

IoT on Heart Arrhythmia Real Time Monitoring

Muhammad Alif Akbar ^{#1}, Satria Mandala ^{*2}

Department of Informatics Engineering,
School of Computing, Telkom University
Bandung, Jawa Barat, Indonesia*

¹ maakbar@student.telkomuniversity.ac.id

² satriamandala@telkomuniversity.ac.id

Abstract

Heart monitoring is popular in the recent 5 years. We can see this with emergence of various cardiovascular monitoring products based on wearable sensors. Those products commonly communicates using radio telemetry which has expensive operational costs. Some research try to implement internet of things (IoT) concept to solve the issue. However, those IoT implementation aren't efficient enough. The research are only focused on how to read the sensors data and allowing it to be monitored on real-time. This research proposed a cloud based IoT architecture to monitor arrhythmia, one type of a common heart attack, which more efficient than previous research. Arrhythmia detector that used in this paper is an improvement of algorithm proposed by Tsipouras et al, which using R peak on ECG. The system proposed on this paper has been tested using MIT-BIH datasets and has result 93.11% accuracy against 3 arrhythmia class, that is PAC, PVC and VT. The interesting result is that by implementing IoT, the R-Peak detection algorithm's execution time decreased up to 30% compared to has been proposed by Pan and Tompkins. The average of execution time of every sample is decreased to 0.00749 ms.

Keywords: IoT, heart monitoring, Arrhythmia

Abstrak

Monitoring jantung telah populer sejak 5 tahun terakhir. Hal ini ditandai dengan munculnya berbagai produk monitoring jantung berbasis wearable sensor. Umumnya komunikasi yang digunakan pada sistem tersebut menggunakan radio telemetry dengan biaya operasional yang mahal. Beberapa riset mencoba menggunakan konsep internet of things (IoT) untuk mengatasi hal tersebut. Namun demikian, desain komunikasi IoT yang ada belum efisien. Ini disebabkan riset yang ada hanya berfokus pada bagaimana hasil baca sensor dapat dipantau secara realtime. Untuk mengatasi hal tersebut, riset ini mengusulkan sebuah arsitektur IoT berbasis cloud untuk memonitor aritmia, salah satu jenis penyakit jantung yang umum ditemukan. Deteksi aritmia yang diusulkan adalah pengembangan algoritma yang sebelumnya diusulkan oleh Tsipouras et al, dengan menggunakan deteksi fitur R dalam ECG. Sistem yang diusulkan pada paper ini telah diuji menggunakan dataset MIT-BIH dan menghasilkan akurasi 93.11% terhadap 3 kelas aritmia, yaitu PAC, PVC dan VT. Menariknya, dengan penerapan IoT, efisiensi waktu eksekusi algoritma pendeteksi fitur R meningkat 30% dibanding yang diusulkan oleh Pan dan Tompkins. Terbukti dengan rendahnya waktu rata-rata eksekusi tiap sampel data, yaitu sekitar 0.00749 ms.

Kata Kunci: IoT, heart monitoring, Arrhythmia

I. PENDAHULUAN

BELAKANGAN ini monitoring jantung mendapatkan perhatian lebih dari publik. Hal ini terlihat dari bermunculannya produk *wearable* yang memungkinkan monitoring jantung dimana saja dan

kapan saja [4], [6], [10]. Umumnya produk tersebut menggunakan *electrocardiography* (ECG) ataupun *photoplethysmography* (PPG) sebagai sensornya. Penggunaan ECG dapat ditemukan pada produk produk holter monitor [6]. Penggunaan PPG dapat ditemukan pada produk produk jam tangan pintar seperti FitBit dan Gear Watch yang menambahkan PPG di bagian belakang jamnya [4], [10]. Sensor-sensor ini akan membaca sinyal jantung lalu menampilkan perkiraan jumlah detak jantung permenit (BPM) pada layar produk ataupun layar ponsel yang terhubung ke produk tersebut. Selain produk *wearable* diatas, produk berupa software yang memanfaatkan kamera pada belakang ponsel juga banyak ditemukan pada *online store* seperti *android play store* [13].

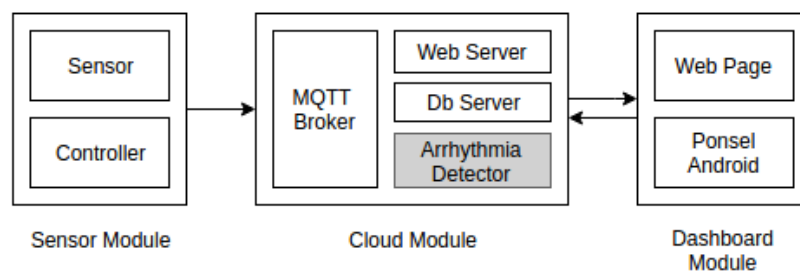
Beberapa penelitian telah dilakukan untuk menerapkan konsep IoT [3], [7], [8], [11]. Namun penelitian tersebut belum dapat memaksimalkan kemampuan yang ditawarkan oleh IoT. Kemampuan yang belum diterapkan seperti *multi user monitoring*, *online recording*, *online analyzing*, dan *real time alerting*. Terlebih penelitian tersebut belum mempertimbangkan optimasi terhadap waktu pengiriman dan pemrosesan data. Penelitian tersebut hanya berfokus pada bagaimana hasil baca sensor dapat dipantau secara realtime oleh orang lain di tempat lain.

Pada paper ini kami merancang sebuah arsitektur IoT yang menerapkan deteksi aritmia pada *cloud*. Deteksi aritmia yang diterapkan merupakan usulan algoritma oleh Tsipuras et al [14] yang menggunakan fitur R pada ECG. Untuk mendeteksi fitur R, diterapkan modifikasi terhadap algoritma usulan Pan-Tompkins [9]. Modifikasi yang dilakukan ditujukan untuk mengurangi waktu eksekusi yang diperlukan algoritma sehingga *cloud* dapat melayani lebih banyak *sensor*. Pembahasan tentang arsitektur monitoring aritmia berbasis iot yang diusulkan dapat dilihat pada bab II. Bab III menjelaskan tentang algoritma yang digunakan untuk mendeteksi terjadinya aritmia. Sedangkan Bab 4 mendiskusikan tentang pengujian sistem dan hasilnya.

II. ARSITEKTUR MONITORING ARITMIA BERBASIS INTERNET OF THINGS (IoT)

Sebuah sistem IoT umumnya memiliki 3 komponen yaitu *Sensor*, *Server* (selanjutnya disebut *cloud*) dan *Actuator*. Pada rancangan arsitektur di paper ini, tidak ada *actuator* yang bersifat sebagai pelaksana suatu perintah. *Actuator* yang diterapkan bersifat sebagai penerima pesan peringatan ketika terjadi aritmia dan modul untuk melakukan *real time monitoring*, selanjutnya disebut *dashboard*.

Secara umum sistem bekerja dimulai dari sebuah sensor membaca sinyal jantung seorang pasien. Kemudian sensor akan terkoneksi ke *cloud* dan mengirimkan hasil bacanya secara periodik. Protokol komunikasi yang digunakan dalam sistem ialah MQTT. *Cloud* akan memroses sampel dan mengirimkan hasil *filtering* dan notifikasi terjadinya aritmia kepada dashboard. *Dashboard* akan berkomunikasi dengan *cloud* tentang sensor mana yang ingin dipantau. Gambaran ini dapat dilihat pada gambar 1.



Gambar 1. Diagram Rancangan Keseluruhan Sistem

A. Sensor Module

Sensor Module dirancang hanya untuk membaca dan mengirimkan sinyal jantung, tidak ada pemrosesan berat pada modul. Hal ini ditujukan agar sensor dapat melakukan sampling dengan frekuensi yang tinggi. Selain itu, hal tersebut memungkinkan kedua jenis sensor (ECG atau PPG) dijadikan *input* pada modul.

Sensor yang dirancang melakukan *sampling* pada 200Hz atau 5ms/sampel. Frekuensi sampel yang tinggi ini membutuhkan protokol komunikasi yang cepat sehingga sampel tidak memenuhi *buffer* yang disediakan. Oleh karena itu, MQTT dipilih sebagai protokol komunikasi pada arsitektur yang dirancang. Setiap pesan dikirim menggunakan skema *fire-and-forget* pada MQTT (MQTT-QOS 0). Namun, karena tidak ada jaminan sampel sampai pada *cloud* maka perlu dilakukan penanganan data hilang. Hal ini dilakukan dengan memberikan index pada setiap sampel yang dikirim. Index ini akan direset setiap 1000 sampel untuk meminimalkan penggunaan memori pada modul dan mengurangi ukuran paket yang dikirim.

B. Cloud Module

Semua pemrosesan dilakukan di *cloud/server*. *Cloud* bekerja dengan membagi pekerjaan kepada unit-unit proses. Sebuah unit proses secara khusus menangani sebuah sensor dengan suatu ID unik. Sistem membuat unit proses ketika sebuah ID baru saja terhubung dan menutupnya ketika ID tersebut berhenti terhubung.

Sebuah unit proses akan memproses sampel berdasarkan index yang menyertai sampel tersebut. Ketika terdeteksi index yang hilang maka akan dilakukan penanganan data yang hilang. Index dikatakan hilang jika terjadi lompatan index yang diterima, misalnya proses menanti index bernomor 5 namun yang diterima ialah index bernomor 10. Penanganan data yang dilakukan ialah menghitung berapa nilai yang hilang berdasarkan garis lurus dari nilai sampel terakhir yang diterima ke nilai sampel sebelumnya (persamaan 1). Nilai sampel ini, baik yang asli ataupun hasil perkiraan akan disimpan dalam **database** pada server.

$$y(n) = \frac{n(v_2 - v_1)}{d} + v_1 \quad (1)$$

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

Sampel kemudian diproses untuk dilakukan **filtering**, **r-peak detection**, dan klasifikasi aritmia. Sampel yang telah difilter akan divisualisasika pada **dashboard**. Karena frekuensi sampel yang diterima sangat tinggi maka perlu dilakukan **downsample** terhadap data sampel tersebut sebelum dapat divisualisasikan pada sebuah monitor. Nilai frekuensi downsampel yang dipilih ialah 20Hz (20 FPS), nilai umum frekuensi sampel banyak monitor yang ada dipasaran.

C. Dashboard Module

Dashboard berfungsi sebagai penerima pesan peringatan dan modul pemantauan. Dengan menerapkan MQTT pada rancangan arsitektur, hubungan antara modul *sensor-cloud-dashboard* dapat digambarkan sebagai pola **subscriber** dan **publisher**. Hal ini memungkinkan banyak **dashboard** untuk **subscribe** kepada ID sensor. Sekaligus memungkinkan dikirimnya *push notification* kepada semua **dashboard** tersebut ketika sebuah ID yang dipantau mengalami serangan aritmia, yang dapat berupa ponsel android maupun halaman web.

III. PENDETEKSI ARITMIA

Telah banyak penelitian dilakukan untuk mengklasifikasikan aritmia berdasarkan sinyal ECG. Terdapat metode klasifikasi yang memanfaatkan metode kecerdasan buatan seperti *support vector machine* (SVM) [2] dan *Artificial Neural Network* (ANN) [12], ada pula yang memanfaatkan aturan yang dibuat oleh dokter ahli jantung [14]. Pada arsitektur yang dirancang diterapkan klasifikasi yang memanfaatkan aturan yang dibuat oleh ahli jantung. Aturan ini pertama kali diusulkan oleh Tsipouras et al, pada percobaan mereka menghasilkan akurasi yang cukup tinggi dibandingkan dengan algoritma yang lebih kompleks [14].

Algoritma *Arrhythmia Detector* akan berjalan pada server (lihat gambar 1, diberi warna abu). Deteksi aritmia terbagi menjadi 2 tahap yaitu *R-Peak detection* dan *Arrhythmia detection*. *R peak detection*

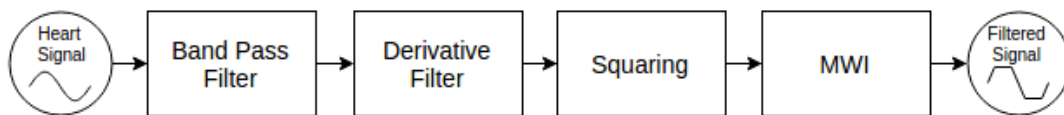
bertujuan untuk menemukan gelombang R pada sinyal ECG. Lalu *R peak* ini akan menjadi inputan pada *Arrhythmia detection*. Deteksi puncak R dilakukan dengan memanfaatkan modifikasi terhadap metode yang diajukan oleh Pan-Tompkins.

A. Deteksi Puncak R oleh Pan and Tompkins

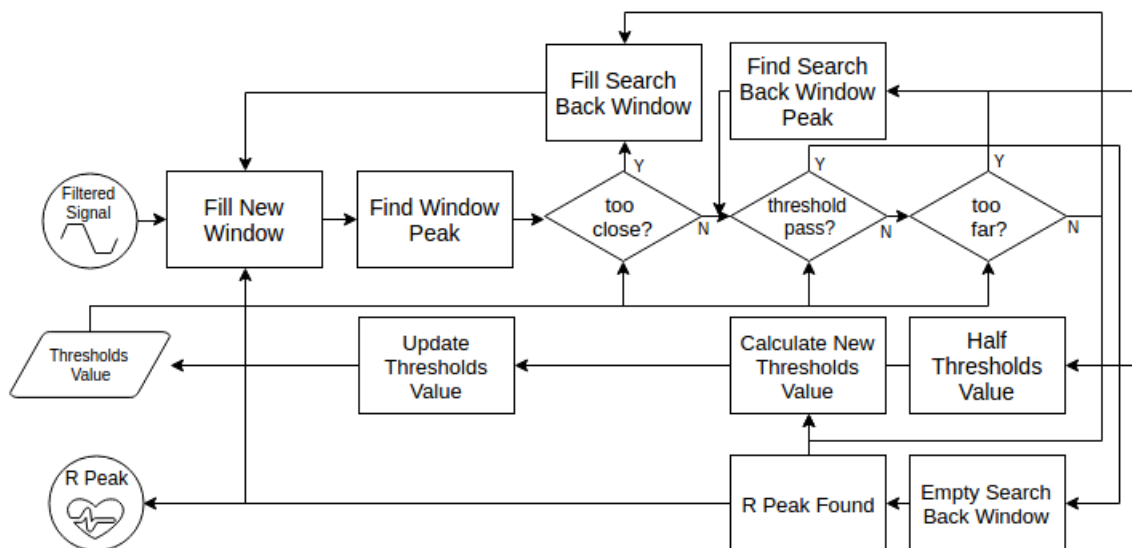
Pan dan Tompkins mengusulkan sebuah algoritma pencarian QRS pada sinyal ECG [9]. Metode ini menjadi populer karena dapat bekerja secara *real-time*. Metode PanTompkins dimulai dari proses *preprocessing* berupa filtering sinyal. Kemudian dilanjutkan dengan pencarian puncak R menggunakan *adaptive thresholding*.

Pada tahap *preprocessing*, Pan-Tompkins menerapkan 4 proses *filtering* yaitu *Band Pass filtering*, *Derived filter*, *Squaring*, dan *Moving Window Integrator (MWI)*, seperti yang ditunjukkan pada gambar 2. Tahapan tersebut dilakukan untuk menghilangkan *noise* pada sinyal dan memudahkan proses pencarian puncak.

Setelah tahap *preprocessing*, dilanjutkan tahap *processing*, Pan-Tompkins menerapkan *adaptive thresholding* terhadap sebuah *window* sinyal yang berukuran kecil (0.3s) [9]. 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 3).



Gambar 2. Preprocessing: Filtering

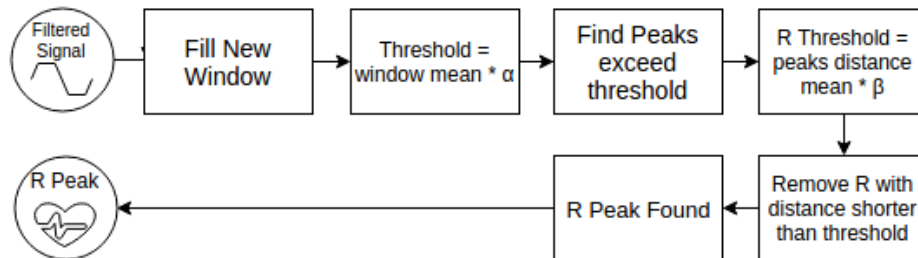


Gambar 3. Processing: Original R Peak Detection

B. Modifikasi Algoritma Pan - Tompkins dalam Deteksi Puncak R

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, menghapus *false beat*

dengan menolak R yang terlalu dekat berdasarkan rata rata jarak R (gambar 4). Pelebaran window berakibat pada meningkatnya *delay* atas munculnya hasil deteksi sejak diterimanya sampel namun dapat meningkatkan akurasi deteksi.



Gambar 4. Processing: Original R Peak Detection

Nilai α , β serta lebar window yang diterapkan pada algoritma modifikasi didapatkan melalui hasil percobaan.

C. Arrhythmia Detection

Tsipouras et al. mengusulkan algoritma deteksi setelah melakukan percobaan bersama dokter spesialis jantung [14]. 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 I.

Tabel I
ARRHYTHMIA BEAT CLASSIFICATION

MIT-BIH Beat Symbol	Symbol Meaning	Category
N	Normal	1
.	Normal Paced Beat	
/	Fusion of paced and normal beat	
L	Left bundle branch block	
R	Right bundle branch block	
Q	Unclassified	
V	Premature Ventricular Contraction	2
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	
x	Non-conducted P-wave (blocked Atrial Premature Contraction)	
!	Ventricular Flutter	3

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 detak. Tiap beat mulanya dikategorikan sebagai kelas 1 dan kemudian mendapatkan kategori baru sesuai *Rule* yang telah ditetapkan (gambar 5).

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 terjadinya VF dan window berikutnya akan dites dengan kedua kondisi berikut:

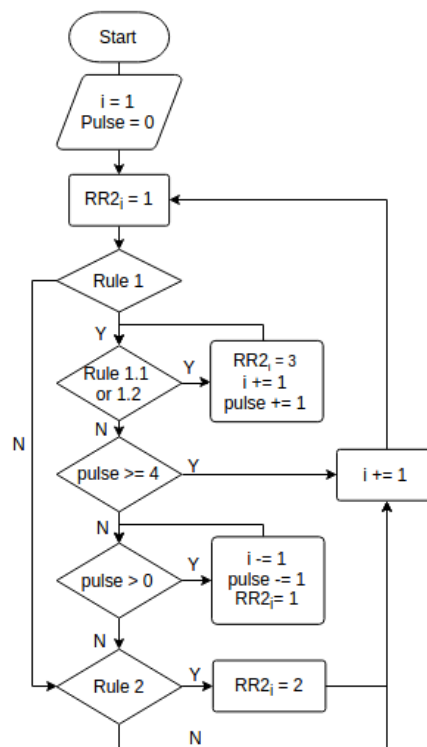
- 1) durasi setiap RR interval dalam satu window lebih kecil dari 0.7s

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 5. Flowchart of Beat Aritmia Classification

IV. EVALUASI PERFORMANSI

Untuk menguji performa rancangan arsitektur, rancangan arsitektur perlu untuk diterapkan. Modul sensor diterapkan menjadi bentuk gelang tangan yang ditanamkan sensor berjenis PPG. ESP-12E lalu dipilih sebagai *controller*-nya. ESP-12E dipilih karena bentuknya yang kecil dan spesifikasinya yang telah memiliki modul Wi-Fi. PPG dipilih hanya untuk mensimulasikan transmisi sinyal jantung, akurasi deteksi akan diukur menggunakan dataset ECG yang telah disediakan oleh MIT-BIH [1], [5]. Modul cloud diterapkan secara lokal pada komputer ASUS K432SD@2.1GHZ-RAM 6GB. Cloud akan diuji menggunakan 2 buah bahasa pemrograman yang populer dalam membangun sistem IoT, yaitu Node.js dan Python yang keduanya terhubung dengan database No-SQL MongoDB. Dashboard modul diterapkan menjadi sebuah halaman web dan menjadi sebuah aplikasi pada ponsel android.

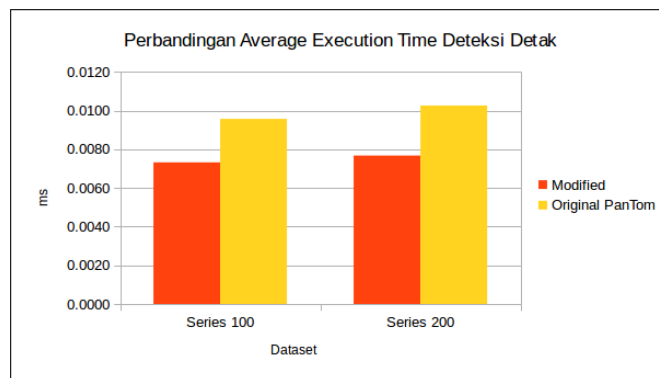
Pengujian dilakukan oleh satu orang yang mengenakan modul sensor dan bergerak secara bebas dalam daerah yang tercakup WiFi. Modul sensor, Router WiFi dan *Cloud* modul masih berada dalam satu wilayah yang sama sehingga diasumsikan tidak ada delay akibat proses *routing*. Sehingga ketika terdeteksi aritmia, dashboard modul dapat segera menerima push notifikasi dari MQTT. Namun karena keterbatasan

maksimum FPS yang dapat dirender oleh monitor, sampel 200Hz harus di downsample hingga 20Hz pada dashboard.

A. Waktu Eksekusi

Setelah diuji, waktu eksekusi ditemukan cukup kecil. Waktu eksekusi mencakup diprosesnya sampel hingga menghasilkan deteksi aritmia (tahapan *filter, save, detect*). Waktu eksekusi tercatat sebesar 0.00749 ms/sampel dengan menggunakan backend Node.JS dan 0.00991 ms/sampel dengan menggunakan backend Python. Ditemukan untuk rancangan arsitektur ini penggunaan nodejs lebih baik karena dengan algoritma dan spesifikasi server yang sama node.js dapat bekerja lebih cepat hingga 13% daripada python.

Modifikasi algoritma deteksi R juga menghasilkan performa yang diharapkan ($\alpha = 1.1; \beta = 0.8; window = 6.5s$). Modifikasi algoritma menghasilkan waktu eksekusi hingga 30% lebih cepat dari algoritma asli PanTompkin (gambar 6). Perbandingan hasil waktu eksekusi lengkap variabel dengan algoritma original dapat dilihat pada tabel II.



Gambar 6. Execution time comparison

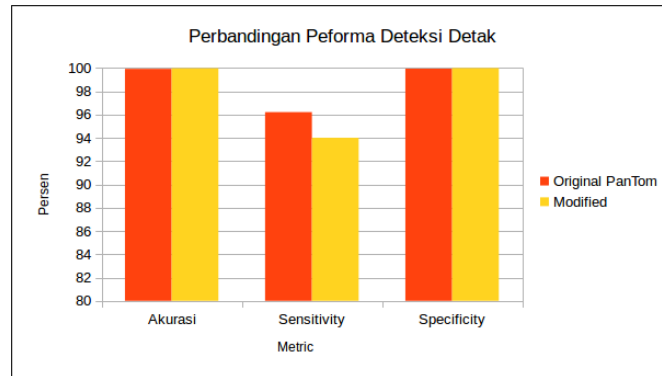
B. Deteksi

Selain menghasilkan waktu eksekusi yang lebih cepat, modifikasi terhadap algoritma pantomkin juga memiliki tingkat akurasi yang menyamai algoritma pantomkin yang asli (gambar 7). Perbandingan lengkap performa deteksi algoritma original dengan modifikasi dapat dilihat pada tabel II.

Tabel II
HASIL PENGUJIAN PERFORMA DETEKSI DETAK

Metric	Experiment					
	1	2	3	4	5	Original
Accuracy (%)	99.97589	99.97600	99.97577	99.97802	99.97188	99.92822
Sensitivity (%)	93.93309	94.00993	94.00535	95.03472	92.70835	96.21294
Specificity (%)	99.99910	99.99895	99.99872	99.99796	99.99940	99.94489
Avg Time (ms)	0.00797	0.00611	0.00749	0.00777	0.00809	0.00991

Variables	Experiment				
	1	2	3	4	5
Window Duration	8	7	6.5	6.5	6.5
Val by Mean (α)	1.1	1.1	1.1	0.65	0.65
Index by R (β)	0.8	0.8	0.8	0.87	0.93

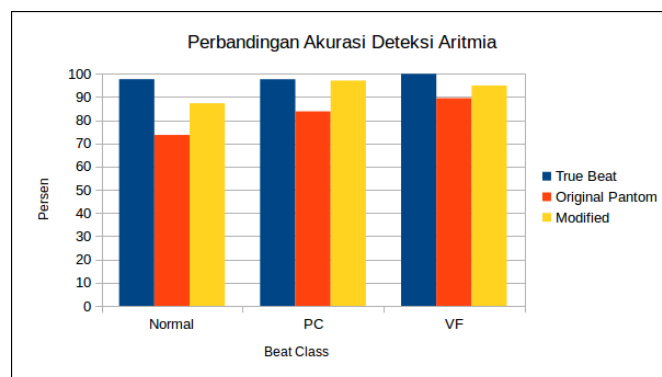


Gambar 7. Perbandingan Performa Deteksi Detak

Dengan menerapkan algoritma deteksi aritmia usulan Tsipouras, sistem berhasil mendeteksi 3 kelas detak dengan performa akurasi yang cukup baik yaitu 93.11% (gambar 8). Perbandingan lengkap performa deteksi algoritma original dengan modifikasi dapat dilihat pada tabel III. *True Beat* adalah input posisi R asli dari dataset MIT-BIH.

Tabel III
 HASIL PENGUJIAN PERFORMA DETEKSI ARITMIA

Peak Detector	Class	Metric		
		Ac	Sp	Se
True Beat	Normal	97.664986772	100	97.359734448
	PC	97.640984468	97.347191948	100
	VF	99.975997696	100	96.828752643
Original -PanTom	Normal	73.621767478	100	70.189966819
	PC	83.783329609	82.745637666	100
	VF	89.47615906	89.416847235	100
Modified -Pantom	Normal	87.298089647	100	85.401888396
	PC	97.080146855	98.843767625	87.93916024
	VF	94.947320409	94.912025085	100



Gambar 8. Perbandingan Performa Deteksi Aritmia

V. KESIMPULAN

Monitoring aritmia jantung secara realtime menggunakan IoT berhasil dilakukan pada rancangan arsitektur, terlihat dari hasil akurasi deteksi yang mencapai 93.11%. Yang menjadi perhatian utama dalam rancangan ini ialah bagaimana melakukan optimasi terhadap *resource* yang terlibat, sehingga memungkinkan semakin banyak *sensor* dan *dashboard* yang dapat terhubung. Pilihan optimasi yang dapat dilakukan terdapat pada pemilihan bahasa *server*, *database*, hingga algoritma. Dengan menerapkan optimasi pada algoritma deteksi detak rancangan arsitektur dapat menangani hingga 1.4 kali lebih banyak sensor (waktu eksekusi berkurang hingga 30%). Penerapan algoritma yang lebih akurat dan mampu mendeteksi lebih banyak kelas dapat menjadi penelitian lanjutan yang menarik.

PUSTAKA

- [1] Goldberger AL, Amaral LAN, Glass L, Hausdorff JM, Ivanov PCh, Mark RG, Mietus JE, Moody GB, Peng C-K, and Stanley HE. Physiobank, physiotookit, and physionet: Components of a new research resource for complex physiologic signals, jun 2000.
- [2] Babak Mohammadzadeh Asl, Seyed Kamaledin Setarehdan, and Maryam Mohebbi. Support vector machine-based arrhythmia classification using reduced features of heart rate variability signal. 2008.
- [3] Daniel Barata, Goncalo Louzada, Andreia Carreiro, and Antonio Damasceno. System of acquisition, transmission, storage and visualization of pulse oximeter and ecg data using android and mqtt. 2013.
- [4] Gareth Beavis. Fitbit blaze review, July 2017.
- [5] Moody GB and Mark RG. The impact of the mit-bih arrhythmia database. 2001.
- [6] Endo Indonesia. Holter system: Edan se-2003, 2017.
- [7] Vasu 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.
- [8] Mamidi Manisha, Katakam Neeraja, Vemuri Sindhura, and Paruchuri Ramya. Iot on heart attack detection and heart rate monitoring. 2016.
- [9] Jiapu Pan and Willis J. Tompkins. A real-time qrs detection algorithm. *BME-32(3)*, 1985.
- [10] James Peckham. Fitbit blaze review, February 2017.
- [11] Paola Pierleoni, Luca Pernini, Alberto Belli, and Lorenzo Palma. An android-based heart monitoring system for the elderly and for patients with heart disease. 2014.
- [12] Andrius Solosenko and Vaidotas Marozas. Automatic premature ventricular contraction detection in photoplethysmographic signals. In *IEEE Biomedical Circuits and Systems Conference*, 2014.
- [13] Android Play Store. Apps, 2017.
- [14] M.G. Tsipouras, D.I. Fotiadis, and D. Sideris. An arrhythmia classification system based on the rr-interval signal. 2005.

