



Klasifikasi Suara Paru Normal dan Abnormal Menggunakan Ekstraksi Fitur Discrete Wavelet Transform dengan Klasifikasi Menggunakan Jaringan Saraf Tiruan yang Dioptimasi dengan Algoritma Genetika

Fathurrahman Rabani¹, Jondri², Achmad Rizal³

¹Universitas Telkom, Teknik Informatika email: thurrabani@student.telkomuniversity.ac.id

²Universitas Telkom, Teknik Informatika, email: jondri@telkomuniversity.ac.id

³Universitas Telkom, Teknik Elektro, email: achmadrizal@telkomuniversity.ac.id

Abstrak

Salah satu cara yang digunakan dokter untuk mendiagnosis gangguan pada paru-paru adalah dengan mendengarkan suara pernapasan menggunakan stetoskop. Namun, dalam penggunaan stetoskop masih bisa terjadi kesalahan mendiagnosis karena beberapa faktor, seperti kemampuan, pengalaman, dan kepekaan pendengaran dokter, serta pola suara yang hampir mirip. Pada penelitian ini suara paru didekomposisi menggunakan metode discrete wavelet transform (DWT) untuk dekomposisi sinyal dan diklasifikasi menggunakan jaringan saraf tiruan (JST) yang dioptimasi oleh algoritma genetika (AG). Suara paru sebagai suara dari sistem pernapasan, kemudian dilakukan normalisasi dan dekomposisi menggunakan DWT. Karakteristik sinyal suara paru diperoleh dengan menghitung entropi pada koefisien DWT. Selanjutnya dilakukan klasifikasi dengan JST dan dioptimasi oleh AG. Berdasarkan solusi tersebut, klasifikasi suara paru-paru dapat diselesaikan dengan akurasi hingga 67,54%. Hasil ini lebih tinggi dibandingkan tanpa menggunakan AG. Metode yang diusulkan diharapkan bisa menjadi salah satu alternatif metode dalam analisis suara paru.

Kata kunci: discrete wavelet transform, jaringan saraf tiruan, algoritma genetika, suara paru

Abstract

One of the methods doctors use to diagnose lung disorders is listening to breathing sounds using a stethoscope. However, the use of a stethoscope can still be misdiagnosed due to several factors, such as the doctor's ability, experience, and hearing sensitivity, as well as almost similar sound patterns. In this study, the pulmonary sound was decomposed using the discrete wavelet transform (DWT) method and classified using an artificial neural network (ANN) optimized by a genetic algorithm (GA). Lung sounds as sounds from the respiratory system, then normalization and decomposition are carried out using DWT. The pulmonary sound signal characteristics are obtained by calculating the entropy at the DWT coefficient. Furthermore, the classification is done by using ANN and optimized by GA. Based on this solution, the classification of lung sounds can be completed with an accuracy of up to 67.54%. This result is higher than without using GA. The proposed method is expected to be an alternative method in lung sound analysis.

Keywords: discrete wavelet transform, artificial neural network, genetic algorithm, lung sound

1. Pendahuluan

Paru-paru adalah salah satu organ vital dalam tubuh manusia. Paru-paru berperan dalam proses pernafasan. Kelainan paru atau pernafasan bisa dideteksi menggunakan beberapa metode. Salah satu cara yang digunakan dokter untuk mendiagnosis gangguan pada sistem pernapasan adalah dengan mendengarkan suara pernapasan menggunakan stetoskop yang dikenal dengan teknik auskultasi [1]. Teknik auskultasi yang baik digunakan pada suasana sunyi dan sesuai dengan prosedur. Permasalahan dalam penggunaan teknik ini tidak lepas dari kemampuan, pengalaman, dan kepekaan pendengaran dokter, serta kemiripan dari pola suara. Hal tersebut bisa mengakibatkan kesalahan pada diagnosis.

Dengan perkembangan teknologi saat ini, beberapa sistem telah dibangun sebagai alat bantu dokter dalam melakukan diagnosis. Melalui rekaman suara paru, sistem dapat mendeteksi adanya gangguan pada sistem pernapasan berdasarkan jenis suara paru. Terdapat 2 metode dalam sistem untuk melakukan klasifikasi suara paru, metode tersebut adalah metode ekstraksi fitur dan metode klasifikasi. Metode yang telah digunakan dalam klasifikasi suara antara lain MFCC dan *backpropagation* [2]. *Multiscale Hjorth Descriptor*, dan *Multilayer Perceptron* [3], dimensi fractal [4] juga digunakan dalam penelitian sebelumnya. Metode-metode tersebut digunakan untuk klasifikasi suara paru normal dan abnormal, maupun untuk klasifikasi suara paru normal dengan beberapa kelas suara dengan kelainan tertentu. Pada penelitian sebelumnya biasanya banyak ciri digunakan secara bersamaan sebagai masukan bagi *classifier*. Pemilihan ciri yang tepat menggunakan metode optimasi belum pernah dibahas pada penelitian terdahulu.

Pada penelitian ini dibahas klasifikasi menggunakan *discrete wavelet transform* (DWT) sebagai metode ekstraksi fitur dan jaringan saraf tiruan (JST) yang dioptimasi oleh Algoritma Genetika (AG) sebagai metode klasifikasi. Tujuan yang hendak dicapai dari penelitian ini yaitu untuk mengetahui level dekomposisi DWT terbaik untuk mendapatkan ciri yang baik dari sinyal suara paru, memahami peran AG dalam mengoptimasi JST, serta mengetahui parameter arsitektur dari JST yang memberikan hasil terbaik berupa akurasi tertinggi.

2. Landasan Teori

2.1 Teori Suara Paru

Suara paru terbagi menjadi 2 jenis, yaitu suara paru normal dan abnormal. Suara paru normal terdengar jelas dan lembut seperti udara melewati pipa yang tanpa ada halangan. Sementara itu, suara paru abnormal terdengar tidak jelas seperti ada penghalang dan lain sebagainya. Suara paru abnormal yang dimaksud antara lain *wheeze* dan *crackle*. *Wheeze* merupakan suara paru yang ditimbulkan karena adanya penyempitan saluran pernapasan atau penebalan dari dinding saluran pernapasan. *Crackle* merupakan suara paru yang ditimbulkan akibat penyimpangan arah udara pada saluran pernapasan. Penyimpangan arah udara terjadi karena pembukaan saluran pernapasan (mekanisme inspirasi) secara mendadak pada saat saluran pernapasan masih dalam keadaan menutup (mekanisme ekspirasi) [5].

2.2 Discrete Wavelet Transform (DWT)

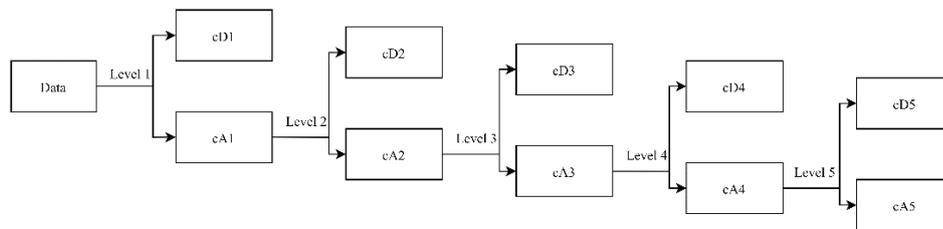
DWT adalah salah satu transformasi wavelet yang membatasi sinyal terhadap waktu dan skala. Transformasi dilakukan dengan dekomposisi sinyal melalui 2 filter, yaitu *Low Pass Filter* (LPF) dan *High Pass Filter* (HPF) [6]. Proses DWT dapat dilihat pada Gambar 1. Keluaran dari HPF adalah detail dari sinyal, dan keluaran dari LPF disebut aproksimasi yang merupakan perkiraan dari sinyal untuk dekomposisi selanjutnya. Berikut persamaan DWT [7].

$$WT_x(j, k) = \int x(t) \hat{\psi}_{j,k}(t) dt \quad (1)$$

Dimana $x(t)$ adalah sinyal, j adalah parameter frekuensi atau skala dalam bilangan bulat, k adalah parameter waktu dalam bilangan bulat, dan $\psi(t)$ adalah fungsi basis wavelet diskrit dengan persamaan berikut.

$$\hat{\psi}_{j,k}(t) = 2^{-j/2} \psi(2^{-j}t - k) \quad (2)$$

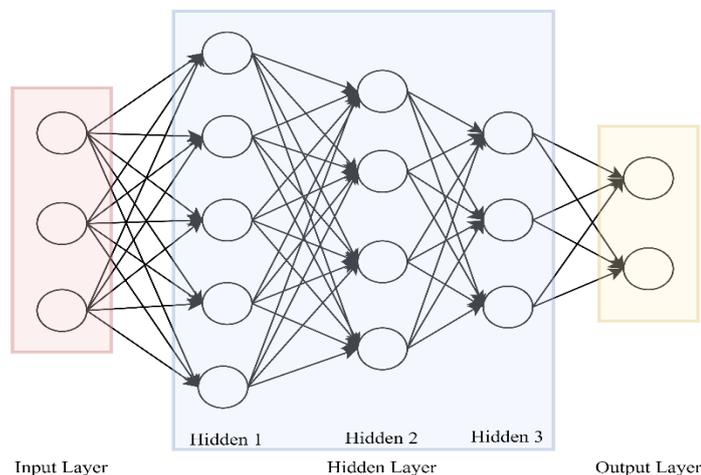
Dimana ψ adalah fungsi basis wavelet seperti Haar, Daubechies, Coiflet, dan lainnya.



Gambar 1. DWT level 5 [6]

2.3 Jaringan Saraf Tiruan (JST)

Arsitektur JST terdiri dari beberapa lapisan yang didalamnya terdapat beberapa node yang menyerupai neuron pada saraf biologis. Lapisan yang dimaksud adalah lapisan masukan (*input layer*), lapisan tersembunyi (*hidden layer*), dan lapisan keluaran (*output layer*). Koneksi setiap neuron membentuk jaringan yang disebut jaringan saraf. Pada JST, neuron yang berada dalam *hidden layer* menerima dan menerapkan fungsi aktivasi jumlah masukan yang dibobotkan. Kemudian *output layer* menerima jumlah keluaran yang dibobotkan dari *hidden layer* [8]. Dengan kata lain, informasi diteruskan dari satu lapisan ke lapisan berikutnya, baik dari *input layer*, *hidden layer* hingga *output layer*. Dimana pada *hidden layer* dapat terdiri lebih dari 1 lapisan, berikut arsitektur JST dengan 3 buah *hidden layer*.



Gambar 1. Struktur JST

Selain pada lapisan masukan, neuron-neuron di lapisan lain melakukan perhitungan fungsi aktivasi ($f(x)$). Dari jumlah masukan yang dibobotkan ($z(x)$) dengan persamaan sebagai berikut:

$$z(x) = \sum_{i=1}^n (x_i w_i) \quad (3)$$

$$f(x) = f(z(x)) \quad (4)$$

Dimana n adalah jumlah neuron pada lapisan sebelumnya, x adalah nilai masukan atau hasil keluaran dari neuron pada lapisan sebelumnya, dan w adalah bobot. Fungsi aktivasi terdiri dari beberapa jenis seperti relu, logistik sigmoid, tanh, dan *identity*. Hasil dari fungsi aktivasi pada output layer menjadi hasil keluaran dari pembelajaran JST. Kemudian didapatkan *error* (E) dari selisih nilai yang diinginkan (T) dengan nilai keluaran dari pembelajaran JST (P), sebagai berikut.

$$E=T-P \quad (5)$$

Perhitungan satu siklus pembelajaran JST dimulai dari nilai masukan pada input layer hingga mendapatkan nilai *error*. Perhitungan dilakukan berulang hingga tercapai jumlah siklus pembelajaran JST yang diinginkan atau telah mencapai nilai *error* yang diinginkan.

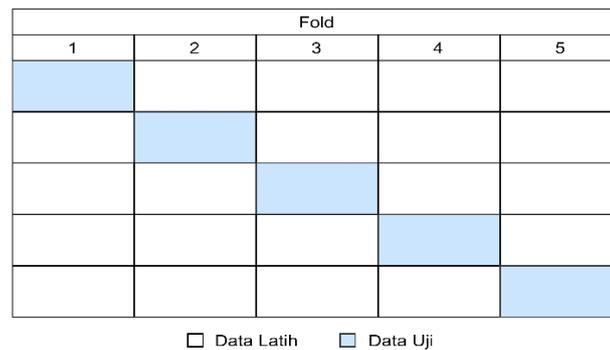
2.4 Algoritma Genetika

Algoritma genetika (AG) merupakan teknik pengoptimasi berdasarkan seleksi alam yang memungkinkan individu untuk bertahan hidup dan beradaptasi dengan lingkungan, serta berkembang biak. Dalam populasi, individu-individu yang menguntungkan kemudian kawin silang dan menghasilkan keturunan dengan karakteristik yang serupa. Dengan mempertahankan individu-individu berkarakteristik yang menguntungkan, sehingga individu yang tidak menguntungkan terseleksi dari populasi [8]. Penggunaan AG dimanfaatkan karena kemampuan AG dalam mengoptimalkan kekuatan metode kecerdasan buatan lainnya seperti JST [9]. Dalam optimasi JST, berikut proses AG mengoptimasi arsitektur JST dan update bobot.

1. Pembentukan kromosom arsitektur JST, yang menyatakan salah satu solusi dari arsitektur JST seperti jumlah hidden layer dan jumlah neuron.
2. Pembentukan populasi arsitektur JST, membangkitkan secara acak kromosom dengan jumlah yang ditentukan.
3. Perhitungan fitness arsitektur JST, merupakan pengukuran baik atau tidaknya solusi yang didapatkan. Dalam hal ini, pengukuran berupa akurasi dari pembelajaran JST yang memerlukan bobot. Berikut proses update bobot yang dilakukan oleh AG.
 - Pembentukan kromosom, yang menyatakan salah satu solusi dalam bilangan integer ataupun bilangan real.
 - Pembentukan populasi, membangkitkan secara acak kromosom dengan jumlah yang ditentukan.
 - Perhitungan fitness dengan perhitungan nilai error pada pembelajaran JST.
 - Menentukan metode selection, seperti roulette wheel selection, tournament selection, rank selection, dan lainnya.
 - Menentukan proses crossover
 - Menentukan proses mutasi
4. Menentukan metode selection arsitektur JST, seperti roulette wheel selection, tournament selection, rank selection, dan lainnya.
5. Menentukan proses crossover arsitektur JST
6. Menentukan proses mutasi arsitektur JST

2.5 K-Fold Cross Validatiion

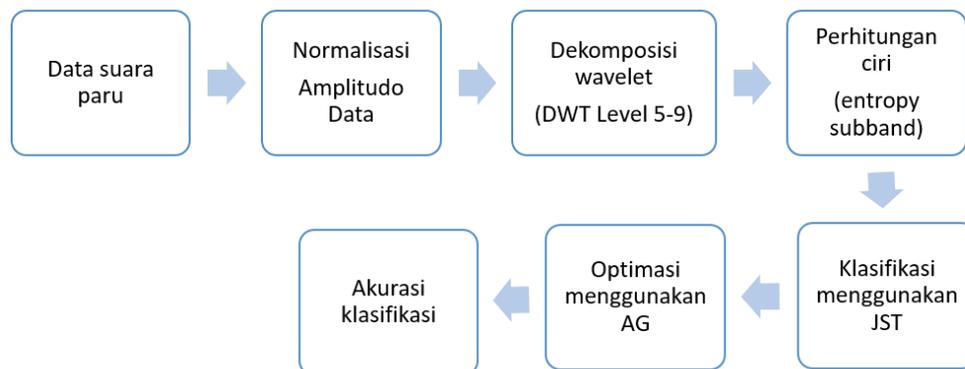
Cross Validation adalah teknik validasi model untuk menilai keakuratan hasil analisis dari penyamarataan kumpulan data independent. Ada beberapa metode dalam *cross-validation*, salah satunya adalah *k-fold cross-validation* yang membagi data menjadi k bagian set data untuk pelatihan dan pengujian. Dimana Pelatihan dan pengujian dilakukan sebanyak k kali [10]. Terdapat (k-1) bagian yang menjadi set data pengujian, dan yang lainnya menjadi set data pelatihan. Berikut pembagian data menggunakan *5-fold cross-validation*. Ilustrasi *5-fold cross-validation* seperti pada Gambar 3.



Gambar 3. Ilustrasi 5-fold cross-validation

3. Sistem yang Dibangun

Gambaran umum dari sistem yang dibangun dapat dilihat pada diagram blok Gambar 4.



Gambar 4. Diagram blok sistem

Secara sederhana proses yang dilakukan sebagai berikut: data suara paru ditransformasikan wavelet dengan metode *discrete wavelevelt transform* (DWT). Pada saat transformasi wavelet, terdapat pengujian terhadap level dekomposisi DWT. Pengujian level dekomposisi diberlakukan kepada 5 level dekomposisi, yaitu level 5 sampai level 9. Masing-masing level menghasilkan koefisien DWT atau fitur dengan jumlah yang berbeda, kemudian perhitungan energi dari fitur dengan perhitungan entropi. Selanjutnya, fitur menjadi data masukkan dalam proses klasifikasi dengan metode JST yang dioptimasi oleh AG. Optimasi AG berperan dalam melakukan *update* bobot dan bias, serta mencari arsitektur terbaik dari JST. Optimiasi arsitektur JST meliputi jumlah *hidden layer*, jumlah neuron, dan fungsi aktivasi. Penjelasan lebih lanjut dari proses di atas dijelaskan pada subbagian berikut ini.

3.1 Data Suara Paru

Data berasal dari database suara penapasan paru yang digunakan pada International Conference on Biomedical and Health Informatics (ICBHI) 2017 [11]. Dalam database tersebut terdiri dari 274 sampel audio berekstensi .wav, serta terdapat file berekstensi .txt, dan .hea. Sampel audio dipecah berdasarkan proses pernapasan yang meliputi mekanisme inspirasi dan mekanisme ekspirasi yang didapatkan informasinya dalam file .txt. Diperoleh dataset sebanyak 2369 data sinyal dengan frekuensi sampling sebesar 44100 Hz dan bit sebesar 32-bit. Kemudian suara paru dibagi menjadi 4 kelas, yaitu kelas suara normal sebanyak 1251 data, kelas suara wheeze sebanyak 304 data, kelas suara crackle sebanyak 642 data, dan kelas suara crackle+wheeze sebanyak 172 data.

3.2 Praproses

Adapun metode praproses yang digunakan pada penelitian ini yaitu metode normalisasi. Untuk data masukkan pada proses ekstraksi fitur, data dinormalisasi dengan normalisasi min-max. Normalisasi ini mengubah data ke dalam rentang 0 sampai 1 melalui persamaan berikut.

$$X_{normalized} = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (6)$$

Dimana X adalah data yang digunakan, X_{min} adalah nilai minimum dari data, dan X_{max} adalah nilai maksimum dari data.

3.3 Ekstraksi Fitur

Data yang telah dinormalisasi selanjutnya diproses dalam transformasi wavelet dengan metode *Discrete Wavelet Transform* (DWT). Fungsi basis *wavelet* yang digunakan dalam penelitian ini, yaitu *wavelet* Haar, dengan jumlah level dekomposisi DWT yang diuji dari level 5 sampai level 9. Koefisien yang telah didapatkan dari setiap level dekomposisi dihitung energinya dengan perhitungan entropi pada persamaan berikut.

$$En = - \sum_{k=1}^n p_k \log(p_k) \quad (7)$$

Dimana n adalah jumlah data, dan p_k adalah probabilitas munculnya elemen k . Berikut rumus dari probabilitas p_k .

$$p_k = \frac{n(k)}{n} \quad (8)$$

Hasil dari perhitungan entropi merupakan fitur dari sinyal suara paru. Selanjutnya, fitur dinormalisasi menggunakan normalisasi *z-score* untuk digunakan sebagai data masukkan pada proses klasifikasi. Normalisasi ini mengubah nilai rata-rata menuju nilai 0 dan standar deviasi menuju nilai 1. Berikut persamaan dari normalisasi *z-score*:

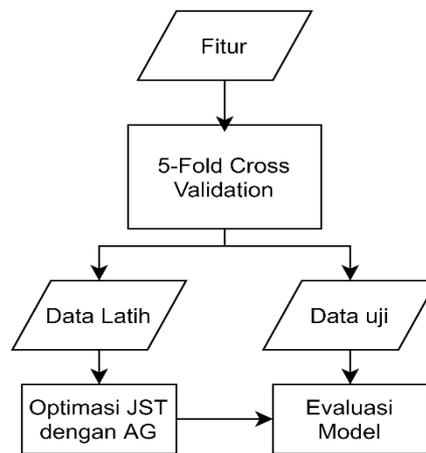
Hasil dari perhitungan entropi merupakan fitur dari sinyal suara paru. Selanjutnya, fitur dinormalisasi menggunakan normalisasi *z-score* untuk digunakan sebagai data masukkan pada proses klasifikasi. Normalisasi ini mengubah nilai rata-rata menuju nilai 0 dan standar deviasi menuju nilai 1. Berikut persamaan dari normalisasi *z-score*:

$$Z_i = \frac{X_i - \mu}{\sigma} \quad (9)$$

Dimana Z_i adalah z -score pada data ke i , μ adalah rata-rata dan σ adalah standar deviasi. Hasil dari standardisasi ini fitur yang dimasukkan dalam metode klasifikasi.

3.4 Klasifikasi

Metode klasifikasi yang digunakan dalam penelitian ini adalah Jaringan Saraf Tiruan (JST) yang dioptimasi oleh Algoritma Genetika (AG). Optimasi yang dilakukan oleh AG yaitu untuk *update* bobot dan bias, serta untuk mencari arsitektur terbaik dari JST, dimana arsitektur meliputi jumlah *hidden layer* dan jumlah neuron yang menghasilkan akurasi terbaik. Proses klasifikasi secara umum dapat dilihat pada Gambar 4. Jumlah input layer yang digunakan menyesuaikan dengan level DWT yang digunakan. Jika digunakan DWT level 5 maka akan ada 6 ciri yang dipakai, jadi akan digunakan 6 input node. Sementara itu jika digunakan DWT level 9 maka akan dipakai 10 input node. Untuk output layer digunakan 4 node sesuai dengan jumlah kelas suara paru. Sementara itu pada hidden layer digunakan layer 1-3 dengan node 1-100. Parameter-parameter ini yang nantinya akan dioptimasi menggunakan AG.



Gambar 2. Gambaran umum proses klasifikasi

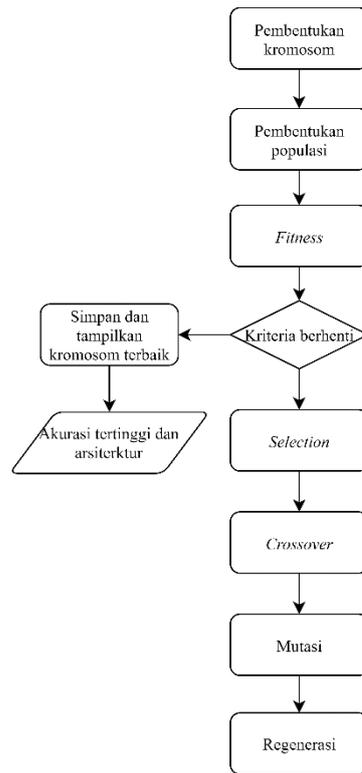
1. Split Data menggunakan *K-fold Cross-Validation*

Data dimasukkan berupa fitur yang telah dinormalisasi selanjutnya dibagi menjadi data latih dan data uji. Data latih digunakan dalam pembelajaran JST untuk melakukan klasifikasi pada data uji. Pembagian data latih dan data uji dengan menggunakan *k-fold cross validation*, dimana $k=5$.

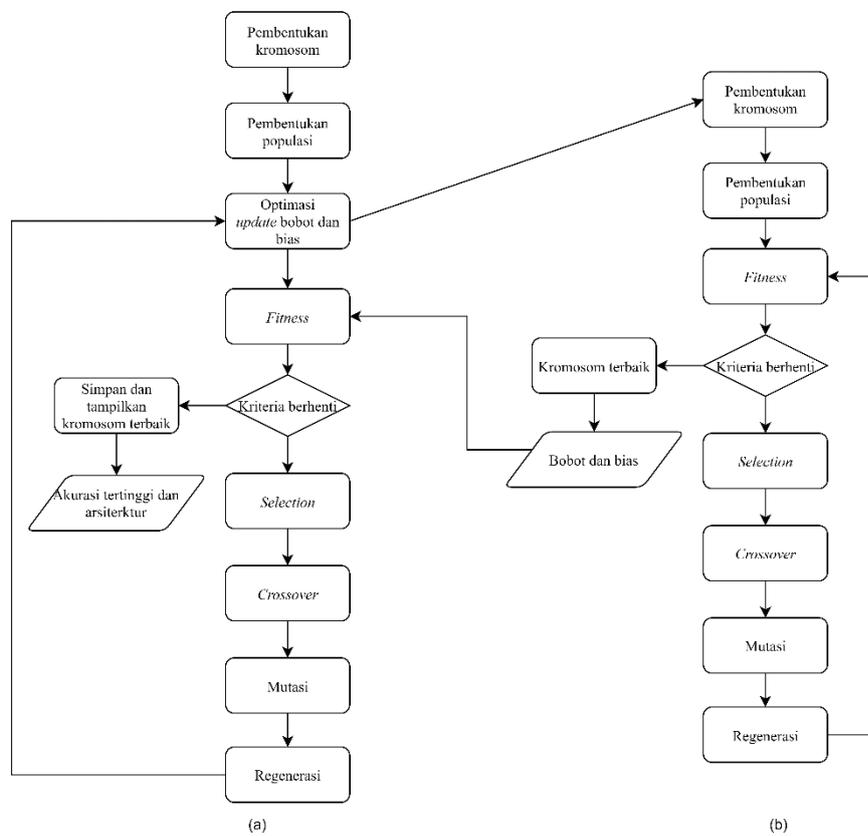
2. Optimasi JST dengan AG

Arsitektur JST yang dibangun memiliki *hidden layer* dengan jumlah minimal 1 *hidden layer* dan maksimal 3 *hidden layer*. Kemudian jumlah neuron pada setiap lapisan sebanyak 1-100 neuron, kecuali *input layer* yang memiliki jumlah neuron berdasarkan data masukkan. Penentuan jumlah *hidden layer* dan jumlah neuron dioptimasi oleh AG, serta *update* bobot juga dapat dilakukan oleh AG. Dimana arsitektur JST yang terbentuk untuk optimasi memiliki jumlah maksimal *hidden layer*, yaitu 3 buah *hidden layer* seperti pada Gambar 3.

Pengujian pada optimasi JST dengan AG dibagi menjadi 2, yang pertama dilakukan pada optimasi arsitektur JST. Kemudian yang kedua dilakukan optimasi pada arsitektur JST serta optimasi *update* bobot dan bias.



Gambar 5. Flowchart AG optimasi arsitektur JST



Gambar 6. Flowchart AG optimasi JST, (a) optimasi arsitektur JST, (b) update bobot dan bias

Berikut penjelasan proses AG dalam pencarian arsitektur terbaik dari JST.

1. Pembentukan kromosom

Kromosom terdiri dari gen fungsi aktivasi berupa *identity*, *logistic sigmoid*, *tanh*, atau *relu*. Kemudian gen jumlah *hidden layer* berupa suatu nilai integer acak dengan rentang 1 sampai 3, dan gen jumlah neuron berupa suatu nilai integer acak dengan rentang 1 sampai 100 untuk masing-masing *hidden layer*. Berikut kromosom yang dibentuk.

Fungsi aktivasi	Jumlah <i>hidden layer</i>	Jumlah neuron <i>hidden 1</i>	Jumlah neuron <i>hidden 2</i>	Jumlah neuron <i>hidden 3</i>
-----------------	----------------------------	-------------------------------	-------------------------------	-------------------------------

Gambar 7. Kromosom arsitektur JST

2. Pembentukan populasi

Populasi terdiri dari 40 kromosom/individu

3. Fitness

Fitness dilakukan untuk memaksimalkan nilai akurasi dari klasifikasi JST dengan arsitektur berdasarkan kromosom-kromosom yang telah dibentuk dalam populasi. Berikut persamaan untuk menghitung akurasi.

$$Akurasi = \frac{\text{jumlah data yang benar}}{\text{jumlah data uji}} \times 100\% \quad (10)$$

4. Selection

Selection dilakukan dengan metode *Rank Selection*, dimana kromosom terpilih didapatkan berdasarkan peringkat dari *fitness* setiap kromosom. Kromosom terpilih sebanyak seperdua dari jumlah kromosom pada populasi atau sebanyak 20 kromosom. Kromosom terpilih selanjutnya dijadikan sebagai orang tua pada proses *crossover*.

5. Crossover

Berdasarkan orang tua yang telah didapat pada proses *selection*, selanjutnya *crossover* dilakukan dengan menukarkan gen. Pertukaran gen diberlakukan pada gen fungsi aktivasi, gen jumlah *hidden layer*, dan gen jumlah neuron di *hidden 2*.

6. Mutasi

Kromosom hasil *crossover* selanjutnya dimutasi dengan mengubah nilai dari gen yang terpilih. Proses mutasi diawali dengan membangun suatu bilangan real acak pada rentang 0-1 untuk setiap gen, lalu membandingkannya dengan tingkat mutasi yang diberlakukan yaitu 0.1 atau 10%. Apabila bilangan real lebih kecil dari tingkat mutasi, maka gen tersebut merupakan gen terpilih untuk dimutasi. Nilai dari gen terpilih selanjutnya diganti sesuai dengan tipe nilainya. Untuk gen fungsi aktivasi, nilai diganti oleh salah satu dari *identity*, *logistic sigmoid*, *tanh*, atau *relu*. Sedangkan gen jumlah *hidden layer* diganti dengan suatu nilai integer acak pada rentang 1-3 dan gen jumlah *hidden layer* dan 1-100 untuk jumlah neuron.

7. Regenerasi

Regenerasi dengan menghasilkan populasi baru yang terbentuk dari kromosom terpilih atau orang tua pada proses *selection* dan kromosom hasil *crossover* yang telah dimutasi. Populasi baru yang telah terbentuk selanjutnya diproses dalam perhitungan *fitness*. Kromosom yang

dilakukan perhitungan *fitness* adalah kromosom yang belum memiliki nilai *fitness*. Dalam hal ini, kromosom orang tua yang telah memiliki nilai *fitness* tidak diikutsertakan pada proses perhitungan *fitness*.

8. Kriteria berhenti

AG berhenti meregenerasi setelah menemui kriteria berhenti. Kriteria tersebut adalah jumlah maksimal regenerasi sebanyak 30 dan/atau nilai akurasi sebesar 95%.

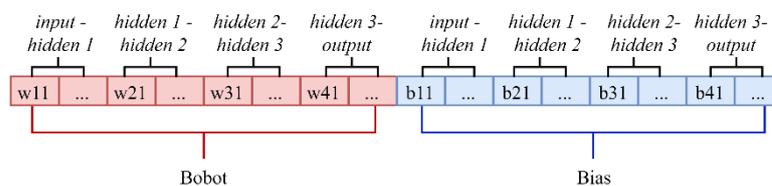
9. Kromosom terbaik

Kromosom terbaik merupakan kromosom yang memiliki nilai *fitness* terbaik, dalam hal ini nilai *fitness* dengan akurasi tertinggi. Kromosom terbaik selanjutnya disimpan dan ditampilkan sebagai hasil dari arsitektur terbaik.

Selanjutnya penjelasan proses AG dalam melakukan *update* bobot dan bias, sebagai berikut.

1) Pembentukan kromosom

Kromosom terdiri dari bobot dan bias dengan jumlah masing-masing gen berdasarkan jaringan yang terbentuk pada optimasi arsitektur. Jaringan terdiri dari 4 bagian yaitu *input layer - hidden 1* sampai dengan *hidden 3 - output layer*. Gen dari bobot dan bias berisikan suatu bilangan real acak pada rentang -1 sampai 1. Berikut kromosom yang terbentuk.



Gambar 8. Kromosom *update* bobot dan bias

2) Pembentukan populasi

Populasi yang dibentuk terdiri dari 20 kromosom/individu

3) Fitness

Fitness dilakukan untuk meminimalkan nilai *error* dari klasifikasi JST berdasarkan arsitektur serta bobot dan bias. Nilai *error* dihitung dengan persamaan *Mean Squared Error* (MSE) pada persamaan (4). *Error* yang dimaksud adalah kesalahan dari setiap data uji dalam memprediksi.

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \tag{11}$$

Dimana y merupakan nilai aktual dan \hat{y} merupakan hasil prediksi, serta n adalah jumlah sampel pada hal ini adalah jumlah dari data uji.

4) Selection

Selection dilakukan dengan metode *Rank Selection*, dimana kromosom terpilih didapatkan berdasarkan peringkat dari *fitness* kromosom. Kromosom terpilih sebanyak seperdua dari jumlah kromosom pada populasi atau sebanyak 10 kromosom. Kromosom terpilih selanjutnya dijadikan sebagai orang tua pada proses *crossover*.

5) **Crossover**

Berdasarkan orang tua yang telah didapat pada proses *selection*, selanjutnya *crossover* dilakukan dengan menukarkan gen. Gen yang ditukar sebanyak seperdua dari jumlah gen pada masing-masing jaringan, baik pada bobot maupun bias.

6) **Mutasi**

Kromosom hasil *crossover* selanjutnya dimutasi dengan mengubah nilai dari gen yang terpilih. Proses mutasi diawali dengan membangun suatu bilangan real acak pada rentang 0-1 untuk setiap gen, lalu membandingkannya dengan tingkat mutasi yang diberlakukan yaitu 0.2 atau 20%. Apabila bilangan real lebih kecil dari tingkat mutasi, maka gen tersebut merupakan gen terpilih untuk dimutasi. Nilai dari gen terpilih diganti dengan suatu bilangan real acak pada rentang -1 sampai 1.

7) **Regenerasi**

Regenerasi dengan menghasilkan populasi baru yang terbentuk dari kromosom terpilih atau orang tua pada proses *selection* dan kromosom hasil *crossover* yang telah dimutasi. Populasi baru yang telah terbentuk selanjutnya diproses dalam perhitungan *fitness*. Kromosom yang dilakukan perhitungan *fitness* adalah kromosom yang belum memiliki nilai *fitness*. Dalam hal ini, kromosom orang tua yang telah memiliki nilai *fitness* tidak diikutsertakan pada proses perhitungan *fitness*.

8) **Kriteria berhenti**

Algoritma genetika berhenti meregenerasi setelah menemui kriteria berhenti. Kriteria tersebut adalah jumlah maksimal regenerasi sebanyak 30 dan/atau nilai MSE sebesar 0.01.

9) **Kromosom terbaik**

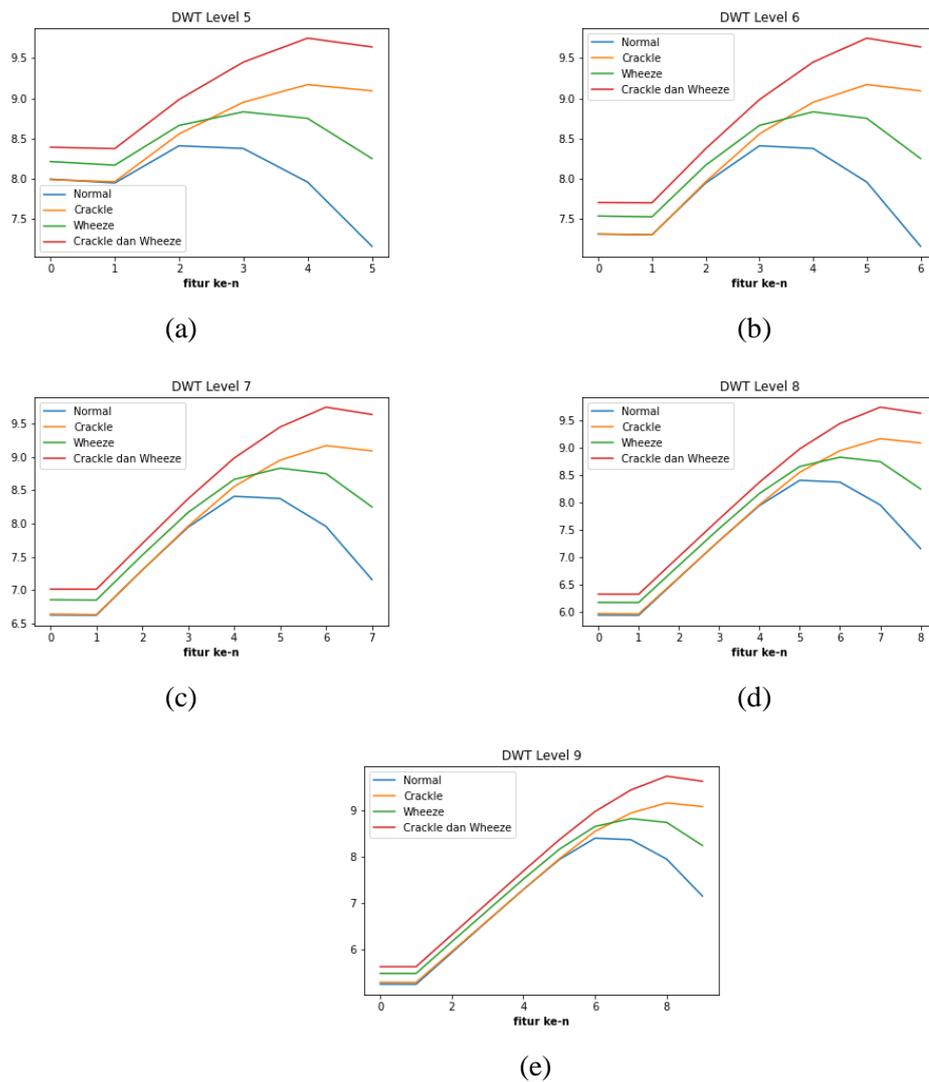
Kromosom terbaik merupakan kromosom yang memiliki nilai *fitness* terbaik, dalam hal ini nilai *fitness* dengan MSE terendah. Kromosom terbaik selanjutnya digunakan dalam perhitungan *fitness* pada optimasi pencarian arsitektur.

Hasil yang didapatkan dari optimasi AG terhadap JST adalah arsitektur terbaik JST yang terdiri dari fungsi aktivasi, jumlah *hidden layer*, dan jumlah neuron. Serta bobot dan bias yang digunakan dalam arsitektur tersebut.

4. Hasil dan Diskusi

4.1 Hasil Pengujian

Berikut hasil dari proses ekstraksi fitur dengan pengujian level dekomposisi DWT dan 4 kelas data (normal, *crackle*, *wheeze*, dan *crackle+wheeze*).



Gambar 9. (a) Fitur DWT level 5, (b) Fitur DWT level 6, (c) Fitur DWT level 7, (d) Fitur DWT level 8, (e) Fitur DWT level 9

Tabel 1. Perbandingan akurasi hasil klasifikasi

Level DWT	JST	AG Optimasi Arsitektur JST	AG Optimasi Arsitektur JST + Update Bobot dan Bias
Level 5	64.96%	66.95%	58.59%
Level 6	66.10%	67.16%	57.32%
Level 7	66.19%	67.20%	57.75%
Level 8	65.22%	67.37%	60.66%
Level 9	66.95%	67.54%	57.87%

Berikut hasil dari F1 Score untuk setiap kelas data pada setiap level dan setiap pengujian.

Table 2. Perbandingan F1 Score dari setiap kelas data

<i>JST tanpa optimasi</i>				
Level	Normal	Crackle	Wheeze	Crackle + Wheeze
Level 5	75.32%	59.36%	44.71%	40.79%
Level 6	77.33%	60.26%	41.17%	40.14%
Level 7	77.03%	60.35%	41.96%	41.80%
Level 8	75.56%	61.51%	42.58%	42.59%
Level 9	77.46%	61.70%	42.86%	41.27%
<i>Optimasi Arsitektur JST</i>				
Level	Normal	Crackle	Wheeze	Crackle + Wheeze
Level 5	76.94%	61.99%	35.62%	42.36%
Level 6	77.11%	62.51%	35.54%	45.14%
Level 7	77.50%	63.53%	30.96%	41.14%
Level 8	77.25%	64.36%	35.61%	40.82%
Level 9	77.67%	64.37%	35.04%	39.12%
<i>Optimasi Arsitektur JST + Update bobot dan Bias</i>				
Level	Normal	Crackle	Wheeze	Crackle + Wheeze
Level 5	71.07%	47.46%	0.0%	0.0%
Level 6	68.58%	53.83%	0.0%	0.0%
Level 7	71.81%	36.33%	3.32%	0.0%
Level 8	72.98%	50.39%	0.0%	0.0%
Level 9	69.42%	54.44%	0.0%	0.0%

Perbandingan arsitektur JST dengan dan tanpa optimasi dapat dilihat pada Tabel 3.

Tabel 3. Perbandingan arsitektur

	JST	AG Optimasi Arsitektur JST	AG Optimasi Arsitektur JST + Update Bobot dan Bias
Fungsi Aktivasi	<i>Relu</i>	<i>Tanh</i>	<i>Identity</i>
Total Hidden Layer	3	3	2
Jumlah Node Hidden 1	100	49	92
Jumlah Node Hidden 2	100	32	53
Jumlah Node Hidden 3	100	38	64
Iterasi	5000	167	1

4.2 Analisis Hasil Pengujian

Berdasarkan hasil ekstraksi fitur pada Gambar 8. Terlihat bahwa fitur dari setiap kelas data memiliki perbedaan yang tidak terlalu signifikan. Kondisi ini mempengaruhi pembelajaran JST dalam klasifikasi. Fitur yang serupa membuat pembelajaran JST kesulitan dalam membedakan kelas data. Hasil dari pengujian level dekomposisi DWT, diketahui bahwa semakin tinggi level dekomposisi bisa mendapatkan hasil yang lebih baik. Hal ini dibuktikan pada Tabel 1. Dimana didapatkan hasil terbaik pada level 9 dengan JST maupun dengan AG optimasi arsitektur JST. Lalu level 8 dengan AG optimasi arsitektur JST serta optimasi update bobot dan bias.

Penggunaan AG dalam optimasi JST dapat dilakukan untuk optimasi arsitektur JST maupun optimasi arsitektur JST dan update bobot dan bias. Namun dalam penelitian ini, optimasi yang memberikan hasil lebih baik adalah optimasi arsitektur JST. Hal ini dikarenakan pada optimasi update bobot dan bias kesulitan dalam memprediksi kelas *wheeze* dan *crackle+wheeze*. Akurasi untuk optimasi arsitektur JST menghasilkan akurasi tertinggi mencapai 67,54% lebih baik daripada hasil pada JST tanpa optimasi yaitu 66,95%.

Sekilas didapat nilai akurasi yang terlihat relatif rendah. Data set yang digunakan adalah data set pada *Int. Conf. on Biomedical Health Informatics - ICBHI 2017 challenge* dengan tingkat kesulitan yang tinggi [10]. Akurasi tertinggi yang dicapai pada kompetisi tersebut hanya mencapai 80%. Fokus pada penelitian ini adalah pengaruh optimasi AG pada pemilihan ciri dan struktur JST terhadap peningkatan akurasi. Dari Tabel 1 terlihat bahwa terdapat peningkatan akurasi pada penggunaan GA untuk optimasi arsitektur JST. Sementara itu akurasi pada aoptimasi arsitektur, update bobot dan bias menghasilkan akurasi yang lebih rendah karena konstrain dari GA terlalu banyak sehingga akurasi tidak mencapai maksimal. Tetapi apabila dilihat pada Tabel 3, maka terlihat bahwa optimasi yang dilakukan menhasilka arsitektur JST yang lebih sederhana dengan jumlah node yang lebih sedikit dan iterasi yang lebih sedikit. Akurasi 66,95% pada JST tanpa optimasi dicapai dengan 100 node dan 5000 iterasi. Sementara itu pada optimasi arsitektur JST dihasilkan akurasi 67,54% pada jumlah node yang lebih sedikit dan iterasi hanya 167.

Peningkatan akurasi yang relative kecil dalam penelitian ini disebabkan beberapa keterbatasan. Keterbatasan dalam penelitian ini adalah tidak adanya penanganan pada *imbalance data* dan eksplorasi pada parameter GA yang terbatas. Selain itu pemilihan subband yang relevan dengan suara paru juga tidak dilakukan seperti pada [12]. Eksplorasi lebih lanjut dari berbagai

parameter GA dan pemilihan subband untuk ekstraksi ciri menjadi penelitian yang menarik di masa berikutnya.

5. Kesimpulan

Pada penelitian ini diusulkan optimasi AG untuk meningkatkan akurasi pada klasifikasi suara paru menggunakan JST. Peran AG dalam optimasi JST meliputi optimasi arsitektur JST dan *update* bobot, dimana optimasi arsitektur JST lebih baik disertai dengan pemilihan fungsi aktivasi. Optimasi yang memberikan hasil terbaik yaitu optimasi arsitektur JST dengan akurasi mencapai 67.54%. Arsitektur yang terbentuk yaitu dengan 3 buah *hidden layer* (49, 32, 38). Hasil ini juga didapatkan dengan penggunaan dekomposisi DWT level 9. Penggunaan GA untuk optimasi arsitektur JST menghasilkan akurasi yang lebih tinggi dibandingkan dengan JST tanpa optimasi. Sementara itu optimasi arsitektur dan *update* bobot dan bias pada JST menghasilkan akurasi yang lebih rendah dari akurasi tanpa optimasi. Hal ini disebabkan oleh sifat GA yang tidak selalu menghasilkan nilai terbaik tetapi nilai optimal. Penelitian selanjutnya dapat menggunakan metode untuk mengatasi *imbalance* pada data pelatihan. Serta meningkatkan batas dari parameter AG yang digunakan, terlebih pada optimasi *update* bobot dan bias.

6. Daftar Pustaka

- [1] A. Rizal and V. Suryani, "Aplikasi Pengolahan Sinyal Digital pada Analisis dan Pengenalan Suara Jantung dan Paru untuk Diagnosis Penyakit Jantung dan Paru Secara Otomatis."
- [2] F. Syafria, A. Buono, and B. P. Silalahi, "Pengenalan Suara Paru-Paru dengan MFCC sebagai Ekstraksi Ciri dan Backpropagation sebagai Classifier," *J. Ilmu Komput. dan Agri-Informatika*, 2017.
- [3] A. Rizal, R. Hidayat, and H. A. Nugroho, "Multiscale Hjorth descriptor for lung sound classification," in *AIP Conference Proceedings*, 2016, vol. 1755.
- [4] A. Rizal, R. Hidayat, and H. A. Nugroho, "Fractal Dimension for Lung Sound Classification in Multiscale Scheme," *J. Comput. Sci.*, vol. 14, no. 8, pp. 1081–1096, 2018.
- [5] A. Bohadana, G. Izbicki, and S. S. Kraman, "Fundamentals of lung auscultation.," *N. Engl. J. Med.*, vol. 370, no. 8, pp. 744–51, Feb. 2014.
- [6] A. Rizal, R. Hidayat, and H. A. Nugroho, "Comparison of Discrete Wavelet Transform and Wavelet Packet Decomposition for The Lung Sound Classification," *Far East J. Electron. Commun.*, vol. 17, no. 5, pp. 1065–1078, Oct. 2017.
- [7] N. Ji, L. Ma, H. Dong, and X. Zhang, "EEG signals feature extraction based on DWT and EMD combined with approximate entropy," *Brain Sci.*, 2019.
- [8] A. B. Correa and A. M. Gonzalez, "Evolutionary algorithms for selecting the architecture of a MLP Neural Network: A credit scoring case," in *Proceedings - IEEE International Conference on Data Mining, ICDM*, 2011.
- [9] C. M. Anderson-Cook, "Practical Genetic Algorithms," *J. Am. Stat. Assoc.*, 2005.
- [10] F. Tempola, M. Muhammad, and A. Khairan, "Perbandingan Klasifikasi Antara KNN dan Naive Bayes pada Penentuan Status Gunung Berapi dengan K-Fold Cross Validation," *J. Teknol. Inf. dan Ilmu Komput.*, 2018.
- [11] B. M. Rocha *et al.*, "An open access database for the evaluation of respiratory sound classification algorithms," *Physiol. Meas.*, vol. 40, no. 3, 2019.
- [12] A. Rizal, R. Hidayat, and H. A. Nugroho, "Lung Sound Classification Using Hjorth Descriptor Measurement on Wavelet Sub-bands," *J. Inf. Process. Syst.*, vol. 15, no. 5, pp. 2–14, 2019.