

# Deteksi Dini Aritmia Jantung Melalui Denyut Nadi Menggunakan Algoritma *Grammatical Evolution*

Aulia el hakim<sup>1</sup>, Sukanto<sup>2</sup>

<sup>1,2</sup>Politeknik Negeri Madiun

Jl. Serayu No.84 Madiun, 0351-452970

aim@pnm.ac.id<sup>1</sup>

**Abstrak** –Aritmia jantung merupakan salah satu kelainan jantung yang menyebabkan detak jantung menjadi terlalu cepat (*Takikardia*) atau terlalu lambat (*Bradikardia*). Gejala aritmia jantung tidak selalu mudah dikenali. Dibutuhkan suatu metode atau algoritma yang dapat digunakan untuk deteksi dini aritmia jantung. Pada penelitian ini digunakan algoritma *grammatical evolution* dalam mendeteksi pola detak jantung berdasarkan data history. Sistem deteksi Aritmia jantung pada penelitian ini terdiri dari alat deteksi detak jantung melalui denyut nadi berupa pulse sensor, Board Arduino sebagai pengolah data input pulse sensor serta algoritma *grammatical evolution* dalam proses menghasilkan fungsi pola detak jantung. Dari hasil ujicoba dihasilkan 3 buah fungsi yang mempunyai nilai fitness yang berbeda-beda serta prosentase eror yang berbeda-beda Untuk GE 1 prosentase eror sebesar 4,01 % dan fitness value sebesar 57,0983. Untuk GE 2 dengan prosentase eror sebesar 6,82% dan fitness value sebesar 54,95, sedangkan untuk GE 3 dengan prosentase eror sebesar 3,62% dan fitness value sebesar 58,2.

**Kata kunci:** Aritmia Jantung, Denyut nadi, pulse sensor, *Grammatical Evolution*

## 1. Pendahuluan

Sejak tahun 1996 penyakit Jantung merupakan penyebab kematian nomor 1 di Indonesia. Tidak hanya di Indonesia, penyakit jantung juga merupakan penyebab kematian nomor 1 di Amerika. Berdasarkan pusat kendali penyakit di amerika terdapat kurang lebih 610,00 orang yang meninggal disebabkan oleh sakit jantung[1]. Peningkatan jumlah kematian di indonesia akibat penyakit jantung disebabkan karena langkanya ahli jantung di Indonesia, sedikitnya alat pendeteksi serangan jantung, tidak rutinnya pengecekan kesehatan jantung secara berkala, serta buruknya gaya hidup masyarakat secara luas.

Salah satu penyebab serangan jantung adalah aritmia atau Gangguan irama jantung merupakan perubahan pada frekuensi dan irama jantung[1]. Aritmia jantung (*heart arrhythmia*) menyebabkan detak jantung menjadi terlalu cepat, terlalu lambat, atau tidak teratur. Pada umumnya gangguan sistem kerja jantung Akan menimbulkan perubahan irama jantung menjadi *terlalu lambat* (**Bradikardia**, jantung berdenyut kurang dari 60 kali permenit) atau *terlalu cepat* (**Takikardia**, jantung berdenyut lebih dari 100 kali permenit). Kedua keadaan tersebut Akan berpengaruh terhadap kerja jantung yang memompa darah ke seluruh tubuh. Bila pada kondisi Bradikardia, jantung berdenyut terlalu lambat, maka jumlah darah yang mengalir di dalam sirkulasi menjadi berkurang, sehingga kebutuhan tubuh tidak terpenuhi. Faktor yang membahayakan adalah bila jumlah darah yang menuju otak menjadi berkurang bahkan minimal sehingga terjadi pingsan. Pada keadaan yang lebih parah dapat menyebabkan stroke. Sebaliknya, bila pada kondisi **Takikardia**, jantung berdenyut terlalu cepat. Maka jantung Akan mengalami kelelahan dan menimbulkan gejala-gejala berdebar yang biasanya disertai perasaan takut karena debaran jantung yang begitu cepat (sampai lebih dari 200 kali permenit). Pada keadaan yang ekstrim dimana bilik jantung berdenyut sangat cepat dan tidak terkendali, maka terjadi kegagalan

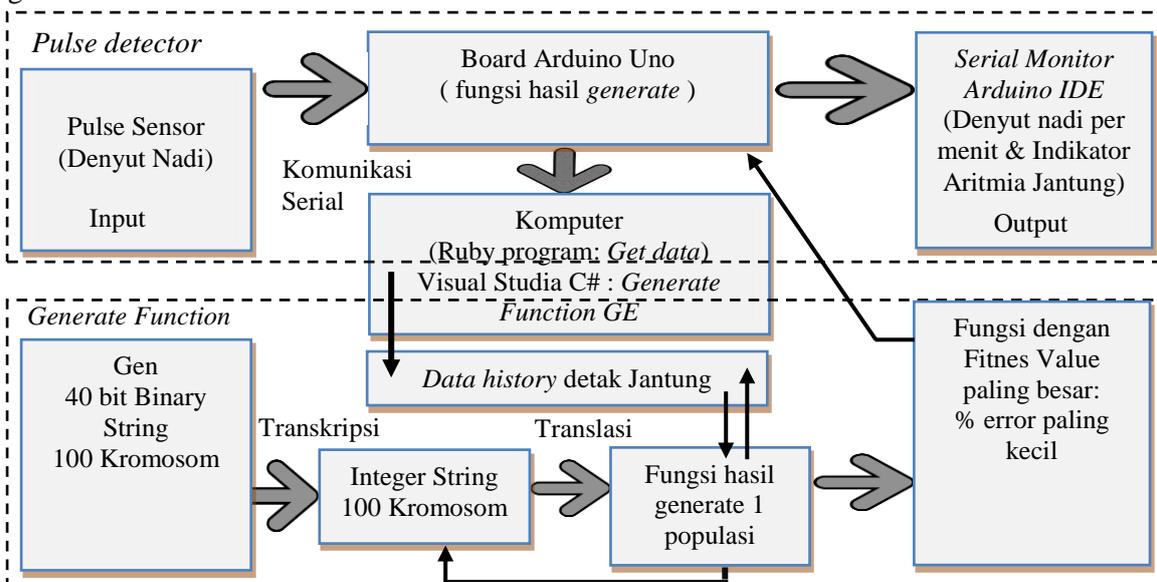
sirkulasi darah yang bila dilakukan pertolongan cepat dengan kejut listrik (DC shock) dapat mengakibatkan kematian.

Tanda dan gejala aritmia jantung atau gangguan irama jantung tidak selalu mudah dikenali. Dari kondisi tersebut maka dibutuhkan suatu metode atau algoritma serta alat yang dapat dipergunakan sebagai pendeteksi dini aritmia jantung. Sehingga pengguna dapat mendeteksi gejala-gejala serangan jantung yang terjadi pada dirinya secara dini dan mandiri. Terdapat beberapa penelitian yang sudah dilakukan terkait deteksi aritmia jantung. Deteksi dapat dilakukan dengan menggunakan ECG yang dihubungkan dengan instrumentasi amplifier dan ditampilkan di komputer melalui software labview[1]. Namun dalam penelitian tersebut tingkat sensitifitas yang didapatkan masih 35% dan membutuhkan perangkat instrumentasi yang mempunyai toleransi error yang kecil. Prediksi atau deteksi dini aritmia jantung juga sudah pernah dilakukan dengan menggunakan Algoritma Neural Network berdasarkan *Heart Rate Time Series*[2] Penelitian tersebut menggunakan Heart Rate variability (HRV) yang didapat dari ECG dan bukan menggunakan sensor Pulse Rate serta penggunaan *Evolutionary Algorithm* untuk deteksi ECG aritmia[3].

Algoritma *Grammatical evolution* merupakan salah satu kecerdasan buatan yang digunakan untuk teknik optimasi dan merupakan salah satu bagian dari *evolutionary algorithm*. *Grammatical evolution* telah banyak digunakan dalam menemukan simbol regresi, Trigonometri identitas, simbol integrasi dan juga digunakan untuk proses prediksi atau peramalan[4], [5]. Pada penelitian ini, *grammatical evolution* digunakan dikarenakan algoritma tersebut dapat menghasilkan fungsi matematis pola aritmia jantung berdasarkan *history* atau data seri pola detak jantung. fungsi matematis dari pola aritmia jantung yang didapat dari Algoritma Grammatical evolution tersebut kemudian dimasukkan kedalam program Arduino. Melalui penelitian yang berjudul “Deteksi Dini Aritmia Jantung Melalui Denyut Nadi menggunakan Algoritma *Grammatical Evolution*”, peneliti bermaksud mengembangkan sebuah sistem deteksi dini Aritmia jantung yang terdiri dari alat pendeteksi detak jantung melalui detak nadi berupa pulse sensor (*heart beat*), Board Arduino sebagai pengolah data input dari *pulse sensor* serta algoritma *grammatical evolution* dalam proses *generate* fungsi pola aritmia jantung.

**2. Metode Penelitian**

Pada dasarnya Penelitian yang berjudul ” Deteksi Dini Aritmia Jantung Melalui Denyut Nadi menggunakan Algoritma Grammatical Evolution”, dapat dibagi menjadi 2 blok diagram yaitu blok diagram “ pulse detector” serta blok diagram “Generate Function” yang dapat dilihat pada gambar 1.



Gambar 1. Blok diagram Sistem

Dari 2 buah blok diagram sistem gambar 1 dapat dijelaskan untuk blok diagram “pulse detector” merupakan blok diagram perangkat hardware yang terdiri dari rangkaian sensor pulse (heart beat), board Arduino, Serial Monitor IDE Arduino serta komunikasi serial dengan Komputer. Fungsi dan Cara kerja dari sistem pada Blok diagram ini adalah sebagai alat untuk mendeteksi denyut nadi dengan menggunakan sensor pulse. Denyut nadi yang terdeteksi tersebut Akan dikirim ke *Board* Arduino dan dihitung berapa jumlah denyut nadi yang terdeteksi dalam 1 menit. Jumlah denyut nadi dalam 1 menit tersebut Akan ditampilkan dalam serial Monitor IDE Arduino dan disimpan kedalam file melalui Software interface Ruby. Pada saat proses pengumpulan data dengan menggunakan pulse detector, dilakukan dengan Cara menghubungkan minimum sistem Arduino dengan komputer. Kemudian dilakukan proses deteksi denyut nadi, banyaknya denyut nadi dalam 1 menit yang terdeteksi oleh rangkaian pulse detector ini kemudian dikirim ke komputer melalui komunikasi serial. Data tersebut Akan disimpan dalam bentuk text dan dimasukkan kedalam kumpulan data *history* rekam medis jantung. Proses penyimpanan data tersebut hanya dilakukan kepada orang yang detak jantungnya normal. Hal ini dilakukan agar didapatkan data-data detak jantung normal yang Akan digunakan untuk proses mendapatkan fungsi pola detak jantung dengan menggunakan grammatical evolution. Data detak jantung normal tersebut digunakan sebagai *Threshold* jantung normal dan sebagai pembanding dengan jantung yang mengalami aritmia atau kelainan ritme jantung.

Tabel 1. Table Grammatical Evolution Untuk deteksi pola detak jantung

Objective [6]:	Mencari Fungsi variabel independen Input (a) dan Variabel Independen Output (B), dengan memberikan sampel data 10 buah data detak jantung melalui denyut Nadi.
Terminal Operands	A,b,c,d ( Variabel independen), 1.0,2.0,3.0,4.0,5.0,6.0
Terminal Operator	x ,+, * ,-,
Fitness Case	Terdapat 10 data input banyaknya BPM dalam 10s seperti { 13,15,15,16} Dan 10 data Output banyaknya BPM dalam 60s seperti {76,77,78,79,80,81,82,83,84,85}
Raw Fitness	Nilai rata-rata prosentase eror dari 10 data yang dibaerikan , semakin besar nilai fitness maka semakin bagus kromosom yang dihasilkan
Standardised Fitness	Sama dengan Raw Fitness
Wrapper	Standard production to generate C function[6]
Parameters	Population Size = 100, Termination when Generation = 50 Prob. Mutation = 1/( jumlah kromosom* jumlah gen), Prob.Crossover = 0.3

Pada Blok diagram ke 2 merupakan Blok diagram “Generate Function” yaitu Blok diagram bagian software yang terdiri mulai dari proses Inisialisasi Populasi, transkripsi kromosom sampai menghasilkan fungsi yang Akan dimasukkan kedalam program Arduino. Cara kerja dari sistem pada Blok diagram ini adalah pertama-tama 100 kromosom Akan dibuat secara acak, dengan 1 kromosom terdiri dari 40 bit bilangan biner. 100 buah kromosom tersebut kemudian Akan dilakukan transkripsi kedalam bentuk 100 bilangan integer. 100 buah kromosom yang sudah berubah bentuk menjadi integer tersebut kemudian dilakukan translasi dengan menggunakan aturan BNF grammar menjadi bentuk fungsi kemudian fungsi yang dihasilkan tersebut dilakukan perhitungan fitness terhadap data input detak jantung, fungsi yang mendapatkan nilai fitness yang besar akan dipertahankan sampai generasi selanjutnya sampai 50 generasi. Rangkuman Informasi Grammatical evolusi yang sudah diadopsi dari[6] dapat dilihat pada Tabel 1. Berikut tahapan-tahapan algoritma grammatical evolution yang digunakan pada penelitian ini.

## 2.1 Inisialisasi Populasi

Pada tahapan inisialisasi ini dapat dibagi menjadi beberapa langkah, antara lain pertama – tama menentukan maksimum gen dalam 1 kromosom [max\_gens: 50]; jumlah populasi [pop\_size:100], maximum depth [max\_depth:5], codon [codon\_bits: 4], and maximum chromosome [num\_bits = 10\*codon\_bits]. Langkah kedua *generate* random bit [0,1] sebanyak maksimum jumlah kromosom dan mengulangi langkah kedua sebanyak jumlah populasi yang ditentukan[5].

**2.2 Transkripsi**

Pada proses transkripsi, kromosom yang masih dalam bentuk *binary* ( dapat dilihat pada gambar 2.) yang sudah di *generate* sebelumnya, kemudian dirubah kedalam bentuk *integer string* (gambar 3 ). Seperti yang terlihat pada contoh berikut ini contoh Satu kromosom dengan Binary string yang sudah didapatkan dari tahapan sebelumnya:

1011	0010	1110	0101	0011	0011	0110	1101	1111	0000
------	------	------	------	------	------	------	------	------	------

Gambar 2. Kromosom dalam *Binary String*

Akan dirubah kedalam bentuk integer string:

11	2	14	5	3	3	6	13	15	0
----	---	----	---	---	---	---	----	----	---

Gambar 3. Kromosom dalam *Integers String*

Proses ini akan diulang sampai semua kromosom dalam 1 populasi berubah kedalam bentuk integer string[5], [6].

**2.3 Translasi menggunakan grammar Backus Naur Form (BNF)**

Pada proses translasi ini, kromosom yang sudah dalam bentuk integer akan dirubah menjadi bentuk fungsi sesuai dengan grammar BNF yang sudah ditentukan. Grammar BNF memiliki tuple { N,T,P,S}, dimana N adalah kumpulan dari non-terminal, T adalah kumpulan dari terminal, P adalah kumpulan dari aturan *Production* yang memetakan elemen dari N sampai T, dan S adalah simbol start yang merupakan bagian dari N[5]. Berikut ini grammar yang digunakan dalam penelitian ini.

$$N = \{ \text{expr, Op, var} \}$$

$$T = \{ x, +, *, -, 1, 2, 3, 4, 5, 6, a, b, c, d, \}$$

$$S = \{ \text{expr} \}$$

P, Production Rules :

$$(1) \text{ <expr> } ::= \text{ <expr><op><expr> } | \quad (0)$$

$$\text{ (<expr><op><expr>)} | \quad (1)$$

$$\text{ <expr><op><var> } | \quad (2)$$

$$\text{ <var> } \quad (3)$$

$$(2) \text{ <op> } ::= + \quad (0)$$

$$| - \quad (1)$$

$$| \div \quad (2)$$

$$| \times \quad (3)$$

$$(3) \text{ <var> } ::= 1 \quad (1) \quad | 6 \quad (6)$$

$$| 2 \quad (2) \quad | a \quad (7)$$

$$| 3 \quad (3) \quad | b \quad (8)$$

$$| 4 \quad (4) \quad | c \quad (9)$$

$$| 5 \quad (5) \quad | d \quad (10)$$

**2.4 Fitness Value (f<sub>i</sub>)**

Pada tahapan ini, *Fitness Value* (f<sub>i</sub>) didapat dari fungsi yang sudah dibuat pada tahapan sebelumnya. Fungsi tersebut diujicoba pada data detak jantung yang ada, pencarian *Fitness value* pada penelitian ini dilakukan dengan menggunakan perumusan sebagai berikut[5]:

$$f_i = \frac{1}{e + b} \quad (1)$$

$$e = \frac{1}{N} \left( \sum_{i=1}^N |R_i - S_i| \right) \quad (2)$$

$$b = 1/N \quad (3)$$

$f$  = Fitness Value

$e$  = Nilai rata-rata dari eror

$R_i$  = Nilai dari fungsi yang sudah dibuat

$S_i$  = Nilai dari data yang disimpan

$b$  = Nilai konstanta yang diberikan untuk menghindari pembagian nilai 1 dengan 0

## 2.5 Parent Selection

Pada tahapan ini, semua individu dalam satu populasi yang masih dalam bentuk fungsi dan mempunyai *fitness value*, dirubah kedalam bentuk integer dan dilakukan pemilihan orang tua dengan menggunakan algoritma *Roulette wheel* dan ditempatkan kedalam mating pool[5] dengan menggunakan rumus 4. Setelah terkumpul di mating pool maka kemudian dilakukan pemilihan secara random berdasarkan besar nilai fitness yang didapatkan.

$$E(n_i) = Nf_i/f_{total} \quad (4)$$

$E(n_i)$  = Orang tua yang terpilih

$N$  = Besar Populasi ( max\_Pop = 100)

$f_i$  = Fitness value dari masing-masing individu

$f_{total}$  = Fitness value dari satu populasi

## 2.6 Rekombinasi

Pada proses ini, setelah didapatkan individu bakal orang tua baru sejumlah 100 kromosom dalam 1 Populasi yang ada di mating pool, maka langkah selanjutnya adalah pemilihan pasangan orang tua tersebut dengan menggunakan metode *crossover* dengan menggunakan probabilitas *crossover* sebesar (p\_cross: 0.3) [5]. dari 100 buah individu tersebut akan terpilih 50 pasangan individu, yang kemudian dilakukan *crossover* pada masing-masing pasangan tersebut, proses *crossover* dilakukan dengan menggunakan teknik 1 point *crossover*.

## 2.7 Mutasi

Pada tahapan mutasi ini dilakukan untuk masing-masing individu hasil rekombinasi. Mutasi dilakukan dengan cara membangkitkan bilangan random pada masing-masing individu dan masing-masing gen. Jika bilangan random tersebut kurang dari nilai probabilitas mutasi ( $P_m$  :  $1/(100*40)$ ), lihat pada tabel 1, maka mutasi akan dilakukan pada gen tersebut dengan mengganti nilai 1 bit dengan bilangan kebalikannya.

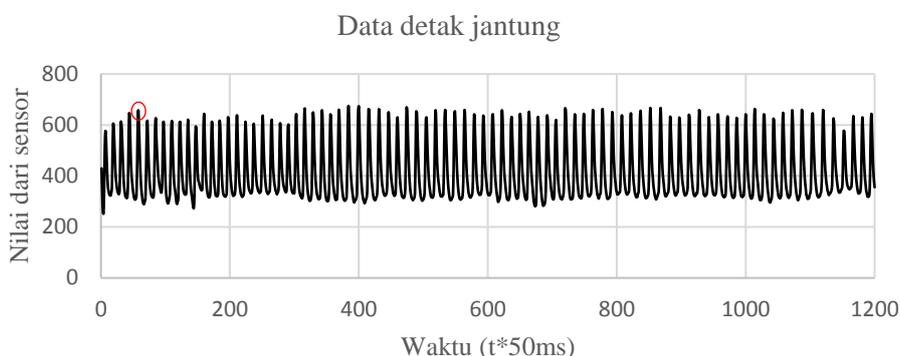
## 2.8 Survivor Selection

Pada tahapan ini dilakukan penggantian terhadap 100 kromosom individu orang tua menjadi 100 kromosom individu anak hasil rekombinasi dan mutasi. Individu anak tersebut kemudian akan dijadikan orang tua pada generasi berikutnya, pada penelitian ini maksimal generasi yang ditentukan sebesar 50 generasi. Penggantian N kromosom orang tua dengan N kromosom anak tidak memperhatikan Fitness values masing-masing kromosom.

## 3. Hasil dan Analisa

Pada penelitian ini, data yang digunakan sebagai subjek penelitian adalah data detak jantung orang normal dengan interval 60 -100 yang dihasilkan dari sinyal denyut nadi. Data diperoleh dengan cara menghubungkan sensor *easy pulse* dengan arduino dan Software ruby. Data denyut nadi diukur selama  $1200*50ms$  :  $60000ms$  atau 60s. Detak jantung diketahui dengan cara menghitung banyaknya titik puncak dalam 1 menit, seperti contoh detak jantung yang terlihat

pada Gambar 4. Pada gambar tersebut banyaknya titik puncak dalam 1 menit sebanyak 85 kali atau 85 BPM. Sedangkan banyaknya detak jantung dalam 10s sebesar 16 Beat.



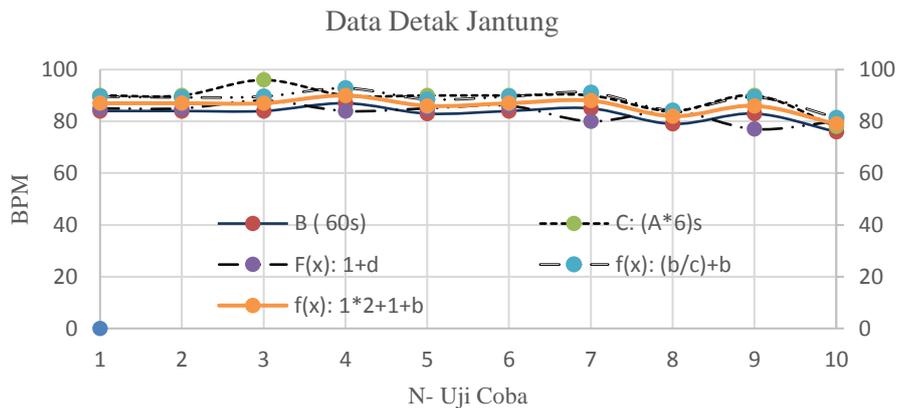
Gambar 4. Sinyal detak jantung dalam 1 Menit

Pada penelitian ini, data yang dijadikan subjek learning sebanyak 10 data, dengan menghitung detak jantung selama 10s (A) dan detak jantung selama 60s (B). Jumlah detak jantung dalam 10s akan dijadikan sebagai input data sedangkan data detak jantung selama 60s akan dijadikan sebagai output data seperti yang terlihat pada Tabel 2. Sebagai data pembandingan learning adalah data detak jantung selama 6s (B) dan data prosentase eror detak jantung hasil perkalian antara data detak jantung selama 10s dengan nilai 6 (C). Pada penelitian ini telah dihasilkan 3 buah fungsi yang mempunyai nilai fitness yang berbeda-beda serta prosentase eror yang berbeda-beda. Untuk GE 1 dengan fungsi  $f(x): 1+d$ , prosentase eror sebesar 4,01 % dan fitness value sebesar 57,0983. Untuk GE 2 dengan fungsi  $f(x): (b/c)+b$  prosentase eror sebesar 6,82% dan fitness value sebesar 54,95, sedangkan untuk GE 3 dengan fungsi  $f(x): 1*2+1+b$ , prosentase eror sebesar 3,62% dan fitness value sebesar 58,23. dari ketiga hasil learning Grammatical Evolution (GE) tersebut, fungsi GE 3 (seperti yang terlihat pada gambar 5) mempunyai nilai prosentase eror paling kecil dan mempunyai nilai Fitness yang paling besar.

Tabel 2. Data percobaan deteksi detak jantung menggunakan Grammatical Evolution

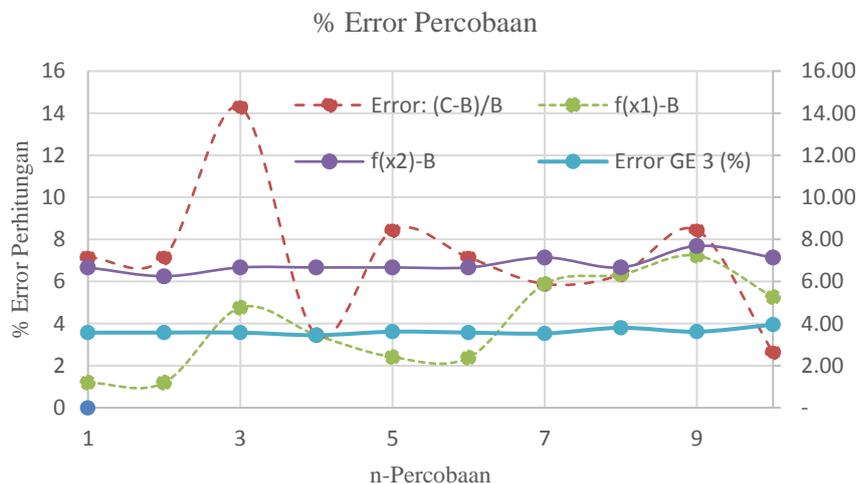
DATA	A	B	C: (A*6)s	Error: (C-B)/B	GE 1	Error GE	GE 2	Error GE2	GE3	Error GE 3
	10s	60s BPM	BPM	( % )	F(x): 1+d	f(x1)-B	f(x): (b/c)+b	f(x2)-B	f(x): 1*2+1+b	f(x3)- B
1	15	84	90	7,14	85	1,19	89,60	6,67	87,00	3,57
2	15	84	90	7,14	85	1,19	89,25	6,25	87,00	3,57
3	16	84	96	14,29	88	4,76	89,60	6,67	87,00	3,57
4	15	87	90	3,45	84	3,45	92,80	6,67	90,00	3,45
5	15	83	90	8,43	85	2,41	88,53	6,67	86,00	3,61
6	15	84	90	7,14	86	2,38	89,60	6,67	87,00	3,57
7	15	85	90	5,88	80	5,88	91,07	7,14	88,00	3,53
8	14	79	84	6,33	84	6,33	84,27	6,67	82,00	3,80
9	15	83	90	8,43	77	7,23	89,38	7,69	86,00	3,61
10	13	76	78	2,63	80	5,26	81,43	7,14	79,00	3,95
11	14	79	84	6,33						
<b>Nilai Rata-rata</b>				<b>7,09</b>		<b>4,01</b>		<b>6,82</b>		<b>3,62</b>

Fungsi yang didapatkan dari hasil learning menggunakan Grammatical evolution (seperti yang terlihat pada Tabel 2) diujicoba kembali dengan 10 data yang sudah ada, sehingga didapatkan hasil seperti pada gambar 5. Data detak jantung antara hasil pembacaan secara langsung selama 60 detik dengan data detak jantung hasil learning Grammatical evolution GE 3 hampir berhimpitan. Sehingga didapatkan bahwa fungsi GE 3 dapat dijadikan sebagai fungsi untuk deteksi pola aritmia jantung. Dari gambar 5 dapat diketahui bahwa kondisi jantung masih dalam keadaan normal, hal ini diketahui dari data detak jantung (BPM) yang masih dalam interval normal yaitu antara 60-100 BPM .



Gambar 5. Sinyal detak jantung 10 kali ujicoba

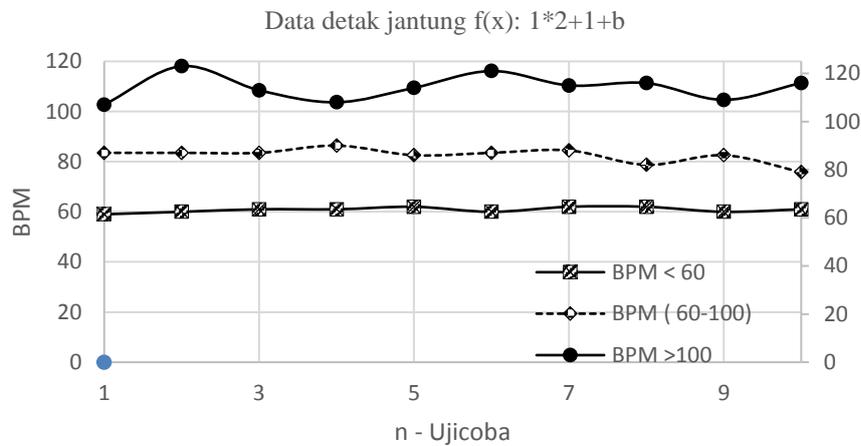
Pada gambar 6 diketahui bahwa data detak jantung yang dihasilkan dari melakukan perhitungan perkalian data detak jantung 10 detik pertama dengan 6, mempunyai prosentase error paling besar, sehingga fungsi tersebut tidak dapat digunakan untuk mendeteksi dini aritmia jantung. Sebaliknya dari gambar 5 dapat diketahui bahwa prosentase error GE 3 mempunyai nilai yang paling kecil yaitu dengan rata-rata error sebesar 3,62 %.



Gambar 6. Prosentase Error perhitungan manual dan menggunakan Grammatical Evolution

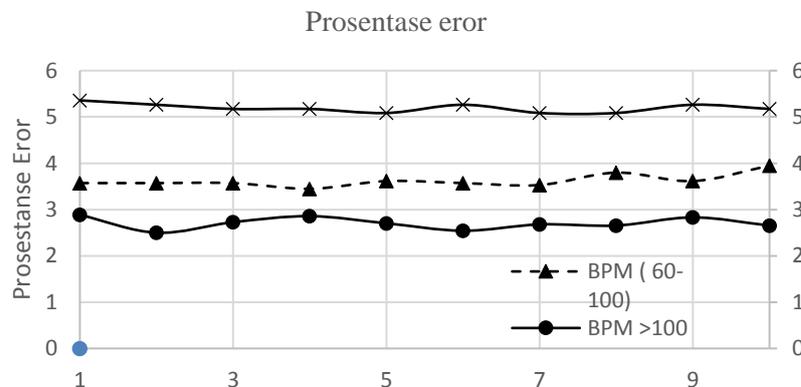
Fungsi yang didapatkan tersebut kemudian dijadikan perumusan untuk mendeteksi pola detak jantung. Dengan melakukan ujicoba terhadap 10 data detak jantung normal dengan interval BPM (60-100), 10 data simulasi detak jantung dengan BPM dibawah 60, dan 10 data simulasi detak jantung dengan BPM diatas 100. Dari data hasil ujicoba detak jantung menggunakan fungsi  $f(x): 1*2+1+b$  (seperti yang terlihat pada gambar 7) didapatkan bahwa pada data detak jantung dengan BPM diatas 100 dan dengan menggunakan fungsi tersebut mempunyai toleransi *error*

sebesar 5,19 %, sedangkan untuk detak jantung dengan BPM dibawah 60, mempunyai toleransi error sebesar 2,70 %, seperti yang terlihat pada gambar 8.



Gambar 7. Pola detak jantung menggunakan fungsi Grammatical Evolution

Pada gambar 7 dapat diketahui bahwa deteksi aritmia jantung dapat dilakukan dengan melihat grafik tersebut. Untuk grafik BPM (60-100) merupakan kondisi jantung dalam keadaan normal. Untuk grafik BPM < 60, kondisi jantung dalam keadaan aritmia dimana detak jantung lebih lambat dari yang seharusnya, kondisi tersebut dapat dikatakan sebagai kondisi Bradikardia, sedangkan untuk grafik BPM > 100, kondisi tersebut dapat dikatakan sebagai Takikardia atau kondisi dimana detak jantung lebih dari 100 kali permenit



Gambar 8. Presentase Error ujicoba menggunakan fungsi Grammatical Evolution

#### 4. Kesimpulan

Dari penelitian yang dilakukan dapat disimpulkan bahwa penggunaan algoritma *grammatical evolution* dalam deteksi dini aritmia dapat dilakukan dengan cara mendeteksi denyut nadi dengan menggunakan sensor *pulse rate*. Data detak jantung tersebut kemudian disimpan dan dikumpulkan sebanyak 10 data ujicoba. Dengan menggunakan 100 Individu dan masing-masing individu mempunyai 40 gen yang berbedebeda serta dengan 50 generasi dalam proses pembelajaran, maka didapatkan fungsi yang mempunyai nilai fitness sebesar 58,23 serta prosentase error sebesar 3,62% yaitu fungsi GE 3  $f(x): 1*2+1+b$ . Fungsi tersebut merupakan fungsi yang terbaik dari 3 fungsi yang sudah dihasilkan. Untuk GE 1 dengan fungsi  $f(x): 1+d$ , prosentase error sebesar 4,01 % dan fitness value sebesar 57,0983. Untuk GE 2 dengan fungsi  $f(x): (b/c)+b$  prosentase error sebesar 6,82% dan fitness value sebesar 54,95. Fungsi GE 1 tersebut dapat digunakan untuk proses deteksi aritmia jantung. Namun untuk hasil yang presisi masih dibutuhkan pembelajaran

yang lebih banyak lagi, dikarenakan fitness value yang terbaik yang dihasilkan dari proses pembelajaran menggunakan algoritma Grammatical Evolution ini masih sekitar 50% dari nilai yang diinginkan yaitu fungsi yang mempunyai fitness value mendekati 100.

### Ucapan Terima Kasih

Ucapan terima kasih kami tujukan kepada Politeknik Negeri Madiun serta Kementerian Riset Teknologi dan Perguruan Tinggi, yang telah memberikan dukungan dan fasilitas terhadap terlaksananya penelitian ini sampai selesai.

### Daftar Pustaka

- [1] S. Rege, T. Barkey, and M. Lowenstern, "Heart arrhythmia detection," in *2015 IEEE Virtual Conference on Applications of Commercial Sensors (VCACS)*, 2015, pp. 1–7.
- [2] E. R. Adams and A. Choi, "Using neural networks to predict cardiac arrhythmias," in *2012 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2012, pp. 402–407.
- [3] K. Waseem, A. Javed, R. Ramzan, and M. Farooq, "Using evolutionary algorithms for ECG Arrhythmia detection and classification," in *2011 Seventh International Conference on Natural Computation*, 2011, vol. 4, pp. 2386–2390.
- [4] M. O'Neill and C. Ryan, "Grammatical evolution," *IEEE Trans. Evol. Comput.*, vol. 5, no. 4, pp. 349–358, Aug. 2001.
- [5] A. E. Hakim, D. N. Ramdan, I. Hidayatulloh, A. S. Prihatmanto, and E. Rijanto, "Grammatical Evolution Algorithm for Position Prediction of the Ball in Robot-Soccer Goal Keeper Optimization," in *Intelligent Robotics Systems: Inspiring the NEXT*, 2013, pp. 147–160.
- [6] "Grammatical evolution - IEEE Journals & Magazine." [Online]. Available: <http://ieeexplore.ieee.org/document/942529/>. [Accessed: 10-Nov-2017].