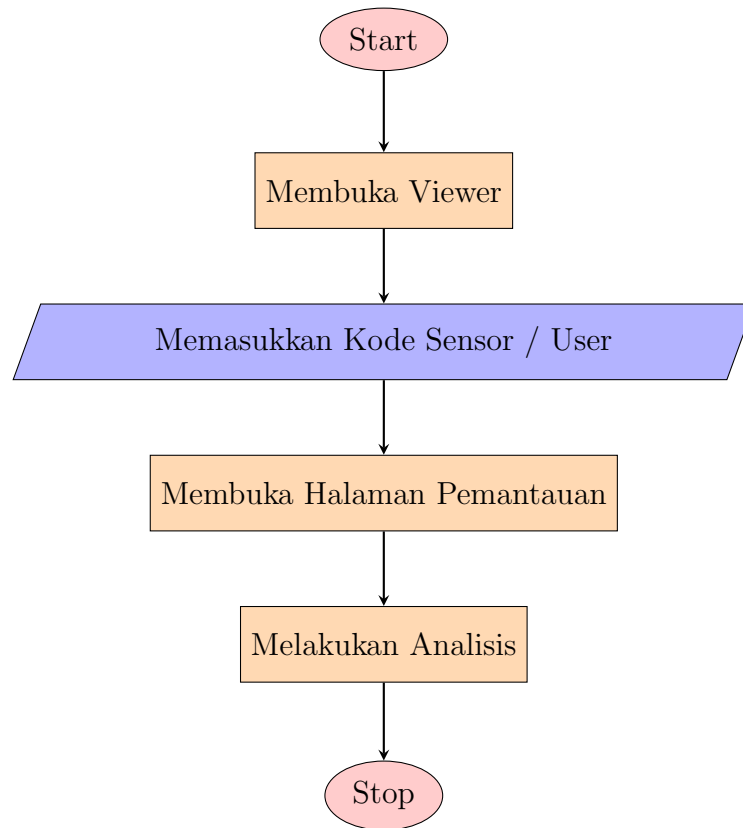
Aplikasi ponsel android dibangun untuk dapat berjalan pada ponsel android ber-OS (*operating system*) minimal Jelly Bean (Android v4.1). Pada aplikasi ini terjadinya aritmia ditandai dengan bunyi atau berubahnya status deteksi dan kode warna ikon seru. Kode warna merah berarti terdeteksi aritmia berbahaya,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

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

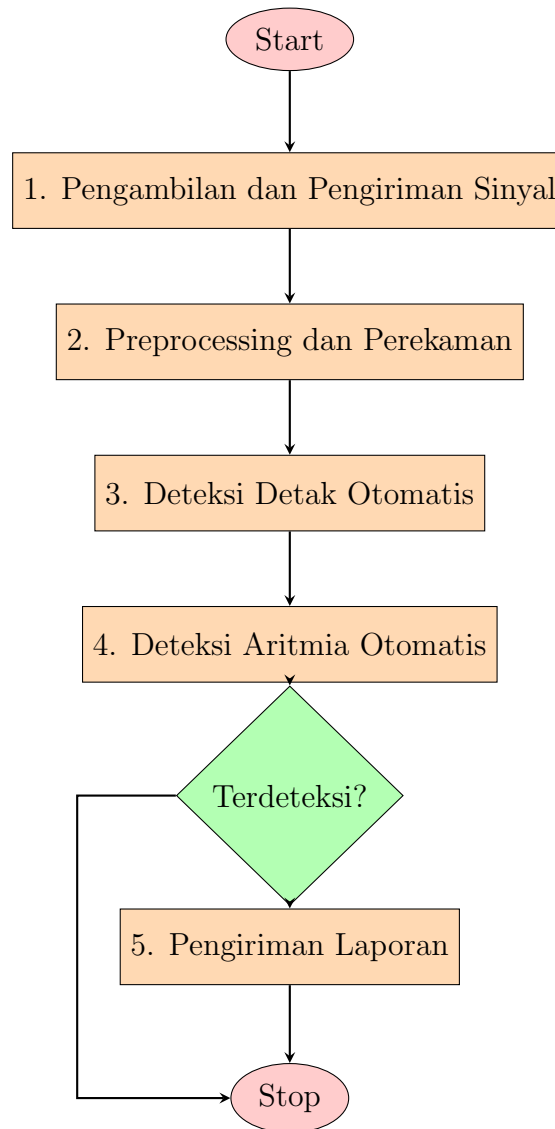


Gambar 3.6: Flowchart Rancangan Algoritma Pemantauan

Seorang pengguna baik pasien, keluarga pasien, maupun dokter perlu membuka sebuah perangkat *viewer*. 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 Algoritma Deteksi

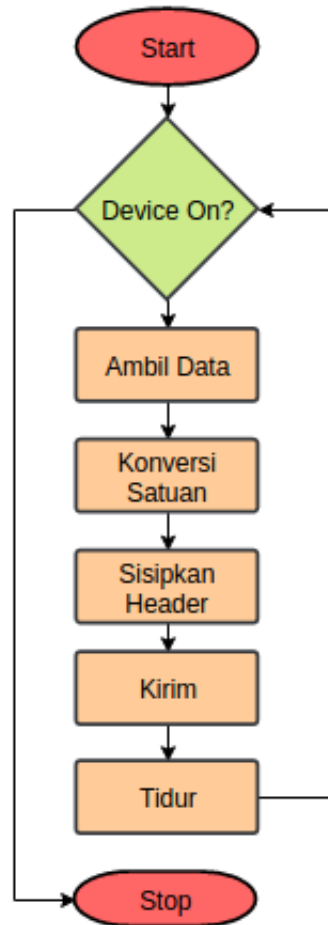
Fungsi berikutnya yang akan diterapkan dalam sistem ialah dapat melakukan pendeteksian aritmia otomatis. Untuk itu sistem perlu menerapkan algoritma deteksi. Algoritma deteksi yang diterapkan pada tugas akhir ini terbagi menjadi 5 tahap yaitu Pengambilan dan Pengiriman Sinyal, Preprocessing dan Perekaman, Deteksi Detak Otomatis, Deteksi Aritmia Otomatis dan Pengiriman Laporan. Alir tahap algoritma deteksi digambarkan pada gambar diagram alir 3.7.



Gambar 3.7: Flowchart Rancangan Algoritma Deteksi

3.5.1 Pengambilan dan Pengiriman Sinyal

Pengambilan dan Pengiriman sinyal dilakukan di Receptor. Langkah pertama ialah *controller* mengambil nilai pada pin analognya. Nilai pada pin analog lalu dikonversi menjadi satuan *Volt*. Nilai yang telah dikonversi kemudian disisipkan header lalu dikirim menggunakan protokol MQTT dengan QoS 0. *Controller* kemudian tidur selama 2 ms lalu mengulang pengambilan dan pengiriman. 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 oleh server. Server melakukan *preprocessing* pada data. *Preprocessing* yang dilakukan terbagi menjadi *data completer*, *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*. Alir algoritma *preprocessing* dapat dilihat pada diagram alir 3.11. Rangkaian *preprocessing* ini merupakan modifikasi pada algoritma yang diusulkan oleh Pan-Tomkin (1985) dan Deshmane (2009) agar kedua algoritma ini bisa bekerja untuk ECG dan PPG.

Data Completer

Data Completion berfungsi ialah algoritma untuk menangani hilangnya data selama pengiriman. Hal ini mungkin terjadi karena data dikirimkan dari *receptor* 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.

$$y(n) = \frac{n(v_2 - v_1)}{d} + v_1 \quad (3.1)$$

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

Filtering

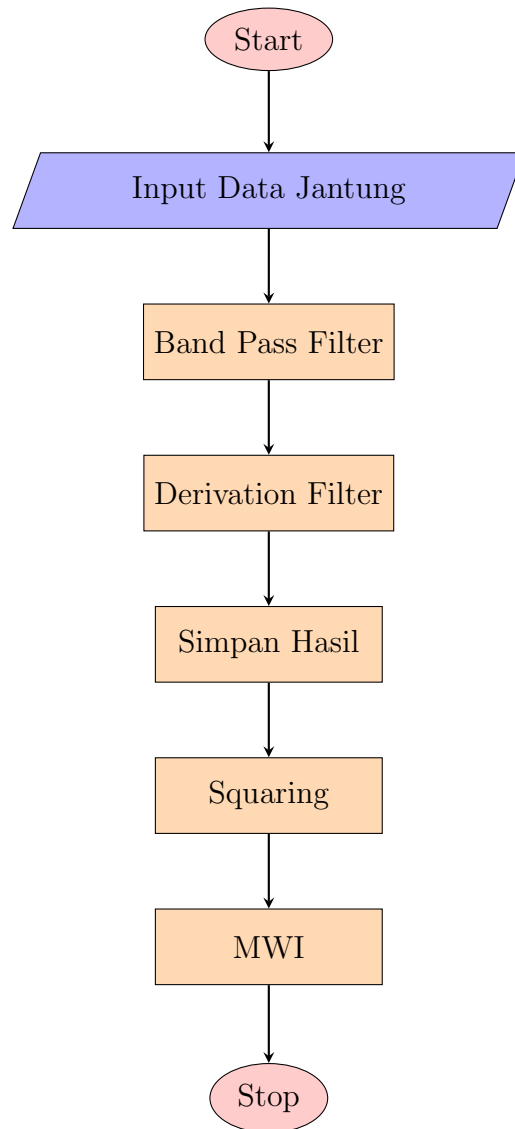
Filtering berfungsi untuk menghilangkan noise yang mempengaruhi sinyal. Noise yang umum terdapat ialah *muscle noise* dan *baseline wander*. Filtering yang diterapkan terbagi menjadi dua tahap yaitu *Band Pass Finite Impulse Response(FIR) Filter* kemudian *First Order Derivation Filter*. Tahap ini sesuai dengan algoritma yang diusulkan oleh Pan-Tomkins (1985)[13] dan Deshmone (2009) [14]. *Band Pass Filter* yang digunakan memiliki frekuensi response 5-15Hz. Daftar koefisien lengkap untuk *Band Pass Filter* dan *Derivation Filter* tercantum pada tabel x.x di bab lampiran.

Squaring dan MWI

Sesuai namanya *Squaring* melakukan penguadratan (persamaan 3.2) terhadap data. Sedangkan MWI melakukan penghalusan data berdasarkan n data sebelum. MWI dilakukan dengan persamaan 3.3.

$$y = x^2 \quad (3.2)$$

$$y(n) = 2v_n + v_{n-1} - v_{n-3} - 2v_{n-4} \quad (3.3)$$



Gambar 3.9: Flowchart Rancangan Algoritma Preprocessing

3.5.3 Deteksi Detak Otomatis

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.

Modifikasi yang diterapkan merubah *threshold* berdasarkan rata-rata nilai pengukuran pada suatu durasi window. Peak ditentukan dengan cara mengambil nilai maksimum dari suatu wilayah yang melewati threshold (persamaan 3.4 dan 3.5). Algoritma deteksi puncak dan penghilang puncak palsu dapat dilihat pada kolom algoritma 1 dan 2.

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

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

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:    $rThreshold := \beta * \text{mean}(\text{distance}(\text{peaksArea}))$ 
3:   for all  $peak_i \in \text{peaksArea}$  do
4:     if  $\text{distance}(peak_i, peak_{i-1}) < rThreshold$  then            $\triangleright$  Calculate
       distance from  $peak_i$  to  $peak_{i-1}$ 
5:        $\text{remove}(peak, \text{peaksArea})$             $\triangleright$  Remove peak from peak area
6:     end if
7:   end for
8:   return  $\text{peaksArea}$ 
9: end function
```

3.5.4 Deteksi Aritmia Otomatis

Deteksi Aritmia merupakan salah satu tujuan dari tugas akhir ini. Untuk melakukan deteksi aritmia pada tugas akhir ini menerapkan algoritma usulan Tsipouras(2005)[15]. 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. Pada penelitian Tsipouras (2005), 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, $RR1_i, RR2_i, RR3_i$.

Algoritma ini hanya dapat mendeteksi 4 kelompok aritmia. Pengelompokan aritmia ini dapat dilihat pada tabel 3.1. *Rule* untuk pengelompokan ini dapat dilihat pada gambar 3.10.

| MIT-BIH Beat Symbol | Arti Symbol | Kategori |
|---------------------------|------------------------------------|----------|
| N | Normal | 1 |
| . | Normal | |
| / | Paced Beat | |
| f | Fusion of paced and normal beat | |
| x | Non-conducted P-wave (blocked APB) | |
| L | Left bundle branch block | |
| R | Right bundle branch block | |
| Q | Unclassified | |
| V | PVC | 2 |
| A | PAC | |
| ! | VF | 3 |
| (BII | Heart block | 4 |

Tabel 3.1: Pengelompokan Tipe beat

```

1. Initialization

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

2. Rule 1 - Ventricular flutter/fibrillation beat classification

a. If  $RR2_i < 0.6$  sec and  $1.8 \cdot RR2_i < RR1_i$  then

    i.  $RR2_i$  is classified in category 3.

    ii. The  $RR2_k$  of all windows  $k = i+1, i+2, \dots, i+n$  with
        ( $RR1_k < 0.7$  and  $RR2_k < 0.7$  and  $RR3_k < 0.7$ ) or
        ( $RR1_k + RR2_k + RR3_k < 1.7$ ) 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 ( $(1.15 \cdot RR2_i < RR1_i)$  and  $(1.15 \cdot RR2_i < RR3_i)$ ) or
( $(|RR1_i - RR2_i| < 0.3)$  and  $(RR1_i < 0.8)$  and  $(RR2_i < 0.8)$ ) and
 $(RR3_i > 1.2 \cdot \text{mean}(RR1_i, RR2_i))$  or
( $(|RR2_i - RR3_i| < 0.3)$  and  $(RR2_i < 0.8)$  and  $(RR3_i < 0.8)$ ) and
 $(RR1_i > 1.2 \cdot \text{mean}(RR2_i, RR3_i))$ 
then  $RR2_i$  is classified in category 2.

4. Rule 3 - 2° heart block beats

If  $(2.2 < RR2_i < 3.0)$  and
 $(|RR1_i - RR2_i| < 0.2$  or  $|RR2_i - RR3_i| < 0.2)$  then
then  $RR2_i$  is classified in category 4

5. Update Window

i = i + 1

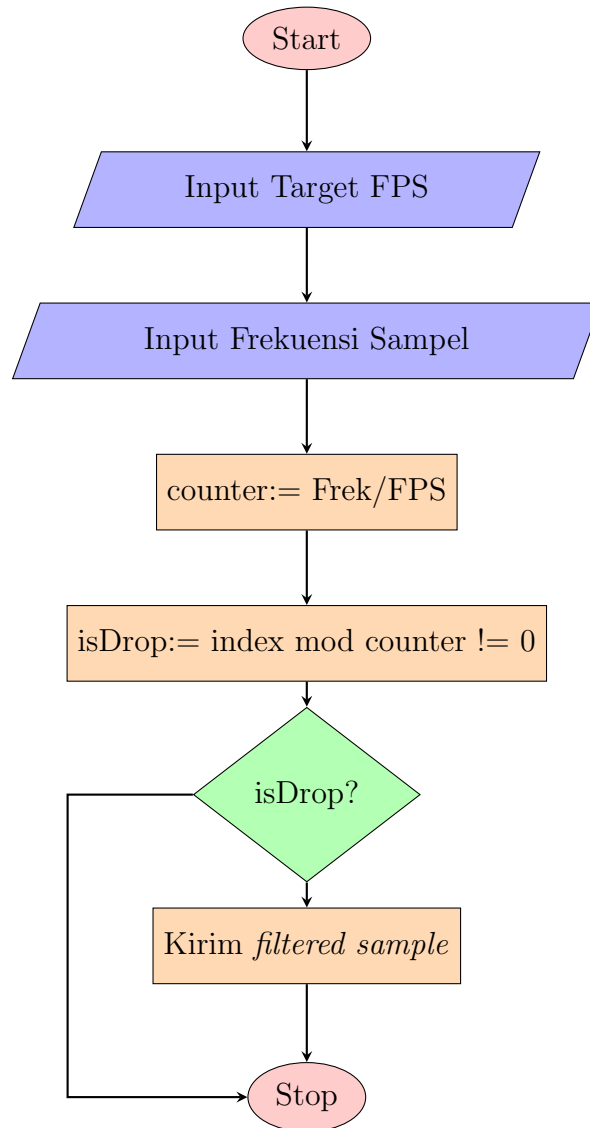
```

Gambar 3.10: *Rule/Algoritma* yang diajukan Tsipouras (2005)

3.5.5 Pengiriman Laporan

Pengguna dapat menerima laporan atau melakukan pemantauan pada *Viewer* mereka. Namun untuk menjaga perangkat *Viewer* dapat *me-render* tampilan dengan baik perlu dilakukan penurunan kecepatan sample (lihat diagram

alir). Selain itu *Viewer* harus terus menerus terkoneksi dengan *Server* untuk bisa mendapatkan data *real-time*. Sehingga kapanpun *server* mendeteksi aritmia pengguna dapat melihat peringatan pada device mereka.

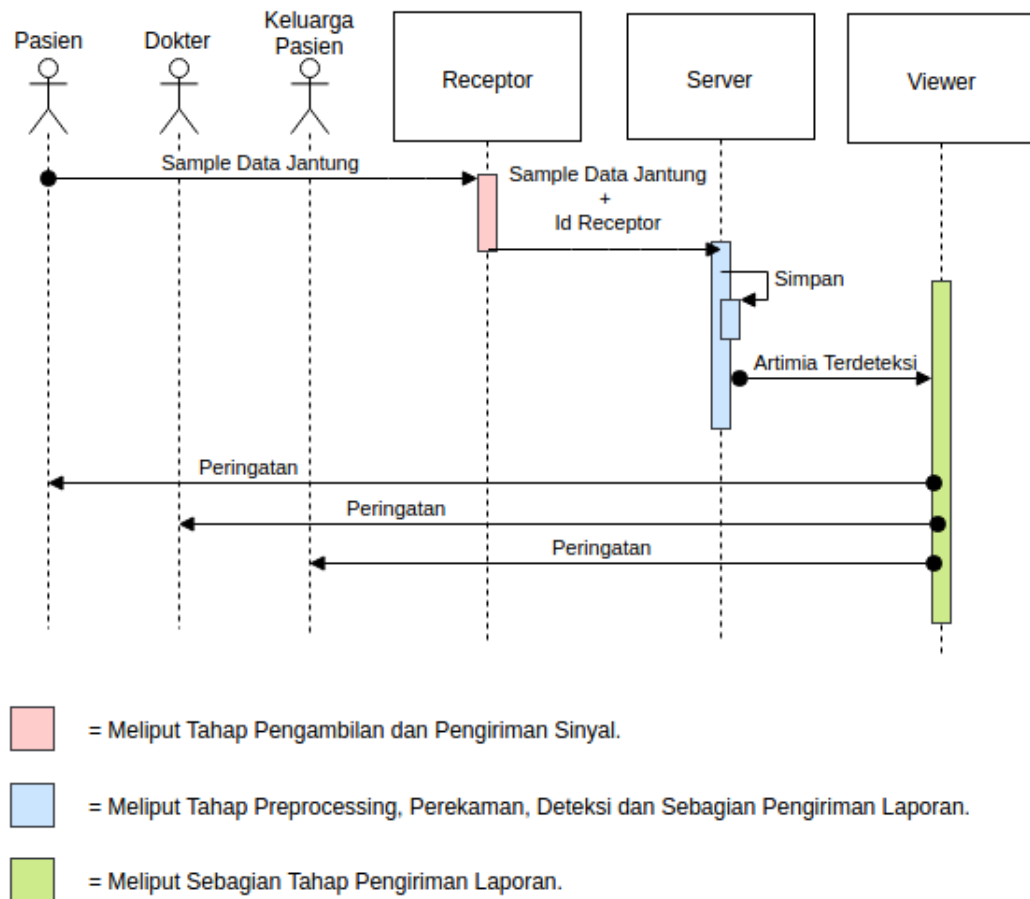


Gambar 3.11: Flowchart Rancangan Algoritma Preprocessing

3.5.6 Hubungan Algoritma Deteksi Dengan Perangkat Keras

Hubungan antara algoritma deteksi dengan perangkat keras digambarkan pada gambar diagram tahap 3.12. Seorang pasien yang mengenakan *receptor* akan diambil data jantungnya kemudian dikirim ke *server*. Ketika terdeteksi

aritmia, *server* akan mengirim *flag* manandakan aritmia terdeteksi ke *viewer* yang kemudian dilihat oleh pasien, dokter dan keluarga pasien.



Gambar 3.12: Diagram Tahap 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:

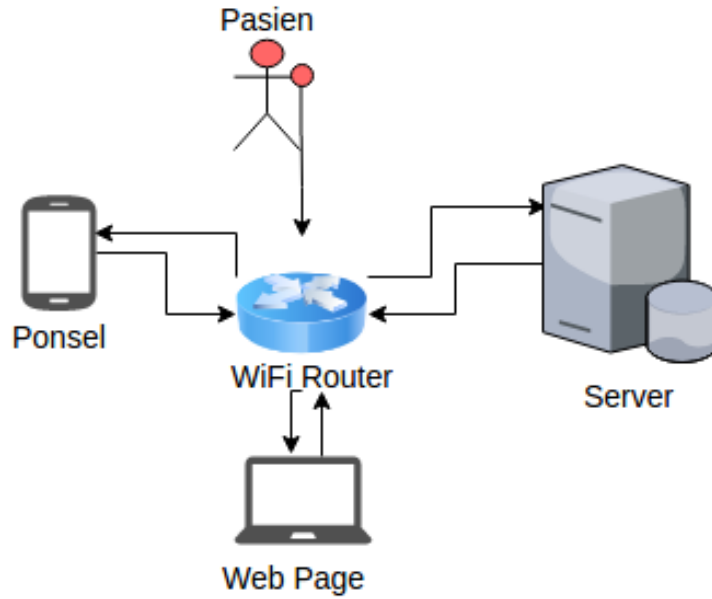
| No | Parameter | Tujuan yg Dicapai | Alasan |
|----|------------------|-------------------|--|
| 1 | Eksistensi Fitur | 2, 4 | Mengetahui apakah fitur yang direncanakan bisa berjalan. |
| 2 | Delay | 1, 3, 4 | Dengan mengukur <i>delay</i> dapat diketahui berapa lama proses pengiriman sehingga dapat diketahui seberapa <i>Real Time</i> sistem dapat berjalan. |
| 3 | Execution Time | 1, 2, 3, 4 | Dengan mengukur execution time dapat diketahui seberapa banyak sensor yang dapat ditangani oleh spesifikasi server tertentu |
| 4 | FPS | 2, 4 | Dengan mengukur FPS dapat diketahui maksimum FPS agar dapat dijalankan pada banyak <i>viewer</i> |
| 5 | 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 3.2. Pengujian dilakukan dengan menghubungkan sebuah *receiver*, kepada sebuah *server* dan dipantau oleh 2 *viewer* (web dan ponsel). Hubungan jaringan perangkat keras digambarkan pada gambar 3.13.

| No | Perangkat | Spesifikasi |
|----|----------------|---|
| 1 | Receptor | <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 | Server | <ul style="list-style-type: none"> • Laptop ASUS A43SD • Intel i3-2350M CPU @ 2.30GHz • RAM DDR3 8GB |
| 3 | Viewer Web | <ul style="list-style-type: none"> • Laptop Lenovo G40S • AMD A8 5550M CPU @ 2.1GHz • RAM DDR3 8GB |
| 4 | Viewer 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 3.2: 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 *viewer*. Pengujian ini ditujukan untuk menguji semua parameter selain parameter 5 (Akurasi).

Delay

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

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

Execution Time

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

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

FPS

FPS diukur dengan mencoba konfigurasi *downsample* pada *server* dan melihat tampilan pada *viewer*.

3.6.4 Pengujian Algoritma Deteksi

Pengujian algoritma deteksi dilakukan dengan memasukkan dataset ECG dari MIT-BIH Arrhythmia Database [16][17] ke dalam sistem. Untuk mempermudah visualisasi dan analisis data, penulis menjalankan algoritma deteksi pada bahasa pemrograman python untuk selanjutnya diimplementasikan ke Node.js. Pengujian ini ditujukan untuk menguji parameter akurasi.

Dataset

Dataset dari MIT-BIH [16][17] 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 3.3. Setiap *record* telah dianotasi (ditandai) oleh dokter ahli jantung[17]. Rekapitulasi anotasi pada dataset dapat dilihat pada tabel 3.4 dan 3.5.

| Series 100 | | Series 200 | |
|------------|-------------|------------|-------------|
| Data Set | Number Beat | Data Set | Number Beat |
| 100 | 2273 | 200 | 2601 |
| 101 | 1865 | 201 | 2000 |
| 102 | 2187 | 202 | 2136 |
| 103 | 2084 | 203 | 2980 |
| 104 | 2229 | 205 | 2656 |
| 105 | 2572 | 207 | 2332 |
| 106 | 2027 | 208 | 2955 |
| 107 | 2137 | 209 | 3005 |
| 108 | 1774 | 210 | 2650 |
| 109 | 2532 | 212 | 2748 |
| 111 | 2124 | 213 | 3251 |
| 112 | 2539 | 214 | 2262 |
| 113 | 1795 | 215 | 3363 |
| 114 | 1879 | 217 | 2208 |
| 115 | 1953 | 219 | 2287 |
| 116 | 2412 | 220 | 2048 |
| 117 | 1535 | 221 | 2427 |
| 118 | 2288 | 222 | 2483 |
| 119 | 1987 | 223 | 2605 |
| 121 | 1863 | 228 | 2053 |
| 122 | 2476 | 230 | 2256 |
| 123 | 1518 | 231 | 1573 |
| 124 | 1619 | 232 | 1780 |
| | | 233 | 3079 |
| | | 234 | 2753 |
| Total Set | Beat Num | Total Set | Beat Num |
| 23 | 47668 | 25 | 62491 |

Tabel 3.3: Tabel Rakpitulasi Dataset

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

Tabel 3.4: Tabel Rakpitulasi Aritmia series 100

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

Tabel 3.5: Tabel Rakpitulasi Aritmia series 200

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 dan 3.9.

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

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

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

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 Eksistensi Fitur

Pemantauan (aktivitas jantung) dan Deteksi (detak dan aritmia) berhasil dilakukan secara realtime (parameter lainnya, dibahas pada sub bab berikutnya). Berikut beberapa perbandingan mode *monitoring* sistem sejenis lainnya yang berada pada puncak 8 Android Playstore (kata pencarian heart rate) [18] ditambah 2 aplikasi medis.

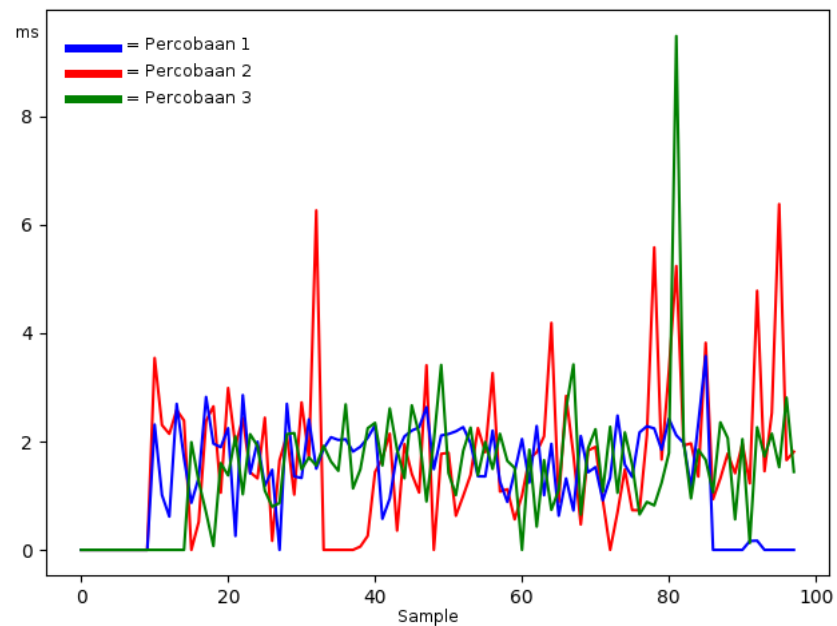
| No | Produk | Sensor | Fitur | | | | | | | |
|----|---|------------|-------|---|---|---|---|---|---|---|
| | | | A | B | C | D | E | F | G | H |
| 1 | Tugas Akhir | PPG | Y | Y | Y | Y | Y | Y | Y | N |
| 2 | Instant Heart Rate : Heart Rate & Pulse Monitor | PPG | Y | N | Y | Y | N | Y | N | Y |
| 3 | iCare Health Monitor (BP & HR) | PPG | Y | N | Y | N | N | Y | N | Y |
| 4 | Heart Rate Monitor(REPS) | PPG | Y | N | Y | Y | N | Y | N | Y |
| 5 | Runtastic Heart Rate Monitor & Pulse Checker | PPG | Y | N | Y | N | N | Y | N | N |
| 6 | Cardiograph - Heart Rate Meter | PPG | Y | N | Y | Y | N | Y | N | Y |
| 7 | ASUS Heart Rate | PPG | N | N | N | N | N | Y | N | N |
| 8 | Samsung Health | PPG | Y | Y | N | N | N | Y | N | Y |
| 9 | Heart Rate Monitor(Meet Your Need Production) | PPG | N | N | N | N | N | Y | N | N |
| 10 | MobECG | ECG | N | Y | Y | Y | N | Y | N | N |
| 11 | CMS50Dplus | ECG | N | Y | Y | Y | N | Y | N | N |

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

4.1.2 Delay

Pengujian dilakukan oleh pengguna yang bergerak secara bebas dalam wilayah cakupan router (*receptor*, *router* dan *server* masih dalam satu wilayah) sehingga tidak ada proses routing antara router. Setelah melakukan pengujian sebanyak 3 kali pada waktu berbeda (pagi, siang, malam) didapatkan rata-rata delay percobaan-1 1.60624 ms, percobaan-2 1.36287 ms dan percobaan-3 1.45066 ms. Hasil pengukuran delay tertera pada gambar 4.1. Rata-rata ketiga percobaan ialah 1.47326 ms per sampel.



Gambar 4.1: Hasil Pengukuran Delay 100 sample

4.1.3 Execution Time

Hasil pengukuran execution time tertera pada grafik ...

4.1.4 FPS

Setelah melakukan pengujian, ditemukan FPS terbaik untuk dapat berjalan dengan lancar pada kedua jenis *viewer* yaitu 20 FPS. Hasil lengkap percobaan FPS dapat dilihat pada tabel ...

4.1.5 Akurasi

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

Akurasi Detak

Dilakukan percobaan untuk menemukan konfigurasi konstanta deteksi α , β dan d (durasi window).

Akurasi Aritmia

4.2 Pembahasan

Dengan terdapatnya mode *monitoring* maka parameter 1 (Eksistensi Fitur) dinyatakan terpenuhi.

Delay dan *Execution time* ini terhitung kecil sehingga tidak menyebabkan fenomena *bottleneck* pada sisi server.

Bab V

Kesimpulan dan Saran

5.1 Kesimpulan

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[xx]. 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.
- [3] 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.
- [4] 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>.
- [5] C. H. SeungMin Lee HyunSoon Shin, “Effective ppg sensor placement for reflected red and green light, and infrared wristband-type photoplethysmography”, in *18th International Conference on Advanced Communication Technology (ICACT)*, 2016.
- [6] 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. [Online]. Available: <http://www.mdpi.com/2079-9292/3/2/282>.
- [7] M. e. a. Pimentel, “Probabilistic estimation of respiratory rate from wearable sensors”, *Springer International Publishing*, 2015.
- [8] V. Kalidas and L. S. Tamil, “Cardiac arrhythmia classification using multi-modal signal analysis”, *Physiological Measurement*, 2016.
- [9] Node.Js. (2017). Node js, [Online]. Available: <https://nodejs.org/en/>.
- [10] I. Gouy. (2016). The computer language benchmarks game, [Online]. Available: <http://benchmarksgame.alioth.debian.org/>.
- [11] DA-14. (2017). Mongodb vs mysql comparison: Which database is better?, [Online]. Available: <http://benchmarksgame.alioth.debian.org/>.
- [12] Y. G. Joel Murphy. (2017). Pulse sensor amped, [Online]. Available: <https://pulsesensor.com>.

- [13] W. J. T. Jia Pu Pan, “A real-time qrs detection algorithm”, *IEEE Transactions on Biomedical Engineering*, 1985.
- [14] A. V. Deshmane, “False arrhythmia alarm suppression using ecg, abp, and photoplethysmogram”, Master Thesis, MIT, 2009.
- [15] D. S. M.G. Tsipouras D.I. Fotiadis, “An arrhythmia classification system based on the rr-interval signal”, *Artificial Intelligence in Medicine*, 2005.
- [16] M. R. Moody GB, “The impact of the mit-bih arrhythmia database”, *IEEE Eng in Med and Biol*, 2001.
- [17] A. L. Goldberger AL. (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>.
- [18] A. P. Store. (2017). Apps, [Online]. Available: <https://play.google.com/store/search?q=heart%20rate&c=apps&hl=en>.

Lampiran