TUGAS AKHIR – EB184803

PERBANDINGAN 10 METODE QRS DETECTION UNTUK DETEKSI HEART BEAT DAN HEART RATE PADA SISTEM ECG PORTABLE

Yeni Wahyu Siswanti
07311540000018

DOSEN PEMBIMBING:
Dr. Rachmad Setiawan, S.T., M.T.
Muhammad Yazid, B.Eng., M.Eng.

PROGRAM SARJANA
DEPARTEMEN TEKNIK BIOMEDIK
FAKULTAS TEKNOLOGI ELEKTRO DAN INFORMATIKA CERDAS
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2020

TUGAS AKHIR – EB184803

PERBANDINGAN 10 METODE QRS DETECTION UNTUK DETEKSI HEART BEAT DAN HEART RATE PADA SISTEM ECG PORTABLE

Yeni Wahyu Siswanti

07311540000018

DOSEN PEMBIMBING:

Dr. Rachmad Setiawan, S.T., M.T.
Muhammad Yazid, B.Eng., M.Eng.

PROGRAM SARJANA
DEPARTEMEN TEKNIK BIOMEDIK
FAKULTAS TEKNOLOGI ELEKTRO DAN INFORMATIKA CERDAS INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2020

FINAL PROJECT – EB184803

COMPARISON OF 10 QRS DETECTION METHODS FOR HEART BEAT DETECTION AND HEART RATE ON PORTABLE ECG SYSTEMS

Yeni Wahyu Siswanti

07311540000018

SUPERVISOR:

Dr. Rachmad Setiawan, S.T., M.T.
Muhammad Yazid, B.Eng., M.Eng.

UNDERGRADUATE PROGRAM
BIOMEDICAL ENGINEERING DEPARTMENT
FACULTY OF INTELLIGENT ELECTRICAL AND INFORMATICS TECHNOLOGY
SEPULUH NOPEMBER INSTITUTE OF TECHNOLOGY SURABAYA
2020

PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa sebagian maupun keseluruhan isi tugas akhir saya yang berjudul "**Perbandingan 10 Metode QRS detection untuk deteksi heart beat dan heart rate pada sistem ECG portable.**" merupakan hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya mandiri. Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka. Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, Agustus 2020

Yeni Wahyu Siswanti

NRP. 07311540000018

Tugas Akhir disusun untuk memenuhi salah satu syarat memperoleh gelar
Sarjana Teknik (S.T.)
di
Institut Teknologi Sepuluh Nopember

Oleh
Yeni Wahyu Siswanti
NRP. 07311540000018

Tanggal Ujian : **15 Juli 2020**
Periode Wisuda : **September 2020**

Disetujui Oleh :

Dosen Penguji :

1. Dr. Achmad Arifin, S.T., M.Eng.
NIP. 197103141997021001

2. Atar Fuady Babgei, S.T., M.Sc.
NIP. 198911112018121001

Dosen Pembimbing :

1. Dr. Rachmad Setiawan, S.T., M.T.
NIP. 196905291995121001

2. Muhammad Yazid, B.Eng., M.Eng.
NIP. 198004202015041001

Kepala Departemen Teknik Biomedik,

Dr. Achmad Arifin, S.T., M.Eng.
NIP. 197103141997021001

PERBANDINGAN 10 METODE QRS DETECTION UNTUK DETEKSI HEART BEAT DAN HEART RATE PADA SISTEM ECG PORTABLE

Nama : Yeni Wahyu Siswanti
NRP : 07311540000018
Pembimbing : 1. Dr. Rachmad Setiawan, S.T., M.T.
2. Muhammad Yazid, B.Eng., M.Eng.

ABSTRAK

Salah satu penyakit kardiovaskular yang berbahaya yaitu aritmia. Aritmia mengacu pada setiap bentuk denyut atau irama jantung yang bersifat tidak normal. Hal yang berbahaya dari aritmia adalah sifatnya yang *asymptomatic* sehingga menyulitkan dokter dalam mendeteksinya. Oleh karena itu penggunaan ECG *portable* dapat membantu dokter karena dapat digunakan dalam jangka waktu yang lebih lama. Selain itu, penggunaan komputasi yang ringan akan turut membantu dalam daya tahan baterai dan memori sehingga dapat digunakan dalam jangka waktu yang lebih lama. Terdapat beberapa metode yang dapat digunakan dalam mendeteksi kompleks QRS, namun pada penelitian ini metode yang digunakan yaitu algoritma berbasis *Amplitude and First Derivative*, *First Derivative*, *First and Second Derivative,* dan *Digital Filter*. Terdapat beberapa parameter statistik kuantitatif untuk melihat perbedaan dari setiap metode deteksi QRS yang digunakan. Beberapa parameternya yaitu *Positive Predictivity* (+P), *Sensitivity* (Se), dan F1 Measure. Dari hasil pengujian didapatkan bahwa metode yang digunakan pada penelitian untuk mendeteksi *heart rate* dan *heart beat* yang paling bagus adalah algortima Pan and Tompkins. Hal ini disimpulkan dari nilai rata-rata error yang didapatkan oleh algortima Pan and Tompkins paling sedikit yakni 14% dan memiliki nilai F1 paling besar yaitu sebesar 98%. Kemudian memori yang digunakan pada sistem ini juga kecil yaitu sebesar 220 KB dari total 1 MB *flash memory* mikrokontroler serta waktu komputasinya yang singkat pada tiap metode.

Kata kunci : *Aritmia, ECG Portable, Embedded System*, Mikrokontroler, *QRS Detection Algorithm.*

COMPARISON OF 10 QRS DETECTION METHODS FOR HEART BEAT DETECTION AND HEART RATE ON PORTABLE ECG SYSTEMS

Name : Yeni Wahyu Siswanti
Student Identity Number : 07311540000018
Advisors : 1. Dr. Rachmad Setiawan, S.T., M.T.
2. Muhammad Yazid, B.Eng., M.Eng.

ABSTRACT

One of the dangerous cardiovascular disease is arrhythmia. Arrhythmia refers to any form of heartbeat or rhythm that is abnormal. The dangerous thing about arrhythmia is that it is asymptomatic, making it difficult for doctors to detect it. Therefore the use of a portable ECG can help doctors because it can be used for a longer period of time. In addition, the use of lightweight computing will also help in battery life and memory so that it can be used in a longer period of time. There are several methods that can be used in detecting QRS complexes, but in this study the method used is an algorithm based on Amplitude and First Derivative, First Derivative, First and Second Derivative, and Digital Filter. There are several quantitative statistical parameters to see the difference of each QRS detection method used. Some parameters are Positive Predictivity (+ P), Sensitivity (Se), and F1 Measure. From the test results it was found that the method used in research to detect the best heart rate and heart beat is the Pan and Tompkins algorithm. This is concluded from the average value of errors obtained by the Pan and Tompkins algorithm at least 14% and has the largest F1 value of 98%. Then the memory used in this system is also small, which is 220 KB from a total of 1 MB of microcontroller flash memory and the short computation time for each method.

Keywords : Arryhthmia, ECG Portable, Embedded System, Microcontroller, QRS Detection Algorithm

KATA PENGANTAR

Penulis mengucapkan puji syukur kepada Tuhan Yang Maha Esa karena hanya atas kasih sayang-Nya penulis mampu menyelesaikan Tugas Akhir ini dengan lancar dan dapat selesai tepat pada waktunya. Penulis juga mengucapkan terima kasih atas bantuan dari berbagai pihak, maka penulis mengucapkan terima kasih kepada:

1. Orang tua, keluarga, dan kucing kesayangan penulis yang selalu memberi doa dan dukungan penuh dalam hal apapun.
2. Dr. Rachmad Setiawan, S.T., M.T. dan M. Yazid, B.Eng., M.Eng. selaku dosen pembimbing yang senantiasa membimbing, mengarahkan, dan menginspirasi penulis selama pengerjaan penelitian ini.
3. Dr. Achmad Arifin, S.T., M.Eng. selaku kepala departemen yang telah memberi inspirasi, motivasi, serta bimbingan selama pengerjaan Tugas Akhir terlebih selama perkuliahan di ITS.
4. Nada Fitrieyatul H., S.T., M.T. dan Norma Hermawan, S.T., M.Sc. selaku dosen wali
5. Rekan-rekan Teknik Biomedik angkatan 2015 dan anggota laboratorium B205 yang saya sayangi.
6. Keluarga besar Kalpataru, terutama K-32 yang saya kasihi.
7. Bapak Ibu dosen pengajar Teknik Biomedik ITS.
8. Keluarga Bapak Rachmad Setiawan yang selalu memotivasi saya untuk segera menyelesaikan Tugas Akhir ini.
9. Serta seluruh pihak yang tidak dapat disebutkan satu per satu yang telah membantu penulis hingga Tugas Akhir ini bisa diselesaikan dengan semaksimal mungkin.

Terlepas dari semua itu, penulis sadar bahwa masih terdapat kekurangan pada berbagai aspek dan masih banyak hal yang perlu diperbaiki. Oleh karena itu, penulis dengan terbuka menerima kritik dan saran dari pembaca agar dapat memperbaiki Tugas Akhir ini. Semoga penelitian ini dapat memberikan manfaat untuk banyak orang dan dapat dikembangkan lebih baik lagi pada penelitian selanjutnya.

Surabaya, Agustus 2020

Penulis

DAFTAR ISI

PERNYATAAN KEASLIAN TUGAS AKHIR i
ABSTRAK iii
ABSTRACT iv
KATA PENGANTAR v
DAFTAR GAMBAR ix
DAFTAR TABEL xii
BAB 1 PENDAHULUAN 1
1.1 Latar Belakang 1
1.2 Perumusan Masalah 2
1.3 Batasan Masalah 2
1.4 Tujuan dan Manfaat 2
1.5 Kontribusi 2
1.6 Sistematika 3
BAB 2 TINJAUAN PUSTAKA 4
2.1 Jantung 4
2.1.1 Anatomi dan Fisiologi Jantung 4
2.1.2 Kelistrikan Jantung 5
2.2 Aritmia 6
2.3 Electrocardiography 7
2.3.1 Sadapan ECG 7
2.3.2 Gelombang, Interval, dan Segmen 8
2.4 Embedded System 9
2.5 Hardware 11
2.5.1 Modul Mikromedia 5 for STM32 Capacitive 11
2.5.2 ADC 11
2.5.3 *ECG Click* 13
2.5.4 Phantom ECG 19
2.6 Software 19
2.6.1 Timer Calculator 20

2.6.2 MikroC Pro for ARM 20
2.6.3 CODEGRIP Suite 21
2.7 Metode Deteksi QRS 22
2.7.1 Amplitude & First Derivative Algorithm 22
2.7.2 First Derivative Algorithm 24
2.7.3 First and Second Derivative Algorithm 25
2.7.4 Digital Filter Algorithm 26
2.8 Parameter yang dianalisa 30
2.8.1 Sensitivity (Se) 30
2.8.2 Positive Predictivity (+P) 30
2.8.3 F1 Measure (F1) 30
2.9 Rasional 31
BAB 3 PERANCANGAN SISTEM 34
3.1 Desain Sistem Secara Umum 34
3.2 Perancangan *Hardware* Sistem 34
3.3 Perancangan Software Sistem 35
3.3.1 Amplitude & First Derivative Algorithm 36
3.3.2 First Derivative Algorithm 39
3.3.3 First & Second Derivative Algorithm 40
3.3.4 Digital Filter Algorithm 42
3.4 Desain Visualisasi *Output* 46
3.5 Pengambilan Data 46
BAB 4 PENGUJIAN SISTEM 48
4.1 Pengujian dengan Menggunakan Simulasi 48
4.2 Pengujian dengan Menggunakan Phantom ECG 55
4.2.1 Pengujian *heart rate* sebesar 40 bpm 59
4.2.2 Pengujian *heart rate* sebesar 60 bpm 61
4.2.3 Pengujian *heart rate* sebesar 80 bpm 63
4.2.4 Pengujian *heart rate* sebesar 100 bpm 65
4.2.5 Pengujian *heart rate* sebesar 120 bpm 67
4.3 Pengujian dengan Menggunakan Phantom ECG dengan noise 69

4.3.1 Pengujian dengan *inference 50 Hz noise* 69
4.3.2 Pengujian dengan *muscle artifact noise* 71
4.3.3 Pengujian dengan *baseline artifact noise* 73
4.3.4 Pengujian dengan respiration artifact noise 75
4.4 Pengujian dengan Menggunakan Sinyal Uji Tubuh 77
4.5 *Computation cost* 79
BAB 5 PEMBAHASAN 80
5.1 Analisa tiap algoritma 80
5.1.1 Moriet-Mahoudeux 80
5.1.2 Fraden and Neuman 81
5.1.3 Gustafson 82
5.1.4 Menard 83
5.1.5 Holsinger 84
5.1.6 Balda 85
5.1.7 Ahlstrom and Tompkins 86
5.1.8 Engelese and Zeelenberg 87
5.1.9 Okada 88
5.1.10 Pan and Tompkins 89
5.2 Analisa tiap parameter 90
5.2.1 Pengujian *heart rate* 90
5.2.2 Pengujian *heart beat* 91
5.3 Analisa dengan Sinyal Uji Tubuh 92
5.4 Analisa Performansi 93
5.5 Klasifikasi Aritmia 93
BAB 6 KESIMPULAN DAN SARAN 94
6.1 Kesimpulan 94
6.2 Saran 95
DAFTAR PUSTAKA 97
LAMPIRAN 100
BIODATA PENULIS 108

DAFTAR GAMBAR

Gambar 2.1 Anatomi jantung [3] 5
Gambar 2.2 Sistem konduksi jantung [5] 6
Gambar 2.3 Bipolar dan augmented limb lead [6] 7
Gambar 2.4 Precordial chest lead [6] 8
Gambar 2.5 ECG *grid*. Gelombang, interval, dan segmen pada ECG [9] 9
Gambar 2.6 Mikromedia 5 for STM32 Capacitive [11] 11
Gambar 2.7 ECG Click [13] 13
Gambar 2.8 *Schematic Diagram* ECG Click [13] 13
Gambar 2.9 Rangkaian proteksi [13] 14
Gambar 2.10 Rangkaian Pre-Amplifier [13] 15
Gambar 2.11 Rangakaian HPF #1 [13] 15
Gambar 2.12 Rangkaian Amplifier [13] 16
Gambar 2.13 Rangkaian HPF #2 [13] 16
Gambar 2.14 Rangkaian LPF [13] 17
Gambar 2.15 Rangkaian DRL [13] 17
Gambar 2.16 Rangkaian *voltage reference* [13] 18
Gambar 2.17 MS400 Multiparameter Simulator [15] 19
Gambar 2.18 Timer Calculator 20
Gambar 2.19 *Workspace* pada MikroC Pro for ARM 21
Gambar 2.20 *Block Diagram* dari Pan-Tompkins Algorithm 28
Gambar 2.21 Diagram *Fishbone* 33
Gambar 3.1 *Block Diagram* Sistem Keseluruhan 34
Gambar 3.2 *Block Diagram* Hardware Sistem 35
Gambar 3.3 *Flow chart* algoritma “Moriet-Mahoudeux” 36
Gambar 3.4 *Flow chart* algoritma “Fraden and Neuman” 37
Gambar 3.5 *Flow chart* algoritma “Gustafson” 38
Gambar 3.6 *Flow chart* algoritma “Menard” 39
Gambar 3.7 *Flow chart* algoritma “Holsinger” 40
Gambar 3.8 *Flow chart* algoritma “Balda” 41
Gambar 3.9 *Flow chart* algoritma “Ahlstrom and Tompkins” 42
Gambar 3.10 *Flow chart* algoritma “Engelese and Zeelenberg” 43
Gambar 3.11 *Flow chart* algoritma “Okada” 44
Gambar 3.12 *Flow chart* algoritma “Pan and Tompkins” 45
Gambar 3.13 (a) Desain visualisasi output pada layar utama (b) Desain visualisasi output pada menu *simulation* dan *measurement*. 46
Gambar 4.1 Interface Mikrokontroler 48

Gambar 4.2 Interface Mikrokontroler setelah ditekan menu “Simulation” 49
Gambar 4.3 Interface Mikrokontroler saat simulasi metode “Moriet-Mahoudeux” yang dipilih..... 49
Gambar 4.4 Interface Mikrokontroler saat simulasi metode “Fraden and Neuman” yang dipilih..... 49
Gambar 4.5 Interface Mikrokontroler saat simulasi metode “Gustafson” yang dipilih. 50
Gambar 4.6 Interface Mikrokontroler saat simulasi metode “Menard” yang dipilih. 50
Gambar 4.7 Interface Mikrokontroler saat simulasi metode “Holsinger” yang dipilih. 50
Gambar 4.8 Interface Mikrokontroler saat simulasi metode “Balda” yang dipilih.. 51
Gambar 4.9 Interface Mikrokontroler saat simulasi metode “Ahlstrom-Tompkins” yang dipilih..... 51
Gambar 4.10 Interface Mikrokontroler saat simulasi metode “Engelese-Zeelenberg” yang dipilih. 51
Gambar 4.11 Interface Mikrokontroler saat simulasi metode “Okada” yang dipilih. 52
Gambar 4.12 Interface Mikrokontroler saat simulasi metode “Pan-Tompkins” yang dipilih. 52
Gambar 4.13 Interface phantom ECG saat simulasi menggunakan *heart rate* sebesar 80 bpm. 55
Gambar 4.14 Sinyal output yang disimulasikan oleh phantom ECG dengan *heart rate* sebesar 80 bpm. 55
Gambar 4.15 Interface Mikrokontroler saat melakukan pengukuran pada metode “Moriet-Mahoudeux” yang dipilih dengan input 80 bpm. 56
Gambar 4.16 Interface Mikrokontroler saat melakukan pengukuran pada metode “Fraden and Neuman” yang dipilih dengan input 80 bpm. 56
Gambar 4.17 Interface Mikrokontroler saat melakukan pengukuran pada metode “Gustafson” yang dipilih dengan input 80 bpm..... 56
Gambar 4.18 Interface Mikrokontroler saat melakukan pengukuran pada metode “Menard” yang dipilih dengan input 80 bpm. 57
Gambar 4.19 Interface Mikrokontroler saat melakukan pengukuran pada metode “Holsinger” yang dipilih dengan input 80 bpm. 57
Gambar 4.20 Interface Mikrokontroler saat melakukan pengukuran pada metode “Balda” yang dipilih dengan input 80 bpm. 57
Gambar 4.21 Interface Mikrokontroler saat melakukan pengukuran pada metode “Ahlstrom and Tompkins” yang dipilih dengan input 80 bpm..... 58

Gambar 4.22 Interface Mikrokontroler saat melakukan pengukuran pada metode "Engelese and Zeelenberg" yang dipilih dengan input 80 bpm..... 58
Gambar 4.23 Interface Mikrokontroler saat melakukan pengukuran pada metode "Okada" yang dipilih dengan input 80 bpm. 58
Gambar 4.24 Interface Mikrokontroler saat melakukan pengukuran pada metode "Pan and Tompkins" yang dipilih dengan input 80 bpm..... 59

DAFTAR TABEL

Tabel 4.1 Hasil pengujian *heart rate* pada menu "*simulation*" 53
Tabel 4.2 Hasil pengujian *heart beat* pada menu "*Measurement"* 54
Tabel 4.3 Hasil pengujian *heart rate* pada saat pengaturan *heart rate* 40 bpm 59
Tabel 4.4 Hasil pengujian *heart beat* pada pengaturan *heart rate* sebesar 40 bpm. 60
Tabel 4.5 Hasil pengujian *heart rate* pada saat pengaturan *heart rate* 60 bpm 61
Tabel 4.6 Hasil pengujian *heart beat* pada pengaturan *heart rate* sebesar 60 bpm. 62
Tabel 4.7 Hasil pengujian *heart rate* pada saat pengaturan *heart rate* 80 bpm 63
Tabel 4.8 Hasil pengujian *heart beat* pada pengaturan *heart rate* sebesar 80 bpm. 64
Tabel 4.9 Hasil pengujian *heart rate* pada saat pengaturan *heart rate* 100 bpm 65
Tabel 4.10 Hasil pengujian *heart beat* pada pengaturan *heart rate* sebesar 100 bpm. 66
Tabel 4.11 Hasil pengujian *heart rate* pada saat pengaturan *heart rate* 120 bpm 67
Tabel 4.12 Hasil pengujian *heart beat* pada pengaturan *heart rate* sebesar 120 bpm. 68
Tabel 4.13 Hasil pengujian *heart rate* dengan *inference 50 Hz noise* 69
Tabel 4.14 Hasil pengujian *heart beat* dengan *inference 50 Hz noise* 70
Tabel 4.15 Hasil pengujian *heart rate* dengan *muscle artifact noise* 71
Tabel 4.16 Hasil pengujian *heart beat* dengan *muscle artifact noise* 72
Tabel 4.17 Hasil pengujian *heart rate* dengan *baseline artifact noise* 73
Tabel 4.18 Hasil pengujian *heart beat* dengan *baseline artifact noise* 74
Tabel 4.19 Hasil pengujian heart rate dengan respiration artifact noise 75
Tabel 4.20 Hasil pengujian heart beat dengan *respiration artifact noise* 76
Tabel 4.21 Hasil pengujian *heart rate* dengan sinyal uji tubuh 77
Tabel 4.22 Hasil pengujian *heart beat* dengan sinyal uji tubuh 78
Tabel 4.23 Tabel penggunaan waktu komputasi tiap algoritma 79
Tabel 5.1 Hasil pengujian *heart rate* pada pengujian algoritma "Moriet-Mahoudeux" 80
Tabel 5.2 Hasil pengujian *heart beat* pada pengujian algoritma "Moriet-Mahoudeux" 81
Tabel 5.3 Hasil pengujian *heart rate* pada pengujian algoritma "Fraden and Neuman" 81
Tabel 5.4 Hasil pengujian *heart beat* pada pengujian algoritma "Fraden and Neuman" 82
Tabel 5.5 Hasil pengujian *heart rate* pada pengujian algoritma "Gustafson" 82
Tabel 5.6 Hasil pengujian *heart beat* pada pengujian algoritma "Gustafson" 83
Tabel 5.7 Hasil pengujian *heart rate* pada pengujian algoritma "Menard" 83

Tabel 5.8 Hasil pengujian *heart beat* pada pengujian algoritma “Menard” 84
Tabel 5.9 Hasil pengujian *heart rate* pada pengujian algoritma “Holsinger”..... 84
Tabel 5.10 Hasil pengujian *heart beat* pada pengujian algoritma “Holsinger” 85
Tabel 5.11 Hasil pengujian *heart rate* pada pengujian algoritma “Balda” 85
Tabel 5.12 Hasil pengujian *heart beat* pada pengujian algoritma “Balda”..... 86
Tabel 5.13 Hasil pengujian *heart rate* pada pengujian algoritma “Ahlstrom and Tompkins” 86
Tabel 5.14 Hasil pengujian *heart beat* pada pengujian algoritma “Ahlstrom and Tompkins” 87
Tabel 5.15 Hasil pengujian *heart rate* pada pengujian algoritma “Engelese and Zeelenberg”..... 87
Tabel 5.16 Hasil pengujian *heart beat* pada pengujian algoritma “Engelese and Zeelenberg”..... 88
Tabel 5.17 Hasil pengujian *heart rate* pada pengujian algoritma “Okada”..... 88
Tabel 5.18 Hasil pengujian *heart beat* pada pengujian algoritma “Okada” 89
Tabel 5.19 Hasil pengujian *heart rate* pada pengujian algoritma “Pan and Tompkins” 89
Tabel 5.20 Hasil pengujian *heart beat* pada pengujian algoritma “Pan and Tompkins” 90
Tabel 5.21 Rata-rata error tiap algoritma setelah 10 kali pengujian *heart rate*. 90
Tabel 5.22 Hasil sensitivity (Se), *positive predictivity* (+P), dan *F1 measure* (F1) tiap algoritma dalam 10 kali pengujian *heart beat.* 91

BAB 1 PENDAHULUAN

1.1 Latar Belakang

Kesehatan merupakan sesuatu yang harus dijaga melalui pola hidup sehat, terutama kesehatan jantung. Laporan Hasil Riset Kesehatan Dasar Badan Penelitian dan Pengembangan Kesehatan Departemen Kesehatan Republik Indonesia tahun 2008 menyatakan bahwa sebanyak 16 provinsi di Indonesia memiliki nilai prevalensi penyakit jantung diatas prevalensi nasional yaitu 7,2%. Data Dinkes Jateng menyebutkan kasus tertinggi penyakit tidak menular pada tahun 2011 adalah penyakit jantung dan pembuluh darah. Dari total 1.409.857 kasus yang dilaporkan, sebesar 62,43% (880.193 kasus) adalah penyakit jantung dan pembuluh darah. Selain itu menurut salah satu website kesehatan dunia, menyatakan bahwa penyakit jantung merupakan 30% penyakit global pada tahun 2005. Sedangkan menurut Data World Health Organization (WHO) tahun 2012 menunjukkan 17,5 juta orang di dunia meninggal akibat penyakit kardiovaskuler atau 31% dari 56,5 juta kematian di seluruh dunia. Lebih dari 3/4 kematian akibat penyakit kardiovaskuler terjadi di negara berkembang yang berpenghasilan rendah sampai sedang [1]. Aritmia merupakan penyakit jantung yang umum terjadi. Secara umum, Aritmia mengacu pada setiap bentuk denyut atau irama jantung yang bersifat tidak normal yang merujuk pada setiap gangguan frekuensi, regularitas, lokasi asal atau konduksi impuls listrik jantung yang mengakibatkan aktivitas listrik pada jantung tidak stabil [2]. Ada berbagai jenis aritmia, di mana fibrilasi atrium merupakan aritmia yang paling umum. Meskipun tidak berbahaya secara langsung, aritmia meningkatkan peluang terjadinya stroke hingga lima kali lipat, meningkatkan peluang terjadinya gagal jantung hingga tiga kali lipat, dan meningkatkan peluang kematian hingga dua kali lipat. Jantung normal berdenyut sebanyak 60-100 kali per menit saat seseorang beristirahat. Dan denyut jantung tersebut memiliki irama yang teratur. Bila seseorang memiliki denyut jantung yang lambat, yaitu kurang dari 60 denyut per menit; atau denyut jantung yang cepat, yaitu lebih dari 100 denyut per menit saat beristirahat, Ia mungkin menderita aritmia. Kadang-kadang, seseorang bisa memiliki denyut jantung yang normal namun dengan irama yang tidak teratur. Mungkin saja Ia menderita penyakit denyut jantung ektopik atau fibrilasi atrium.

Electrocardiogram (ECG) merupakan alat diagnostik yang penting. Pemantauan denyut jantung secara berkelanjutan dengan ECG selama 24 jam mungkin diperlukan untuk mendeteksi aritmia karena sifatnya yang *asymptomatic*. Penyakit yang memiliki sifat *asymptomatic* memiliki risiko tidak terdeteksi karena gejalanya yang bisa tiba-tiba muncul dan tiba-tiba hilang. Sehingga penggunaan

alat diagnostik berupa ECG *portable* dapat membantu untuk mendeteksi kelainan pada jantung, terutama aritmia.

1.2 Perumusan Masalah

Berdasarkan latar belakang yang telah dijelaskan sebelumnya, maka rumusan masalahnya adalah bagaimana merancang sistem ECG *portable* yang didalamnya tertanam sistem pemrosesan sinyal digital sehingga dapat membantu dalam deteksi aritmia. Kemudian masalah yang kedua adalah bagaimana menerapkan metode-metode deteksi QRS pada mikrokontroler. Terakhir, rumusan masalahnya yaitu keakurasian tiap metode yang digunakan.

1.3 Batasan Masalah

Terdapat tiga masalah dari penelitian Tugas Akhir ini, dimana batasan masalah yang pertama adalah memfokuskan pemrosesan sinyal digital yang tertanam pada mikrokontroler yang digunakan. Batasan masalah yang kedua yaitu pendeteksian penyakit jantung yang dilakukan hanya sebatas mendeteksi apakah sinyal jantung yang dideteksi tersebut termasuk ke dalam kategori penyakit jantung Aritmia (Bradikardia ataupun Takikardia) dari parameter *heart rate* yang dihasilkan. Batasan masalah yang terakhir adalah data yang digunakan pada penelitian ini dilakukan secara *real time* sehingga dapat terjadi kesalahan akurasi dalam mendeteksi metode QRS yang digunakan seperti *False Positive* (FP) dan *False Negative* (FN).

1.4 Tujuan dan Manfaat

Tujuan dari penelitian ini adalah mengembangkan ECG portable yang di dalamnya tertanam pemrosesan sinyal digital. Dengan begitu maka setelah menggunakan metode yang diajukan pada penelitian ini harapannya dapat dilihat metode-metode mana yang performanya bagus diimplementasikan pada mikrokontroler sehingga dapat dikembangkan sistem ECG portable yang lebih baik.

Manfaat yang ingin dicapai dari penelitian ini meliputi aspek teoritis dan aspek praktis. Manfaat dalam aspek teoritis yang diharapkan adalah dapat menambah wawasan yang berkaitan tentang ECG *portable* berbasis mikrokontroler tersebut dan membuat terobosan baru kedepannya sehingga membuat pendeteksian menjadi lebih cepat dan akurat.

Manfaat dalam aspek praktis yang diharapkan adalah memudahkan dokter dapat menemukan indikasi-indikasi abnormalitas pada jantung, terutama penyakit yang bersifat *asymptomatic* sehingga dapat membantu dalam proses monitoring dan deteksi penyakit agar nantinya dapat dilakukan penanganan lebih lanjut dari penyakit yang telah didiagnosis tersebut.

1.5 Kontribusi

Kontribusi dari penelitian ini meliputi kontribusi ilmiah dan kontribusi praktis. Kontribusi ilmiah yang diharapkan dalam penelitian ini adalah dapat memperkaya

khasanah pengetahuan mengenai pengembangan instrumentasi ECG *portable*, terutama mengenai perkembangan metode deteksi yang digunakan pada instrumentasi ECG *portable*.

Kontribusi praktis yang diharapkan dalam penelitian ini meliputi sektor pemerintah, sektor swasta, dan sektor publik (masyarakat). Kontribusi praktis yang bagi pemerintah adalah dapat mendukung program pemerintah untuk meningkatkan kualitas hidup masyarakat, khususnya pada bidang kesehatan. Kontribusi praktis bagi pihak swasta adalah dapat memberikan peluang pengembangan teknologi dalam instrumentasi alat-alat kesehatan, khususnya teknologi yang berkaitan dengan monitoring pada pasien penyakit kardiovaskular. Kontribusi praktis bagi masyarakat adalah dapat membantu meningkatkan harapan hidup orang yang mengalami penyakit kardiovaskular, terutama aritmia.

1.6 Sistematika

Sistematika penulisan pengerjaan dari laporan Tugas Akhir (TA) oleh mahasiswa mengenai penelitian sebagai langkah awal untuk menentukan tugas akhir yang akan diambil adalah sebagai berikut :

- **BAB 1 PENDAHULUAN**
  Pada bab 1 dijelaskan mengenai latar belakang pengambilan judul penelitian, perumusan masalah mengenai penelitian ini, Batasan-batasan masalah yang terdapat pada penelitian ini, tujuan dan manfaat dari diberlakukannya penelitian ini serta kontribusi apa saja yang akan diberikan pada penelitian tersebut.
- **BAB 2 TINJAUAN PUSTAKA**
  Pada bab 2 dijelaskan mengenai metode-metode apa saja yang digunakan pada penelitian ini, beserta penjelasan mengenai aritmia, *electrocardiograph*, beserta hal-hal apa saja yang akan dijadikan parameter dalam analisis setiap metode.
- **BAB 3 PERANCANGAN SISTEM**
  Pada bab 3 dijelaskan mengenai desain pengerjaan metode penelitian ini beserta perancangan sistem yang diajukan pada penelitian ini.
- **BAB 4 PENGUJIAN SISTEM**
  Pada bab 4 dijelaskan mengenai cara dan pelaksanaan pengujian system yang diajukan beserta menampilkan hasil dari sistem tersebut.
- **BAB 5 HASIL DAN PEMBAHASAN**
  Pada bab 5 dijelaskan mengenai hasil Analisa dari hasil yang sudah ditampilkan pada bab sebelumnya, yaitu bab 4.
- **BAB 6 KESIMPULAN DAN SARAN**
  Pada bab 6 dijelaskan mengenai kesimpulan dari penelitian ini beserta saran yang diajukan sehingga dapat digunakan untuk penelitian selanjutnya.

BAB 2 TINJAUAN PUSTAKA

2.1 Jantung

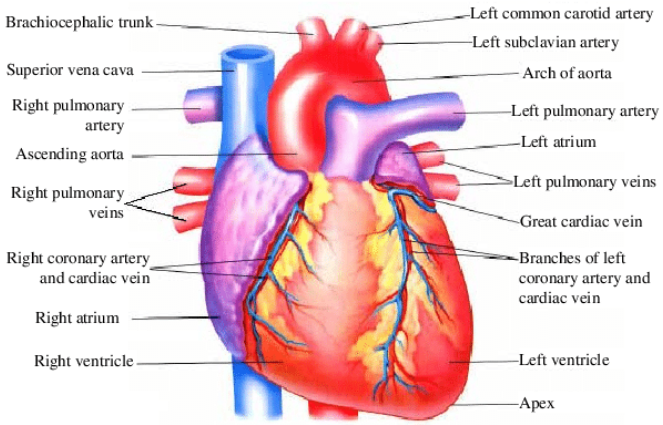
2.1.1 Anatomi dan Fisiologi Jantung

Jantung adalah organ otot yang berongga dan berukuran sebesar kepalan tangan. Fungsi utama jantung adalah memompa darah ke pembuluh darah dengan kontraksi ritmik dan berulang. Jantung normal terdiri dari empat ruang, 2 ruang jantung atas dinamakan atrium dan 2 ruang jantung di bawahnya dinamakan ventrikel, yang berfungsi sebagai pompa. Dinding yang memisahkan kedua atrium dan ventrikel menjadi bagian kanan dan kiri dinamakan septum.

Darah dipompakan melalui semua ruang jantung dengan bantuan keempat katup yang mencegah agar darah tidak kembali ke belakang dan menjaga agar darah tersebut mengalir ke tempat yang dituju [3]. Keempat katup ini adalah katup trikuspid yang terletak di antara atrium kanan dan ventrikel kanan, katup pulmonal, terletak di antara ventrikel kanan dan arteri pulmonal, katup mitral yang terletak di antara atrium kiri dan ventrikel kiri dan katup aorta, terletak di antara ventrikel kiri dan aorta. Katup mitral memiliki 2 daun (*leaflet*), yaitu *leaflet* anterior dan posterior.

Jantung dapat dianggap sebagai 2 bagian pompa yang terpisah terkait fungsinya sebagai pompa darah. Masing-masing terdiri dari satu atrium-ventrikel kiri dan kanan. Berdasarkan sirkulasi dari kedua bagian pompa jantung tersebut, pompa kanan berfungsi untuk sirkulasi paru sedangkan bagian pompa jantung yang kiri berperan dalam sirkulasi sistemik untuk seluruh tubuh. Kedua jenis sirkulasi yang dilakukan oleh jantung ini adalah suatu proses yang berkesinambungan dan berkaitan sangat erat untuk asupan oksigen manusia demi kelangsungan hidupnya.

Ada 5 pembuluh darah mayor yang mengalirkan darah dari dan ke jantung. Vena cava inferior dan vena cava superior mengumpulkan darah dari sirkulasi vena (disebut darah biru) dan mengalirkan darah biru tersebut ke jantung sebelah kanan. Darah masuk ke atrium kanan, dan melalui katup trikuspid menuju ventrikel kanan, kemudian ke paru-paru melalui katup pulmonal [4].



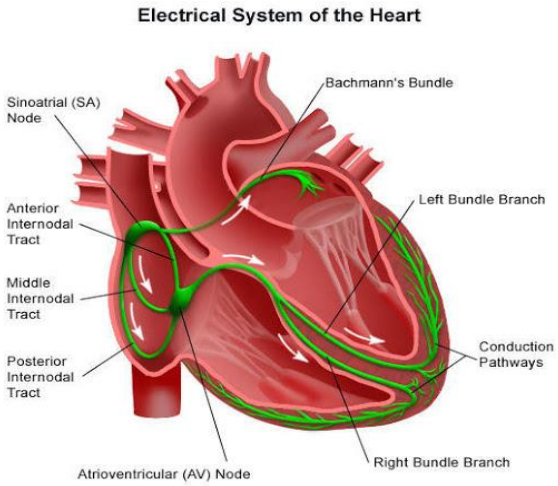
Gambar 2.1 Anatomi jantung **[3]**

2.1.2 Kelistrikan Jantung

Kontraksi sel otot jantung dalam siklus di picu oleh aksi potensial yang menyebar ke seluruh membran sel otot. Terdapat dua jenis sel otot jantung yaitu sel kontraktil dan sel otoritmik. Sel kontraktil yang membentuk 99% dari sel-sel otot jantung, melakukan kerja mekanis memompa darah. Dalam keadaan normal, sel ini tidak membentuk sendiri potensial aksinya. Sedangkan sel otoritmik, yang tidak berkontraksi tapi khusus memulai dan menghantarkan potensial aksi yang menyebabkan kontraksi sel-sel jantung kontraktil.

Sel otoritmik jantung merupakan sel otot khusus yang berbeda dari sel saraf dan sel otot rangka di mana sel otoritmik jantung tidak memiliki potensial istirahat. Sel ini memperlihatkan aktivitas pemicu yaitu potensial membran secara perlahan terdepolarisasi sampai ke ambang (potensial pemicu). Dengan siklus yang berulang tersebut, sel otoritmik memicu potensial aksi yang kemudian menyebar ke seluruh jantung untuk memicu denyut berirama tanpa rangsangan saraf apapun. Sel-sel jantung otoritmik ini membentuk area tersendiri pada [5]:

a) *Nodus Sinoatrial* (nodus SA), suatu daerah kecil khusus di dinding atrium kanan dekat pintu masuk vena cava superior.
b) *Nodus Atrioventrikuler* (nodus AV), suatu berkas kecil sel-sel otot jantung khusus yang terdapat pada dasar atrium kanan dekat septum, tepat diatas pertemuan atrium dan ventrikel.
c) Berkas His (berkas atrioventrikuler), suatu jaras sel-sel khusus yang berasal dari nodus AV dan masuk ke septum antar ventrikel. Disini berkas tersebut terbagi menjadi cabang berkas kanan dan kiri yang turun menyusuri septum, melengkung mengelilingi ujung rongga ventrikel dan berjalan balik kearah atrium di sepanjang dinding luar.
d) Serat Purkinje, serat-serat halus terminal yang menjulur ke seluruh miokardium ventrikel seperti ranting kecil dari suatu cabang pohon.



Gambar 2.2 Sistem konduksi jantung **[5]**

Sistem konduksi diatas di mulai dari nodus sinoatrial sebagai *pacemaker* yang berguna untuk memicu setiap siklus jantung. Nodus SA ini biasa dipengaruhi oleh sistem saraf pusat, seperti impuls dari saraf simpatis akan menambah kecepatannya dan saraf parasimpatis akan memperlambatnya. Hormon tiroid dan epinefrin yang dibawa oleh darah juga dapat mempengaruhi kecepatan impuls nodus SA. Setelah impuls listrik yang diinisiasi oleh nodus SA, impulnya akan menyebar melalui kedua atrium sehingga menyebabkan kedua atrium berkontraksi secara berkesinambungan. Pada saat yang sama impuls tersebut mendepolarisasi nodus atrioventrikular yang berada dibawah atrium kanan.

Dari nodus AV ini, cabang dari serat konduksi yaitu berkas His melalui otot jantung sampai septum interventrikular. Berkas His ini kemudian bercabang menjadi cabang kanan (*right bundle*) dan cabang kiri (*left bundle*). Walaupun berkas His mendistribusikan energi listrik ini sampai melewati permukaan medial ventrikel, kontraksi sesungguhnya distimulasi oleh berkas purkinje (serat otot konduksi) yang muncul dari cabang bundle yang dilanjutkan ke sel miokardium ventrikel.

2.2 Aritmia

Istilah aritmia mengacu pada perubahan dari mekanisme penjalaran impuls listrik jantung yang menyebabkan gangguan irama denyut jantung. Ada dua bentuk mendasar dari aritmia yaitu Takikardi, jika denyut jantung >100x/menit dan Bradikardi, jika denyut jantung <60x/menit [2]. Beberapa aritmia berlangsung secara singkat sehingga denyut jantung keseluruhan tidak terlalu terpengaruhi. Namun jika aritmia berlangsung cukup lama dapat mengakibatkan denyut jantung

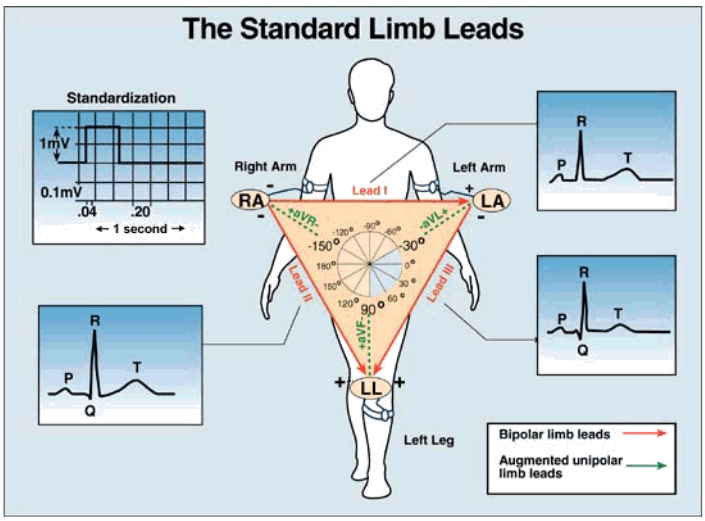
menjadi terlalu lambat ataupun terlalu cepat sehingga kemampuan jantung untuk memompa darah menjadi kurang efektif. Takikardi mengurangi curah jantung dengan memperpendek waktu pengisian ventrikel dan volume sekuncup, sedangkan bradikardi mengurangi curah jantung dengan mengurangi frekuensi ejeksi ventrikel. Aritmia dapat terjadi jika *pacemaker* (nodus SA) menghasilkan irama yang abnormal, kemudian adanya gangguan pada jalur konduksi normal, dan bagian jantung selain nodus SA mengambil alih sebagai *pacemaker*.

2.3 Electrocardiography

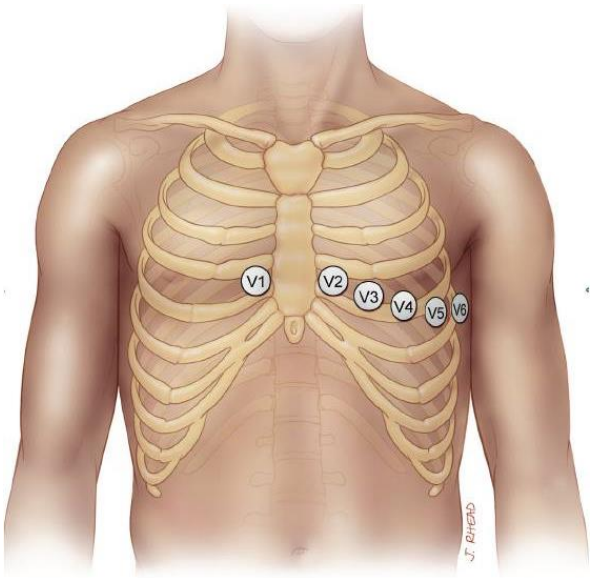
Electrocardiography adalah proses (atau bisa diartikan sebagai ilmu) menghasilkan *electrocardiogram*, sebuah grafik tegangan terhadap waktu aktivitas listrik jantung menggunakan alat yang dinamakan *electrocardiograph*. ECG mencatat dari permukaan tubuh dan mencatat perbedaan potensial listrik yang dihasilkan oleh jantung. Sinyal yang direkam ditentukan oleh potensial aksi yang dihasilkan oleh jutaan sel individu dan urutan aktivasinya.

2.3.1 Sadapan ECG

Electrocardiograph mendeteksi sinyal listrik yang terkait dengan aktivitas jantung dan menghasilkan ECG, catatan grafis dari tegangan terhadap waktu. Mereka digunakan untuk mendiagnosis dan membantu mengobati beberapa jenis penyakit jantung dan aritmia, menentukan respons pasien terhadap terapi obat, dan mengungkap tren atau perubahan fungsi jantung. Unit ECG terdiri dari unit ECG, elektroda, dan kabel. Sistem 12-lead mencakup tiga jenis lead: bipolar, augmented atau unipolar, dan precordial. Masing-masing dari 12 lead standar menyajikan perspektif yang berbeda dari aktivitas listrik jantung; menghasilkan bentuk gelombang EKG di mana gelombang P, kompleks QRS, dan gelombang T bervariasi dalam amplitudo dan polaritas. 12 lead bisa didapatkan dengan



Gambar 2.3 Bipolar dan augmented limb lead **[6]**



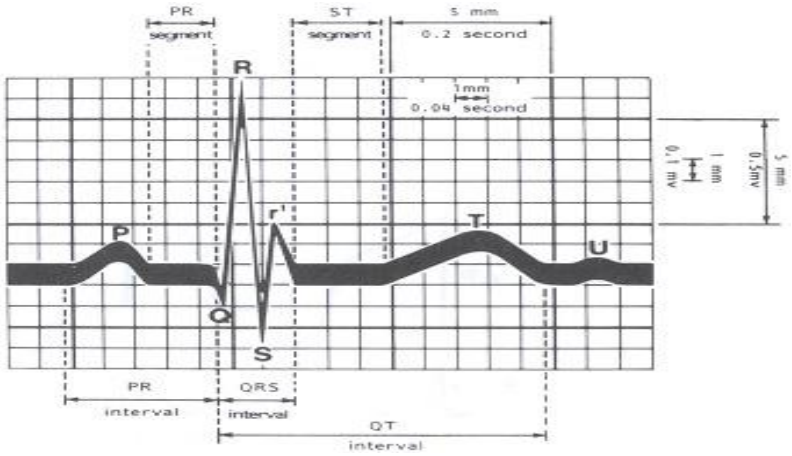
Gambar 2.4 Precordial chest lead **[6]**

pengambilan sinyal yang berasal dari sepuluh buah elektroda tersebut dihubungkan ke tubuh manusia yaitu, Right Arm (RA), Left Arm (LA), Left Leg (LL), Right Leg (RL), Chest 1 (C1), C2, C3, C4, C5 dan C6 [6].

Multi-channel ECG merekam sinyal dari dua atau lebih *lead* secara bersamaan. Sebaliknya, pada single-channel ECG hanya merekam sinyal listrik jantung dari satu konfigurasi *lead* pada satu waktu [7]. Multi-channel ECG memang banyak digunakan karena hasilnya yang lebih lengkap untuk melihat keseluruhan sinyal jantung dari beberapa posisi. Namun sistem multi-channel 12 Lead sulit dilakukan untuk *ambulatory system* sehingga digunakanlah sistem single-channel [8]. Lead yang digunakan dalam penelitian ini yaitu lead I. Lead I merekam beda potensial antara tangan kanan (RA) dengan tangan kiri (LA) yang mana tangan kanan bermuatan (-) dan tangan kiri bermuatan (+). Untuk keperluan pemantauan, sistem single-channel dianggap cukup untuk digunakan. Lead I merupakan jenis lead yang didapatkan dari bagian *bipolar limb*. Adapun Lead I didapatkan dari elektroda-elektroda yang diletakkan pada lengan kiri (LA), lengan kanan (RA), dan kaki kiri (LL).

2.3.2 Gelombang, Interval, dan Segmen

Dengan konvensi, defleksi ke atas pertama dari garis dasar disebut gelombang P, dan ini mencerminkan depolarisasi atrium. Gelombang P sebaiknya tidak melebihi 0,11 detik. Depolarisasi ventrikel diwakili oleh kompleks QRS [9]. Gelombang Q adalah defleksi negatif pertama dari garis dasar setelah gelombang P, tetapi sebelum defleksi ke atas. Biasanya, gelombang Q mencerminkan depolarisasi septum ventrikel, dan durasinya tidak melebihi 0,03 detik. Gelombang R adalah defleksi positif pertama setelah gelombang P, yang mencerminkan depolarisasi massa ventrikel. Gelombang S adalah defleksi negatif setelah



Gambar 2.5 ECG *grid*. Gelombang, interval, dan segmen pada ECG **[9]**

gelombang R positif yang mewakili depolarisasi ventrikel kemudian. Setiap defleksi positif setelah gelombang S diberi label R ′ (baca "R-*prime*"); setiap defleksi negatif setelah R ′ dilabeli S ′. Dengan konvensi, huruf kapital R atau S menyimpulkan defleksi besar, sedangkan huruf kecil r atau s menyimpulkan defleksi yang lebih kecil.

Gelombang T mencerminkan repolarisasi ventrikel dan dapat direpresentasikan sebagai defleksi positif atau negatif setelah kompleks QRS. Area yang tergabung dalam gelombang T kira-kira sama dengan di dalam kompleks QRS, dan polaritasnya kira-kira sama dengan polaritas QRS utama.

Kadang-kadang, gelombang lain, gelombang U, dapat mengikuti gelombang T, dan umumnya memiliki polaritas yang sama dengan gelombang T. Mekanisme gelombang U tidak diketahui, meskipun mungkin mencerminkan repolarisasi otot papiler, atau hanya mewakili *afterpotential*.

Interval PR adalah waktu dari awal gelombang P ke awal QRS, apakah diprakarsai oleh Q atau R, dan interval ini menunjukkan waktu yang diperlukan untuk atria untuk mendepolarisasi, dan untuk arus listrik untuk melakukan melalui simpul atrioventrikular dan cabang bundel sampai ventrikel mengalami depolarisasi. Interval QRS adalah interval itu dari awal gelombang Q ke akhir gelombang S, menggabungkan depolarisasi ventrikel. Interval QT adalah waktu dari awal gelombang Q ke akhir gelombang T, menggabungkan depolarisasi ventrikel dan repolarisasi. Segmen PR adalah bagian dari rekaman antara akhir gelombang P dan awal QRS. Segmen ST adalah bagian rekaman itu, yang pada umumnya diwakili oleh garis horizontal, dari ujung depolarisasi ventrikel, baik diwakili oleh gelombang R atau gelombang S, hingga awal gelombang T [9].

2.4 Embedded System

Seperti namanya, *embedded system* adalah sistem berbasis mikroprosesor atau mikrokontroler, yang dirancang untuk fungsi tertentu dan tertanam ke dalam sistem mekanik atau listrik yang lebih besar. Karena *embedded system*

dikembangkan untuk beberapa tugas khusus daripada menjadi sistem tujuan umum untuk beberapa tugas, mereka biasanya berukuran terbatas, daya rendah, dan biaya rendah. *Embedded system* banyak digunakan dalam berbagai keperluan, seperti aplikasi komersial, industri, dan militer. Pada Tugas Akhir ini, instrumen ini yang merupakan *embedded system* berfungsi pula sebagai analyzer. Dikatakan seperti itu karena pada sistem instrumentasi ini juga dilakukan perbandingan metode-metode *QRS detection* secara bersamaan.

Biasanya, sistem tertanam terdiri dari komponen *hardware* dan *software* aplikasi. Beberapa *embedded system* memiliki *Real-Time Operating System* (RTOS). Beberapa sistem tertanam kecil mungkin tidak memiliki RTOS. Oleh karena itu, *embedded system* dapat didefinisikan sebagai sistem berbasis mikroprosesor atau mikrokontroler, berbasis *software*, andal, dan *real-time control*. Salah satu contoh *embedded system* misalnya pada kartu plug-in dengan beberapa komponen seperti prosesor, memori, *power supply*, dan *external interfaces*.

Embedded system berisi mikroprosesor atau mikrokontroler yang biasanya dirancang untuk melakukan komputasi untuk operasi *real-time*. Umumnya, mikroprosesor hanya merupakan *central processing unit* (CPU). Oleh karena itu, komponen lain (misalnya, memori dan *communication interfaces*) perlu diintegrasikan dan bekerja dengan mikroprosesor sebagai keseluruhan sistem. Sebaliknya, mikrokontroler adalah sistem mandiri, yang mencakup CPU, memori (mis., RAM, memori flash), dan periferal (mis., port komunikasi serial).

Mikroprosesor atau mikrokontroler yang digunakan dalam *embedded system* pada umumnya tidak semaju jika dibandingkan dengan prosesor tujuan umum yang dirancang untuk mengelola banyak tugas. Mereka sering bekerja pada lingkungan program yang sederhana. Akibatnya, *software* pada *embedded system* memiliki persyaratan dan kemampuan *hardware* tertentu. Ini dirancang untuk *hardware* tertentu dan memiliki keterbatasan waktu dan memori. Program dan sistem operasi umumnya disimpan dalam memori flash di dalam *embedded system*.

Secara garis besar, karakteristik dari *embedded system* dapat dilihat dari beberapa hal, yaitu fungsi spesifik, keterbatasan, desain *hardware/software*, memori, periferal yang terhubung, dan berbasis mikrokontoler/mikroprosesor [10]. *Embedded system* biasanya dirancang untuk fungsi tertentu/spesifik. Kemudian untuk keterbatasan dari *embedded system* yaitu pada sumber daya dan waktu. Misalnya, *embedded system* harus cepat dan toleran terhadap tugas terhadap sedikit variasi dalam waktu reaksi (secara *real-time* atau hampir *real-time*), dengan memori terbatas dan konsumsi daya minimum. *Embedded system* biasanya merupakan sistem *hardware* komputer dengan *software* tertanam di dalamnya. *Hardware* dirancang untuk kinerja dan keamanan, sementara *software* dirancang

untuk lebih banyak fitur dan fleksibilitas. Lalu *embedded system* memerlukan memori untuk program dan sistem operasi pada umumnya dimuat dan disimpan dalam memori. Periferal juga diperlukan untuk menghubungkan perangkat *input* dan *output*. Mikroprosesor atau mikrokontroler sering digunakan sebagai jantung *embedded system* dan dirancang untuk melakukan operasi.

2.5 Hardware

2.5.1 Modul Mikromedia 5 for STM32 Capacitive

Penelitian ini menggunakan modul Mikromedia 5 for STM32 Capacitive dengan bagian utamanya menggunakan mikrokontroler STM32F407ZGT6 (32 bit) dengan *flash memory* sebesar 1 MB dan 192 + 4 KB SRAM (termasuk 64 KB dari Core Coupled Memory). STM32F407ZGT6 merupakan mikrokontroler yang masuk dalam keluarga mikrokontroler ARM Cortex-M4 yang diproduksi oleh STMicroelectronics. Mikrokontroler ini memiliki ADC 12-bit sebanyak 3 buah. Kemudian besaran *operating frequency* dapat mencapai 168 MHz untuk kecepatan pemrosesan sinyal yang lebih baik. Modul ini dilengkapi dengan LCD *touch screen* beresolusi 800x480px [11]. Dengan ukuran tampilan sebesar 5” maka modul ini dapat digunakan sebagai development board ataupun sebagai final product untuk berbagai macam keperluan bidang seperti industri, medis, automotif, dan lain-lain. Selain itu, modul ini juga dilengkapi dengan slot mikroSD card, konektor miniUSB, konektor baterai, dan lain-lain. Dengan kelengkapan yang cukup banyak, penggunaan MCU yang *powerful*, memori yang cukup besar, kristalnya besar sehingga cocok untuk pemrosesan sinyal digital yang memerlukan kecepatan *processing*, serta modul dengan daya tahan yang tinggi maka modul ini dianggap cocok untuk keperluan penelitian ini.

2.5.2 ADC

Meskipun pada mikrokontroler STM32F408ZGT6 telah terdapat juga ADC di dalamnya, pada sub bab ini akan dijelaskan sedikit mengenai ADC secara umum.

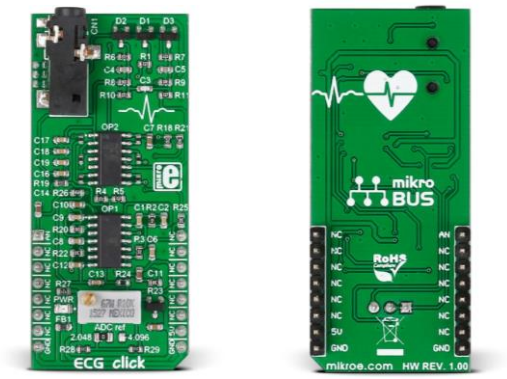
Gambar 2.6 Mikromedia 5 for STM32 Capacitive **[11]**

Konverter Analog-ke-Digital adalah sirkuit "sinyal campuran" yang melakukan digitalisasi sinyal analog eksternal [12]. Pada pengolahan sinyal digital yang menggunakan input berupa sinyal analog, penting untuk mendigitalisasi sinyal masukan tadi. Untuk mengubah sinyal analog menjadi sinyal digital dapat digunakan ADC (Analog-to-Digital Converter), dimana sinyal analog harus melalui proses sampling, quantizing dan coding. Demikian juga output dari processor digital harus melalui perangkat Digital-to-Analog Converter (DAC) agar outputnya kembali menjadi bentuk analog. Proses sampling merupakan proses yang penting karena proses ini mengubah sinyal analog yang berbentuk sinyal waktu kontinu menjadi sinyal waktu diskrit. Untuk mendapatkan sinyal waktu diskrit yang mampu mewakili sifat sinyal aslinya, proses sampling harus memenuhi syarat Nyquist. Apabila proses sampling tidak sesuai dengan syarat Nyquist maka akan terjadi aliasing sinyal.

Pada 2 versi mikrokontroler yang dipertimbangkan untuk digunakan pada penelitian ini yaitu A/D *converter* 10-bit dan A/D *converter* 12-bit. Seperti telah ditunjukkan dalam praktek bahwa untuk sistem kontrol atau pengaturan, konverter A/D 10-bit memberikan solusi yang memuaskan. Bit tambahan di konverter A/D dalam aplikasi ini hampir tidak dapat digunakan karena mereka ditutupi oleh tingkat noise elektro-magnetik yang tinggi. Dalam sistem pengukuran, praktik menunjukkan bahwa dengan konverter 12-bit, seseorang dapat mencapai hasil yang baik antara tingkat sampling dan akurasi (0,02% untuk konverter 12-bit).

Mikrokontroler A/D *converter* 10-bit dapat mengambil sampel hingga 16 input analog dengan frekuensi 500kHz yang menerapkan sistem *successive approximation* (SAR). Metode *successive approximation conversion* adalah salah satu tipe dasar konversi A/D. Metode ini dilakukan dalam beberapa langkah sampai tegangan input sama dengan tegangan yang dihasilkan secara internal. Kecepatan konversi dari jenis konverter A/D ini terbatas, tetapi cukup cepat untuk sebagian besar aplikasi umum. Konverter A/D 10-bit ini memiliki kemampuan menggunakan referensi tegangan eksternal, pengambilan sampel secara bersamaan hingga 4 input analog di amplifier *Sample/Hold* (pengambilan sampel beberapa sinyal pada satu waktu), mode pemindaian saluran otomatis, dan dapat dipilih sumber pemicu konversi. Hasil konversi dimuat ke buffer 16-word dalam 4 format output yang dapat dipilih (dua integer dan dua *floating point*). Selain itu, konverter A/D dapat beroperasi dalam mode SLEEP atau IDLE.

Kinerja mikrokontroler yang memiliki 12-bit A/D *converter* agak lebih rendah daripada 10 bit, tetapi akurasi konversinya lebih tinggi, bisa mengambil sampel hingga 16 input analog dengan frekuensi 100kHz menggunakan sistem SAR. Konverter dapat menggunakan referensi tegangan eksternal, hanya sampel salah satu dari 16 input analog dalam penguat *Sample / Hold*, memiliki mode

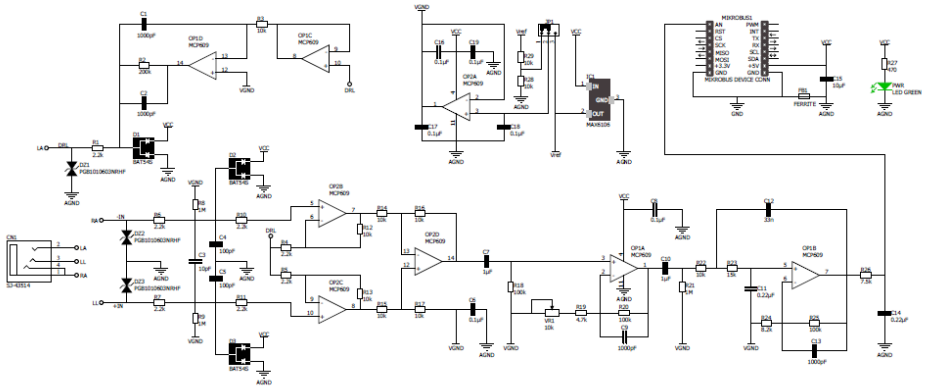


Gambar 2.7 ECG Click **[13]**

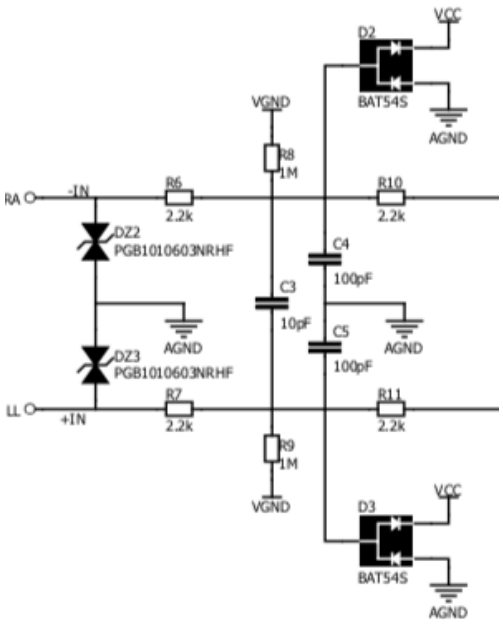
pemindaian saluran otomatis dan sumber pemicu konversi yang dapat dipilih. Seperti konverter 10-bit, hasil konversi dimuat ke register buffer 16-word dalam 4 format output yang dapat dipilih (2 bilangan bulat dan 2 *floating point*). Selain itu, konverter A/D dapat beroperasi dalam mode SLEEP atau IDLE.

2.5.3 *ECG Click*

ECG *click* merupakan modul ECG yang berfungsi untuk untuk mengukur aktivitas listrik jantung yang berdetak melalui elektroda yang ditempelkan pada permukaan kulit. Modul ini menggunakan OpAmp MCP609 dan MAX6106 sebagai *Voltage Reference* IC. Modul ini mendapatkan input dari pembacaan 3 elektroda yang digunakan. Biasanya elektroda-elektroda ini ditempatkan di bagian tangan kiri (LA), tangan kanan (RA), dan kaki kiri (LL). ECG Click ini berisi jack 3.5mm yang terhubung dengan tiga elektroda. Elektroda akan mendeteksi aktivitas listrik jantung dan mengarahkan sinyal ke jack 3.5mm. Setelah itu, sinyal perlu diperkuat dan disaring, oleh karena itu ECG click ini juga berisi dua amplifier, dua *high-pass filter* dan satu *low-pass filter*. Setelah data diperoleh, diperkuat dan difilter, itu dikirim ke satu pin analog.



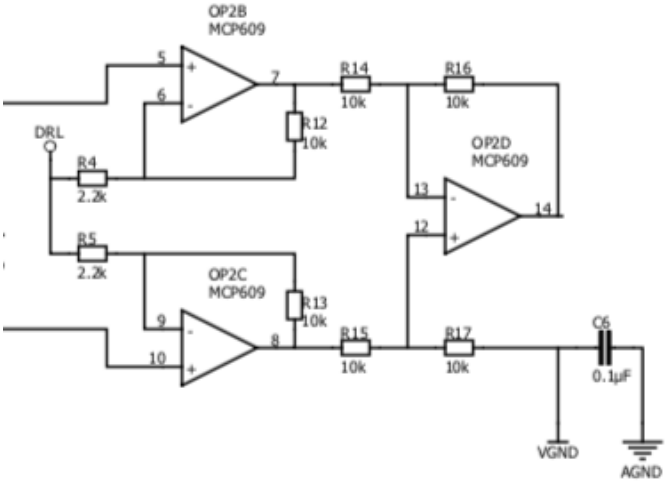
Gambar 2.8 *Schematic Diagram* ECG Click **[13]**



Gambar 2.9 Rangkaian proteksi **[13]**

Mikrokontroler kemudian membaca data analog ini, mengubahnya menjadi digital, data digital ini menyajikan nilai aktivitas listrik jantung pada saat tertentu.

Modul ECG click ini terdiri dari design 7-block yaitu rangkaian proteksi, preamplifier, high-pass filter #1, amplifier, high-pass filter #2, low-pass filter, dan DRL circuit [13] yang dapat dilihat pada Gambar 2.8. Pada Gambar 2.9 sampai Gambar 2.16 merupakan pecahan rangkaian-rangkaian yang ada pada Gambar 2.8. Rangkaian proteksi berfungsi untuk melindungi rangkaian dari tegangan tinggi yang muncul pada input ECG pada kondisi tertentu agar tidak merusak alat dan melukai pasien. Pada rangkaian proteksi yang ditunjukkan seperti pada Gambar 2.9 terdapat filter LPF yang digunakan untuk menghilangkan frekuensi RF (*Radio Frequency*). Kemudian rangkaian ini juga dilengkapi dengan dioda BAT54S sebagai pelindung agar tidak ada arus bocor ataupun tegangan lebih atau arus lebih yang menuju pasien. Kemudian output dari rangkaian proteksi akan masuk ke rangkaian pre-amplifier untuk dikuatkan sinyalnya. Hal ini berfungsi untuk memperbesar tegangan sinyal karena sinyal tubuh memiliki nilai tegangan yang kecil sehingga akan menyulitkan untuk diambil informasinya secara langsung. Penguatan yang dilakukan pada rangkaian ini yaitu Y kali. Gambar rangkaian pre-amplifier dapat dilihat pada Gambar 2.10. Kemudian proses dilanjutkan dengan menuju rangkaian HPF #1 yang dapat dilihat pada Gambar 2.11. Rangkaian HPF #1 merupakan HPF pasif karena menggunakan komponen pasif. Nilai frekuensi cut-off yang digunakan dapat dicari dengan menggunakan Persamaan 2.1



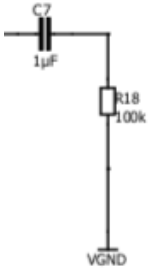
Gambar 2.10 Rangkaian Pre-Amplifier **[13]**

$$f_c = \frac{1}{2\pi RC} \tag{2.1}$$

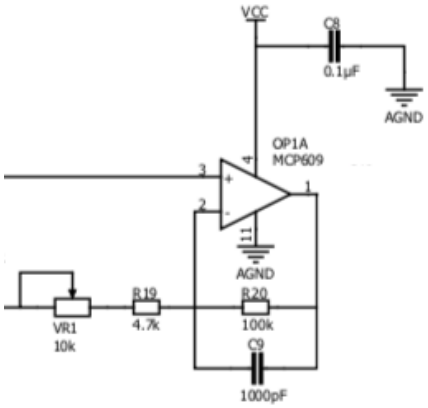
Dengan menggunakan Persamaan 2.1 maka dapat diketahui nilai frekuensi cut-off pada rangkaian HPF #1 sebesar 1.6Hz. Penggunaan HPF pasif ketika sinyal frekuensi hasil filter tidak perlu lagi di kuatkan menggunakan penguat, artinya sinyal frekuensi hasil filter telah dapat diterjemahkan atau dibaca. Selanjutnya hasil filter tadi dikuatkan dengan masuk ke rangkaian amplifier seperti pada Gambar 2.12. Tipe penguatan yang digunakan pada rangkaian ini yaitu non-inverting amplifier. Penguatan yang dilakukan oleh rangkaian non-inverting amplifier dihitung dengan rumus seperti pada Persamaan 2.2.

$$V_{out} = \left[1 + \frac{R_f}{R_{in}}\right] V_{in} \tag{2.2}$$

Penguatan dilakukan dengan mengubah besar hambatan potensiometer VR1. Penguatan dilakukan agar tegangan sinyal terbaca oleh ADC yang berkisar di tegangan 0-5V. Hasil keluaran dari rangkaian non-inverting amplifier ini akan masuk ke rangkaian HPF #2 yang ditunjukkan pada Gambar 2.13. Rangkaian



Gambar 2.11 Rangakaian HPF #1 **[13]**



Gambar 2.12 Rangkaian Amplifier **[13]**

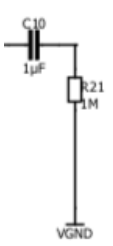
HPF #2 ini juga merupakan rangkaian pasif karena menggunakan komponen-komponen pasif. Nilai frekuensi cut-off nya dapat dihitung dengan menggunakan Persaman 2.1. Dari persamaan tersebut dapat diketahui jika nilai frekuensi cut-off pada rangkaian HPF #2 yaitu sebesar 0.16Hz. Selanjutnya hasil keluarannya akan masuk ke rangkaian LPF yang ditunjukkan seperti pada Gambar 2.14. Pada rangkaian yang berfungsi untuk meredam frekuensi tinggi dari sinyal listrik di atas frekuensi cut-off. Rangkaian LPF tersebut terdiri dari rangkaian LPF aktif dan LPF pasif. Untuk rangkaian LPF aktif menggunakan orde 2 dengan morfologi sallen key dan untuk rangkaian LPF pasif menggunakan orde 1. Frekuensi cut-off yang digunakan yaitu seperti pada Persamaan 2.3.

$$f_c = \frac{1}{2\pi\sqrt{R_1 R_2 C_1 C_2}} \tag{2.3}$$

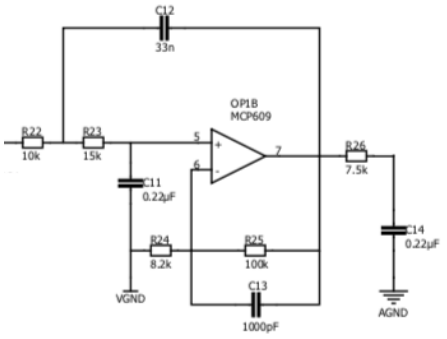
Kemudian untuk penguatan tegangannya dihitung dengan menggunakan Persamaan 2.4.

$$Av = 1 + \frac{R_A}{R_B} \tag{2.4}$$

Terakhir, terdapat rangkaian *driven right leg* (DRL) seperti pada Gambar 2.15. Rangkaian DRL adalah rangkaian listrik yang sering ditambahkan ke penguat sinyal biologis untuk mengurangi *noise* berupa *common-mode*



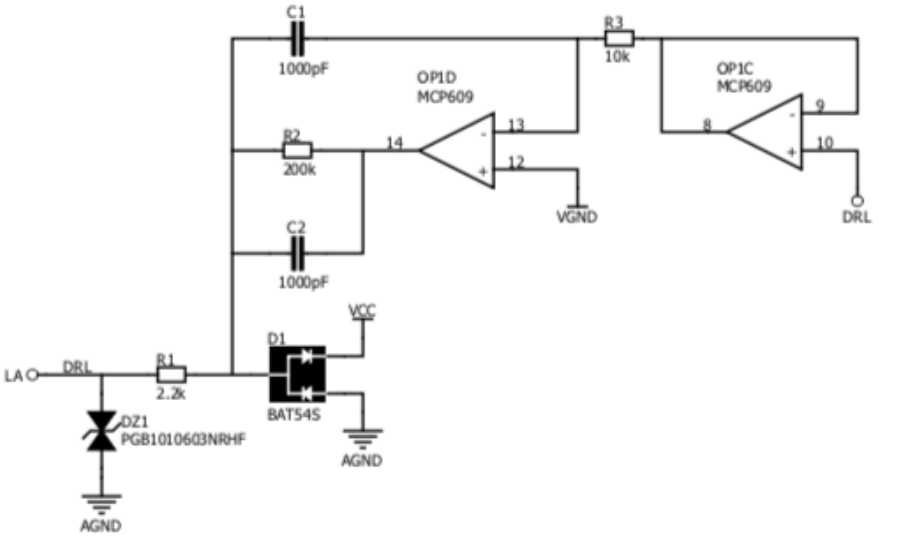
Gambar 2.13 Rangkaian HPF #2 [13]



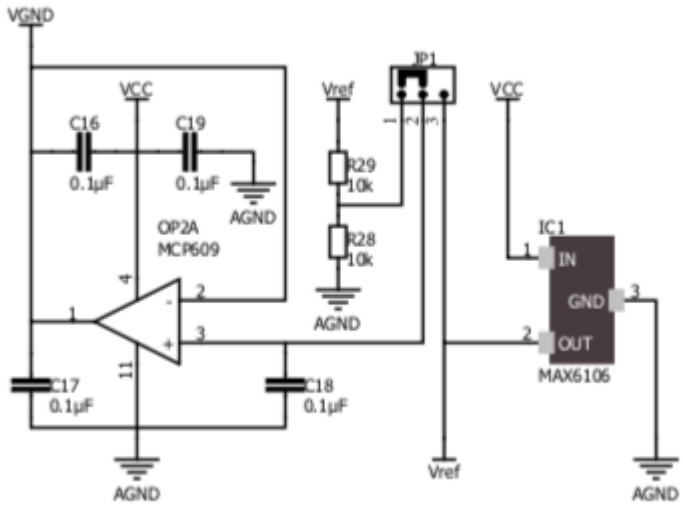
Gambar 2.14 Rangkaian LPF **[13]**

interference. Penguat sinyal biologis seperti ECG (elektrokardiogram) mengukur sinyal listrik yang sangat kecil yang dipancarkan oleh tubuh, seringkali sekecil beberapa mikro-volt (sepersejuta volt). Namun tubuh pasien juga dapat bertindak sebagai antena yang menerima interferensi elektromagnetik, terutama noise interferensi 50/60 Hz dari saluran listrik. Gangguan ini dapat mengaburkan sinyal biologis, membuatnya sangat sulit untuk diukur. Rangkaian DRL digunakan untuk menghilangkan gangguan interferensi dengan secara aktif membatalkan interferensi. Pada bagian paling kanannya terdapat rangkaian buffer dengan 1x penguatan untuk menurunkan arus dan mempertahankan tegangan agar tidak turun.

Kemudian modul ECG click ini dilengkapi juga dengan MAX6106 yang merupakan *voltage reference IC.* Rangkaiannya dapat dilihat pada Gambar 2.16. IC ini merupakan perangkat elektronik yang idealnya menghasilkan tegangan tetap (konstan) terlepas dari pemuatan pada perangkat, variasi *power supply*,



Gambar 2.15 Rangkaian DRL **[13]**



Gambar 2.16 Rangkaian *voltage reference* **[13]**

perubahan suhu, dan berlalunya waktu. Referensi tegangan mode seri ini hanya menghasilkan 90µA arus pasokan dan dapat mengambil 5mA dan menenggelamkan arus beban 2mA. Tidak seperti referensi mode-shunt (dua terminal) konvensional yang memboroskan arus pasokan dan memerlukan resistor eksternal, perangkat ini menawarkan arus pasokan yang hampir tidak tergantung pada tegangan suplai (dengan hanya variasi 4µA / V dengan tegangan suplai) dan tidak memerlukan resistor eksternal. Selain itu, perangkat yang dikompensasi secara internal ini tidak memerlukan kapasitor kompensasi eksternal dan stabil dengan kapasitansi beban. Menghilangkan kapasitor kompensasi eksternal menghemat area pada board yang juga terbatas. Tegangan *low-dropout* dan suplai independen, kemudian arus suplai yang rendah membuat perangkat ini ideal untuk sistem yang menggunakan baterai, berkinerja tinggi, bertegangan rendah. MAX6106 tersedia dalam SOT23 *package* dengan 3 pin kecil.

Gambar 2.17 MS400 Multiparameter Simulator **[15]**

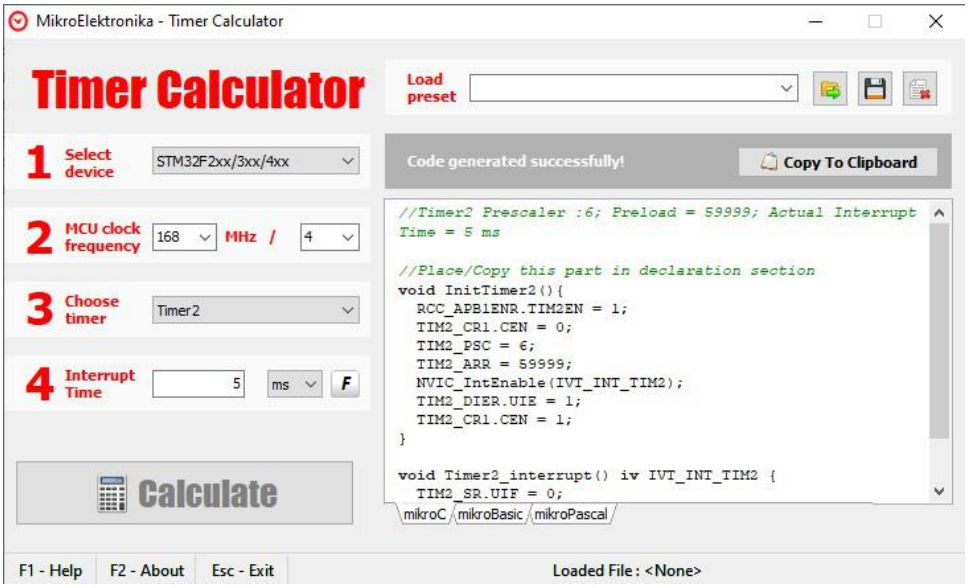
2.5.4 Phantom ECG

Sinyal ECG merupakan salah satu obyek yang paling sering dipakai dalam mempelajari fenomena bioelektrik. Sinyal ini dipelajari dalam bidang kedokteran untuk aplikasi kesehatan atau dalam bidang teknik biomedika untuk mempelajari teknik instrumentasi biomedik. Sinyal ini terjadi karena proses depolarisasi sel-sel jantung yang terjadi spontan diawali oleh Sinoatrial node (SA node). Untuk mempelajari ECG dibutuhkan alat bantu berupa gambar sinyal ECG atau perangkat ECG untuk menampilkan sinyal ECG. Perangkat ECG cukup mahal karena pengunaan yang terbatas. Selain itu, keterbatasan subjek yang digunakan dalam penelitian pun terbatas. Untuk itu diperlukan simulator ECG atau suatu perangkat untuk menampilkan sinyal ECG guna memudahkan proses mempelajari sinyal ECG. Simulator ECG atau biasa disebut dengan phantom ECG, biasanya digunakan di rumah sakit untuk kalibrasi perangkat ECG. Perangkat ini bisa dianggap sebagai generator sinyal dengan sinyal keluaran berupa sinyal ECG [14].

Phantom ECG yang digunakan untuk pengujian pada penelitian ini yaitu Contec MS400 Multiparameter Simulator. MS400 Multiparameter Simulator adalah perangkat portabel, dengan volume kecil dan ringan [15]. MS400 Multiparameter Simulator dapat mensimulasikan secara akurat 12-lead EKG, Respirasi, Temperatur, 2-channel IBP atau 4-channel IBP. Selain itu dapat digunakan pula beberapa mode seperti *noise.* Dan itu bisa diuji di mana monitor digunakan. Phantom ini diproduksi oleh Contect Medical System.

2.6 Software

Software yang digunakan untuk mendukung Tugas Akhir ini diantaranya yaitu Timer Calculator, MikroC Pro for ARM, dan CODEGRIP.



Gambar 2.18 Timer Calculator

2.6.1 Timer Calculator

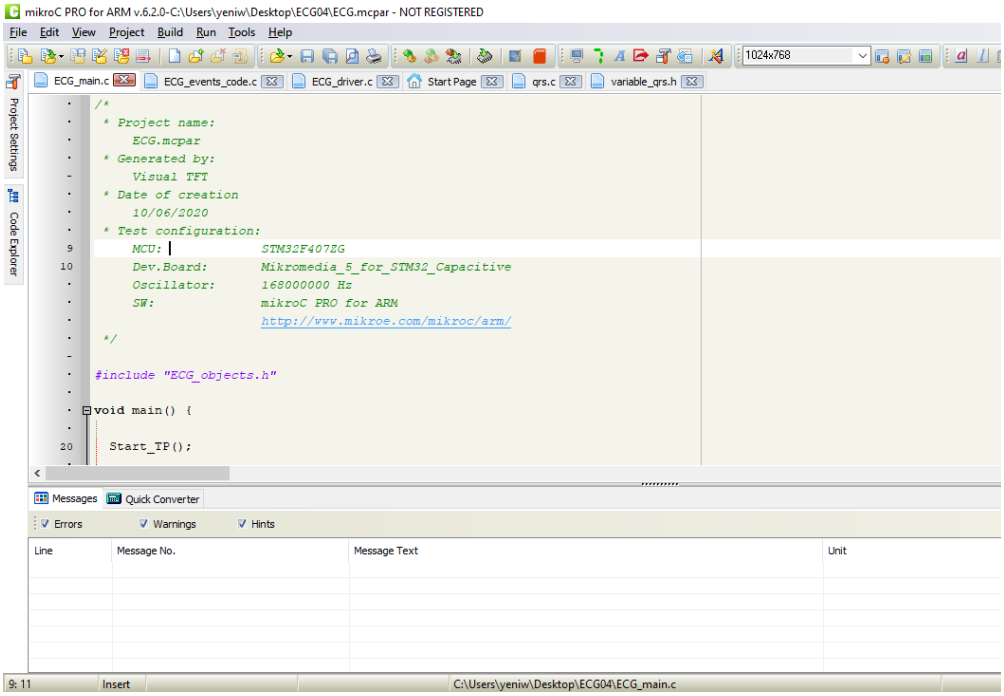
Timer Calculator merupakan software pendukung yang digunakan untuk membuat *interrupts code* untuk beberapa software programming mikrokontroler, misalnya seperti MikroC, MikroBasic, dan MikroPascal. *Interface* Timer Calculator dapat dilihat pada Gambar 2.18. Dapat dilihat pada Gambar 2.18 bahwa dalam penggunaan Timer Calculator diperlukan input berupa jenis mikrokontroler yang digunakan, frekuensi *clock* dari mikrokontroler yang digunakan, timer, dan interrupt time. Interrupt time adalah waktu yang digunakan untuk menginterupsi proses yang sedang berjalan. Untuk interrupt time didapatkan dengan cara membagi 1 dengan besar frekuensi sampling seperti pada Persamaan 2.5.

$$t = \frac{1}{fs} \tag{2.5}$$

Frekuensi sampling sendiri merupakan kecepatan pengambilan sample (sampling). Sehingga hal ini berhubungan juga dengan waktu pemrosesan sinyal yang dilakukan. Pada penelitian ini digunakan 2 besaran frekuensi sampling. Besar frekuensi sampling yang pertama yaitu besar 320Hz dan yang kedua yaitu 200Hz. Dimana besar frekuensi sampling yang digunakan tergantung pada metode yang digunakan. Sehingga bisa dikatakan penggunaan besar frekuensi sampling yang berbeda akan berpengaruh pada waktu komputasi metode tersebut.

2.6.2 MikroC Pro for ARM

MikroC Pro for ARM adalah sebuah *Integrated Development Environment* (IDE) yang digunakan untuk membuat project dengan menggunakan bahasa pemrograman C pada mikrokontroler ARM. MikroC PRO untuk ARM saat ini mendukung lebih dari 1312 ARM dengan bermacam-macam tipe seperti Cortex-



Gambar 2.19 *Workspace* pada MikroC Pro for ARM

M0, M0 +, M3, M4, dan mikrokontroler M7 dari manufaktur terkemuka, dan akan terus dikembangkan. MikroC Pro for ARM ini juga dilengkapi dengan library yang dapat memudahkan dalam pengerjaan project. Kumpulan library tersebut, yang berjumlah 1200 *library function*, tersedia pada menu "Library Manager". Kemudian IDE ini juga digunakan untuk menulis kode, *compile*, *execute*, dan men-*debug* kode program yang ditulis. Fitur-fitur yang tersedia pada IDE ini diantaranya selain Library Manager yaitu Code Assistant, Wild Card Code Completion, Active Comments, Interrupt Assistant, Build-in tools, In-Circuit Debugging, Software simulator, Statistics. IDE ini juga bisa diintegrasikan dengan software VisualTFT yang dapat mempermudah pengerjaan pembuatan GUI (*Graphical User Interface*).

2.6.3 CODEGRIP Suite

CODEGRIP Suite merupakan sebuah software pendukung yang digunakan untuk memasukkan program yang telah dibuat sebelumnya ke mikrokontroler. CODEGRIP Suite digunakan juga bersama dengan alat penunjangnya yang bernama CODEGRIP yang berfungsi sebagai *external programmer*. Pada CODEGRIP ini ada beberapa pilihan jalur yang dapat dipilih untuk memasukkan program ke mikrokontroler, contohnya bisa digunakan kabel USB C-type atau *wireless* menggunakan WiFi.

2.7 Metode Deteksi QRS

Pada metode deteksi QRS yang akan dijelaskan dibawah ini, dapat dibagi menjadi beberapa bagian. Bagian ini dibagi berdasarkan teknik pemrosesan sinyal yang digunakan untuk menekankan segmen QRS dalam *time series*, *frequency series*, dan *time-frequency series*. Langkah ini disebut *QRS enhancement*. Pentingnya *QRS enhancement* yaitu sebagai prasyarat dalam mendeteksi kompleks QRS [16]. Ada 4 teknik *QRS enhancement* yang akan digunakan dalam penelitian ini, yaitu *Amplitude*, *First Derivative Only*, *First and Second Derivative*, dan *Digital Filter* [17].

2.7.1 Amplitude & First Derivative Algorithm

Algoritma ini merupakan algoritma yang paling tua dalam pendeteksian puncak R pada sinyal ECG, dimana metode ini lazim digunakan selama 30 tahun. Algoritma ini biasanya tidak hanya digunakan sendiri, melainkan diikuti dengan langkah selanjutnya yaitu diferensiasi untuk mengurangi pengaruh gelombang P dan T terhadap gelombang R [16]. Derivatif pertama ini akan diterapkan setelah *amplitude threshold* untuk menonjolkan *slope* dari kompleks QRS. *Amplitude threshold* dihitung dengan rumus seperti pada Persamaan 2.6.

$$X_{th} = \beta \max\{X[n]\} \tag{2.6}$$

Dimana β adalah persentase sinyal ECG yang perlu dihilangkan dan $1 > \beta > 0$. Algoritma *amplitude & first derivative* ini menjadi dasar *QRS enhancement* pada algoritma yang dilakukan oleh Moriet-Mahoudeux, Fraden and Neuman, dan Gustafson.

2.7.1.1 Moriet-Mahoudeux

Pada penelitian yang dilakukan oleh Moriet-Mahoudeux, penelitian ini menggunakan teknik *QRS enhancement* berupa *amplitude* dan dilanjutkan dengan *first derivative* [18]. Oleh karena itu, dalam mendapatkan *amplitude threshold* digunakan rumus seperti pada Persamaan 1. Pada penelitian ini digunakan nilai β sebesar 0.3, sehingga didapatkan persamaan QRS detector seperti pada Persamaan 2.7

$$X_{th} = 0.3 \max\{X[n]\} \tag{2.7}$$

Dimana persamaan di atas dapat diartikan jika nilai X[n] dibawah 30% dari amplitudo sinyal positif maksimum akan terpotong dari sinyal.

Selanjutnya dilakukan tahap *first derivative* Y[n] yang dihitung pada setiap titik sinyal X[n] dengan pendekatan *central finite-difference*. Dimana langkah ini menggunakan Persamaan 2.8

$$Y[n] = X[n+1] - X[n-1] \quad (2.8)$$

Sebuah kandidat QRS terjadi ketika tiga poin berturut-turut dalam array derivatif pertama melebihi threshold slope positif dan diikuti dalam 100 ms berikutnya oleh dua poin berturut-turut yang melebihi threshold negatif (descending slope), semua titik data dalam ECG antara terjadinya ascending slope dan sebelum akhir dari descending slope harus memenuhi atau melebihi threshold amplitude seperti pada Persamaan 2.9

$$Y(i), Y(i+1), Y(i+2) > 0.5 \text{ dan } Y(j), Y(j+1) > 0.3 \quad (2.9)$$

Dimana $(i+2) < j < (i+25)$ dan $X(i), X(i+1), \dots, X(j+1) \geq threshold\ amplitudo$

2.7.1.2 Fraden and Neuman

Penelitian yang dilakukan oleh Fraden and Newman juga menggunakan teknik *QRS enhancement* berupa *amplitudo threshold* dan *first derivative* [19]. Perbedaan antara algoritma yang dilakukan oleh Fraden dan Neuman dengan algoritma yang dilakukan oleh Moriet-Mahoudeux terletak pada nilai β. Dimana nilai β pada algoritma yang dikemukakan oleh Fraden dan Neuman sebesar 0.4, sehingga dapat diartikan jika nilai X[n] dibawah 40% dari amplitudo sinyal positif maksimum akan terpotong dari sinyal. Dimana dapat ditulis persamaan QRS detectornya seperti pada Persamaan 2.10,

$$X_{th} = 0.4 \max\{X[n]\} \quad (2.10)$$

Langkah selanjutnya pun mirip dengan algoritma Moriet-Mahoudeux yaitu dilakukan *first derivative*. *First derivative*, Y[n], dihitung pada setiap titik sinyal ECG X[n] yang menggunakan pendekatan *central finite-difference* [20], [21] sehingga didapatkan persamaan *first derivative* seperti pada Persamaan 2.11,

$$Y[n] = X[n+1] - X[n-1] \quad (2.11)$$

Dimana Y[n] merupakan hasil sinyal *first derivative*. Untuk *threshold* kedua ditetapkan nilai absolut sebesar 0.7 [22] sehingga puncak R akan terdeteksi ketika titik *derivative* melebihi nilai *threshold* kedua (0.7 * amplitudo maksimum dari sinyal yang telah diderivatif).

2.7.1.3 Gustafson

Pada algoritma yang diusulkan oleh Gustafson, teknik QRS enhancement yang digunakan juga menggunakan amplitudo dan first derivative [23]. Dimana persamaan first derivative yang digunakan dihitung pada setiap titik ECG.

Persamaan first derivative ini juga menggunakan pendekatan *central finite-difference* dengan persamaan seperti pada Persamaan 2.12

$$Y[n] = X[n+1] - X[n-1] \quad (2.12)$$

Untuk nilai $1 < n < N - 1$. Threshold konstan yang ditetapkan pada algoritma ini sebesar 0.15. Dimana array first derivative ini kemudian akan mencari point-point yang nilainya melebihi threshold konstan yang telah ditetapkan sebelumnya. Kemudian nilai derivative pada tiga titik selanjutnya ($Y(i+1), Y(i+2), Y(i+3)$) juga harus melebihi threshold konstan sebesar 0.15. Ketika kedua kondisi tersebut sudah terpenuhi maka point i akan bisa diklasifikasikan sebagai kandidat kompleks QRS jika perkalian dari hasil derivative dua sample point selanjutnya dengan sinyal ECG pada titik yang sama memiliki amplitude slope positif, seperti pada Persamaan 2.13

$$Y(i+1)X(i+1) > 0 \text{ dan } Y(i+2)X(i+2) > 0 \quad (2.13)$$

2.7.2 First Derivative Algorithm

Pada algoritma ini, teknik *QRS enchancement* menggunakan algoritma first derivative. Dimana pada teknik *QRS enhancement* ini hanya digunakan first derivative. *First-order* differentiator umumnya digunakan sebagai *high-pass filter*, meningkatkan *baseline wander* dan menghilangkan *noise* dengan frekuensi tinggi yang tidak diinginkan, memodifikasi fase sinyal ECG, dan membuat *zero crossing* pada lokasi puncak R. Ada beberapa algoritma deteksi QRS yang menggunakan *first derivative*, namun yang akan digunakan dalam penelitian ini yaitu algoritma yang telah dilakukan oleh Menard dan Holsinger.

2.7.2.1 Menard

Algoritma yang diusulkan oleh Menard menggunakan teknik QRS *enhancement* berupa *first derivative* [24]. Untuk menghitung hasil derivatif sinyal ECG, Menard menggunakan rumus seperti pada Persamaan 2.14,

$$Y[n] = -2X[n-2] - X[n-1] + X[n+1] + 2X[n+2] \quad (2.14)$$

Dimana n merupakan titik sinyal ke-n. Kemudian ditetapkan nilai *threshold* sebesar 0.7. Dimana deteksi gelombang R diperoleh jika titik turunan melebihi *threshold* 0.7 sebanding dengan amplitudo maksimum dari sinyal yang diturunkan [22].

2.7.2.2 Holsinger

Pada tahun 1971 W.P. Holsinger mengusulkan algoritma untuk mendeteksi QRS kompleks menggunakan *first derivative* sebagai teknik QRS *enhancement* [25]. Rumus *first derivative* yang digunakan berbeda dengan algoritma Menard,

dimana Holsinger menggunakan pendekatan *central finite-difference* [16] seperti pada Persamaan 2.15,

$$Y[n] = X[n+1] - X[n-1] \quad (2.15)$$

Dimana X[n] adalah sample ke-n dari sinyal ECG. *Threshold* yang digunakan yaitu 0.45. Array ini diperiksa sampai satu titik melampaui *threshold* dengan *slope* $y[i] > 0.45$. Poin ini dianggap sebagai kandidat kompleks QRS. Calon kompleks QRS secara efektif adalah kompleks QRS jika satu (1) dari tiga (3) titik pengukuran berikut memungkinkan hasil derivatifnya untuk melampaui *detection threshold* [20]. Ini jika $y[i+1] > 0.45$ atau $y[i+2] > 0.45$ atau $y[i+3] > 0.45$. Sehingga kompleks QRS nya adalah $y[i] > 0.45$ dan salah satu dari ketiga persamaan sebelumnya.

2.7.3 First and Second Derivative Algorithm

Algoritma *QRS enhancement* menghitung derivatif pertama dan kedua dari sinyal ECG secara independen. Sebuah kombinasi linear dari besaran derivatif ini kemudian digunakan untuk menegaskan daerah kompleks QRS relatif terhadap fitur ECG lainnya.

2.7.3.1 Balda

Algoritma ini merupakan penyederhanaan skema pendeteksian QRS yang disajikan oleh Balda [26]. Nilai absolut turunan pertama dan kedua dihitung dari ECG seperti pada Persamaan 2.16 dan 2.17

$$Y_0(n) = ABS[X(n+1) - X(n-1)] \qquad 2 < n < N-2 \quad (2.16)$$

$$Y_1(n) = ABS[X(n+2) - 2X(n) + X(n-2)] \qquad 2 < n < N-2 \quad (2.17)$$

Kedua persamaan ini kemudian membentuk kombinasi linear dari kedua derivatif seperti pada Persamaan 2.18

$$Y_2(n) = 1.3Y_0(n) + 1.1Y_1(n) \qquad 2 < n < N-2 \quad (2.18)$$

Persamaan di atas di-scan hingga memenuhi atau melampaui threshold sehingga seperti pada Persamaan 2.19

$$Y_2(i) \geq 1 \quad (2.19)$$

Setelah ini terjadi, delapan poin berikutnya dibandingkan dengan threshold. Jika enam atau lebih dari delapan poin ini memenuhi atau melampaui threshold, kriteria untuk identifikasi kandidat QRS terpenuhi.

2.7.3.2 Ahlstrom and Tompkins

Ahlstrom dan Tompkins [27] menghitung derivatif pertama yang diperbaiki dari ECG seperti pada Persamaan 2.20

$$Y_0(n) = ABS[X(n+1) - X(n-1)] \qquad 3 < n < N-3 \quad (2.20)$$

Derivatif pertama yang diperbaiki kemudian dihaluskan seperti pada Persamaan 2.21

$$Y_1(n) = \frac{[Y_0(n-1) + 2Y_0(n) + Y_0(n+1)]}{4} \qquad 3 < n < N-3 \qquad (2.21)$$

Turunan kedua yang diperbaiki kemudian dihitung seperti pada Persamaan 2.22

$$Y_2(n) = ABS[X(n+2) - 2X(n) + X(n-2)] \qquad 3 < n < N-3 \quad (2.22)$$

Kemudian hasil derivatif pertama yang diperbaiki dan dihaluskan (Persamaan 2.21) ditambahkan ke turunan kedua yang diperbaiki (Persamaan 2.22) sehingga seperti pada Persamaan 2.23

$$Y_3(n) = Y_1(n) + Y_2(n) \qquad 3 < n < N-3 \qquad (2.23)$$

Nilai maksimum array di atas ditentukan dan diskalakan sehingga berfungsi sebagai *primary threshold* dan *secondary threshold*:

$$Primary\ Threshold = 0.8 \max [Y_3(n)] \qquad 3 < n < N-3 \quad (2.24)$$

$$Secondary\ Threshold = 0.1 \max [Y_3(n)] \qquad 3 < n < N-3 \quad (2.25)$$

Array yang disimpulkan pada derivatif pertama dan kedua dipindai sampai titik melebihi *primary threshold*. Agar dapat diklasifikasikan sebagai kandidat QRS, enam poin berturut-turut harus memenuhi atau melampaui *secondary threshold*:

$$Y_3(i) \geq Primary\ Threshold \qquad (2.26)$$

$$Y_3(i+1), Y_3(i+2), \dots, Y_3(i+6) > Secondary\ Threshold \qquad (2.27)$$

2.7.4 Digital Filter Algorithm

2.7.4.1 Engelese and Zeelenberg

Algoritma yang diusulkan oleh Engelese dan Zeelenberg [28] dimulai dengan sinyal ECG yang dilewati melalui diferensiator seperti pada Persamaan 2.28

$$Y(n) = X(n) - X(n-4), \quad 4 < n < N-1 \qquad (2.28)$$

Kemudian sinyal difilter menggunakan digital low-pass filter

$$Y_1(n) = Y_0(n) + 4Y_0(n-1) + 6Y_0(n-2) + 4Y_0(n-3) + Y_0(n-4) \quad (2.29)$$

Dua *threshold* yang digunakan, sama besarnya tapi berlawanan dalam polaritas. Output *Low-Pass filter* dipindai sampai titik dengan amplitudo lebih besar dari *threshold* positif tercapai. Titik ini adalah onset 160 ms daerah pencarian. Jumlah *threshold crossing* alternatif digunakan untuk mengklasifikasikan persimpangan awal sebagai pergeseran *baseline*, calon QRS, atau sebagai *noise*.

Jika $Y_1(i) > 21$ kemudian cari *region onset* = i. Jika tidak ada penyeberangan ambang lain yang terjadi dalam wilayah pencarian 160 ms, kejadian diklasifikasikan sebagai pergeseran baseline. Jika tidak, berikut tiga kondisi yang diuji:

Kondisi 1 : If $Y_1(i+j) < -21 \quad 0 < j < 40$ (2.30)

Kondisi 2 : If $Y_1(i+j) < -21 \quad 0 < j < 40$ (2.31)

And $Y_1(1+k) > 21 \quad j < k < 40$

Kondisi 3 : If $Y_1(i+j) < -21 \quad 0 < j < 40$ (2.32)

And $Y_1(1+k) > 21 \quad j < k < 40$

And $Y_1(i+1) < -21 \quad k < i < 40$

Jika salah satu ketentuan di atas berlaku, kejadian diklasifikasikan sebagai kandidat QRS. Jika terjadi penambahan *threshold crossing*, kejadian diklasifikasikan sebagai *noise*.

2.7.4.2 Okada

Pada penelitian yang dilakukan Okada untuk mendeteksi kompleks QRS, Ia menggunakan *digital filter algorithm* sebagai teknik *QRS enhancement* [29]. Algoritma ini dimulai menghaluskan sinyal ECG dengan *menggunakan three-point moving average filter*

$$Y_0(n) = \frac{[X(n-1) + 2X(n) + X(n+1)]}{4} \quad (2.33)$$

Untuk $1 < n < N - 1$. Output dari MAV ini akan difilter menggunakan *low-pass filter*

$$Y_1(n) = \left[\frac{1}{(2m+1)}\right] \sum_{k=n-m}^{n+m} Y_0(k) \quad (2.34)$$

Untuk $m < n < N - m$. Kemudian perbedaan antara input dan output dari Low-Pass filter dikuadratkan:

$$Y_2(n) = (Y_0(n) - Y_1(n))^2 \quad (2.35)$$

Untuk $m < n < N - m$. Lalu hasil kuadrat tadi (Y_2) difilter

$$Y_3(n) = Y_2(n)\left[\sum_{k=n-m}^{n+m} Y_2(k)\right]^2 \quad (2.36)$$

Kemudian untuk nilai Y_4 didapatkan dengan menggunakan rumus,

Jika $\left[\left(Y_0(n) - Y_0(n-m)\right)\right] > 0$ dan $\left[\left(Y_0(n) - Y_0(n+m)\right)\right] > 0$ (2.37)

Maka nilai $Y_4(n) = Y_3(n)$, selain itu maka nilai $Y_4(n) = 0$.

Nilai maksimum dari array ini ditentukan dan diskalakan untuk membentuk *threshold*:

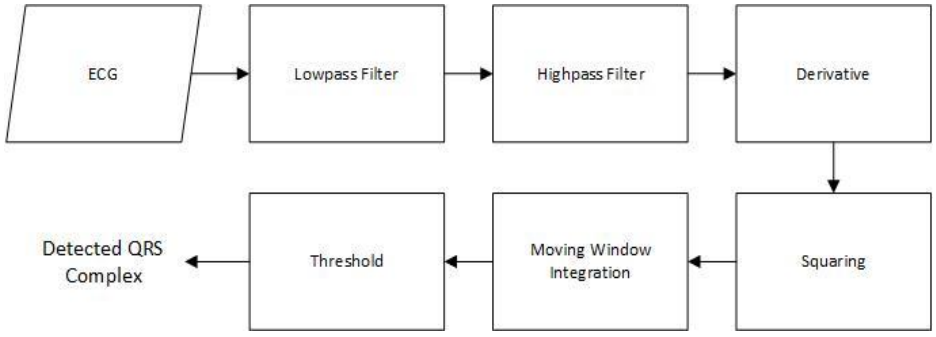
$$Threshold = 0.125 \max\left[Y_4(n)\right], \quad m < n < N - m \quad (2.38)$$

Kandidat QRS terjadi ketika titik di $Y_4(n)$ melebihi *threshold*. Variabel m diusulkan bernilai enam untuk hasil yang baik dan menjaga persyaratan komputasi pada tingkat yang wajar.

2.7.4.3 Pan-Tompkins Algorithm

Algoritma ini merupakan algoritma yang paling umum digunakan untuk mendeteksi kompleks QRS. Algorima yang diusulkan oleh Jiapu Pan dan Willis J. Tompkins pada tahun 1985 ini menggunakan 2 tahap yaitu proses pre-processing dan tahap decision rules [30]. Secara gambaran besarnya dapat dilihat pada Gambar 2.20 yang merupakan block diagram dari algoritma ini. Proses pre-processing meliputi proses low-pass filter hingga moving window integration. Kemudian untuk proses decision rules nya ada pada tahap threshold di Gambar 2.20.

Sebagai langkah awal, filter band-Pass diterapkan untuk meningkatkan rasio SNR. Sebuah filter bandwidth 5-15 Hz dianjurkan untuk memaksimalkan kontribusi QRS dan mengurangi *muscle noise*, *baseline wander*, *interferance powerline* dan gelombang P/T. Pada algoritma ini, filter band-Pass diperoleh dengan Low-Pass filter dan High-Pass filter yang di-cascade untuk mengurangi biaya komputasi dan memungkinkan deteksi real-time, sambil memastikan passband 3 dB dalam kisaran frekuensi 5-12 Hz, sehingga cukup mendekati



Gambar 2.20 *Block Diagram* dari Pan-Tompkins Algorithm

dengan tujuan desain. Transfer function untuk low pass filter yang diusulkan oleh Pan Tompkins dengan frekuensi sampling sebesar 200Hz adalah sebagai berikut

$$H(z) = \frac{(1 - z^{-6})^2}{(1 - z^{-1})^2} \tag{2.39}$$

Persamaan *transfer function* untuk *low-pass filter* di atas jika dibuat persamaan filternya menjadi seperti Persamaan 2.40

$$Y(n) = 2Y(n-1) - Y(n-2) + X(n) - 2X(n-6) + X(n-12) \tag{2.40}$$

Dimana *cut-off frequency* yang digunakan 11 Hz dengan *gain* sebesar 36 dan *processing delay* sebesar 6 sample. Kemudian untuk *transfer function* pada *high-pass filter* yaitu seperti Persamaan 2.41

$$H(z) = \frac{(-1 + 32z^{-16} + z^{-32})}{(1 + z^{-1})} \tag{2.41}$$

Persamaan *transfer function* untuk *high-pass filter* di atas jika dibuat persamaan filternya menjadi seperti Persamaan 2.42

$$Y(n) = 32X(n-16) - [Y(n-1) + X(n) - X(n-32)] \tag{2.42}$$

Dimana *cut-off frequency* yang digunakan 5 Hz dengan *gain* sebesar 32 dan *delay* sebesar 16 sample. Kemudian sinyal diderivatifkan untuk mengetahui lebih jelas informasi mengenai slope dari QRS complex. Pada sampling sinyal pada 200Hz, Pan Tompkins menggunakan transfer function seperti Persamaan 2.43

$$H(z) = \frac{1}{8}(-z^{-2} - 2z^{-1} + 2z^{1} + z^{2}) \tag{2.43}$$

Persamaan *transfer function* untuk *derivative* di atas jika dibuat persamaan filternya menjadi seperti Persamaan 2.44

$$Y(n) = \frac{1}{8}[-X(n-2) - 2X(n-1) + 2X(n+1) + X(n+2)] \tag{2.44}$$

Setelah sinyal diderivatifkan, maka sinyal akan dikuadratkan untuk meningkatkan puncak dominan (QRS) dan mengurangi kemungkinan keliru mengenali gelombang T sebagai R peak dengan Persamaan 2.45

$$Y(n) = [X(n)]^2 \tag{2.45}$$

Kemudian, filter Moving Average diterapkan untuk memberikan informasi tentang durasi dari kompleks QRS. Jumlah sampel untuk rata-rata dipilih dalam rangka untuk rata-rata *window* 150 ms sinyal yang diperoleh disebut *integrated signal*.

$$Y(n) = \frac{1}{N}[X(n-(N-1) + X(n-(N-2) + \cdots + X(n)] \tag{2.46}$$

Setelah ini sinyal masuk ke tahap decision rule, dimana dalam tahap ini akan diberlakukan thresholding.

2.8 Parameter yang dianalisa

Berdasarkan metode-metode yang telah dipakai tersebut maka didapatkanlah perbedaan hasil dengan membandingkan parameter-parameter deteksi.

2.8.1 Sensitivity (Se)

Parameter *Sensitivity* (Se) merupakan rasio prediksi benar positif dibandingkan dengan keseluruhan data yang benar positif sehingga dapat diketahui seberapa sering kompleks QRS terlewati (tidak terdeteksi). Parameter tersebut dapat dihitung dengan menggunakan Persamaan 2.47 sebagai berikut :

$$Se = \frac{TP}{TP + FN} \times 100\% \qquad (2.47)$$

Dimana TP merupakan jumlah kompleks QRS yang benar terdeteksi dan FN merupakan jumlah *false negative* (kompleks QRS yang tidak terdeteksi).

2.8.2 Positive Predictivity (+P)

Parameter *Positive Predictivity* (+P) merupakan rasio prediksi benar positif dibandingkan dengan keseluruhan hasil yang diprediksi positif sehingga dapat diketahui seberapa sering kompleks QRS yang salah terdeteksi. Seperti pada Persamaan 2.48 berikut :

$$+P = \frac{TP}{TP + FP} \times 100\% \qquad (2.48)$$

Dimana TP merupakan jumlah kompleks QRS yang benar terdeteksi dan FP merupakan jumlah *false positive* (kompleks QRS yang salah terdeteksi).

2.8.3 F1 Measure (F1)

F1 Measure (F1) merupakan perbandingan rata-rata positive predictivity dan sensitivity yang dibobotkan. Sehingga dapat dikatakan jika semakin besar nilainya maka akan semakin baik. Semakin besar nilai *F1 Measure* maka algoritma tersebut makin bagus dalam pendeteksian *heart beat* (semakin banyak titik yang terdeteksi benar-benar QRS dan lebih sedikit titik yang salah terdeteksi ataupun terlewati saat deteksi). *F1 Measure* dapat dihitung dangan Persamaan 2.49 sebagai berikut :

$$F1 = \frac{2 \times TP}{(2 \times TP + FP + FN)} \times 100\% \qquad (2.49)$$

Dimana TP merupakan jumlah kompleks QRS yang benar terdeteksi, FP merupakan jumlah *false positive* (kompleks QRS yang salah terdeteksi), dan FN merupakan jumlah *false negative* (kompleks QRS yang tidak terdeteksi).

2.9 Rasional

Pada penyakit *cardiovascular*, terdapat kondisi yang dinamakan aritmia, dimana aritmia mengacu pada bentuk denyut dan irama jantung yang tidak normal. Karena sifat aritmia yang *asymptomatic* maka diperlukan pemantauan dengan menggunakan ECG *portable*. Penelitian ini akan mendesain sistem ECG *portable* menggunakan mikrokontroler. Fokusan pada penelitian ini ada pada pemrosesan sinyal digital tertanam. Penelitian ini akan menggunakan 10 metode deteksi QRS yang akan diimplementasikan pada mikrokontroler dan hasilnya akan dibandingkan untuk mengetahui metode mana yang bagus jika diimplementasikan pada mikrokontroler.

Metode-metode yang digunakan merupakan metode-metode yang tingkat komputasinya relative rendah karena metode-metode ini akan diimplentasikan pada mikrokontroler yang memiliki keterbatasan sumber daya dan waktu dengan memori yang juga terbatas. Metode-metode ini dibagi menjadi 4 kelas besar yaitu *amplitude and first derivative algorithm*, *first derivative algorithm*, *first and second derivative algortihm*, dan *digital filter algorithm*. Keempat kelas besar ini perbedaan mendasarnya ada pada proses *pre-processing*. Pada proses ini akan didapatkan sinyal ECG yang *slope* QRS lebih jelas.

Algoritma amplitude and first derivative merupakan algoritma yang relative sederhana dan berisi threshold fix pada amplitude thresholdnya. Pada algoritma ini, proses *pre-processing*nya akan dimulai dengan membatasi sinyal dengan menggunakan amplitude threshold, kemudian diikuti dengan proses first derivative. Proses first derivative berfungsi untuk mengurangi pengaruh gelombang P dan T terhadap puncak R sehingga dapat menonjolkan slope QRS. Noise pada algoritma ini tidak dihapus dengan benar dan tidak dipertimbangkan oleh algoritma first derivative untuk ekstraksi fitur. Algoritma-algoritma yang masuk pada kelas amplitude and first derivative algorithm yaitu Moriet-Mahoudeux algorithm, Fraden and Neuman algorithm, dan Gustafson algorithm.

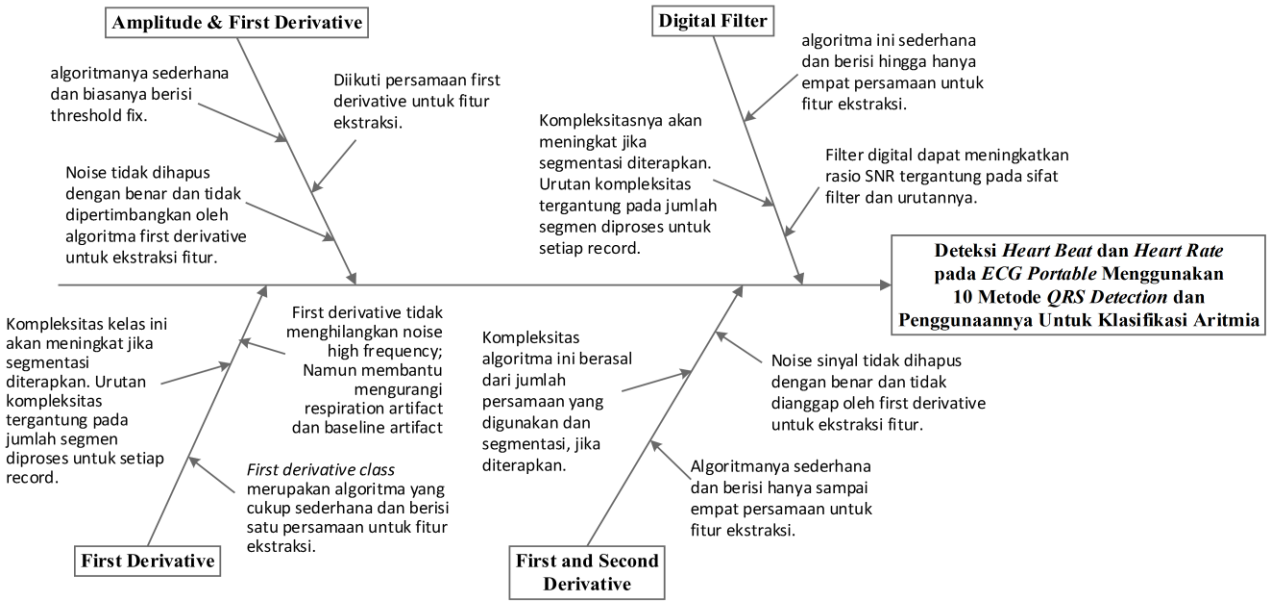
Pada kelas besar selanjutnya yaitu first derivative algorithm. First derivative class merupakan algoritma yang cukup sederhana dan berisi satu persamaan untuk fitur ekstraksi. Kompleksitas kelas ini akan meningkat jika segmentasi diterapkan. Urutan kompleksitas tergantung pada jumlah segmen diproses untuk setiap record. First derivative tidak menghilangkan noise high frequency namun membantu mengurangi respiration artifact dan baseline artifact. Algoritma-algoritma yang masuk pada kelas first derivative algorithm yaitu Menard algorithm dan Holsinger algorithm.

Kelas algoritma selanjutnya yaitu first and second derivative algorithm. Algoritma ini tetap menggunakan first derivative pada proses pre-processingnya kemudian diikuti dengan second derivative. Algoritmanya sederhana dan berisi hanya sampai empat persamaan untuk fitur ekstraksi. Kompleksitas algoritma ini berasal dari jumlah persamaan yang digunakan dan segmentasi, jika diterapkan. Noise sinyal tidak dihapus dengan benar dan tidak dianggap oleh first derivative

untuk ekstraksi fitur. Algoritma-algoritma yang masuk pada kelas first and second derivative algorithm yaitu Balda algorithm dan Ahlstrom and Tompkins algorithm.

Dan kelas algoritma yang terakhir yaitu digital filter algorithm. Pada algoritma ini, proses pre-processingnya menggunakan digital filter untuk menghilangkan noise terutama noise frekuensi tinggi yang tidak disediakan pada kelas-kelas algoritma sebelumnya. Filter digital yang dilakukan dapat meningkatkan rasio SNR tergantung pada sifat filter dan urutannya. Algoritma-algoritma yang masuk pada kelas digital filter algorithm yaitu Engelese and Zeelenberg algorithm, Okada algorithm, dan Pan-Tompkins algorithm.

Pada penelitian ini kesepuluh algoritma tadi akan diimplementasikan dalam sistem mikrokontroler untuk membuat ECG portable. Output dari sistem ini berupa heart rate dan heart beat yang akan dibandingkan mana metode yang bagus jika diimplementasikan dalam mikrokontroler dan penggunaannya untuk deteksi aritmia.



Gambar 2.21 Diagram *Fishbone*

BAB 3 PERANCANGAN SISTEM

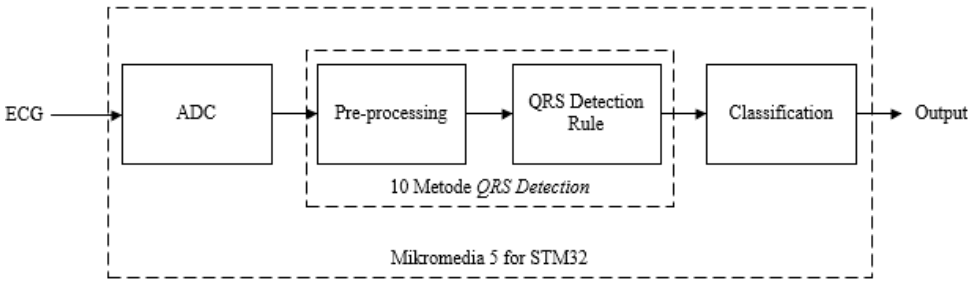
Secara garis besar, kerangka dari Tugas Akhir ini dapat dibagi menjadi 2 bagian yaitu perancangan *hardware* dan *software*. Dimana *software* sistem ini akan berfokus pada pengolahan sinyal digital tertanam pada mikrokontroler menggunakan 10 metode *QRS detection*.

3.1 Desain Sistem Secara Umum

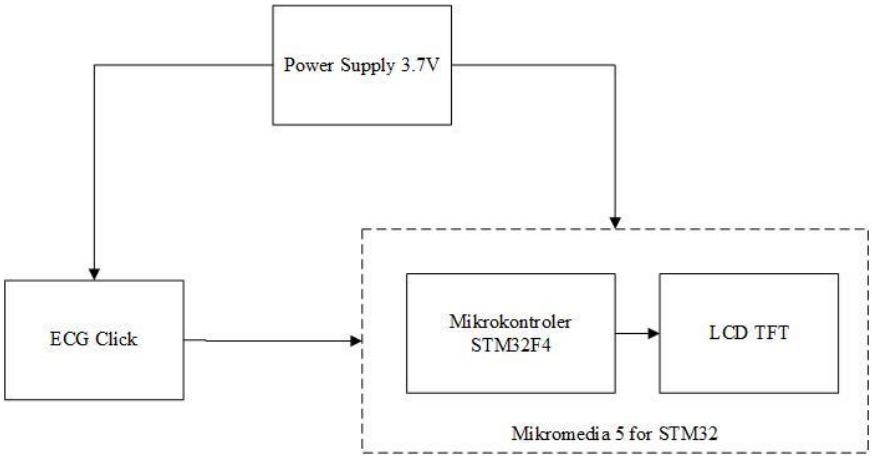
Desain sistem yang digunakan secara umum dijelaskan berdasarkan diagram blok seperti yang telah tertera pada Gambar 3.1. Berdasarkan diagram blok pada Gambar 3.1, dapat dilihat fokusan penelitian ini ada pada proses pengolahan sinyal digital tertanam pada mikrokontroler. Dimana pada penelitian ini diajukan 10 metode *QRS detection* agar dapat dilihat perbedaan penggunaan metode-metode ini dalam membantu pendeteksian penyakit aritmia. Pertama, sinyal ECG yang didapatkan masuk ke ADC mikrokontroler, kemudian dilakukan proses pengolahan sinyal digital dengan 10 metode *QRS detection* yang terdiri dari 2 bagian utama yaitu *pre-processing* sinyal dan *QRS Detection Rule*. Keluaran dari metode-metode ini akan diklasifikasikan sesuai parameter yang digunakan sehingga menghasilkan input apakah sinyal masukan tersebut mengalami aritmia atau tidak.

3.2 Perancangan *Hardware* Sistem

Sistem ini menggunakan beberapa hardware untuk mendukungnya. Komponen utamanya yaitu modul Mikromedia 5 for STM32 Capacitive dimana dalam modul ini menggunakan mikrokontroler STM32F407ZGT6 serta dilengkapi juga dengan LCD TFT yang memiliki resolusi sebesar 800x480 px. Modul ini yang akan menjadi tempat proses pengolahan sinyal digital sekaligus menampilkan hasilnya di layar LCD TFT. Kemudian digunakan juga modul ECG yaitu ECG Click yang digunakan untuk mengambil sinyal ECG dari tubuh secara langsung. ECG Click



Gambar 3.1 *Block Diagram* Sistem Keseluruhan



Gambar 3.2 *Block Diagram* Hardware Sistem

ini juga dilengkapi dengan 3 elektroda yang digunakan untuk mendapatkan sinyal listrik jantung. Ketiga elektroda ini akan digunakan di lengan kiri (LA), lengan kanan (RA), dan kaki kiri (LL). Untuk mendukung 2 komponen utama ini maka diperlukan *power supply* sebagai sumber daya kedua komponen ini. *Power supply* yang digunakan yaitu *li-polymer battery* 3.7V 2000mAh. Dengan begitu maka rancangan *hardware* pada sistem ini yaitu power supply akan memberikan daya kepada komponen-komponen terkait, dalam hal ini yaitu kedua modul Mikromedia 5 for STM32 dan modul ECG Click. Kemudian sistem bekerja dengan ECG Click yang akan mengambil informasi sinyal listrik jantung. *Output* pada ECG Click akan menjadi *input* pada modul Mikromedia 5 for STM32. Dimana karena *output*nya berupa sinyal analog maka untuk masuk dan diproses pada modul Mikromedia 5 for STM32 harus berupa sinyal digital diperlukan ADC (*Analog to Digital Converter*). ADC ini terdapat di dalam mikrokontroler sehingga sinyal akan langsung diproses oleh mikrokontroler setelah dikonversi menjadi sinyal digital. Kemudian hasil pemrosesan yang dihasilkan oleh mikrokontroler akan ditampilkan dengan GUI (*Graphical User Interface*) dengan menggunakan LCD TFT yang sudah terintegrasi di dalam modul Mikromedia 5 for STM32.

3.3 Perancangan Software Sistem

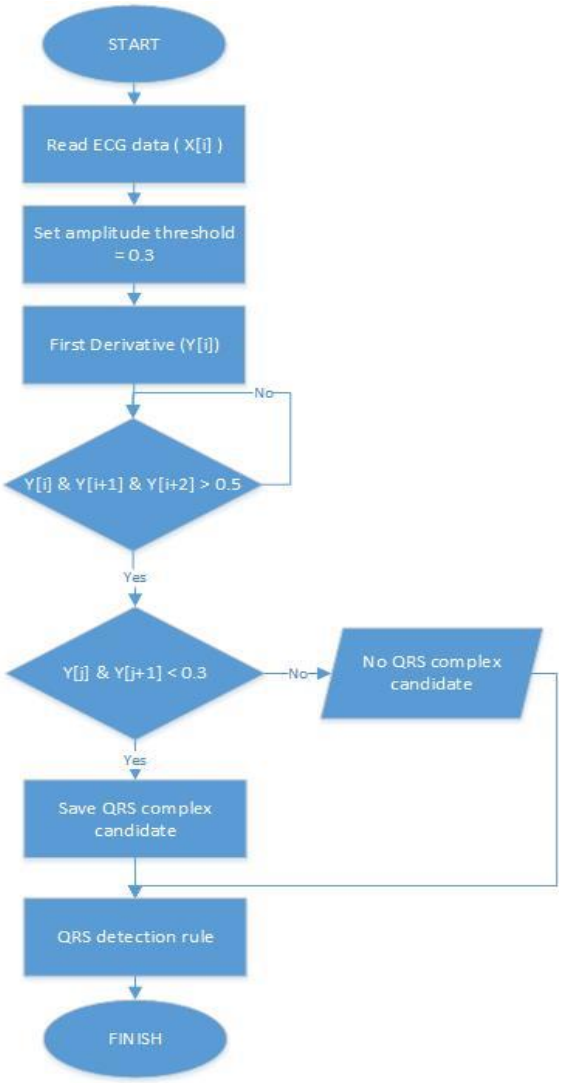
Proses merancang software pengolahan sinyal digital tertanam di mikrokontroler menggunakan *compiler* MikroC Pro for ARM. *Compiler* ini menggunakan bahasa C yang kemudian setelah *project* di-*build* maka akan dimasukkan ke dalam mikrokontroler. Untuk mengunggah *project* ke dalam mikrokontroler dibutuhkan *software* penunjang yaitu CODEGRIP. Fokusan perancangan *software* pada sistem ini yaitu pada sistem pengolahan sinyal digital menggunakan 10 metode *QRS detection*. Dimana seperti yang telah dijelaskan pada sub bab 3.1, ada 2 proses utama pada setiap metode ini yaitu proses *pre-processing* sinyal dan *QRS detection rule* untuk mendeteksi kompleks QRS yang nantinya informasi ini dapat digunakan untuk mengetahui berapakah *heart rate*

dari *input* sistem sehingga dapat digunakan dalam pendeteksian penyakit aritmia. Kemudian pada metode-metode ini juga digunakan beberapa besar frekuensi sampling tergantung metode apa yang digunakan. Hampir semua algoritma yang digunakan menggunakan frekuensi sampling sebesar 320 Hz, hanya algoritma Pan and Tompkins saja yang menggunakan frekuensi sampling sebesar 200 Hz.

3.3.1 *Amplitude & First Derivative Algorithm*

3.3.1.1 Moriet-Mahoudeux

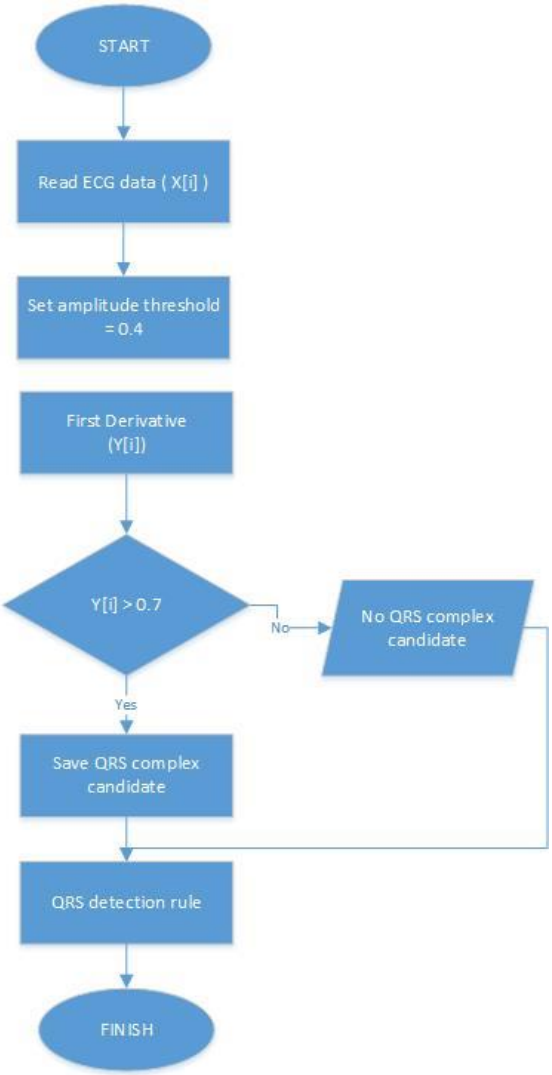
Dari teori yang telah dijelaskan sebelumnya pada Sub bab 2.5.1.1 maka dibuatlah *flow chart* agar memudahkan dalam membuat algoritma pemrogramannya. *Flow chart* pada algoritma Moriet-Mahoudeux dapat dilihat pada Gambar 3.3.



Gambar 3.3 *Flow chart* algoritma "Moriet-Mahoudeux"

3.3.1.2 Fraden and Neuman

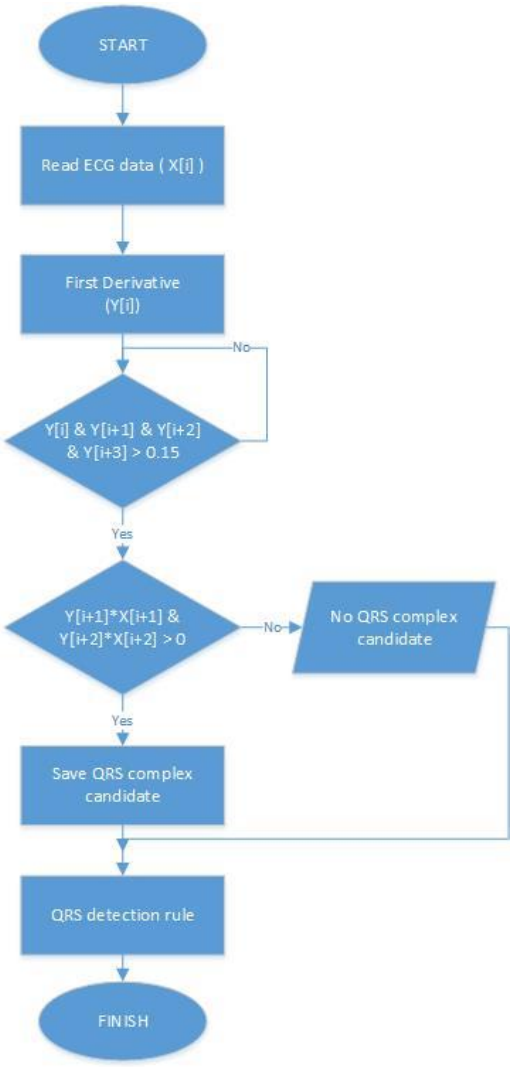
Dari teori yang telah dijelaskan sebelumnya pada Sub bab 2.5.1.2 maka dibuatlah *flow chart* agar memudahkan dalam membuat algoritma pemrogramannya. *Flow chart* pada algoritma Fraden and Neuman dapat dilihat pada Gambar 3.4.



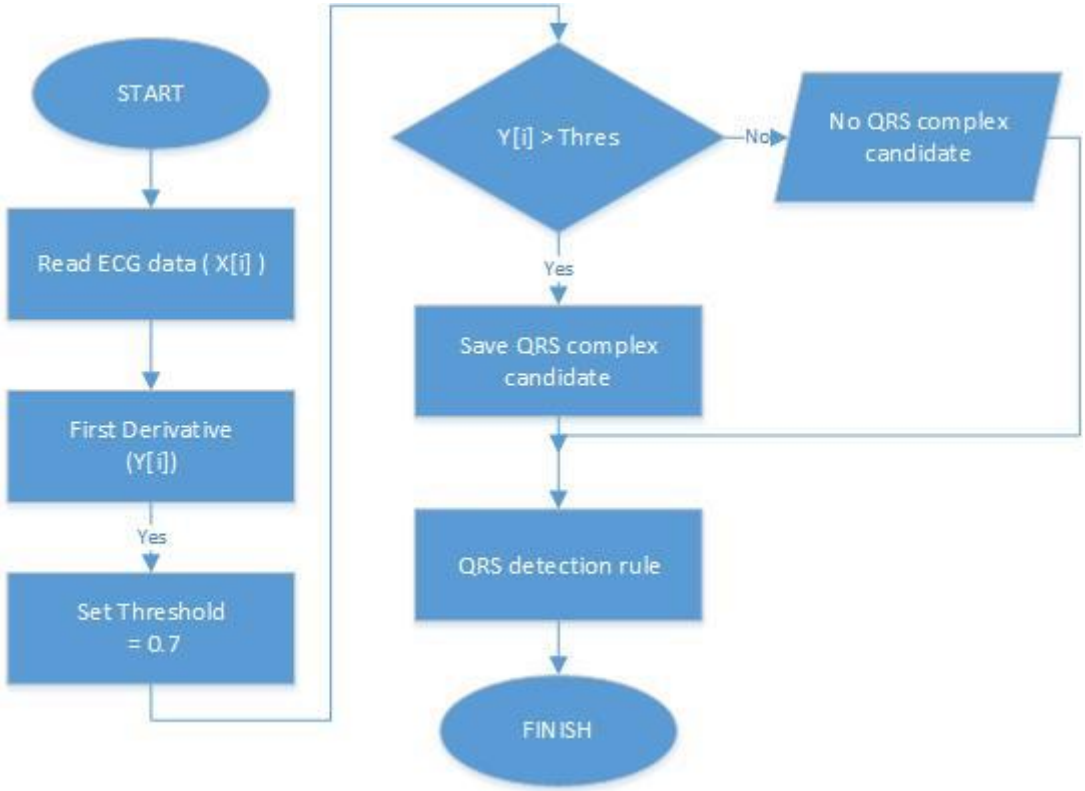
Gambar 3.4 *Flow chart* algoritma "Fraden and Neuman"

3.3.1.3 Gustafson

Dari teori yang telah dijelaskan sebelumnya pada Sub bab 2.5.1.3 maka dibuatlah *flow chart* agar memudahkan dalam membuat algoritma pemrogramannya. *Flow chart* pada algoritma Gustafson dapat dilihat pada Gambar 3.5.



Gambar 3.5 *Flow chart* algoritma "Gustafson"



Gambar 3.6 *Flow chart* algoritma "Menard"

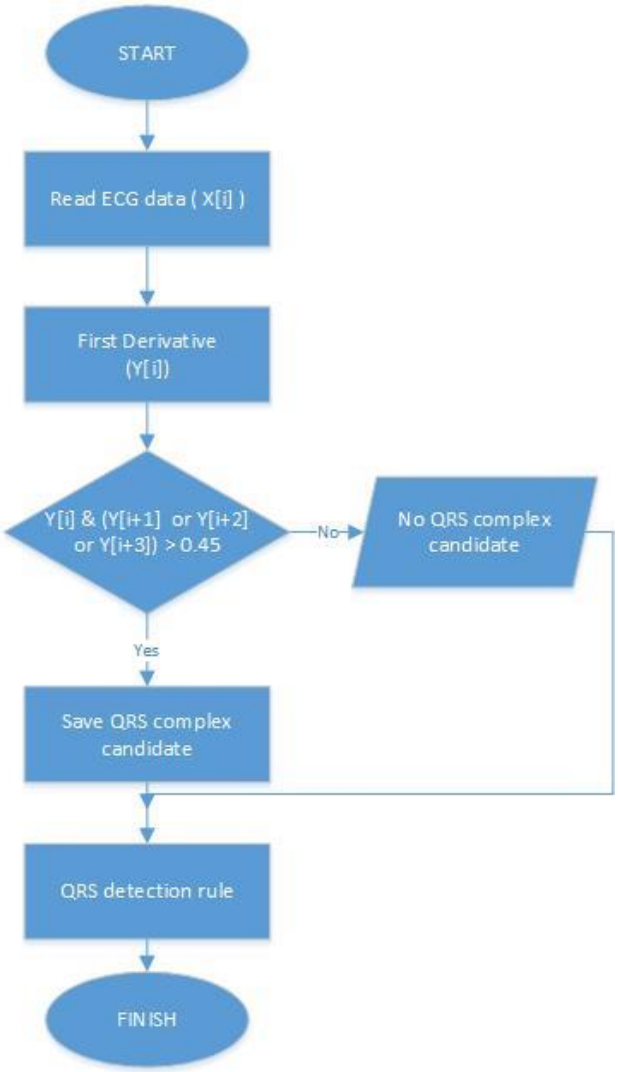
3.3.2 *First Derivative Algorithm*

3.3.2.1 Menard

Dari teori yang telah dijelaskan sebelumnya pada Sub bab 2.5.2.1 maka dibuatlah *flow chart* agar memudahkan dalam membuat algoritma pemrogramannya. *Flow chart* pada algoritma Menard dapat dilihat pada Gambar 3.6.

3.3.2.2 Holsinger

Dari teori yang telah dijelaskan sebelumnya pada Sub bab 2.5.2.2 maka dibuatlah *flow chart* agar memudahkan dalam membuat algoritma pemrogramannya. *Flow chart* pada algoritma Holsinger dapat dilihat pada Gambar 3.7.

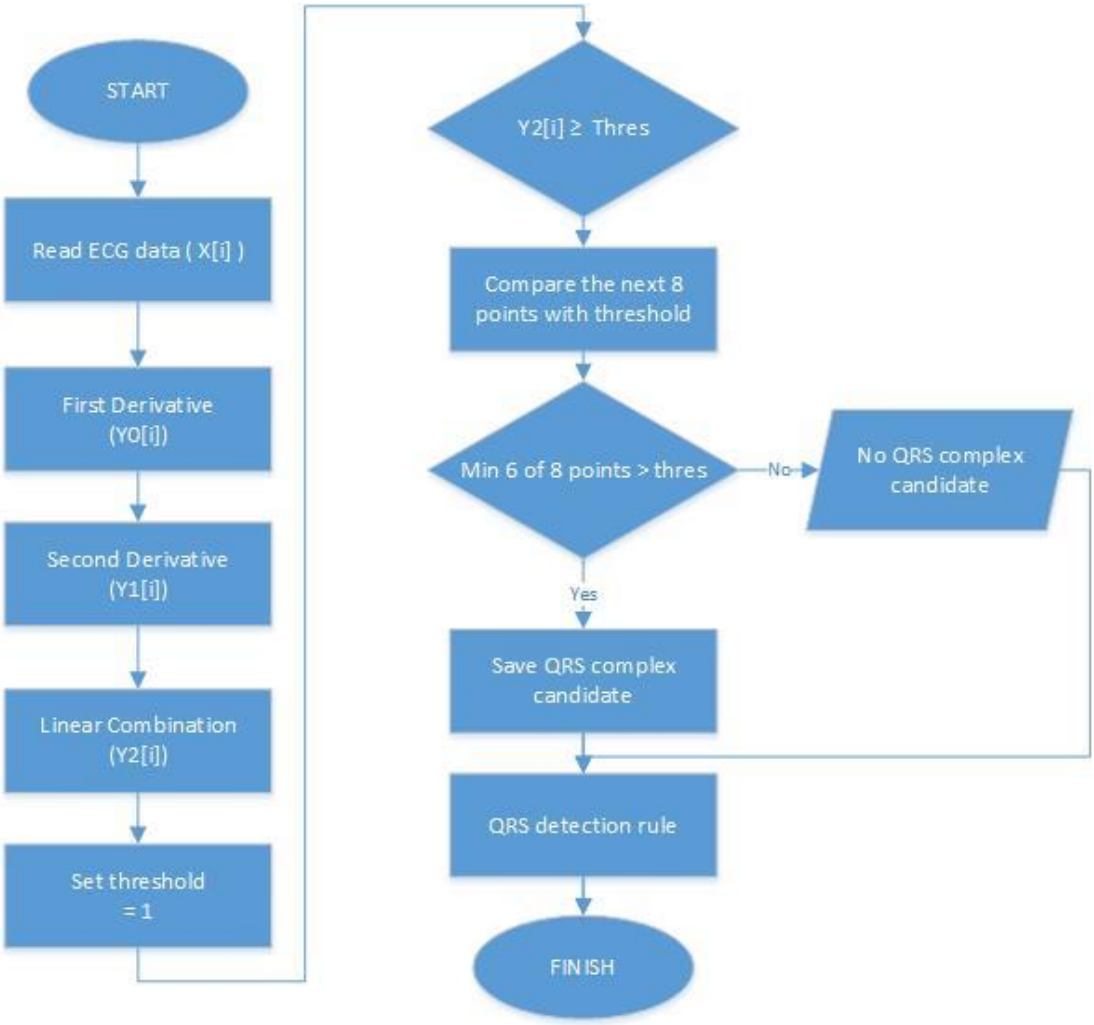


Gambar 3.7 *Flow chart* algoritma "Holsinger"

3.3.3 *First & Second Derivative Algorithm*

3.3.3.1 Balda

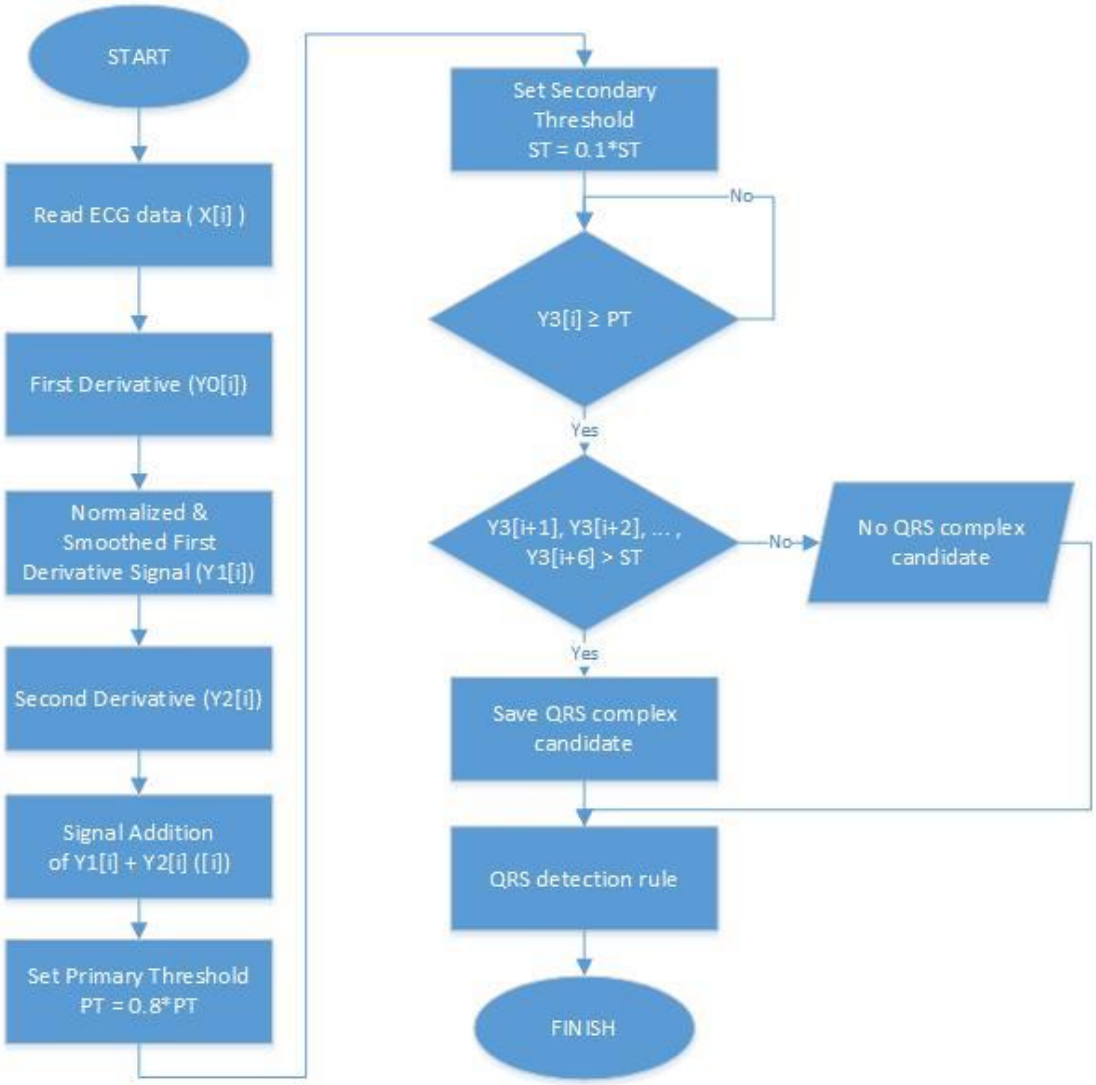
Dari teori yang telah dijelaskan sebelumnya pada Sub bab 2.5.3.1 maka dibuatlah *flow chart* agar memudahkan dalam membuat algoritma pemrogramannya. *Flow chart* pada algoritma Balda dapat dilihat pada Gambar 3.8.



Gambar 3.8 *Flow chart* algoritma "Balda"

3.3.3.2 Ahlstrom and Tompkins

Dari teori yang telah dijelaskan sebelumnya pada Sub bab 2.5.3.2 maka dibuatlah *flow chart* agar memudahkan dalam membuat algoritma pemrogramannya. *Flow chart* pada algoritma Ahlstrom and Tompkins dapat dilihat pada Gambar 3.9.

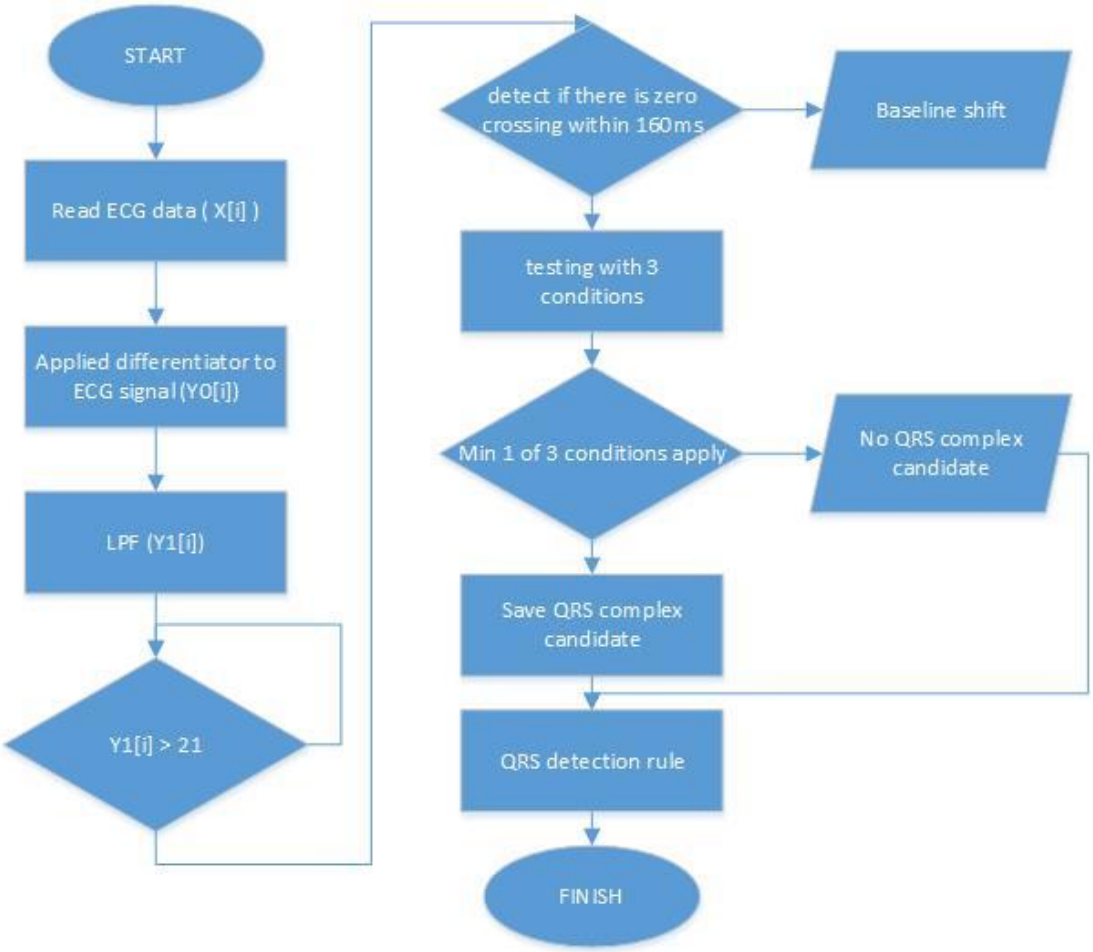


Gambar 3.9 *Flow chart* algoritma "Ahlstrom and Tompkins"

3.3.4 *Digital Filter Algorithm*

3.3.4.1 Engelese and Zeelenberg

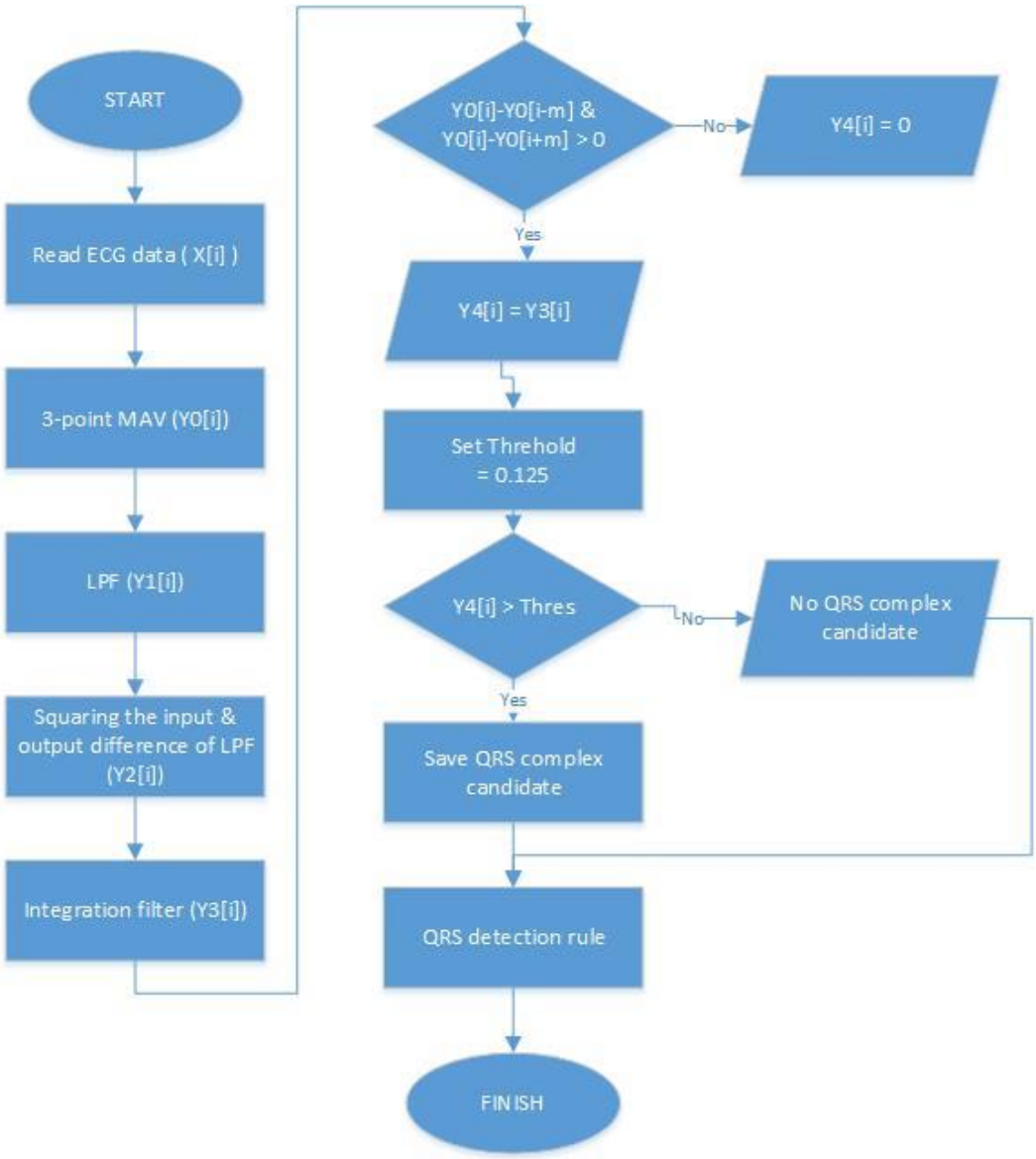
Dari teori yang telah dijelaskan sebelumnya pada Sub bab 2.5.4.1 maka dibuatlah *flow chart* agar memudahkan dalam membuat algoritma pemrogramannya. *Flow chart* pada algoritma Engelese and Zeelenberg dapat dilihat pada Gambar 3.10.



Gambar 3.10 *Flow chart* algoritma "Engelese and Zeelenberg"

3.3.4.2 Okada

Dari teori yang telah dijelaskan sebelumnya pada Sub bab 2.5.4.2 maka dibuatlah *flow chart* agar memudahkan dalam membuat algoritma pemrogramannya. *Flow chart* pada algoritma Okada dapat dilihat pada Gambar 3.11.



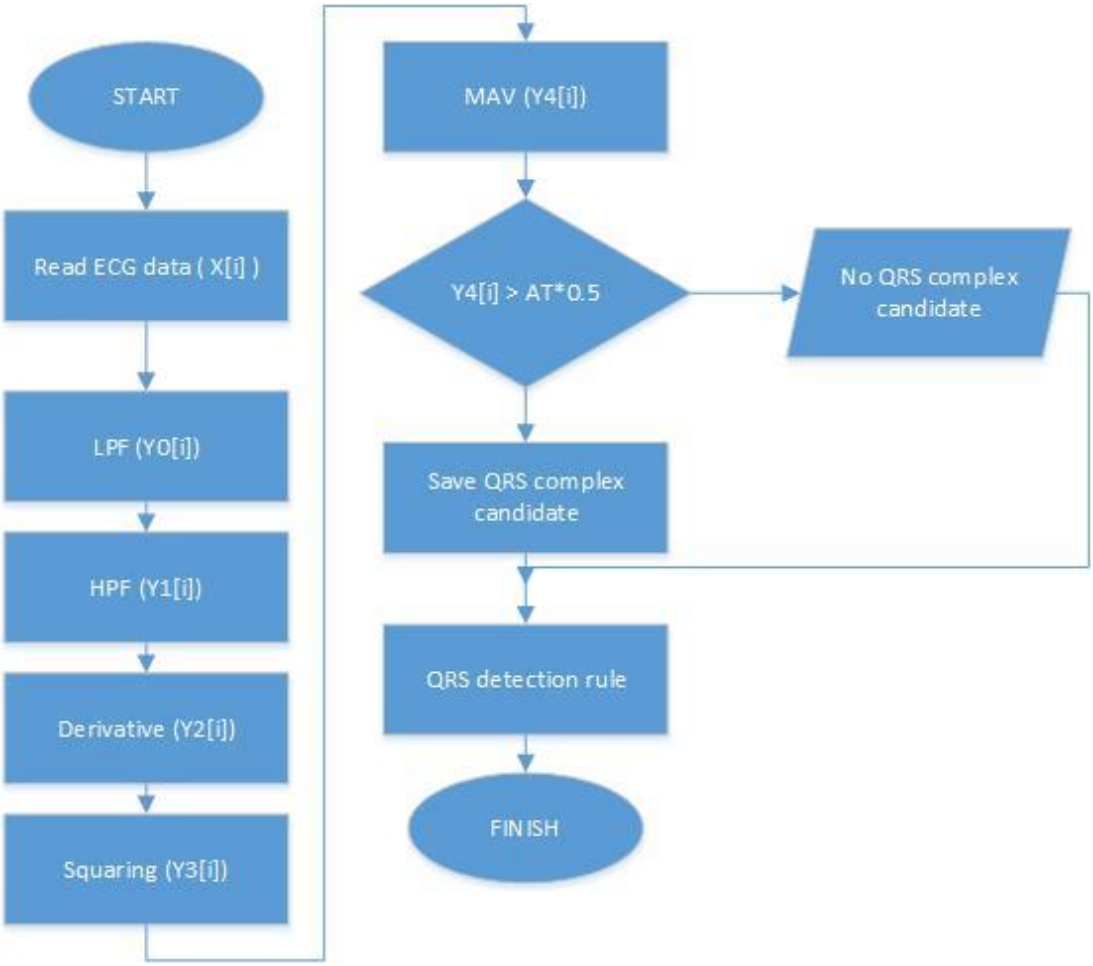
Gambar 3.11 *Flow chart* algoritma "Okada"

3.3.4.3 Pan Tompkins Algorithm

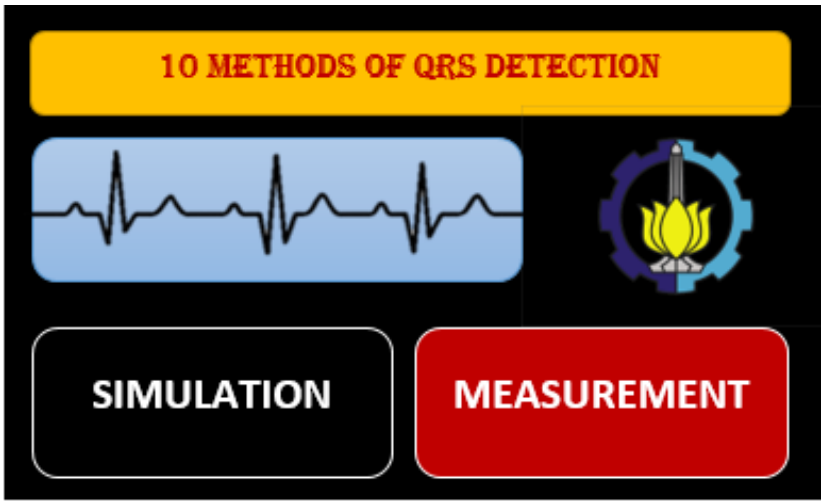
Pada metode Pan-Tompkin Algorithm, proses pre-processing sinyalnya terdiri dari low pass filter, high pass filter, derivative, squaring, dan moving average. Kemudian dilanjutkan dengan proses QRS detection rule yang akan dilakukan pada proses threshold. Awalnya, sinyal input ECG akan dimasukkan pada sistem, kemudian sinyal input ini akan difilter untuk menghilangkan noise dengan menggunakan low pass filter dengan cut-off frequency sebesar 11 Hz sehingga frekuensi dibawah 11 Hz akan dilewatkan dan frekuensi diatasnya akan dipotong.

Setelah itu dilakukan high pass filter dengan cut-off frequency sebesar 5 Hz sehingga frekuensi dibawah 5 Hz akan dipotong dan melewatkan frekuensi diatas 5 Hz. Selanjutnya dilakukan proses derivative untuk mempertajam slope sehingga lebih mudah untuk mendeteksi kompleks QRS nya. Dilanjutkan dengan proses squaring yang bertujuan untuk meningkatkan kompleks QRS serta mengurangi kemungkinan mengenali gelombang T sebagai puncak R. Kemudian, proses selanjutnya yaitu MAV (Moving Average Filter) yang berguna untuk memberikan informasi tentang durasi kompleks QRS. Jumlah sampel rata-rata dipilih untuk rata-rata pada windows 150 ms.

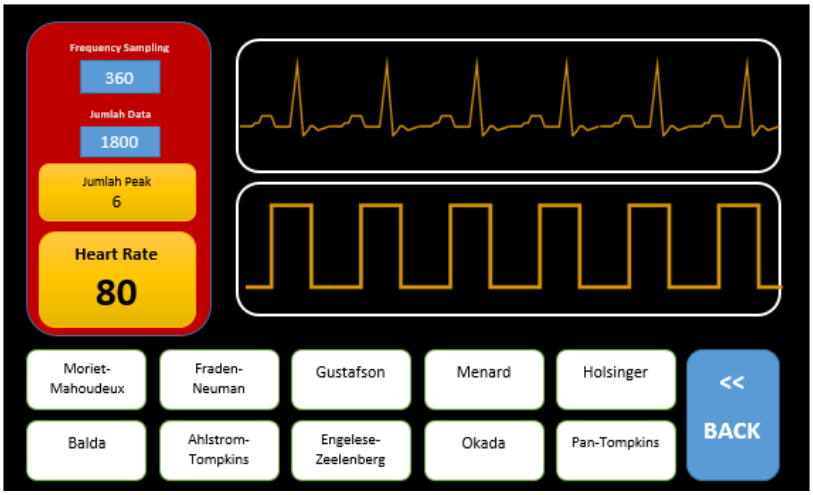
Dari teori yang telah dijelaskan sebelumnya pada Sub bab 2.5.4.3 maka dibuatlah *flow chart* agar memudahkan dalam membuat algoritma pemrogramannya. *Flow chart* pada algoritma Pan and Tompkins dapat dilihat pada Gambar 3.12.



Gambar 3.12 *Flow chart* algoritma "Pan and Tompkins"



(a)



(b)

Gambar 3.13 (a) Desain visualisasi output pada layar utama (b) Desain visualisasi output pada menu *simulation* dan *measurement*.

3.4 Desain Visualisasi *Output*

Hasil keluaran dari sistem akan ditampilkan pada layar LCD TFT yang juga terintegrasi dalam modul Mikromedia 5 for STM32F4. Output yang akan ditampilkan dapat dilihat pada Gambar 3.13. Untuk output setiap step dalam metode tidak ditampilkan dengan pertimbangan besar layar LCD yang terbatas.

3.5 Pengambilan Data

Data yang digunakan dalam penelitian ini menggunakan data yang diperoleh dengan menggunakan phantom ECG (simulator ECG) dan sinyal tubuh manusia. Kelebihan dalam menggunakan phantom ECG yaitu dapat diketahui berapa heart

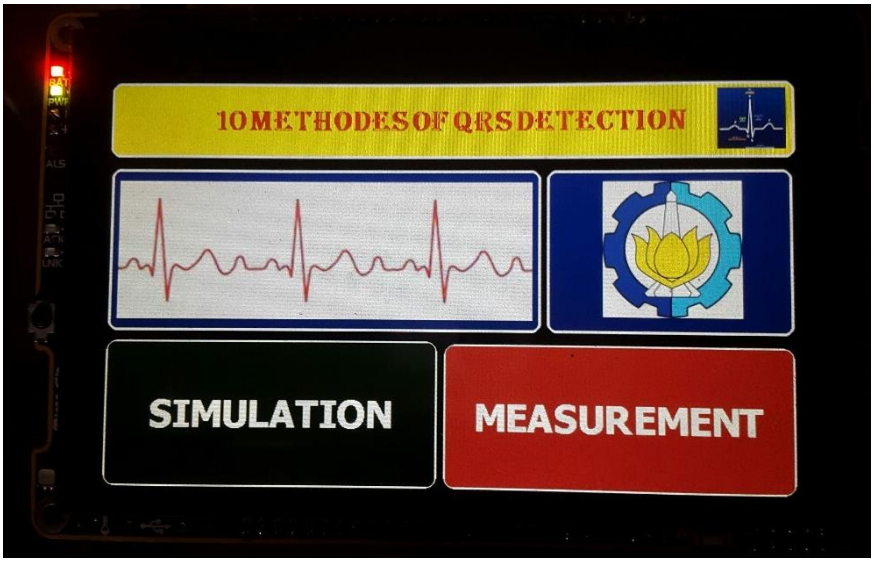
rate yang dihasilkan, kemudian bisa dilihat juga bentuk sinyalnya sehingga memudahkan dalam melakukan pengujian dibanding menggunakan data dari sinyal tubuh manusia. Dalam pengambilan data, dilakukan beberapa skenario pembacaan data. Pengambilan pertama menggunakan phamtom ECG dengan *heart rate* sebesar 40 bpm, pengambilan kedua menggunakan phantom ECG dengan *heart rate* sebesar 60 bpm, pengambilan ketiga menggunakan phantom ECG dengan *heart rate* sebesar 80 bpm, pengambilan keempat menggunakan phantom ECG dengan *heart rate* sebesar 100 bpm, dan pengambilan kelima menggunakan phantom ECG dengan *heart rate* sebesar 120 bpm. Kemudian sistem diuji juga dengan menggunakan sinyal yang telah diberi beberapa jenis noise. Macam-macam *noise* yang digunakan yaitu *inference 50Hz noise*, *muscle artifact noise, baseline artifact noise,* dan *respiration artifact noise*. Setelah itu dilakukan juga pengujian dengan menggunakan sinyal tubuh untuk melihat apakah sistem ini dapat memproses data dengan input sinyal tubuh manusia.

BAB 4 PENGUJIAN SISTEM

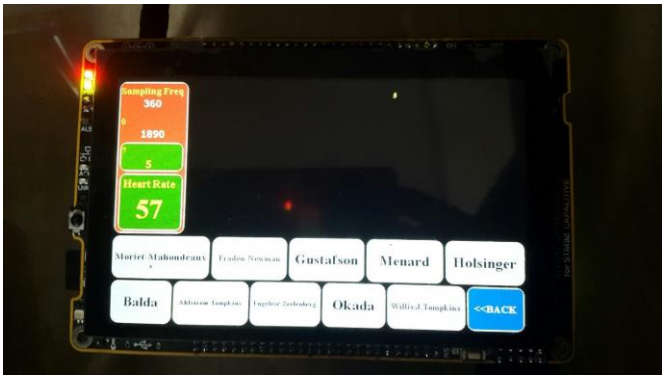
Untuk pengujian sistem pada mikrokontroler digunakan 2 menu, yaitu "*Simulation*" dan "*Measurement*". Perbedaan 2 menu ini ada pada sumber sinyal uji yang digunakan. Pada menu "*Simulation*", data yang digunakan dalam pengujian berupa data yang telah dimasukkan ke dalam mikrokontroler dalam bentuk *array*. Kemudian pada menu "*Measurement*" digunakan phantom ECG yang akan dilakukan beberapa pengaturan untuk menguji sistem dan digunakan juga sinyal uji yang berasal dari subjek manusia.

4.1 Pengujian dengan Menggunakan Simulasi

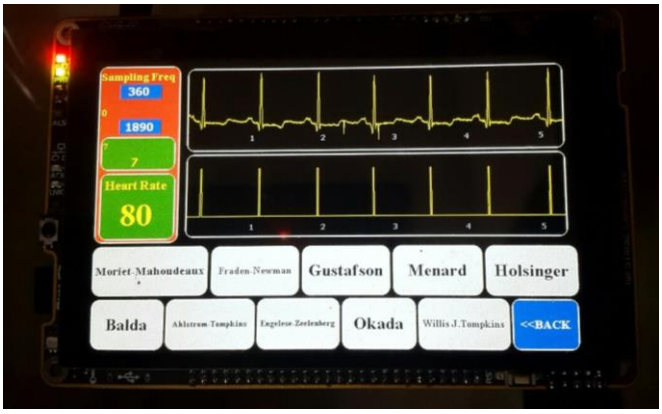
Pada pengujian dengan menggunakan simulasi, sinyal ECG yang digunakan langsung dimasukan ke dalam program sebagai array input sinyal ECG. Data yang digunakan berasal dari database MIT-BIH Normal Sinus Rhythm [31], dengan nilai heart rate sebesar 80 bpm. Kemudian data tersebut digunakan untuk simulasi setiap metode. Setelah itu didapatkan hasil pengujian yang dapat dilihat pada Gambar 4.1 sampai Gambar 4.12. Untuk lebih ringkasnya dapat dilihat pada Tabel 4.1 yang berisi hasil pengujian heart rate dan Tabel 4.2 yang berisi hasil pengujian heart beat.



Gambar 4.1 Interface Mikrokontroler



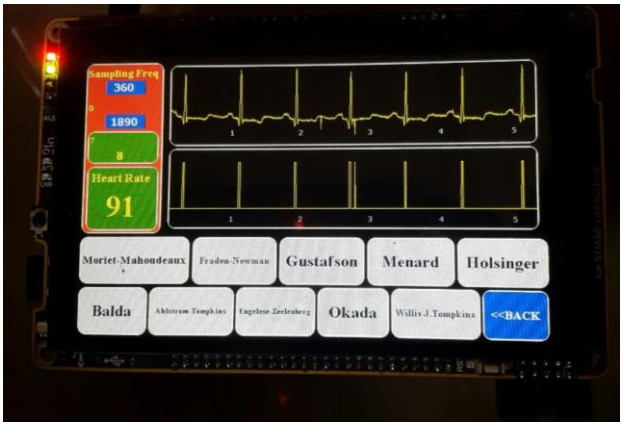
Gambar 4.2 Interface Mikrokontroler setelah ditekan menu “Simulation”



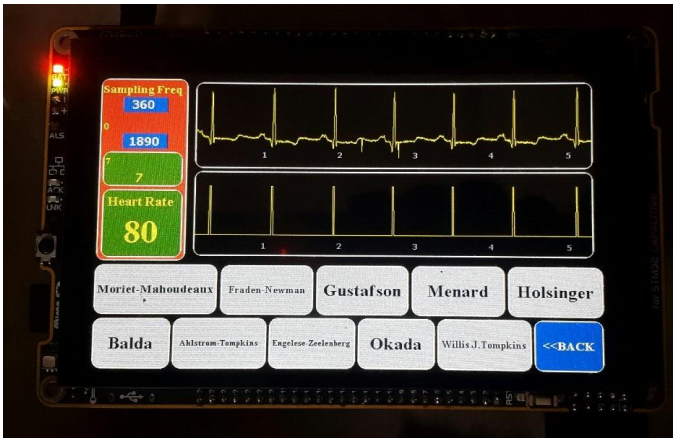
Gambar 4.3 Interface Mikrokontroler saat simulasi metode “Moriet-Mahoudeux” yang dipilih.

Gambar 4.4 Interface Mikrokontroler saat simulasi metode “Fraden and Neuman” yang dipilih.

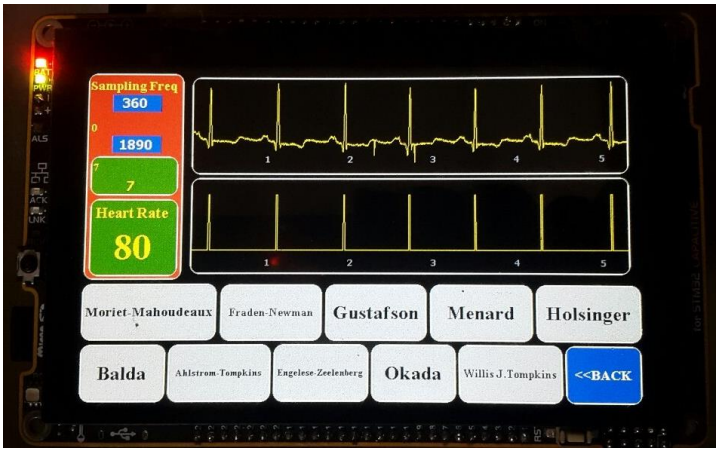
Gambar 4.5 Interface Mikrokontroler saat simulasi metode “Gustafson” yang dipilih.



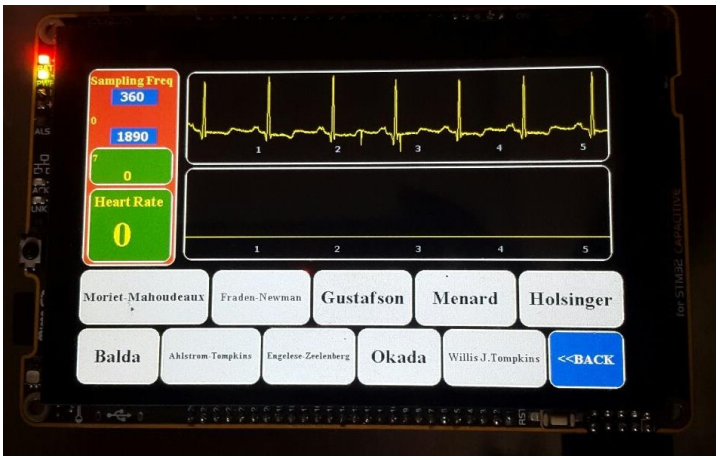
Gambar 4.6 Interface Mikrokontroler saat simulasi metode “Menard” yang dipilih.



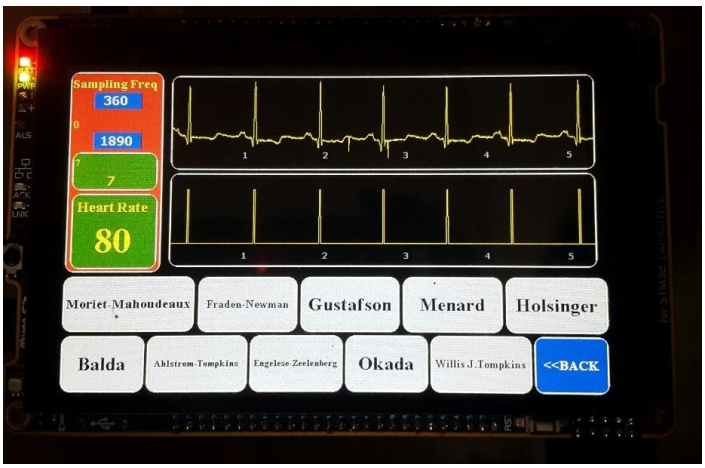
Gambar 4.7 Interface Mikrokontroler saat simulasi metode “Holsinger” yang dipilih.



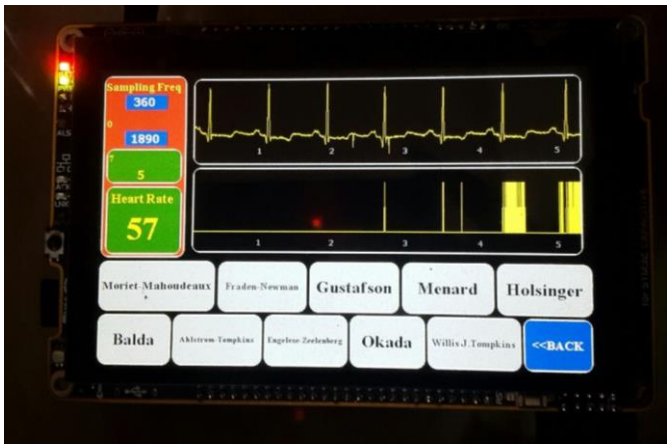
Gambar 4.8 Interface Mikrokontroler saat simulasi metode "Balda" yang dipilih.



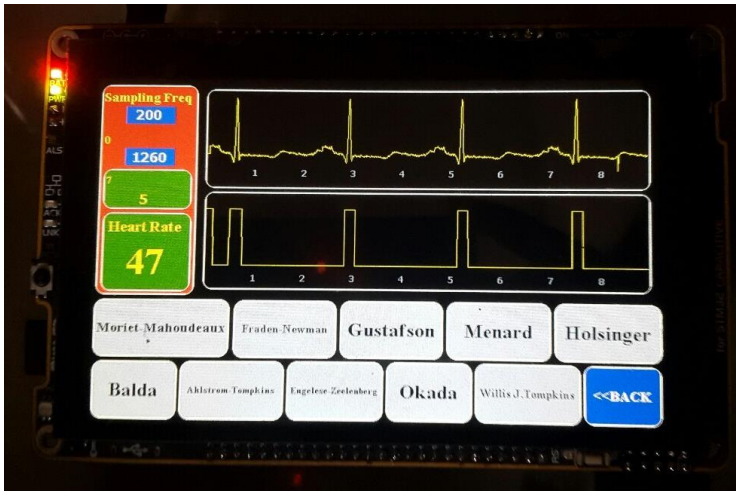
Gambar 4.9 Interface Mikrokontroler saat simulasi metode "Ahlstrom-Tompkins" yang dipilih.



Gambar 4.10 Interface Mikrokontroler saat simulasi metode "Engelese-Zeelenberg" yang dipilih.



Gambar 4.11 Interface Mikrokontroler saat simulasi metode "Okada" yang dipilih.



Gambar 4.12 Interface Mikrokontroler saat simulasi metode "Pan-Tompkins" yang dipilih.

Tabel 4.1 Hasil pengujian *heart rate* pada menu "*simulation*"

No	Algoritma	Target Output (bpm)	Realisasi (bpm)	Error
1	Moriet-Mahoudeux	80	80	0%
2	Fraden and Neuman	80	91	14%
3	Gustafson	80	80	0%
4	Menard	80	91	14%
5	Holsinger	80	80	0%
6	Balda	80	80	0%
7	Ahlstrom and Tompkins	80	0	100%
8	Engelese and Zeelenberg	80	80	0%
9	Okada	80	57	29%
10	Pan and Tompkins	80	47	41%

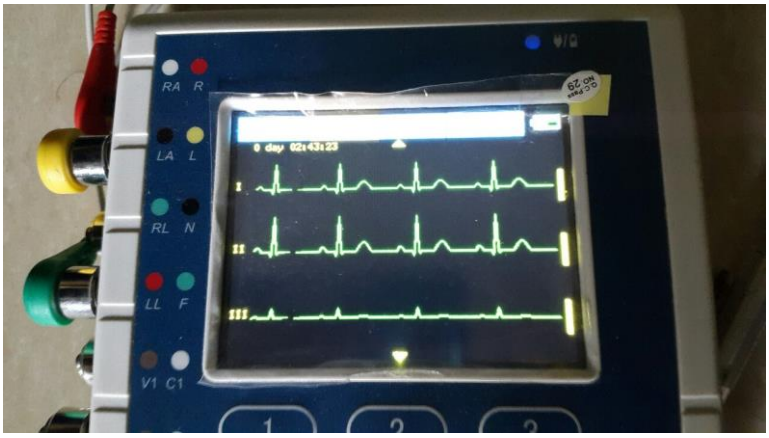
Tabel 4.2 Hasil pengujian *heart beat* pada menu "*Measurement"*

No	Algoritma	TP	FP	FN	Sensitivity	Positive Predictivity	F1 Measure
1	Moriet-Mahoudeux	7	0	0	100%	100%	100%
2	Fraden and Neuman	7	2	0	100%	78%	88%
3	Gustafson	7	0	0	100%	100%	100%
4	Menard	7	1	0	100%	88%	93%
5	Holsinger	7	0	0	100%	100%	100%
6	Balda	7	0	0	100%	100%	100%
7	Ahlstrom and Tompkins	0	0	7	0%	#DIV/0!	0%
8	Engelese and Zeelenberg	7	0	0	100%	100%	100%
9	Okada	4	28	3	57%	13%	21%
10	Pan and Tompkins	4	1	0	100%	80%	89%

4.2 Pengujian dengan Menggunakan Phantom ECG

Pada bagian pengujian mengguanakan phantom, diuji coba dengan menggunakan normal sinus rhythm dengan berbagai nilai heart rate yang berbeda. Nilai heart rate yang diuji yaitu 40 bpm, 60 bpm, 80 bpm, 100 bpm, dan 120 bpm.Berikut ini adalah hasil pengujian menggunakan phantom dengan heart rate sebesar 80 bpm yang dapat dilihat pada Gambar 4.13 sampai Gambar 4. 24. Untuk hasil pengujian *heart rate* dan *heart beat* pada tiap parameter *heart rate* yang diuji, akan ditampilkan pada Tabel 4.3 sampai Tabel 4.12

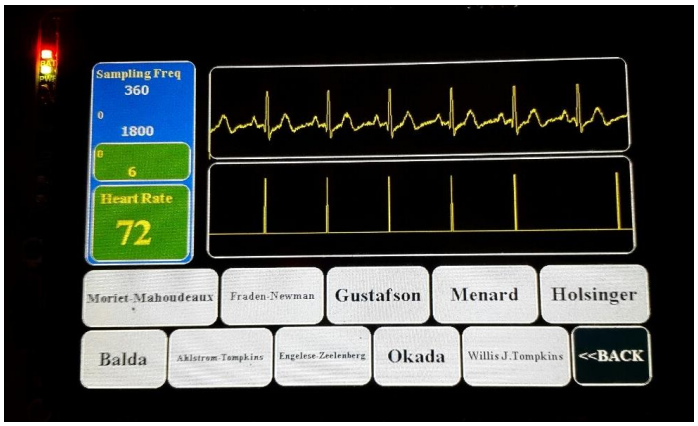
Gambar 4.13 Interface phantom ECG saat simulasi menggunakan *heart rate* sebesar 80 bpm.



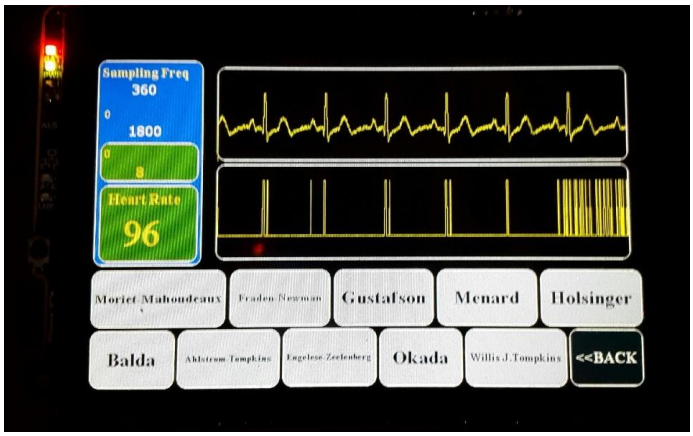
Gambar 4.14 Sinyal output yang disimulasikan oleh phantom ECG dengan *heart rate* sebesar 80 bpm.

Gambar 4.15 Interface Mikrokontroler saat melakukan pengukuran pada metode “Moriet-Mahoudeux” yang dipilih dengan input 80 bpm.

Gambar 4.16 Interface Mikrokontroler saat melakukan pengukuran pada metode “Fraden and Neuman” yang dipilih dengan input 80 bpm.



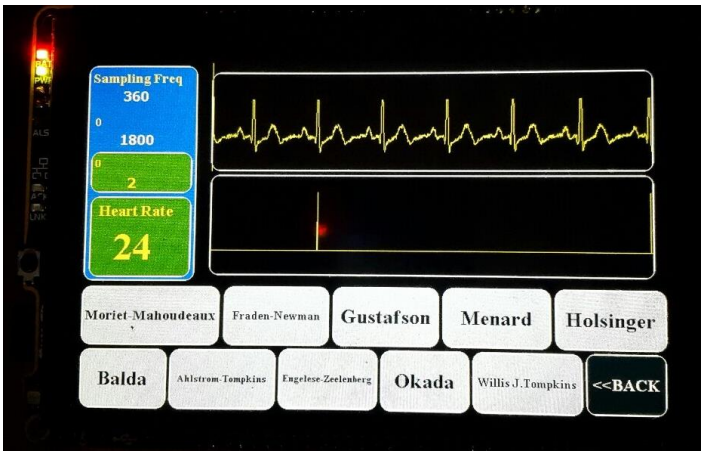
Gambar 4.17 Interface Mikrokontroler saat melakukan pengukuran pada metode “Gustafson” yang dipilih dengan input 80 bpm.



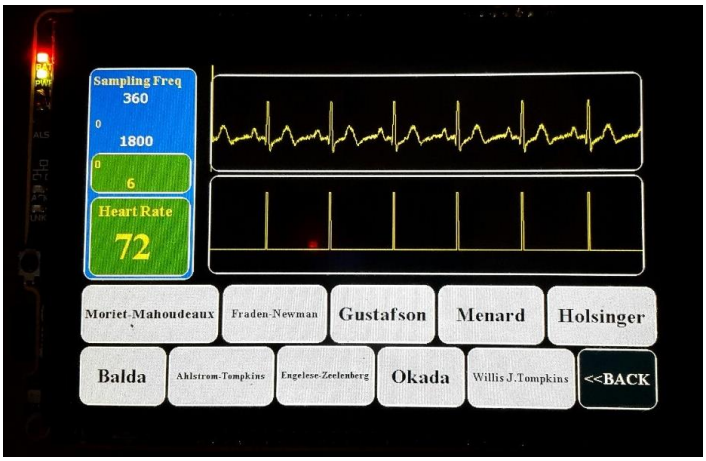
Gambar 4.18 Interface Mikrokontroler saat melakukan pengukuran pada metode “Menard” yang dipilih dengan input 80 bpm.

Gambar 4.19 Interface Mikrokontroler saat melakukan pengukuran pada metode “Holsinger” yang dipilih dengan input 80 bpm.

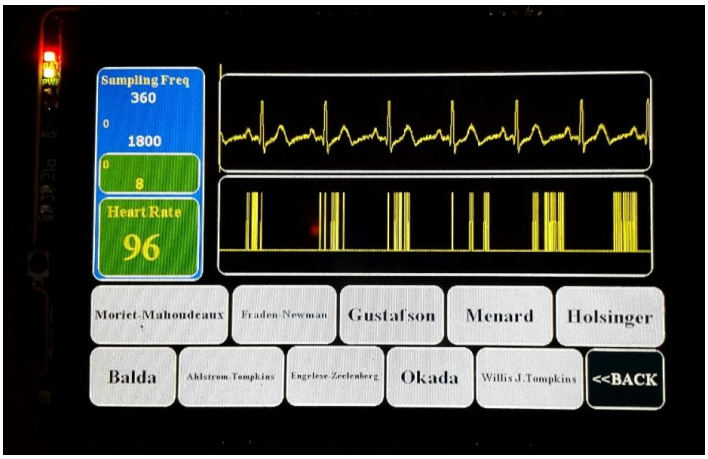
Gambar 4.20 Interface Mikrokontroler saat melakukan pengukuran pada metode “Balda” yang dipilih dengan input 80 bpm.



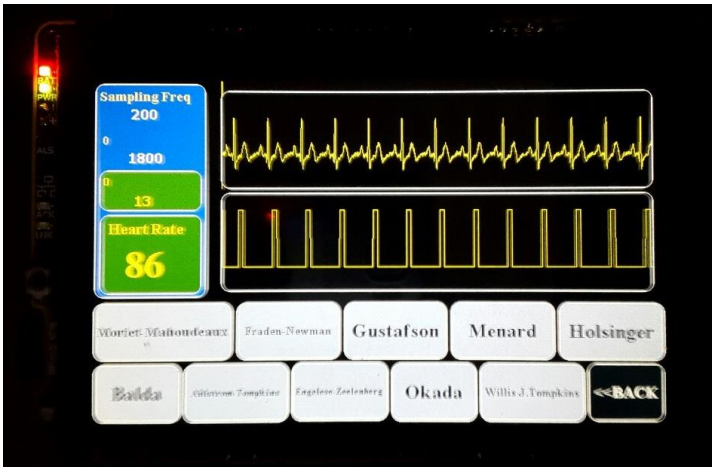
Gambar 4.21 Interface Mikrokontroler saat melakukan pengukuran pada metode "Ahlstrom and Tompkins" yang dipilih dengan input 80 bpm.



Gambar 4.22 Interface Mikrokontroler saat melakukan pengukuran pada metode "Engelese and Zeelenberg" yang dipilih dengan input 80 bpm.



Gambar 4.23 Interface Mikrokontroler saat melakukan pengukuran pada metode "Okada" yang dipilih dengan input 80 bpm.



Gambar 4.24 Interface Mikrokontroler saat melakukan pengukuran pada metode "Pan and Tompkins" yang dipilih dengan input 80 bpm.

4.2.1 Pengujian *heart rate* sebesar 40 bpm

Pada pengujian mengguanakan phantom ECG digunakan pengaturan *heart rate* sebesar 40 bpm. Dari hal ini dapat diketahui hasil pengujian *heart rate* nya yang ditampilkan pada Tabel 4.3 dan hasil pengujian *heart beat* nya yang ditampilkan pada Tabel 4.4.

Tabel 4.3 Hasil pengujian *heart rate* pada saat pengaturan *heart rate* 40 bpm

No	Algoritma	Target Output (bpm)	Realisasi (bpm)	Error
1	Moriet-Mahoudeux	40	36	10%
2	Fraden and Neuman	40	48	20%
3	Gustafson	40	48	20%
4	Menard	40	96	140%
5	Holsinger	40	24	40%
6	Balda	40	36	10%
7	Ahlstrom and Tompkins	40	0	100%
8	Engelese and Zeelenberg	40	36	10%
9	Okada	40	108	170%
10	Pan and Tompkins	40	46	15%

Tabel 4.4 Hasil pengujian *heart beat* pada pengaturan *heart rate* sebesar 40 bpm.

No	Algoritma	TP	FP	FN	Sensitivity	Positive Predictivity	F1 Measure
1	Moriet-Mahoudeux	3	0	0	100%	100%	100%
2	Fraden and Neuman	4	0	0	100%	100%	100%
3	Gustafson	4	0	0	100%	100%	100%
4	Menard	3	31	0	100%	9%	16%
5	Holsinger	2	0	1	67%	100%	80%
6	Balda	3	0	0	100%	100%	100%
7	Ahlstrom and Tompkins	0	0	3	0%	#DIV/0!	0%
8	Engelese and Zeelenberg	3	0	0	100%	100%	100%
9	Okada	3	78	0	100%	4%	7%
10	Pan and Tompkins	7	0	0	100%	100%	100%

4.2.2 Pengujian *heart rate* sebesar 60 bpm

Pada pengujian mengguanakan phantom ECG digunakan pengaturan *heart rate* sebesar 60 bpm. Dari hal ini dapat diketahui hasil pengujian *heart rate* nya yang ditampilkan pada Tabel 4.5 dan hasil pengujian *heart beat* nya yang ditampilkan pada Tabel 4.6.

Tabel 4.5 Hasil pengujian *heart rate* pada saat pengaturan *heart rate* 60 bpm

No	Algoritma	Target Output (bpm)	Realisasi (bpm)	Error
1	Moriet-Mahoudeux	60	48	20%
2	Fraden and Neuman	60	60	0%
3	Gustafson	60	72	20%
4	Menard	60	72	20%
5	Holsinger	60	60	0%
6	Balda	60	48	20%
7	Ahlstrom and Tompkins	60	0	100%
8	Engelese and Zeelenberg	60	60	0%
9	Okada	60	84	40%
10	Pan and Tompkins	60	66	10%

Tabel 4.6 Hasil pengujian *heart beat* pada pengaturan *heart rate* sebesar 60 bpm.

No	Algoritma	TP	FP	FN	Sensitivity	Positive Predictivity	F1 Measure
1	Moriet-Mahoudeux	4	0	1	80%	100%	89%
2	Fraden and Neuman	5	0	0	100%	100%	100%
3	Gustafson	5	1	0	100%	83%	91%
4	Menard	5	31	0	100%	14%	24%
5	Holsinger	5	0	0	100%	100%	100%
6	Balda	5	1	0	100%	83%	91%
7	Ahlstrom and Tompkins	0	0	5	0%	#DIV/0!	0%
8	Engelese and Zeelenberg	5	0	0	100%	100%	100%
9	Okada	3	62	2	60%	5%	9%
10	Pan and Tompkins	9	0	0	100%	100%	100%

4.2.3 Pengujian *heart rate* sebesar 80 bpm

Pada pengujian mengguanakan phantom ECG digunakan pengaturan *heart rate* sebesar 80 bpm. Dari hal ini dapat diketahui hasil pengujian *heart rate* nya yang ditampilkan pada Tabel 4.7 dan hasil pengujian *heart beat* nya yang ditampilkan pada Tabel 4.8.

Tabel 4.7 Hasil pengujian *heart rate* pada saat pengaturan *heart rate* 80 bpm

No	Algoritma	Target Output (bpm)	Realisasi (bpm)	Error
1	Moriet-Mahoudeux	80	72	10%
2	Fraden and Neuman	80	96	20%
3	Gustafson	80	72	10%
4	Menard	80	96	20%
5	Holsinger	80	84	5%
6	Balda	80	84	5%
7	Ahlstrom and Tompkins	80	24	70%
8	Engelese and Zeelenberg	80	72	10%
9	Okada	80	96	20%
10	Pan and Tompkins	80	86	8%

Tabel 4.8 Hasil pengujian *heart beat* pada pengaturan *heart rate* sebesar 80 bpm.

No	Algoritma	TP	FP	FN	Sensitivity	Positive Predictivity	F1 Measure
1	Moriet-Mahoudeux	6	0	1	86%	100%	92%
2	Fraden and Neuman	7	0	0	100%	100%	100%
3	Gustafson	5	1	1	83%	83%	83%
4	Menard	6	24	0	100%	20%	33%
5	Holsinger	6	1	1	86%	86%	86%
6	Balda	7	0	0	100%	100%	100%
7	Ahlstrom and Tompkins	1	0	6	14%	100%	25%
8	Engelese and Zeelenberg	6	0	0	100%	100%	100%
9	Okada	4	49	3	57%	8%	13%
10	Pan and Tompkins	13	0	0	100%	100%	100%

4.2.4 Pengujian *heart rate* sebesar 100 bpm

Pada pengujian mengguanakan phantom ECG digunakan pengaturan *heart rate* sebesar 100 bpm. Dari hal ini dapat diketahui hasil pengujian *heart rate* nya yang ditampilkan pada Tabel 4.9 dan hasil pengujian *heart beat* nya yang ditampilkan pada Tabel 4.10.

Tabel 4.9 Hasil pengujian *heart rate* pada saat pengaturan *heart rate* 100 bpm

No	Algoritma	Target Output (bpm)	Realisasi (bpm)	Error
1	Moriet-Mahoudeux	100	84	16%
2	Fraden and Neuman	100	96	4%
3	Gustafson	100	96	4%
4	Menard	100	96	4%
5	Holsinger	100	96	4%
6	Balda	100	96	4%
7	Ahlstrom and Tompkins	100	0	100%
8	Engelese and Zeelenberg	100	96	4%
9	Okada	100	156	56%
10	Pan and Tompkins	100	113	13%

Tabel 4.10 Hasil pengujian *heart beat* pada pengaturan *heart rate* sebesar 100 bpm.

No	Algoritma	TP	FP	FN	Sensitivity	Positive Predictivity	F1 Measure
1	Moriet-Mahoudeux	7	0	1	88%	100%	93%
2	Fraden and Neuman	8	0	0	100%	100%	100%
3	Gustafson	8	0	0	100%	100%	100%
4	Menard	9	26	0	100%	26%	41%
5	Holsinger	8	1	0	100%	89%	94%
6	Balda	8	7	0	100%	53%	70%
7	Ahlstrom and Tompkins	0	0	8	0%	#DIV/0!	0%
8	Engelese and Zeelenberg	8	0	0	100%	100%	100%
9	Okada	6	32	3	67%	16%	26%
10	Pan and Tompkins	16	0	0	100%	100%	100%

4.2.5 Pengujian *heart rate* sebesar 120 bpm

Pada pengujian mengguanakan phantom ECG digunakan pengaturan *heart rate* sebesar 120 bpm. Dari hal ini dapat diketahui hasil pengujian *heart rate* nya yang ditampilkan pada Tabel 4.11 dan hasil pengujian *heart beat* nya yang ditampilkan pada Tabel 4.12.

Tabel 4.11 Hasil pengujian *heart rate* pada saat pengaturan *heart rate* 120 bpm

No	Algoritma	Target Output (bpm)	Realisasi (bpm)	Error
1	Moriet-Mahoudeux	120	96	20%
2	Fraden and Neuman	120	120	0%
3	Gustafson	120	108	10%
4	Menard	120	144	20%
5	Holsinger	120	108	10%
6	Balda	120	120	0%
7	Ahlstrom and Tompkins	120	0	100%
8	Engelese and Zeelenberg	120	120	0%
9	Okada	120	132	10%
10	Pan and Tompkins	120	126	5%

Tabel 4.12 Hasil pengujian *heart beat* pada pengaturan *heart rate* sebesar 120 bpm.

No	Algoritma	TP	FP	FN	Sensitivity	Positive Predictivity	F1 Measure
1	Moriet-Mahoudeux	8	0	2	80%	100%	89%
2	Fraden and Neuman	10	0	0	100%	100%	100%
3	Gustafson	8	1	2	80%	89%	84%
4	Menard	10	30	0	100%	25%	40%
5	Holsinger	8	2	2	80%	80%	80%
6	Balda	10	9	0	100%	53%	69%
7	Ahlstrom and Tompkins	0	0	10	0%	#DIV/0!	0%
8	Engelese and Zeelenberg	10	0	0	100%	100%	100%
9	Okada	6	36	4	60%	14%	23%
10	Pan and Tompkins	19	0	0	100%	100%	100%

4.3 Pengujian dengan Menggunakan Phantom ECG dengan noise

Pada bagian pengujian mengguanakan phantom, diuji coba juga dengan menggunakan sinyal ECG yang diberikan beberapa jenis noise dengan nilai heart rate sebesar 80 bpm. Jenis-jenis *noise* yang diuji yaitu *inference 50 Hz noise*, *muscle artifact noise, baseline artifact noise,* dan *respiration artifact noise*. Untuk hasil pengujian *heart rate* dan *heart beat* pada tiap jenis *noise* yang diuji, akan ditampilkan pada Tabel 4.13 sampai Tabel 4.20.

4.3.1 Pengujian dengan *inference 50 Hz noise*

Pada pengujian mengguanakan phantom ECG digunakan pengaturan *noise* dengan jenis *inference 50 Hz noise* dengan *heart rate* sebesar 80 bpm. Dari hal ini dapat diketahui hasil pengujian *heart rate* nya yang ditampilkan pada Tabel 4.13 dan hasil pengujian *heart beat* nya yang ditampilkan pada Tabel 4.14.

Tabel 4.13 Hasil pengujian *heart rate* dengan *inference 50 Hz noise*

No	Algoritma	Target Output (bpm)	Realisasi (bpm)	Error
1	Moriet-Mahoudeux	80	24	70%
2	Fraden and Neuman	80	12	85%
3	Gustafson	80	276	245%
4	Menard	80	12	85%
5	Holsinger	80	12	85%
6	Balda	80	36	55%
7	Ahlstrom and Tompkins	80	96	20%
8	Engelese and Zeelenberg	80	12	85%
9	Okada	80	84	5%
10	Pan and Tompkins	80	86	8%

Tabel 4.14 Hasil pengujian *heart beat* dengan *inference 50 Hz noise*

No	Algoritma	TP	FP	FN	Sensitivity	Positive Predictivity	F1 Measure
1	Moriet-Mahoudeux	1	167	5	17%	1%	1%
2	Fraden and Neuman	2	175	4	33%	1%	2%
3	Gustafson	0	35	7	0%	0%	0%
4	Menard	7	170	0	100%	4%	8%
5	Holsinger	4	173	3	57%	2%	4%
6	Balda	5	172	2	71%	3%	5%
7	Ahlstrom and Tompkins	5	148	2	71%	3%	6%
8	Engelese and Zeelenberg	1	173	6	14%	1%	1%
9	Okada	7	98	0	100%	7%	13%
10	Pan and Tompkins	13	2	0	100%	87%	93%

4.3.2 Pengujian dengan *muscle artifact noise*

Pada pengujian mengguanakan phantom ECG digunakan pengaturan *noise* dengan jenis *muscle artifact noise* dengan *heart rate* sebesar 80 bpm. Dari hal ini dapat diketahui hasil pengujian *heart rate* nya yang ditampilkan pada Tabel 4.15 dan hasil pengujian *heart beat* nya yang ditampilkan pada Tabel 4.16

Tabel 4.15 Hasil pengujian *heart rate* dengan *muscle artifact noise*

No	Algoritma	Target Output (bpm)	Realisasi (bpm)	Error
1	Moriet-Mahoudeux	80	36	55%
2	Fraden and Neuman	80	168	110%
3	Gustafson	80	72	10%
4	Menard	80	12	85%
5	Holsinger	80	276	245%
6	Balda	80	108	35%
7	Ahlstrom and Tompkins	80	0	100%
8	Engelese and Zeelenberg	80	156	95%
9	Okada	80	84	5%
10	Pan and Tompkins	80	86	8%

Tabel 4.16 Hasil pengujian *heart beat* dengan *muscle artifact noise*

No	Algoritma	TP	FP	FN	Sensitivity	Positive Predictivity	F1 Measure
1	Moriet-Mahoudeux	3	0	4	43%	100%	60%
2	Fraden and Neuman	5	11	1	83%	31%	45%
3	Gustafson	6	1	1	86%	86%	86%
4	Menard	1	171	6	14%	1%	1%
5	Holsinger	4	44	3	57%	8%	15%
6	Balda	7	98	0	100%	7%	13%
7	Ahlstrom and Tompkins	0	0	7	0%	#DIV/0!	0%
8	Engelese and Zeelenberg	7	6	0	100%	54%	70%
9	Okada	6	58	0	100%	9%	17%
10	Pan and Tompkins	13	0	0	100%	100%	100%

4.3.3 Pengujian dengan *baseline artifact noise*

Pada pengujian mengguanakan phantom ECG digunakan pengaturan *noise* dengan jenis *baseline artifact noise* dengan *heart rate* sebesar 80 bpm. Dari hal ini dapat diketahui hasil pengujian *heart rate* nya yang ditampilkan pada Tabel 4.17 dan hasil pengujian *heart beat* nya yang ditampilkan pada Tabel 4.18.

Tabel 4.17 Hasil pengujian *heart rate* dengan *baseline artifact noise*

No	Algoritma	Target Output (bpm)	Realisasi (bpm)	Error
1	Moriet-Mahoudeux	80	72	10%
2	Fraden and Neuman	80	84	5%
3	Gustafson	80	60	25%
4	Menard	80	120	50%
5	Holsinger	80	96	20%
6	Balda	80	72	10%
7	Ahlstrom and Tompkins	80	0	100%
8	Engelese and Zeelenberg	80	72	10%
9	Okada	80	120	50%
10	Pan and Tompkins	80	93	16%

Tabel 4.18 Hasil pengujian *heart beat* dengan *baseline artifact noise*

No	Algoritma	TP	FP	FN	Sensitivity	Positive Predictivity	F1 Measure
1	Moriet-Mahoudeux	6	0	1	86%	100%	92%
2	Fraden and Neuman	6	1	0	100%	86%	92%
3	Gustafson	5	0	1	83%	100%	91%
4	Menard	7	22	0	100%	24%	39%
5	Holsinger	6	3	1	86%	67%	75%
6	Balda	6	6	0	100%	50%	67%
7	Ahlstrom and Tompkins	0	0	7	0%	#DIV/0!	0%
8	Engelese and Zeelenberg	6	0	0	100%	100%	100%
9	Okada	5	36	2	71%	12%	21%
10	Pan and Tompkins	13	0	0	100%	100%	100%

4.3.4 Pengujian dengan respiration artifact noise

Pada pengujian mengguanakan phantom ECG digunakan pengaturan *noise* dengan jenis *respiration artifact noise* dengan *heart rate* sebesar 80 bpm. Dari hal ini dapat diketahui hasil pengujian *heart rate* nya yang ditampilkan pada Tabel 4.19 dan hasil pengujian *heart beat* nya yang ditampilkan pada Tabel 4.20.

Tabel 4.19 Hasil pengujian heart rate dengan respiration artifact noise

No	Algoritma	Target Output (bpm)	Realisasi (bpm)	Error
1	Moriet-Mahoudeux	80	72	10%
2	Fraden and Neuman	80	84	5%
3	Gustafson	80	84	5%
4	Menard	80	84	5%
5	Holsinger	80	72	10%
6	Balda	80	72	10%
7	Ahlstrom and Tompkins	80	0	100%
8	Engelese and Zeelenberg	80	72	10%
9	Okada	80	96	20%
10	Pan and Tompkins	80	93	16%

Tabel 4.20 Hasil pengujian heart beat dengan *respiration artifact noise*

No	Algoritma	TP	FP	FN	Sensitivity	Positive Predictivity	F1 Measure
1	Moriet-Mahoudeux	6	0	1	86%	100%	92%
2	Fraden and Neuman	6	0	0	100%	100%	100%
3	Gustafson	6	1	1	86%	86%	86%
4	Menard	7	24	0	100%	23%	37%
5	Holsinger	5	2	1	83%	71%	77%
6	Balda	6	4	0	100%	60%	75%
7	Ahlstrom and Tompkins	0	0	7	0%	#DIV/0!	0%
8	Engelese and Zeelenberg	6	0	0	100%	100%	100%
9	Okada	6	64	1	86%	9%	16%
10	Pan and Tompkins	13	0	0	100%	100%	100%

4.4 Pengujian dengan Menggunakan Sinyal Uji Tubuh

Pada pengujian dengan menggunakan subjek manusia, dilakukan pengujian heart rate dan heart beat dengan menggunakan sinyal uji tubuh. Subjek yang dicoba hanya sebanyak 1 orang. Pengujian ini dilakukan dengan menempelkan elektroda-elektroda yang ada ke 3 area, yaitu tangan kiri (LA), tangan kanan (RA), dan kaki kiri (LL). Kemudian posisi subjek saat dilakukan pengujian yaitu berdiri. Sebagai pembanding, juga dilakukan pengujian dengan menggunakan fitur 'heart rate' dengan menggunakan pulse oximeter yang ada dalam gawai. Namun sebelumnya tidak diketahui keakurasian dari fitur ini sehingga tidak akan dihitung error sistem ini terhadap fitur yang ada di gawai. Untuk melihat hasil pengujian dengan menggunakan sinyal tubuh dapat dilihat pada Tabel 4.21 dan Tabel 4.22.

Tabel 4.21 Hasil pengujian *heart rate* dengan sinyal uji tubuh

No	Algoritma	Hasil dari Mikrokontroler (bpm)	Hasil dari Gawai (bpm)
1	Moriet-Mahoudeux	48	102
2	Fraden and Neuman	84	
3	Gustafson	72	
4	Menard	132	
5	Holsinger	144	
6	Balda	228	
7	Ahlstrom and Tompkins	0	
8	Engelese and Zeelenberg	96	
9	Okada	108	
10	Pan and Tompkins	100	

Tabel 4.22 Hasil pengujian *heart beat* dengan sinyal uji tubuh

No	Algoritma	TP	FP	FN	Sensitivity	Positive Predictivity	F1 Measure
1	Moriet-Mahoudeux	4	0	3	57%	100%	73%
2	Fraden and Neuman	6	1	2	75%	86%	80%
3	Gustafson	6	5	2	75%	55%	63%
4	Menard	8	86	0	100%	9%	16%
5	Holsinger	7	10	1	88%	41%	56%
6	Balda	8	47	0	100%	15%	25%
7	Ahlstrom and Tompkins	0	0	8	0%	#DIV/0!	0%
8	Engelese and Zeelenberg	8	1	0	100%	89%	94%
9	Okada	7	54	1	88%	11%	20%
10	Pan and Tompkins	15	0	0	100%	100%	100%

4.5 *Computation cost*

Computation cost ini mengacu pada penggunaan waktu yang digunakan pada tiap metode dan memori yang terpakai pada penelitian ini. Untuk penggunaan waktu yang digunakan pada tiap metode mengacu pada interrupt time yang digunakan pada tiap metode karena *interrupt time* ini merupakan waktu yang digunakan untuk menginterupsi proses yang sedang berjalan sebelumnya. Informasi penggunaan waktu pada tiap algoritma dapat dilihat lengkapnya pada Tabel 4.23.

Tabel 4.23 Tabel penggunaan waktu komputasi tiap algoritma

No	Algoritma	Frekuensi sampling (Hz)	Waktu interupsi (ms)
1	Moriet-Mahoudeux	360	2.778
2	Fraden and Neuman	360	2.778
3	Gustafson	360	2.778
4	Menard	360	2.778
5	Holsinger	360	2.778
6	Balda	360	2.778
7	Ahlstrom and Tompkins	360	2.778
8	Engelese and Zeelenberg	360	2.778
9	Okada	360	2.778
10	Pan and Tompkins	200	5

Kemudian untuk penggunaan memori dapat diketahui pada saat proses *build* program di IDE yang digunakan. Penggunaan memori total keseluruhan sistem ini yaitu sebesar 220 KB dari total 1 MB *flash memory* yang dimiliki oleh mikrokontroler. Dimana untuk sebuah sistem mikrokontroler, maksimal penggunaan memori idealnya yaitu setengah dari total memori untuk menghindari *stack overflow* pada mikrokontroler.

BAB 5 PEMBAHASAN

5.1 Analisa tiap algoritma

Pada tahap pengujian, sistem diuji coba dengan menggunakan berbagai macam besar *heart rate*. Besar nilai *heart rate* yang digunakan dalam pengujian yaitu 40 bpm, 60 bpm, 80 bpm, 100 bpm, dan 120 bpm. Selain itu dengan menggunakan *phantom* ECG, dibuat pula skenario pengujian dengan menggunakan *noise.* Ada beberapa macam *noise* yang akan diuji cobakan pada sistem. Macam-macam *noise* tersebut yaitu *inference 50 Hz noise, muscle artifact noise, baseline artifact noise*, dan *respiration artifact noise*. Dari kelima besar *heart rate* tersebut dan uji coba menggunakan sinyal yang telah diberi *noise* maka didapatkanlah 2 informasi utama yaitu mengenai akurasi *heart rate* pada tiap algoritma dan informasi mengenai ketepatan *heart beat* yang dalam hal ini digunakanlah parameter terkait seperti *sensitivity* (Se), *positive predictivity* (+P), dan *F1 measure* (F1). Adanya error pada tiap kali pengujian *heart rate* bisa terjadi karena jumlah data yang diambil untuk pengolahan sinyal terbatas.

5.1.1 Moriet-Mahoudeux

Pada algoritma ini ketika dilakukan simulasi dan pengujian menggunakan phantom ECG didapatkan hasil pengujian *heart rate* dan *heart beat*. Untuk lebih ringkasnya dapat dilihat di Tabel 5.1 yang merupakan hasil pengujian *heart rate* pada algoritma ini dan Tabel 5.2 yang merupakan hasil pengujian *heart beat* pada algoritma ini. Lalu waktu yang digunakan oleh algoritma ini untuk melakukan prosesnya yaitu sebesar 2.778 ms berdasarkan perhitungan *interrupt time*nya.

Tabel 5.1 Hasil pengujian *heart rate* pada pengujian algoritma “Moriet-Mahoudeux”

No	Pengujian	Target Output (bpm)	Realisasi (bpm)	Error
1	Simulasi	80	80	0%
2	Phantom 40 bpm	40	36	10%
3	Phantom 60 bpm	60	48	20%
4	Phantom 80 bpm	80	72	10%
5	Phantom 100 bpm	100	84	16%
6	Phantom 120 bpm	120	96	20%
7	50 Hz noise	80	24	70%
8	muscle artifact	80	36	55%
9	baseline artifact	80	72	10%
10	respiration artifact	80	72	10%

Tabel 5.2 Hasil pengujian *heart beat* pada pengujian algoritma "Moriet-Mahoudeux"

No	Pengujian	TP	FP	FN	Se	P	F1
1	Simulasi	7	0	0	100%	100%	100%
2	Phantom 40 bpm	3	0	0	100%	100%	100%
3	Phantom 60 bpm	4	0	1	80%	100%	89%
4	Phantom 80 bpm	6	0	1	86%	100%	92%
5	Phantom 100 bpm	7	0	1	88%	100%	93%
6	Phantom 120 bpm	8	0	2	80%	100%	89%
7	50 Hz noise	1	167	5	17%	1%	1%
8	muscle artifact	3	0	4	43%	100%	60%
9	baseline artifact	6	0	1	86%	100%	92%
10	respiration artifact	6	0	1	86%	100%	92%

5.1.2 Fraden and Neuman

Pada algoritma ini ketika dilakukan simulasi dan pengujian menggunakan phantom ECG didapatkan hasil pengujian *heart rate* dan *heart beat*. Untuk lebih ringkasnya dapat dilihat di Tabel 5.3 yang merupakan hasil pengujian *heart rate* pada algoritma ini dan Tabel 5.4 yang merupakan hasil pengujian *heart beat* pada algoritma ini.

Tabel 5.3 Hasil pengujian *heart rate* pada pengujian algoritma "Fraden and Neuman"

No	Pengujian	Target Output (bpm)	Realisasi (bpm)	Error
1	Simulasi	80	91	14%
2	Phantom 40 bpm	40	48	20%
3	Phantom 60 bpm	60	60	0%
4	Phantom 80 bpm	80	96	20%
5	Phantom 100 bpm	100	96	4%
6	Phantom 120 bpm	120	120	0%
7	50 Hz noise	80	12	85%
8	muscle artifact	80	168	110%
9	baseline artifact	80	84	5%
10	respiration artifact	80	84	5%

Tabel 5.4 Hasil pengujian *heart beat* pada pengujian algoritma "Fraden and Neuman"

No	Pengujian	TP	FP	FN	Se	P	F1
1	Simulasi	7	2	0	100%	78%	88%
2	Phantom 40 bpm	4	0	0	100%	100%	100%
3	Phantom 60 bpm	5	0	0	100%	100%	100%
4	Phantom 80 bpm	7	0	0	100%	100%	100%
5	Phantom 100 bpm	8	0	0	100%	100%	100%
6	Phantom 120 bpm	10	0	0	100%	100%	100%
7	50 Hz noise	2	175	4	33%	1%	2%
8	muscle artifact	5	11	1	83%	31%	45%
9	baseline artifact	6	1	0	100%	86%	92%
10	respiration artifact	6	0	0	100%	100%	100%

5.1.3 Gustafson

Pada algoritma ini ketika dilakukan simulasi dan pengujian menggunakan phantom ECG didapatkan hasil pengujian *heart rate* dan *heart beat*. Untuk lebih ringkasnya dapat dilihat di Tabel 5.5 yang merupakan hasil pengujian *heart rate* pada algoritma ini dan Tabel 5.6 yang merupakan hasil pengujian *heart beat* pada algoritma ini.

Tabel 5.5 Hasil pengujian *heart rate* pada pengujian algoritma "Gustafson"

No	Pengujian	Target Output (bpm)	Realisasi (bpm)	Error
1	Simulasi	80	80	0%
2	Phantom 40 bpm	40	48	20%
3	Phantom 60 bpm	60	72	20%
4	Phantom 80 bpm	80	72	10%
5	Phantom 100 bpm	100	96	4%
6	Phantom 120 bpm	120	108	10%
7	50 Hz noise	80	276	245%
8	muscle artifact	80	72	10%
9	baseline artifact	80	60	25%
10	respiration artifact	80	84	5%

Tabel 5.6 Hasil pengujian *heart beat* pada pengujian algoritma "Gustafson"

No	Pengujian	TP	FP	FN	Se	P	F1
1	Simulasi	7	0	0	100%	100%	100%
2	Phantom 40 bpm	4	0	0	100%	100%	100%
3	Phantom 60 bpm	5	1	0	100%	83%	91%
4	Phantom 80 bpm	5	1	1	83%	83%	83%
5	Phantom 100 bpm	8	0	0	100%	100%	100%
6	Phantom 120 bpm	8	1	2	80%	89%	84%
7	50 Hz noise	0	35	7	0%	0%	0%
8	muscle artifact	6	1	1	86%	86%	86%
9	baseline artifact	5	0	1	83%	100%	91%
10	respiration artifact	6	1	1	86%	86%	86%

5.1.4 Menard

Pada algoritma ini ketika dilakukan simulasi dan pengujian menggunakan phantom ECG didapatkan hasil pengujian *heart rate* dan *heart beat*. Untuk lebih ringkasnya dapat dilihat di Tabel 5.7 yang merupakan hasil pengujian *heart rate* pada algoritma ini dan Tabel 5.8 yang merupakan hasil pengujian *heart beat* pada algoritma ini.

Tabel 5.7 Hasil pengujian *heart rate* pada pengujian algoritma "Menard"

No	Pengujian	Target Output (bpm)	Realisasi (bpm)	Error
1	Simulasi	80	91	14%
2	Phantom 40 bpm	40	96	140%
3	Phantom 60 bpm	60	72	20%
4	Phantom 80 bpm	80	96	20%
5	Phantom 100 bpm	100	96	4%
6	Phantom 120 bpm	120	144	20%
7	50 Hz noise	80	12	85%
8	muscle artifact	80	12	85%
9	baseline artifact	80	120	50%
10	respiration artifact	80	84	5%

Tabel 5.8 Hasil pengujian *heart beat* pada pengujian algoritma "Menard"

No	Pengujian	TP	FP	FN	Se	P	F1
1	Simulasi	7	1	0	100%	88%	93%
2	Phantom 40 bpm	3	31	0	100%	9%	16%
3	Phantom 60 bpm	5	31	0	100%	14%	24%
4	Phantom 80 bpm	6	24	0	100%	20%	33%
5	Phantom 100 bpm	9	26	0	100%	26%	41%
6	Phantom 120 bpm	10	30	0	100%	25%	40%
7	50 Hz noise	7	170	0	100%	4%	8%
8	muscle artifact	1	171	6	14%	1%	1%
9	baseline artifact	7	22	0	100%	24%	39%
10	respiration artifact	7	24	0	100%	23%	37%

5.1.5 Holsinger

Pada algoritma ini ketika dilakukan simulasi dan pengujian menggunakan phantom ECG didapatkan hasil pengujian *heart rate* dan *heart beat*. Untuk lebih ringkasnya dapat dilihat di Tabel 5.9 yang merupakan hasil pengujian *heart rate* pada algoritma ini dan Tabel 5.10 yang merupakan hasil pengujian *heart beat* pada algoritma ini.

Tabel 5.9 Hasil pengujian *heart rate* pada pengujian algoritma "Holsinger"

No	Pengujian	Target Output (bpm)	Realisasi (bpm)	Error
1	Simulasi	80	80	0%
2	Phantom 40 bpm	40	24	40%
3	Phantom 60 bpm	60	60	0%
4	Phantom 80 bpm	80	84	5%
5	Phantom 100 bpm	100	96	4%
6	Phantom 120 bpm	120	108	10%
7	50 Hz noise	80	12	85%
8	muscle artifact	80	276	245%
9	baseline artifact	80	96	20%
10	respiration artifact	80	96	20%

Tabel 5.10 Hasil pengujian *heart beat* pada pengujian algoritma "Holsinger"

No	Pengujian	TP	FP	FN	Se	P	F1
1	Simulasi	7	0	0	100%	100%	100%
2	Phantom 40 bpm	2	0	1	67%	100%	80%
3	Phantom 60 bpm	5	0	0	100%	100%	100%
4	Phantom 80 bpm	6	1	1	86%	86%	86%
5	Phantom 100 bpm	8	1	0	100%	89%	94%
6	Phantom 120 bpm	8	2	2	80%	80%	80%
7	50 Hz noise	4	173	3	57%	2%	4%
8	muscle artifact	4	44	3	57%	8%	15%
9	baseline artifact	6	3	1	86%	67%	75%
10	respiration artifact	5	2	1	83%	71%	77%

5.1.6 Balda

Pada algoritma ini ketika dilakukan simulasi dan pengujian menggunakan phantom ECG didapatkan hasil pengujian *heart rate* dan *heart beat*. Untuk lebih ringkasnya dapat dilihat di Tabel 5.11 yang merupakan hasil pengujian *heart rate* pada algoritma ini dan Tabel 5.12 yang merupakan hasil pengujian *heart beat* pada algoritma ini.

Tabel 5.11 Hasil pengujian *heart rate* pada pengujian algoritma "Balda"

No	Pengujian	Target Output (bpm)	Realisasi (bpm)	Error
1	Simulasi	80	80	0%
2	Phantom 40 bpm	40	36	10%
3	Phantom 60 bpm	60	48	20%
4	Phantom 80 bpm	80	84	5%
5	Phantom 100 bpm	100	96	4%
6	Phantom 120 bpm	120	120	0%
7	50 Hz noise	80	36	55%
8	muscle artifact	80	108	35%
9	baseline artifact	80	72	10%
10	respiration artifact	80	72	10%

Tabel 5.12 Hasil pengujian *heart beat* pada pengujian algoritma "Balda"

No	Pengujian	TP	FP	FN	Se	P	F1
1	Simulasi	7	0	0	100%	100%	100%
2	Phantom 40 bpm	3	0	0	100%	100%	100%
3	Phantom 60 bpm	5	1	0	100%	83%	91%
4	Phantom 80 bpm	7	0	0	100%	100%	100%
5	Phantom 100 bpm	8	7	0	100%	53%	70%
6	Phantom 120 bpm	10	9	0	100%	53%	69%
7	50 Hz noise	5	172	2	71%	3%	5%
8	muscle artifact	7	98	0	100%	7%	13%
9	baseline artifact	6	6	0	100%	50%	67%
10	respiration artifact	6	4	0	100%	60%	75%

5.1.7 Ahlstrom and Tompkins

Pada algoritma ini ketika dilakukan simulasi dan pengujian menggunakan phantom ECG didapatkan hasil pengujian *heart rate* dan *heart beat*. Untuk lebih ringkasnya dapat dilihat di Tabel 5.13 yang merupakan hasil pengujian *heart rate* pada algoritma ini dan Tabel 5.14 yang merupakan hasil pengujian *heart beat* pada algoritma ini.

Tabel 5.13 Hasil pengujian *heart rate* pada pengujian algoritma "Ahlstrom and Tompkins"

No	Pengujian	Target Output (bpm)	Realisasi (bpm)	Error
1	Simulasi	80	0	100%
2	Phantom 40 bpm	40	0	100%
3	Phantom 60 bpm	60	0	100%
4	Phantom 80 bpm	80	24	70%
5	Phantom 100 bpm	100	0	100%
6	Phantom 120 bpm	120	0	100%
7	50 Hz noise	80	96	20%
8	muscle artifact	80	0	100%
9	baseline artifact	80	0	100%
10	respiration artifact	80	0	100%

Tabel 5.14 Hasil pengujian *heart beat* pada pengujian algoritma "Ahlstrom and Tompkins"

No	Pengujian	TP	FP	FN	Se	P	F1
1	Simulasi	0	0	7	0%	#DIV/0!	0%
2	Phantom 40 bpm	0	0	3	0%	#DIV/0!	0%
3	Phantom 60 bpm	0	0	5	0%	#DIV/0!	0%
4	Phantom 80 bpm	1	0	6	14%	100%	25%
5	Phantom 100 bpm	0	0	8	0%	#DIV/0!	0%
6	Phantom 120 bpm	0	0	10	0%	#DIV/0!	0%
7	50 Hz noise	5	148	2	71%	3%	6%
8	muscle artifact	0	0	7	0%	#DIV/0!	0%
9	baseline artifact	0	0	7	0%	#DIV/0!	0%
10	respiration artifact	0	0	7	0%	#DIV/0!	0%

5.1.8 Engelese and Zeelenberg

Pada algoritma ini ketika dilakukan simulasi dan pengujian menggunakan phantom ECG didapatkan hasil pengujian *heart rate* dan *heart beat*. Untuk lebih ringkasnya dapat dilihat di Tabel 5.15 yang merupakan hasil pengujian *heart rate* pada algoritma ini dan Tabel 5.16 yang merupakan hasil pengujian *heart beat* pada algoritma ini.

Tabel 5.15 Hasil pengujian *heart rate* pada pengujian algoritma "Engelese and Zeelenberg"

No	Pengujian	Target Output (bpm)	Realisasi (bpm)	Error
1	Simulasi	80	80	0%
2	Phantom 40 bpm	40	36	10%
3	Phantom 60 bpm	60	60	0%
4	Phantom 80 bpm	80	72	10%
5	Phantom 100 bpm	100	96	4%
6	Phantom 120 bpm	120	120	0%
7	50 Hz noise	80	12	85%
8	muscle artifact	80	156	95%
9	baseline artifact	80	72	10%
10	respiration artifact	80	72	10%

Tabel 5.16 Hasil pengujian *heart beat* pada pengujian algoritma "Engelese and Zeelenberg"

No	Pengujian	TP	FP	FN	Se	P	F1
1	Simulasi	7	0	0	100%	100%	100%
2	Phantom 40 bpm	3	0	0	100%	100%	100%
3	Phantom 60 bpm	5	0	0	100%	100%	100%
4	Phantom 80 bpm	6	0	0	100%	100%	100%
5	Phantom 100 bpm	8	0	0	100%	100%	100%
6	Phantom 120 bpm	10	0	0	100%	100%	100%
7	50 Hz noise	1	173	6	14%	1%	1%
8	muscle artifact	7	6	0	100%	54%	70%
9	baseline artifact	6	0	0	100%	100%	100%
10	respiration artifact	6	0	0	100%	100%	100%

5.1.9 Okada

Pada algoritma ini ketika dilakukan simulasi dan pengujian menggunakan phantom ECG didapatkan hasil pengujian *heart rate* dan *heart beat*. Untuk lebih ringkasnya dapat dilihat di Tabel 5.17 yang merupakan hasil pengujian *heart rate* pada algoritma ini dan Tabel 5.18 yang merupakan hasil pengujian *heart beat* pada algoritma ini.

Tabel 5.17 Hasil pengujian *heart rate* pada pengujian algoritma "Okada"

No	Pengujian	Target Output (bpm)	Realisasi (bpm)	Error
1	Simulasi	80	57	29%
2	Phantom 40 bpm	40	108	170%
3	Phantom 60 bpm	60	84	40%
4	Phantom 80 bpm	80	96	20%
5	Phantom 100 bpm	100	156	56%
6	Phantom 120 bpm	120	132	10%
7	50 Hz noise	80	84	5%
8	muscle artifact	80	84	5%
9	baseline artifact	80	120	50%
10	respiration artifact	80	96	20%

Tabel 5.18 Hasil pengujian *heart beat* pada pengujian algoritma "Okada"

No	Pengujian	TP	FP	FN	Se	P	F1
1	Simulasi	4	28	3	57%	13%	21%
2	Phantom 40 bpm	3	78	0	100%	4%	7%
3	Phantom 60 bpm	3	62	2	60%	5%	9%
4	Phantom 80 bpm	4	49	3	57%	8%	13%
5	Phantom 100 bpm	6	32	3	67%	16%	26%
6	Phantom 120 bpm	6	36	4	60%	14%	23%
7	50 Hz noise	7	98	0	100%	7%	13%
8	muscle artifact	6	58	0	100%	9%	17%
9	baseline artifact	5	36	2	71%	12%	21%
10	respiration artifact	6	64	1	86%	9%	16%

5.1.10 Pan and Tompkins

Pada algoritma ini ketika dilakukan simulasi dan pengujian menggunakan phantom ECG didapatkan hasil pengujian *heart rate* dan *heart beat*. Untuk lebih ringkasnya dapat dilihat di Tabel 5.19 yang merupakan hasil pengujian *heart rate* pada algoritma ini dan Tabel 5.20 yang merupakan hasil pengujian *heart beat* pada algoritma ini.

Tabel 5.19 Hasil pengujian *heart rate* pada pengujian algoritma "Pan and Tompkins"

No	Pengujian	Target Output (bpm)	Realisasi (bpm)	Error
1	Simulasi	80	47	41%
2	Phantom 40 bpm	40	46	15%
3	Phantom 60 bpm	60	66	10%
4	Phantom 80 bpm	80	86	8%
5	Phantom 100 bpm	100	113	13%
6	Phantom 120 bpm	120	126	5%
7	50 Hz noise	80	86	8%
8	muscle artifact	80	86	8%
9	baseline artifact	80	93	16%
10	respiration artifact	80	93	16%

Tabel 5.20 Hasil pengujian *heart beat* pada pengujian algoritma "Pan and Tompkins"

No	Pengujian	TP	FP	FN	Se	P	F1
1	Simulasi	4	1	0	100%	80%	89%
2	Phantom 40 bpm	7	0	0	100%	100%	100%
3	Phantom 60 bpm	9	0	0	100%	100%	100%
4	Phantom 80 bpm	13	0	0	100%	100%	100%
5	Phantom 100 bpm	16	0	0	100%	100%	100%
6	Phantom 120 bpm	19	0	0	100%	100%	100%
7	50 Hz noise	13	2	0	100%	87%	93%
8	muscle artifact	13	0	0	100%	100%	100%
9	baseline artifact	13	0	0	100%	100%	100%
10	respiration artifact	13	0	0	100%	100%	100%

5.2 Analisa tiap parameter

Ada 2 jenis pengujian yang digunakan dalam penelitian ini yaitu pengujian *heart rate* dan pengujian *heart beat.* Dari Tabel 5.1 sampai Tabel 5.20 di atas, maka dapat dilihat rata-rata hasilnya sehingga dapat diketahui metode mana yang paling baik dari aspek hasil pengujian.

5.2.1 Pengujian *heart rate*

Pada pengujian *heart rate*, dihitung error tiap algoritma pada 6 kali pengujian yang telah dilakukan. Dari error tersebut, dihitung rata-rata error tiap algoritmanya. Kemudian dari rata-rata error tersebut dapat diketahui bahwa pada pengujian *heart*

Tabel 5.21 Rata-rata error tiap algoritma setelah 10 kali pengujian *heart rate*.

No	Algoritma	Rata-rata error
1	Pan and Tompkins	14%
2	Balda	15%
3	Moriet-Mahoudeux	22%
4	Engelese and Zeelenberg	22%
5	Fraden and Neuman	26%
6	Gustafson	35%
7	Okada	40%
8	Holsinger	43%
9	Menard	44%
10	Ahlstrom and Tompkins	89%

rate ini, algortima Engelese and Zeelenberg mendapatkan rata-rata error paling sedikit yakni 4%. Kemudian algoritma yang mendapatkan rata-rata error paling besar yaitu algoritma Ahlstrom and Tompkins sebesar 95%.

Kemudian dilakukan juga pengujian dengan menggunakan *noise* untuk mengetahui algortima mana yang paling tahan terhadap *noise* dan mana algoritma yang tidak tahan terhadap *noise*. Dari 4 percobaan tambahan dengan menggunakan *noise* dengan bermacam jenis *noise* maka dapat disimpulkan bahwa algoritma Pan and Tompkins memiliki rata-rata error sebesar 12%, yang artinya algoritma Pan and Tompkins paling tahan terhadap *noise*. Sebaliknya, pada algoritma Holsinger yang memiliki rata-rata error terbesar yakni 93%, yang artinya algoritma Holsinger paling tidak tahan terhadap *noise*.

Sehingga dapat disimpulkan pada Tabel 5.21 di atas bahwa dari total keseluruhan pengujian yang berjumlah 10 maka dapat dirata-rata sehingga menghasilkan kesimpulan yang dapat ditarik. Algoritma Pan and Tompkins memiliki rata-rata error terkecil dalam 10 kali pengujian baik dengan *noise* maupun tidak, yakni sebesar 14% dan algoritma Ahlstrom and Tompkins memiliki rata-rata error terbesar dalam 10 kali pengujian baik dengan *noise* maupun tidak, yaitu sebesar 89%. Sehingga bisa disimpulkan bahwa untuk mendeteksi *heart rate* algoritma Pan and Tompkins bagus untuk digunakan karena error nya yang relatif kecil.

5.2.2 Pengujian *heart beat*

Sedangkan pada pengujian heart beat, dihitung *sensitivity* (Se), *positive predictivity* (+P), dan *F1 measure* (F1). Tabel 34 di atas diurutkan berdasarkan nilai *F1 Measure* yang paling besar karena *F1 Measure* merupakan rasio

Tabel 5.22 Hasil sensitivity (Se), *positive predictivity* (+P), dan *F1 measure* (F1) tiap algoritma dalam 10 kali pengujian *heart beat.*

No	Algoritma	Se	P	F1
1	Pan and Tompkins	100%	97%	98%
2	Engelese and Zeelenberg	91%	85%	87%
3	Fraden and Neuman	92%	80%	83%
4	Gustafson	82%	83%	82%
5	Moriet-Mahoudeux	76%	90%	81%
6	Holsinger	82%	70%	71%
7	Balda	97%	61%	69%
8	Menard	91%	23%	33%
9	Okada	76%	10%	16%
10	Ahlstrom and Tompkins	9%	#DIV/0!	3%

perbandingan *sensitivity* dan *positive predictivity*. Semakin besar nilai *F1 Measure* maka algoritma tersebut makin bagus dalam pendeteksian *heart beat*. Oleh karena itu pada 6 pengujian pertama dengan bermacam-macam besar *heart rate*, algoritma Engelese and Zeelenberg merupakan algoritma yang paling besar nilai F1 nya yaitu sebesar 100% sehingga algoritma ini juga bagus untuk digunakan dalam deteksi *heart beat* untuk penggunaan sinyal tanpa *noise*. Sedangkan untuk algoritma Ahlstrom and Tompkins kurang begitu bagus untuk deteksi *heart beat* tanpa *noise* karena nilai *F1 Measure*nya paling kecil yaitu 4%.

Kemudian dilakukan juga pengujian dengan menggunakan *noise* untuk mengetahui algortima mana yang paling tahan terhadap *noise* dan mana algoritma yang tidak tahan terhadap *noise*. Dari 4 percobaan tambahan dengan menggunakan *noise* dengan bermacam jenis *noise* maka dapat disimpulkan bahwa algoritma Pan and Tompkins memiliki rata-rata F1 sebesar 98%, yang artinya algoritma Pan and Tompkins paling tahan terhadap *noise* dalam hal mendeteksi *heart beat*. Sebaliknya, pada algoritma Ahlstrom and Tompkins yang memiliki rata-rata F1 terkecil yakni 2%, yang artinya algoritma Ahlstrom and Tompkins paling tidak tahan terhadap *noise*.

Sehingga dapat disimpulkan pada Tabel 5.22 di atas bahwa dari total keseluruhan pengujian yang berjumlah 10 maka dapat dirata-rata sehingga menghasilkan kesimpulan yang dapat ditarik. Algoritma Pan and Tompkins memiliki rata-rata F1 terbesar dalam 10 kali pengujian baik dengan *noise* maupun tidak, yakni sebesar 98% dan algoritma Ahlstrom and Tompkins memiliki rata-rata F1 terkecil dalam 10 kali pengujian baik dengan *noise* maupun tidak, yaitu sebesar 3%. Sehingga bisa disimpulkan bahwa untuk mendeteksi *heart beat* algoritma Pan and Tompkins bagus untuk digunakan karena F1 yang besar.

5.3 Analisa dengan Sinyal Uji Tubuh

Pada pengujian dengan menggunakan sinyal uji berupa sinyal ECG dari tubuh manusia, dalam percobaan dibandingkan pula dengan hasil dari fitur ‘heart rate’ dengan menggunakan pulse oximeter yang ada pada gawai. Namun karena ketidak tahuan akan berapa besar akurasi fitur ini maka penulis sengaja tidak membuat hasil dari fitur yang ada di gawai sebagai acuan untuk menghitung error. Dapat dilihat pada Tabel 4.21, dalam pengujian heart rate ada beberapa algoritma yang mendekati hasil yang sama yang ditunjukkan oleh fitur pada gawai. Algoritma-algoritma tersebut adalah algoritma Pan and Tompkins, Engelese and Zeelenberg, dan Okada. Sedangkan untuk hasil pengujian heart beat yang dapat dilihat pada Tabel 4.22, ada beberapa algoritma yang nilai F1 nya cukup tinggi (diatas 90%), yakni Pan Tompkins dan Engelese and Zeelenberg. Sehingga bisa kita simpulkan

bahwa sistem ECG portable yang ada pada mikrokontroler ini bisa digunakan juga dengan input sinyal tubuh manusia.

5.4 Analisa Performansi

Dari proses dan pengujian yang telah dilakukan maka dapat dilakukan analisa dari hasil performansi yang telah didapatkan. Sistem ini merupakan sistem yang menggunakan mikrokontroler yang memiliki beberapa keterbatasan sehingga selain akurasi yang tinggi diperlukan juga penggunaan memori yang sedikit dan waktu komputasi yang cepat. Untuk hasil akurasi, seperti pada yang telah dijelaskan pada penelitian yang dilakukan Mohamed Elgendi dkk [16], hasil pengujian sensitivity (Se) dan positive predictivity (+P) algoritma yang memiliki performansi paling bagus dalam sistem ini yakni Pan and Tompkins yaitu 99.76% dan 99.56%. Sedangkan pada sistem ini hasil Se dan +P pada algoritma Pan and Tompkins yaitu sebesar 100% dan 97%. Kemudian untuk penggunaan waktu komputasi, meskipun algoritma Pan and Tompkins merupakan metode yang memiliki waktu komputasi yang lebih banyak dibanding algoritma lain yang digunakan pada sistem ini, namun masih termasuk cepat yaitu 5 ms. Terakhir untuk memorinya, total keseluruhan sistem ini yang berisi 10 metode hanya menggunakan 220 KB dari total 1 MB *flash memory* yang dimiliki mikrokontroler. Sehingga dari hal ini dapat disimpulkan bahwa untuk total 10 metode yang digunakan dalam sistem ini memiliki akurasi yang baik pada beberapa metode, komputasi yang ringan, waktu komputasi yang cepat, dan penggunaan memori yang sedikit.

5.5 Klasifikasi Aritmia

Klasifikasi aritmia dilakukan dengan membandingkan nilai hasil *heart rate* dengan kisaran *heart rate* normal. Nilai *heart rate* normal besarnya ada dalam rentang 60-100 bpm. Sehingga jika nilai *heart rate* lebih kecil daripada 60 bpm maka sistem akan mengklasifikasikan sebagai aritmia bradikardia, begitu pula sebaliknya jika nilai *heart rate* lebih besar daripada 100 bpm maka sistem akan mengklasifikasikan sebagai aritmia takikardia.

BAB 6 KESIMPULAN DAN SARAN

6.1 Kesimpulan

Berdasarkan beberapa hasil percobaan yang dilakukan dapat diambil beberapa kesimpulan, yaitu hasil percobaan menunjukkan bahwa metode yang digunakan pada penelitian untuk mendeteksi *heart rate* (dengan sinyal tanpa noise) yang paling bagus adalah algortima Engelese and Zeelenberg. Hal ini disimpulkan dari nilai rata-rata error yang didapatkan oleh algortima Engelese and Zeelenberg paling sedikit yakni 4%. Kemudian algoritma yang mendapatkan rata-rata error paling besar yaitu algoritma Ahlstrom and Tompkins sebesar 95%. Namun hal berbeda didapatkan jika menggunakan sinyal uji yang telah diberi *noise*. Dari hasil percobaan dengan menggunakan 4 jenis *noise*, maka dapat dilihat bahwa algoritma Pan and Tompkins memiliki rata-rata error sebesar 12%, yang artinya algoritma Pan and Tompkins paling tahan terhadap *noise*. Sebaliknya, pada algoritma Holsinger yang memiliki rata-rata error terbesar yakni 93%, yang artinya algoritma Holsinger paling tidak tahan terhadap *noise*. Sehingga jika dirata-rata semua error dari 10 kali percobaan didapatkan kesimpulan bahwa algoritma Pan and Tompkins memiliki rata-rata error terkecil dalam 10 kali pengujian baik dengan *noise* maupun tidak, yakni sebesar 14% dan algoritma Ahlstrom and Tompkins memiliki rata-rata error terbesar dalam 10 kali pengujian baik dengan *noise* maupun tidak, yaitu sebesar 89%. Sehingga bisa disimpulkan bahwa untuk mendeteksi *heart rate* algoritma Pan and Tompkins bagus untuk digunakan karena error nya yang relatif kecil. Kemudian sistem juga diuji coba dengan menggunakan input sinyal tubuh. Dari hasil percobaan bisa diketahui jika sistem bisa digunakan untuk mendeteksi sinyal yang berasal dari sinyal tubuh manusia. Dan metode yang hasil pengujiannya paling mendekati hasil pengujian dengan menggunakan gawai adalah metode Pan and Tompkins.

Selain percobaan untuk mendeteksi *heart rate,* dilakukan pula percobaan untuk mendeteksi *heart beat.* Untuk mendapatkan algoritma paling bagus untuk mendeteksi *heart beat* dipilih algoritma yang memiliki nilai *F1 Measure* paling besar karena *F1 Measure* merupakan rasio perbandingan *sensitivity* dan *positive predictivity*. Semakin besar nilai *F1 Measure* maka algoritma tersebut makin bagus dalam pendeteksian *heart beat*. Ketika dilakukan uji coba untuk mengetahui algoritma mana yang bagus untuk mendeteksi heart beat, awalnya dilakukan uji dengan sinyal tanpa *noise* dan dapat diketahui jika algoritma Engelese and Zeelenberg merupakan algoritma yang paling besar nilai F1 nya yaitu sebesar 100% sehingga algoritma ini juga bagus untuk digunakan dalam deteksi *heart beat*

dengan sinyal uji tanpa *noise*. Sedangkan untuk algoritma Ahlstrom and Tompkins kurang begitu bagus untuk deteksi *heart beat* karena nilai *F1 Measure*nya paling kecil yaitu 4%. Namun hal berbeda didapatkan jika menggunakan sinyal uji yang telah diberi *noise*. Dari hasil percobaan dengan menggunakan 4 jenis *noise*, maka dapat dilihat bahwa algoritma Pan and Tompkins memiliki nilai F1 sebesar 98%, yang artinya algoritma Pan and Tompkins paling tahan terhadap *noise*. Sebaliknya, pada algoritma Ahlstrom and Tompkins yang memiliki rata-rata F1 terkecil yakni 2%, yang artinya algoritma Ahlstrom and Tompkins paling tidak tahan terhadap *noise*. Sehingga jika dirata-rata semua error dari 10 kali percobaan didapatkan kesimpulan bahwa algoritma Pan and Tompkins memiliki rata-rata nilai F1 terbesar dalam 10 kali pengujian baik dengan *noise* maupun tidak, yakni sebesar 98% dan algoritma Ahlstrom and Tompkins memiliki rata-rata nilai F1 terkecil dalam 10 kali pengujian baik dengan *noise* maupun tidak, yaitu sebesar 3%. Sehingga bisa disimpulkan bahwa untuk mendeteksi *heart beat* algoritma Pan and Tompkins bagus untuk digunakan karena nilai F1 yang besar dari beberapa kali percobaan. Kemudian sistem juga diuji coba dengan menggunakan input sinyal tubuh. Dari hasil percobaan bisa diketahui jika sistem bisa digunakan untuk mendeteksi sinyal yang berasal dari sinyal tubuh manusia. Dan metode yang hasil pengujiannya F1 nya paling besar adalah metode Pan and Tompkins.

Aspek penting lainnya selain akurasi yaitu penggunaan memori dan waktu komputasi. Untuk waktu komputasi, penggunaan algoritma-algoritma ini memiliki waktu komputasi yang relative kecil yaitu 2.778 ms dan 5 ms. Algoritma yang memiliki waktu komputasi terbesar yaitu Pan and Tompkins sebesar 5 ms namun ini masih dapat dikatakan cepat. Selain itu untuk penggunaan memorinya, juga kurang dari setengah kapasitas memori yang dimiliki mikrokontroler yaitu 220 KB dari 1 MB flash memory yang dimiliki oleh mikrokontroler STM32F407. Sehingga dari hal ini dapat disimpulkan bahwa untuk total 10 metode yang digunakan dalam sistem ini memiliki akurasi yang baik pada beberapa metode, komputasi yang ringan, waktu komputasi yang cepat, dan penggunaan memori yang sedikit.

Adanya error pada tiap kali pengujian *heart rate* bisa terjadi karena jumlah data yang diambil untuk pengolahan sinyal terbatas. Selain itu, untuk klasifikasi aritmia digunakan parameter *heart rate*. Jika nilai *heart rate* yang terdeteksi lebih kecil daripada 60 bpm maka akan diklasifikasikan sebagai aritmia bradikardia, begitu pula sebaliknya jika nilai *heart rate* lebih besar daripada 100 bpm akan diklasifikasikan sebagai aritmia takikardia.

6.2 Saran

Saran yang dapat penulis berikan untuk penelitian selanjutnya yaitu untuk meningkatkan keakurasian deteksi *heart* rate perlu digunakan semakin banyak

jumlah data karena jumlah data yang sedikit kurang merepresentasikan keseluruhan data. Selain itu untuk pengklasifikasian aritmia dapat dilakukan selain membandingkan nilai *heart rate* saja. Dan untuk pengembangan selanjutnya bisa dilakukan penelitian dengan metode-metode yang lebih baru.

DAFTAR PUSTAKA

[1] "Kemenkes," Biro Komunikasi dan Pelayanan Masyarakat, Kementerian Kesehatan RI, 29 July 2017. [Online]. Available: https://www.kemkes.go.id/article/view/17073100005/penyakit-jantung-penyebab-kematian-tertinggi-kemenkes-ingatkan-cerdik-.html. [Diakses 27 June 2020].

[2] "Alodokter," 1 August 2019. [Online]. Available: https://www.alodokter.com/aritmia. [Diakses 27 June 2020].

[3] L. O. Alifariki, "Jantung, Pembuluh Darah, dan Tekanan Darah," dalam *Epidemiologi Hipertensi : Sebuah Tinjauan Berbasis Riset*, LeutikaPrio, pp. 1-2.

[4] V. Essianda, W. Istanto dan A. Ismail, "MORTALITAS OPERASI JANTUNG GANTI KATUP DI RSUP DR. KARIADI SEMARANG PERIODE JANUARI 2014," 2015.

[5] L. Irawati, "Aktifitas Listrik pada Otot Jantung," *Jurnal Kesehatan Andalas,* vol. 4, no. 2, 2015.

[6] "ECG Learning Center," Eccles Health Sciences Library University of Utah, [Online]. Available: https://ecg.utah.edu/lesson/1#spacialorientation. [Diakses 6 June 2020].

[7] "WHO," 2011. [Online]. Available: https://www.who.int/medical_devices/innovation/electrocardiograph.pdf. [Diakses 23 July 2020].

[8] "Maxim Integrated," Maxim Integrated, 10 May 2010. [Online]. Available: https://www.maximintegrated.com/en/design/technical-documents/tutorials/4/4693.html. [Diakses 23 July 2020].

[9] R. J. Noble, J. S. Hillis dan D. A. Rothbaum, "Chapter 33: Electrocardiography," dalam *Clinical Methods: The History, Physical, and Laboratory Examinations. 3rd edition.*, Boston, Butterworths, 1990.

[10] S. Bhunia dan M. Tehranipoor, "Chapter 2 : A Quick Overview of Electronic Hardware," dalam *Hardware Security : A Hands-on Learning Approach*, Morgan Kaufmann Publisher, 2019, pp. 23-45.

[11] "MikroE," MikroElektronika, [Online]. Available: https://www.mikroe.com/mikromedia-5-for-stm32-capacitive. [Diakses 28 June 2020].

[12] R. Setiawan, "Bab 7 : A/D Converter," dalam *Mikrokontroler dsPIC,* Indomedia Pustaka, 2019.

[13] "MikroE," MikroElektronika, [Online]. Available: https://www.mikroe.com/ecg-click. [Diakses 28 June 2020].

[14] A. Rizal, I. Y. Setiadi, R. Magdalena dan V. Suryani, “Telkom University,” June 2014. [Online]. Available: https://achmadrizal.staff.telkomuniversity.ac.id/wp-content/uploads/sites/11/2014/06/16651090-Simulator-Ecg-Snpte.pdf. [Diakses 5 July 2020].

[15] “Contec Medical,” Contec Medical System, [Online]. Available: http://www.contecmed.com/index.php?page=shop.product_details&flypage=flypage.tpl&product_id=102&category_id=18&option=com_virtuemart&Itemid=601. [Diakses 5 July 2020].

[16] M. Elgendi, B. Eskofier, S. Dokos dan D. Abbott, “Revisiting QRS Detection Methodologies for Portable, Wearable, Battery-Operated, and Wireless ECG Systems,” *PLOS ONE,* vol. 9, no. 1, 2014.

[17] “Shodhganga,” [Online]. Available: https://shodhganga.inflibnet.ac.in/bitstream/10603/125937/19/13_chapter%204.pdf. [Diakses 24 June 2020].

[18] P. Morizet-Mahoudeux, C. Moreau dan D. Moreau, “Simple microprocessor-based system for on-line e.c.g. arrhythmia analysis,” *Medical & Biological Engineering & Computing,* vol. 19, pp. 497-500, 1981.

[19] J. Fraden dan M. R. Neuman, “QRS wave detection,” *Medical & Biological Engineering & Computing,* vol. 18, pp. 125-132, 1980.

[20] C. Vidal, P. Charnay dan P. Arce, “Enhancement of a QRS detection algorithm based on the first derivative, using techniques of a QRS detector algorithm based on non-linear transformations,” dalam *IFMBE Proceedings,* 2008.

[21] R. Kher, D. Vala, T. Pawar dan V. K. Thakar, “Implementation of Derivative Based QRS Complex Detection Methods,” 2014.

[22] I. Nouira, A. B. Abdallah, I. Kouaja dan M. H. Bedoui, “Comparative Study of QRS Complex Detection in ECG,” *International Journal of Biomedical and Biological Engineering,* vol. 6, no. 11, pp. 593-597, 2012.

[23] D. Gustafson, “Automated VCG interpretation studies using signal analysis technique,” 1977.

[24] A. Menrad, “Dual microprocessor system for cardiovascular data acquisition, processing and recording,” *IEEE Int. Conf. Industrial Elect. Contr. Instrument,* pp. 64-69, 1981.

[25] W. Holsinger, K. Kempner dan M. Miller, “QRS preprocessor based on digital differentiation,” *IEEE Transactions on Biomedical Engineering,* vol. 18, pp. 212-217, 1971.

[26] R. Balda, G. Diller, E. Deardoff, J. Doue dan P. Hsieh, “The HP ECG analysis program,” *Trends in Computer-Processed Electrocardiograms,* pp. 197-205, 1977.

[27] M. L. Ahlstrom dan W. J. Tompkins, “Automated high-speed analysis of Holter tapes with microcomputers,” *IEEE Transactions on Biomedical Engineering,* vol. 30, pp. 651-657, 1983.

[28] W. A. H. Engelse dan C. Zeelenberg, “A single scan algorithm for QRS-detection and feature extraction,” *Proc IEEE Computers in Cardiology,* vol. 6, pp. 37-42, 1979.

[29] M. Okada, “A digital filter for the QRS complex detection,” *IEEE Transactions on Biomedical Engineering,* vol. 26, pp. 700-703, 1979.

[30] J. Pan dan W. J. Tompkins, “A real-time QRS detection algorithm,” *IEEE Transactions on Biomedical Engineering,* vol. 32, pp. 230-236, 1985.

[31] “MIT-BIH Normal Sinus Rhythm Database,” Physionet, 1999.

LAMPIRAN

A. *Source Code* Program untuk Algoritma Moriet-Mahoudeux

Algoritma Moriet-Mahoudeux

```
void AFD1()
{
       reset();
       AT=-9999;
       for (i=0;i<=ndata;i++)  if (signal[i]>AT) AT=signal[i];

       AT= 0.3*AT;
       for (i=1;i<= ndata;i++) dev[i]=(signal[i+1]-signal[i-1]);
       for (i=1;i<= ndata;i++)
       {
         if ((dev[i]>0.5) && (dev[i+1]>0.5) && (dev[i+2]>0.5))
         {
           for (j=i+3;j<= i+24;j++)
           {
            if ((dev[j]<0.3) && (dev[j+1]<0.3))
             y[i]=1;
            else
             y[i]=0;
           }
         }
       }
       Draw_signal();
       QRS();
}
```

B. *Source Code* Program untuk Algoritma Fraden and Neuman

Algoritma Fraden-Neumann

```
void AFD2()
{
       reset();
       AT=-9999;
       for (i=0;i<=ndata;i++)  if (signal[i]>AT) AT=signal[i];

       AT= 0.4*AT;
       for (i=0;i<=ndata;i++)
       {
         if (signal[i]>=0)
           y0[i]=signal[i];
         else
           y0[i]=-signal[i];
       }

       for (i=0;i<=ndata;i++)
       {
         if (y0[i]>=AT)
```

```
            y1[i]=y0[i];
          else
            y1[i]=AT;
        }
        for (i=0;i<=ndata;i++)    dev[i]=y1[i+1]-y1[i-1];

        for (i=0;i<=ndata;i++)
        {
          if (dev[i]>0.7)
            y[i]=1;
          else
            y[i]=0;
        }
        Draw_signal();
        QRS();
}
```

C. *Source Code* Program untuk Algoritma Gustafson

Algoritma Gustafson

```
void AFD3()
{       reset();

        for (i=0;i<=ndata;i++)    dev[i]=signal[i+1]-signal[i-1];

        for (i=0;i<=ndata;i++)
        {
          if ((dev[i]>0.15) && (dev[i+1]>0.15) && (dev[i+2]>0.15)
&& (dev[i+3]>0.15))
          {
            if  ((dev[i+1]*signal[i+1]>0) &&
(dev[i+2]*signal[i+2]>0))
              y[i]=1;
            else
              y[i]=0;
          }
        }
        Draw_signal();
        QRS();
}
```

D. *Source Code* Program untuk Algoritma Menard

Algoritma Menard

```
void Menard()
{
        reset();
       for (i=0;i<=ndata;i++)
      dev[i]=-2*signal[i-2]-signal[i-1]+signal[i+1]+2*signal[i+2];
```

```
        AT=-9999;
        for (i=2;i<=ndata;i++)
          if (signal[i]>AT)  AT=signal[i];

        AT= 0.7*AT;
        for (i=0;i<=ndata;i++)
        {
          if (dev[i]>AT)
            y[i]=1;
          else
            y[i]=0;
        }
        Draw_signal();
        QRS();
}
```

E. *Source Code* Program untuk Algoritma Holsinger

Algoritma Holsinger

```
void Holsinger()
{
        reset();
        for (i=0;i<=ndata;i++)     dev[i]=signal[i+1]-signal[i-1];

        for (i=0;i<=ndata;i++)
        {
          if ((dev[i]>0.45) && ((dev[i+1]>0.45) || (dev[i+2]>0.45)
|| (dev[i+3]>0.45)))
            y[i]=1 ;
          else
            y[i]=0;
        }
        Draw_signal();
        QRS();
}
```

F. *Source Code* Program untuk Algoritma Balda

Algoritma Balda

```
void Balda()
{
        reset();
        for (i=2;i<=ndata;i++)
       y0[i]=abs(signal[i+1]-signal[i-1]);
        for (i=2;i<=ndata;i++)
          y1[i]=abs(signal[i+2]-2*signal[i]+signal[i-2]);
        for (i=2;i<=ndata;i++)
      y2[i]=1.3*y0[i]+1.1*y1[i];
```

```
        for (i=2;i<=ndata;i++)
        {
          if (y2[i]>=1)
          {
            k=0;
            for (j=i+1;j<=i+8;j++)
              if (y2[j]>=1) k=k+1;
                 if (k>=6)
                    y[i]=1;
                 else
                    y[i]=0;
          }
        }
        Draw_signal();
        QRS();
}
```

G. *Source Code* Program untuk Algoritma Ahlstrom and Tompkins

Algoritma Ahlstorm and Tompkin

```
void Ahlstrom_Tompkins()
{
        reset();
        for (i=3;i<=ndata;i++)
       y0[i]=abs(signal[i+1]-signal[i-1]);

        for (i=3;i<=ndata;i++)
       y1[i]=(y0[i-1]+2*y0[i]+y0[i+1])/4;

        for (i=3;i<=ndata;i++)
       y2[i]=abs(signal[i+2]-2*signal[i]+signal[i-2]);

        for (i=3;i<=ndata;i++)     dev[i]=y1[i]+y2[i];

        PT=-9999;
        for (i=3;i<=ndata;i++)
          if (dev[i]>PT) PT=dev[i];

        PT= 0.8*PT;
        SST=-9999;
        for (i=3;i<=ndata;i++)
          if (dev[i]>SST) SST=dev[i];

        SST= 0.1*SST;
        for (i=3;i<=ndata;i++)
        {
          if (dev[i]>=PT)
          {
            k=0;
            for (j=i+1;j<= i+6;j++)
              if (dev[j]>=SST) k=k+1;

            if (k==6)
              y[i]=1;
```

```
            else
              y[i]=0;
           }
       }
       Draw_signal();
       QRS();
}
```

H. *Source Code* Program untuk Algoritma Engelese and Zeelenberg

Algoritma Engelese and Zeelenberg

```
void Engelese_Zeelenberg()
{
       reset();
       for (i=4;i<=ndata;i++)
      y0[i]=signal[i]-signal[i-4];

       for (i=4;i<=ndata;i++)
      y1[i]=y0[i]+4*y0[i-1]+6*y0[i-2]+4*y0[i-3]+y0[i-4];

       for (i=4;i<=ndata;i++)
       {
         if (y1[i]>21)
         {
           k=0;
           mm=0;
        //0.16->160ms deteksi ad overcross atau tidak dalam waktu
160ms
           for (j=1;j<=floor(0.16*ndata/fs);j++)
             if (y1[i+j]>21) k=k+1;

           if (k==0)
           {
            for (j=1;j<=40;j++)
            {
             if (y1[i+j]<-21) mm=mm+1;
             for (kk=j;kk<=40;kk++)
               if (((y1[i+j]<-21) && (y1[i+kk]>21) && (y1[i+1]<-
21)) ||
                   ((y1[i+j]<-21) && (y1[i+kk]>21) && (y1[i+1]<-21)))
                         mm=mm+1;
            }
            if (mm>0)
              y[i]=1;
            else
              y[i]=0;
           }
         }
       }
       Draw_signal();
       QRS();
}
```

I. *Source Code* Program untuk Algoritma Okada

Algoritma Okada

```
void Okada()
{
       reset();
       mm=3;
       sum=0;
       for (i=1;i<=ndata;i++)
      y0[i]=(signal[i-1]+2*signal[i]+signal[i+1])/4;

       for (i=mm;i<=ndata-mm;i++)
       {
         for(k=i-mm;k<=i+mm;k++)
           sum=sum+y0[k];

         y1[i]=(1/(2*mm+1))*sum;
       }
       sum=0;

       for (i=mm;i<=ndata-mm;i++)    y2[i]=(y0[i]-y1[i])*(y0[i]-
y1[i]);
       for (i=mm;i<=ndata-mm;i++)
       {
         for (k=i-mm;k<=i+mm;k++)
           sum=sum+y2[k];

         y3[i]=y2[i]*sum*sum;
       }
       for (i=mm;i<=ndata-mm;i++)
       {
         if (((y0[i] - y0[i-mm])>0) && ((y0[i] - y0[i+mm])>0))
           y4[i]=y3[i];
         else
           y4[i]=0;
       }

       AT=-9999;
       for (i=mm;i<=ndata-mm;i++)
         if (y4[i]>AT) AT=y4[i];

       AT= 0.125*AT;
       for (i=mm;i<=ndata-mm;i++)
       {
         if (y4[i]>AT)
           y[i]=1;
         else
           y[i]=0;
       }
       Draw_signal();
       QRS();
}
```

J. *Source Code* Program untuk Algoritma Pan and Tompkins

Algoritma Pan-Tompkin

```
void qrs_pan_tompkin()
{
      fs =200;t_scale=2;
      ndata =1260;
      ClearRect2(40,35,110,60);
      IntToStr(fs, Label17_Caption);
      DrawLabel(&Label17);

      ClearRect2(40,95,110,120);
      IntToStr(ndata, Label19_Caption);
      DrawLabel(&Label19);

      // Tutup jumlah data dan heart rate
      ClearRect(40,155,100,175);
      ClearRect(30,220,120,280);

      for (i=0;i<=ndata;i++)
       y[i]=0;y0[i]=0;y1[i]=0;y2[i]=0;y3[i]=0;y4[i]=0;y5[i]=0;

      for (i=32;i<=ndata;i++)
      {
      // LPF
         y[i]= 2*y[i-1]-y[i-2]+signal2[i]-2*signal2[i-
6]+signal2[i-12];
         y0[i]= y0[i-1]+signal2[i]-signal2[i-32];
      // HPF
         y1[i]= y[i-16]-(y0[i-1]+y[i]-y[i-32])/32;
      // DERIVATIVE
         y2[i]= (2*y1[i]+y1[i-1]-y1[i-3]-2*y1[i-4])/8;
      // SQUARING
         y3[i]= y2[i]*y2[i];
      }
      mm=30;     //150 ms = 30 sample (fs =200 Hz)
      for (i=mm;i<=ndata;i++)
      {
         for (k=1; k<= mm; k++)
         {
        // MAV
           y4[i]= y4[i]+y3[i-(mm-k)];
         }
         y4[i]= y4[i]/mm;
      }

      AT=-9999;
      for (i=0;i<=ndata;i++)
      {
          if (y4[i]>AT) AT=y4[i];
      }
      for (i=0;i<=ndata;i++)
      {
       // THRESHOLD
          if (y4[i]>AT*0.5)
              y[i]=1;
          else
```

```
            y[i]=0;
      }
      // Gambar Semua Sinyal
      TFT_Set_Pen(CL_YELLOW, 1);
      time_2();
      for (i=1;i<=(ndata-18);i++)
      {
         TFT_Line(((i-1)/t_scale+155),80-(signal2[i-
1]*70),(i/t_scale)+155,80-(signal2[i]*70));
         TFT_Line(((i-1)/t_scale+155),260-
(y[i+31]*80),(i/t_scale)+155,260-(y[i+32]*80));
         Delay_ms(5);
      }
      QRS();
}
```

BIODATA PENULIS

Penulis dilahirkan di Bogor pada tanggal 30 September 1997. Penulis merupakan anak tunggal. Penulis menempuh pendidikan sekolah dasar di SD Tarakanita Citra Raya dan meneruskan ke jenjang sekolah menengah pertama di SMP Tarakanita Citra Raya dan melanjutkan ke sekolah menengah atas di SMAN 8 Kota Tangerang. Kemudian penulis meneruskan ke jenjang pendidikan yang lebih tinggi ke Departemen Teknik Biomedik, Institut Teknologi Sepuluh Nopember. Selama menempuh masa perkuliahan, penulis juga aktif dalam berbagai organisasi dan kepanitiaan, salah satunya yaitu Kalpataru Elektro ITS dan departemen MSDM HIMATEKTRO ITS. Selain itu, penulis juga aktif sebagai asisten Laboratorium Instrumentasi dan Pengolahan Sinyal Biomedika.

E-mail : yeni.ws30@gmail.com

BIOMEDICAL ENGINEERING ITS
2020