

Klasifikasi Arritmia pada Sinyal EKG menggunakan Deep Neural Network

Bayu Wijaya Putra^{*1}, Rahmat Fadli Isnanto², Purwita Sari³, Ariansyah Saputra⁴, M. Rudi Sanjaya⁵, Akhmad Noviar Satria Mukti⁶

^{1,2,3,5,6} Fakultas Ilmu Komputer, Universitas Sriwijaya; Palembang

⁴Jurusan Teknik Komputer, Politeknik Negeri Sriwijaya, Palembang

e-mail: ^{*}bayuwisata@gmail.com, ²fadlyfadfad@gmail.com, ³wita@ilkom.unsri.ac.id,

⁴ariansyah@polsri.ac.id, ⁵m.rudi.sjy@ilkom.unsri.ac.id, ⁶ahmadnoviar.19@gmail.com

Abstrak

Penelitian yang dikembangkan saat ini memfokuskan klasifikasi sinyal Electrocardiogram (EKG) pada gangguan arritmia detak jantung. Monitoring ini bertujuan agar dapat menjadi penanganan dini terhadap berbagai jenis gangguan arritmia. Klasifikasi yang diajukan dapat mengklasifikasi 9 jenis gangguan arritmia dengan menggunakan metode Deep Neural Network (DNN). Teknik preprocessing data pada sinyal EKG sebelum proses klasifikasi, yaitu segmentasi, normalisasi menggunakan normalize bound, dan fitur extraction dengan menggunakan autoencoder. Hasil menunjukkan bahwa metode yang digunakan mendapatkan nilai akurasi yang sangat baik sebesar 99.62% dan sensitivity about 97.18%.

Kata kunci—EKG, Arritmia, Klasifikasi, Deep Neural Network

Abstract

The research developed today focuses the classification of Electrocardiogram (ECG) signals on heart rate arrhythmia disorders. This monitoring aims to be an early treatment of various types of arrhythmia disorders. Using the Deep Neural Network (DNN) process, the proposed classification will identify 9 kinds of arrhythmia disorders. Preprocessing of the ECG signal data technique before the classification process, namely segmentation, normalization using bound normalization, and autoencoder extraction function. Results showed that the system used gained an outstanding 99.62 percent precision value and about 97.18 percent sensitivity.

Kata kunci—ECG, Arrhythmia, Classification, Deep Neural Network

1. PENDAHULUAN

Penyakit jantung perlu pengawasan lebih oleh praktisi kesehatan. Pengawasan dan prosesing data kesehatan jantung pasien dapat mendeteksi berbagai penyakit kardiovaskular dan juga mencegah masalah kesehatan tepat waktu. Namun, kedua pengawasan dan prosesing data jantung pasien secara terus menerus itu merepotkan. Pengawasan jantung terus menerus menghasilkan data berjumlah besar dan menganalisa data besar ini menggunakan metode konvensional itu sangat sulit dilakukan [1]. Peneliti penyakit jantung tidak hanya tentang proteksi masyarakat, tetapi juga merupakan tantangan teknologi yang memotivasi pada pengembangan teknologi analisis. Perangkat yang dapat menghasilkan record monitoring kesehatan jantung berupa elektrokardiogram (EKG) [2].

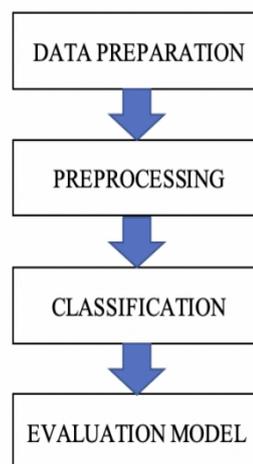
Analisis menggunakan EKG dapat mendeteksi dan mengkategorikan bentuk gelombang yang berbeda dan morfologi dalam sinyal [3]. Sinyal EKG memiliki tiga bentuk gelombang, yaitu gelombang puncak P, kompleks QRS dan gelombang T dalam keadaan normal [4]. Gelombang P merepresentasikan depolarisasi atrial, gelombang QRS merepresentasikan depolarisasi ventrikular dan gelombang T merepresentasikan repolarisasi dari ventricle [5]. Pada kasus tertentu, bentuk EKG berubah bentuk gelombang T yaitu pada, panjang interval ST, elevasi ST, dimana morfologi ini yang menyebabkan cardiac abnormality [6]. Deteksi abnormal detak jantung bergantung pada pemeriksaan sinyal EKG selama proses pengambilan sampel yang memadai. Proses sampling membutuhkan data yang cukup yang dapat di ekstrak menjadi fitur. Fitur tersebut dapat menghasilkan pengukuran akurat untuk mendeteksi abnormal sinyal EKG [7].

Dalam beberapa tahun sebelumnya, muncul tantangan penelitian deteksi sinyal normal dan abnormal. Pendekatan yang digunakan menggunakan proses klasifikasi bersumber dari data berupa ritme dan beat. Penggunaan beat sebagai data klasifikasi telah digunakan banyak penelitian dengan hasil yang baik. Metode yang digunakan menggunakan pendekatan deep learning, seperti hybrid neural network, kombinasi neural network, dan Artificial Neural Network [7], serta deep neural network [8][9].

Metode Deep Neural Network (DNN) digunakan pada teknik supervised machine learning dengan menggunakan tiga atau lebih layer [10][9]. DNN telah digunakan pada banyak riset seperti image processing, face recognition, dan klasifikasi ECG signal [8]. Kelebihan metode DNN pada klasifikasi ECG adalah mengekstraksi fitur dengan mempelajari data dan tidak memaksakan fitur berdasarkan hasil preprocessing [11]. Pada Preprocessing beat ECG terdapat proses fitur extraction. Fitur extraction merupakan tahap penting dalam proses learning untuk mendapatkan fitur yang bagus dan kuat [12]. Pada penelitian [8][12] menunjukkan bahwa metode autoencoder untuk meningkatkan hasil akurasi klasifikasi.

2. METODE PENELITIAN

Pada penelitian ini melalui lima tahap dalam klasifikasi abnormal pada sinyal EKG (Gambar 1) , Dataset untuk persiapan data, preprocessing data, fitur ekstraksi, klasifikasi (training data, testing hasil training data and validasi data) dan di akhiri dengan model evaluasi.



Gambar 1. Blok Diagram Tahapan Penelitian Deteksi pada Sinyal EKG

Dataset

Dataset EKG berlabel yang digunakan adalah MIT-BIH Arrhythmia Database bersumber dari physionet.org yang dikembangkan oleh MIT berisi 48 rekaman elektrokardiogram saat observasi pasien dari yang didukung oleh MIT [13] [3]. Rekaman disimpan dalam bentuk digital berjumlah 360 sampel/detik/kanal dengan resolusi 11-bit dengan jangkauan lebih dari 10 mV [8]. Oleh karena itu database ini cocok untuk mengevaluasi kinerja dan akurasi perangkat keras yang dikembangkan untuk spektrum penyakit jantung yang luas.

EKG record adalah kumpulan dari berbagai bentuk gelombang, artefak, kompleks ventricular dan konduksi abnormal. Setiap record disertai dengan file anotasi dimana tiap detak EKG telah dilabeli oleh lebih dari dua ahli kardiologist [16]. Label ini, disebut sebagai anotasi kebenaran dan digunakan untuk menjadi model evaluasi pada bagian 3.5 Result. Hal ini memungkinkan deteksi serial komponen yang berbeda dari sinyal EKG dalam mendeteksi sinyal gangguan arritmia.

Tabel 1. Data label pada kelas gangguan arritmia

Label	Deskripsi
V	Ventricular premature contraction
F	Fusion of ventricular and normal
O	Ventricular flutter wave
e	Atrial escape beat
j	Nodal (junctional) escape beat
E	Ventrikular escape beat
P	Paced beat
f	Fusion of paced and normal beat
x	Non-conducted P-wave

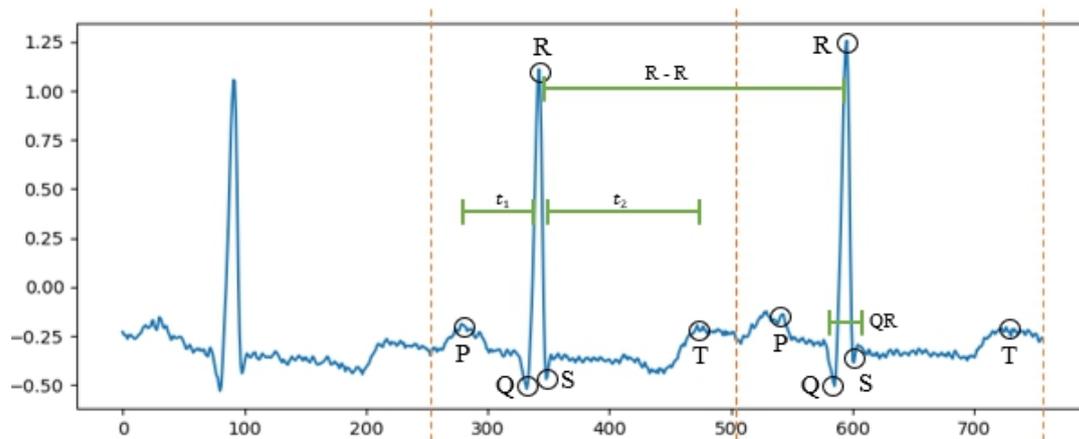
Preprocessing

Proses pertama dalam penelitian ini adalah preprocessing guna membantu menghilangkan berbagai interupsi dari sinyal original EKG. Proses eliminasi noise dapat menggunakan metode yang berbeda. Preprocessing dilakukan sebelum proses klasifikasi, yang bertujuan untuk meningkatkan efisiensi sistem. Preprocessing terdiri dari segmentasi, data normalisasi, dan ekstraksi fitur.

Segmentasi

Proses segmentasi sinyal dimulai dengan mengetahui jarak antara 2 puncak R pada ritme sinyal original (gambar 2), jika telah ditemukan posisi R ke R maka bisa menentukan gelombang Q dan S kemudian akan di dapatkan gelombang QRS, sehingga dapat ditentukan gelombang P dan gelombang T. [17].

Pada proses segmentasi *ritme menjadi beat* ditentukan dari 0,25 detik sebelum puncak R disebut durasi T_1 dan 0,45 detik setelah puncak R disebut dengan durasi T_2 (Gambar 2). Hasil akumulasi panjang 1 detak adalah 0,7 detik atau 252 node yang meliputi gelombang P, QRS kompleks, dan gelombang T [6].

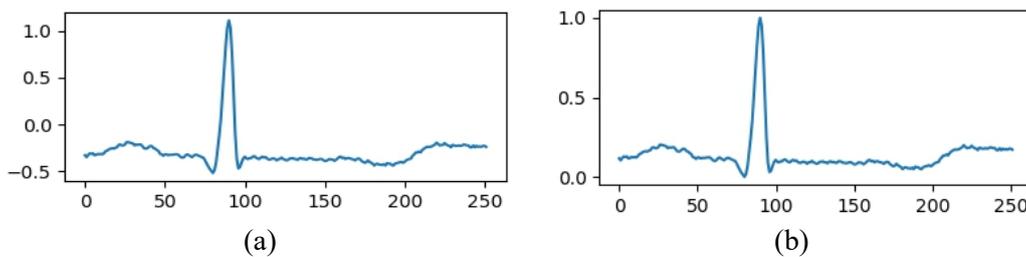


Gambar 2 Segmentasi dari sinyal Ritme menjadi sinyal beat

Normalisasi Data

Normalisasi data digunakan untuk mengubah nilai rentang amplitudo sinyal yang telah disegmentasi agar memiliki nilai amplitudo seragam tanpa mengubah morfologi [18]. Pada penelitian ini metode normalisasi data yang digunakan adalah *Normalize Bound*. *Normalize Bound* mengubah nilai batas bawah (lb) dan batas atas (ub) pada amplitudo sinyal tersebut ke rentang yang lebih kecil tanpa mengubah pola atau bentuk dari sinyal awalnya. Data yang telah

disegmentasi diberikan batas atas dengan batas bawah mulai dari 0 sampai 1 beserta kedua dengan batas atas (gambar 3.b). Setelah semua data sudah dikelompokkan berdasarkan batas atas dan batas bawahnya. Maka masing-masing data baru didapat di *Normalize Bound* dengan properti yang telah disebutkan diatas. Setelah seluruh data di *Normalize Bound* maka data tersebut sudah siap untuk di ekstrak fiturnya.



Gambar 3. Perubahan rentang sinyal tanpa mengubah bentuk morfologi (a) menjadi (b)

Feature Ekstraksi

Ekstraksi fitur digunakan dalam proses pengambilan ciri khas fitur yang dapat menggambarkan karakteristik dari beat yang telah dinormalisasi [19]. Rahhal [8] menunjukkan bahwa processing menggunakan autoencoder dapat meningkatkan hasil accuracy. Preprocessing penelitian ini digunakan model *autoencoder*. Penelitian ini mengajukan 4 model *Autoencoder* yang masing-masing memiliki perbedaan jumlah fitur pada Hidden Layer. Arsitektur *Autoencoder* yang digunakan terdiri dari empat model yang memberikan perbedaan pada output panjang fitur, yaitu 1,3,5,7,9,11 hidden layer (Tabel 2). Arsitektur *Autoencoder* yang digunakan pada *Input* layer dan *Output* layer memiliki 252 fitur yang jumlah tersebut merupakan panjang 1 sinyal detak jantung.

Tabel 2. Model Fitur Ekstraksi

Model	Jumlah Hidden Layer	Jumlah Node pada setiap Layer	Input ke Fitur DNN
1	1	[252, 126, 252]	126
2	3	[252, 126, 63, 126, 252]	63
3	5	[252, 126, 63, 32, 63, 126, 252]	32
4	7	[252, 126, 63, 32, 16, 32, 63, 126, 252]	16
5	9	[252, 126, 63, 32, 16, 8, 16, 32, 63, 126, 252]	8
6	11	[252, 126, 63, 32, 16, 8, 4, 8, 16, 32, 63, 126, 252]	4

Goodfellow [20] menjelaskan persamaan (1) yang digunakan dalam fitur extraction menggunakan persamaan berikut

$$L(x, g(f(x))) \quad (1)$$

Pada *hidden* layer yang pertama digunakan fungsi aktivasi *Rectified Linear Unit (ReLU)*. Kemudian pada *Output* layer digunakan fungsi aktivasi *Sigmoid*.

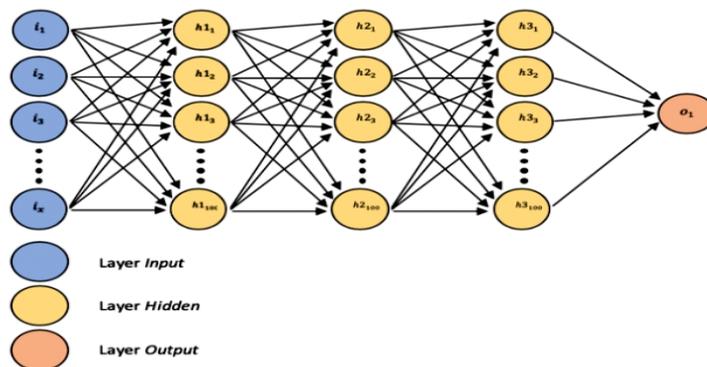
Penyusunan (*COMPILE*) pada sebuah model *Autoencoder* di butuhkan dua buah argumen, yaitu *pengoptimalan (Optimizer)* dan *Fungsi Loss (Loss Function)*. Adapun *pengoptimalan (Optimizer)* yang digunakan pada *Autoencoder* ini adalah *Adaptive Moment Estimation (ADAM)*. Kemudian fungsi *loss* yang digunakan di model *Autoencoder* ini adalah *mean squared error*. Proses membuat arsitektur dan menyusun arsitektur tersebut maka *Autoencoder* tersebut siap untuk *Training (FIT)* setidaknya minimal argumen yang dibutuhkan ada 3 yaitu, 1. Data yang digunakan untuk training, 2. Epoch atau perulangan, 3. Batch size. Epoch yang dilakukan pada sebanyak 400 dan Batch Size yang digunakan sebesar 64.

Klasifikasi

Metode klasifikasi yang digunakan pada adalah *Deep Neural Network (DNN)*. Metode *Deep Neural Network* termasuk kedalam bagian *Deep Learning* karena fitur diproses menggunakan banyak layer dan menggunakan algoritma *Back-Propagation* [16].

Sama seperti *autoencoder*, *Deep Neural Network* juga berbentuk sekuensial karena pada dasarnya mereka merupakan *Neural Network*. Model *Deep Neural Network* yang digunakan pada penelitian ini memiliki arsitektur yaitu, 5 layer secara keseluruhan dimana 2 diantaranya adalah *Input Layer* dan *Output Layer* [18] (Gambar 4).

Pada *Input Layer* terdapat sebanyak panjang x yang merujuk pada panjang tiap 1 sinyal fitur. Nilai x merepresentasikan panjang fitur yang bersesuaian dengan panjang fitur output hasil fitur ekstraksi sebelumnya (tabel 1). Pada *Output Layer* terdapat 1 node dimana node tersebut merupakan banyaknya kelas yang diklasifikasikan.



Gambar 4. Arsitektur klasifikasi DNN

Pada model *klasifikasi* ini digunakan fungsi aktivasi *Rectified Linear Unit (ReLU)* pada seluruh layer *Hidden* dan fungsi aktivasi *Softmax* pada Layer *Output* [21]. Bersamaan dengan fungsi *loss* yang digunakan ada dua yaitu, fungsi *loss Categorical Crossentropy* [22] dan fungsi *loss Mean Squared Error (MSE)*, dan *optimizer ADAM*. Persamaan fungsi aktivasi *Softmax(2)*:

$$f_i(\vec{x}) = \frac{e^{x_i}}{\sum_{j=1}^J e^{x_j}} \text{ for } i = 1, \dots, J \quad (2)$$

Pada tahap ini arsitektur dari model *Deep Neural Network* sudah jadi dan siap untuk digunakan training. Fitur yang digunakan untuk training merupakan yang sudah diekstrak pada tahap sebelumnya atau tahap ekstraksi fitur dan label yang digunakan adalah label yang telah dipilah pada tahap persiapan data. Sebelum Model ini di-Training seluruh data fitur dan label dibagi menjadi masing-masing 72:18:10 data, dimana 72 adalah persentase jumlah fitur dan label yang digunakan untuk Training Model, 18 adalah persentase jumlah fitur dan label yang digunakan untuk Testing, sedangkan 10 adalah persentase jumlah fitur dan label yang digunakan untuk Validation. Model ini di-Training sebanyak 150 *Epoch* dan *batch size* 48.

Pada Tahap *Training*, Model dari *Autoencoder* tersebut sudah bisa digunakan untuk merekonstruksikan sinyal. Untuk merekonstruksi sinyal tersebut dibutuhkan 2 tahapan, pada tahap pertama dilakukan prediksi sinyal input pada layer *Hidden* atau disebut juga dengan *encoder*. Pada tahap kedua dilakukan prediksi sinyal keluaran dari layer *Hidden* pada layer *Output* atau disebut juga dengan *decoder*. Hasil prediksi dari layer *Output* adalah sinyal yang sudah di rekonstruksi. Setelah didapat sinyal *encoder* sinyal inilah yang akan dijadikan fitur untuk klasifikasi.

Model *Deep Neural Network* sudah di-Training maka akan didapati model *akurasi* dan model *loss* pada model evaluasi. Dengan kedua nilai tersebut dapat diukur performa dari Model yang dibuat. Untuk mengukur performa dari Model ini akan dijelaskan pada Model Evaluasi.

Model evaluasi

Performa dari Model yang dibuat menggunakan *Confussion Matrix* (Matriks kebingungan). Sebelum bisa mendapatkan *Confussion Matrix* harus lebih dahulu mem-Predict kelas dari fitur Testing untuk mendapatkan nilai dari *Confussion Matrix*.

Setelah mendapatkan nilai *Confussion Matrix* sekarang dapat mengukur performa dari Model ini. Ada beberapa tahap sebelum bisa mendapatkan nilai performa akhir. Pertama, mendapatkan nilai *True Positif* (Benar Positif), *False Positif* (Salah Positif), *False Negatif* (Salah Negatif) dan *True Negatif* (Benar Negatif). Kedua, mendapatkan nilai Sensitivitas (5), Spesifisitas (7), Presisi (4), Nilai F1 (6), Rasio Error (8) dan Akurasi (3). Berikut adalah persamaan-persamaan dari alat ukur diatas.

$$\text{Akurasi (Acc)} = \frac{tp+tn}{tp+fp+fn+tn} \quad (3)$$

$$\text{Presisi (Pre)} = \frac{tp}{tp+fp} \quad (4)$$

$$\text{Sensitivitas (Sen)} = \frac{tp}{tp+fn} \quad (5)$$

$$\text{Nilai F1 (F1)} = \frac{(\beta^2+1)tp}{(\beta^2+1)tp+\beta^2fn+fp} \quad (6)$$

$$\text{Spesifisitas (Spe)} = \frac{tn}{fp+tn} \quad (7)$$

$$\text{Rasio Error (E)} = \frac{\sum_{i=1}^I \frac{fp_i+fn_i}{tp_i+fp_i+fn_i+tn_i}}{I} \quad (8)$$

3. HASIL DAN PEMBAHASAN

Percobaan yang dilakukan dengan membuat perbedaan pada jumlah hidden layer pada proses ekstraksi fitur yang menjadi input pada proses klasifikasi DNN. Penerapan Coding dan analisis menggunakan bahasa pemrograman Python. Percobaan yang dilakukan menggunakan 6 model evaluasi dengan memaparkan 1-10 hidden layer agar dapat menghasilkan nilai akurasi terbaik [18].

Tabel 3. Performa Training Data pada 6 model

Training	Model Evaluasi (%)					
metrik	1	2	3	4	5	6
loss	15,09	9,80	8,13	8,37	8,33	9,72
akurasi	95,72	97,22	97,68	97,53	97,53	97,21

Tabel 4. Performa Testing Data pada 6 model

Testing	Model Evaluasi (%)					
metrik	1	2	3	4	5	6
loss	15,80	11,04	9,72	10,58	12,93	12,87
akurasi	95,75	97,02	97,40	97,05	96,59	96,52

Tabel 5. Performa Validasi Data pada 6 model

Validasi	Model Evaluasi (%)					
Metrik	1	2	3	4	5	6
Akurasi	99,55	99,59	99,62	99,38	99,55	99,57
Sensitivitas	94,93	96,12	97,18	89,81	91,82	97,18
Spesifisitas	99,71	99,78	99,74	99,60	99,72	99,66
Presisi	82,74	93,71	95,40	91,82	89,84	82,26
Nilai F1	91,02	96,15	95,88	90,32	90,80	92,99
Rasio Error	0,04	0,40	0,37	0,61	0,44	0,52

Selama klasifikasi telah dijalankan proses training dan testing data sebanyak 90% dari jumlah total data, dimana dari jumlah tadi dibagi 80% training dan 20% testing. Pada bagian proses ini, memberikan nilai rata-rata akurasi dan loss pada tahap training (Tabel 3) dan testing data (Tabel 4). Hasil training dan testing, dalam bentuk grafik model accuracy dan model loss (Gambar 5) ketika proses training. Sisanya 10% dari total data untuk validasi data. Pada (Tabel 5) dipaparkan bahwa nilai terbaik terdapat pada model 6 hidden layer dimana akurasi mencapai 99,62%, Sensitivitas mencapai 97,18%, Spesifisitas mencapai 99,74%, Presisi mencapai 95,40%, Nilai F1 mencapai 95,88% dan Rasio Error 0,37%.

Tabel 6. Confusion Matriks pada Training Data

Kelas	V	F	O	e	j	E	P	f	x
V	5111	22	0	0	0	0	0	0	0
F	49	528	0	0	0	0	0	1	0
O	1	0	339	0	0	0	0	0	0
e	1	0	0	9	0	0	0	1	0
j	3	0	0	0	160	0	0	2	0
E	6	0	0	0	0	70	0	0	0
P	0	0	0	0	0	0	5056	1	0
f	3	0	0	0	0	0	11	695	0
x	2	0	0	0	0	0	0	0	137

Tabel 7. Confusion Matriks pada Testing Data

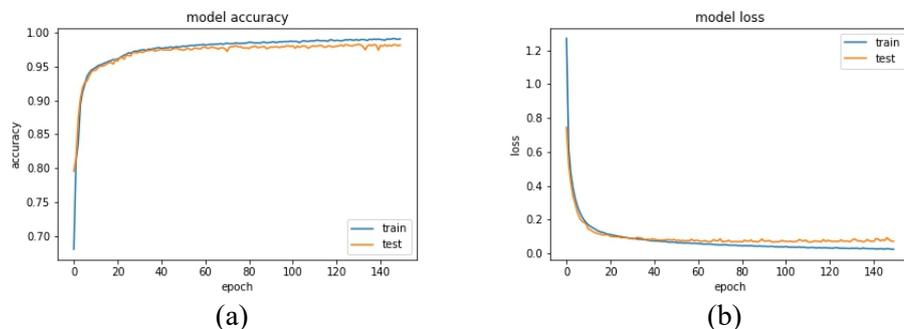
Kelas	V	F	O	e	j	E	P	f	x
V	1269	8	4	0	1	0	0	3	0
F	8	132	0	0	0	0	0	2	0
O	0	0	76	0	0	0	1	0	2
e	0	0	0	2	0	0	0	1	0
j	0	1	0	0	38	0	0	2	0
E	0	0	1	0	1	17	0	1	0
P	0	0	1	0	1	0	1263	1	0
f	3	0	0	0	0	0	4	170	0
x	1	0	0	0	0	0	0	0	34

Tabel 8. Confusion Matriks pada Validasi Data

Kelas	V	F	O	e	j	E	P	f	x
V	702	8	0	0	0	0	1	2	0
F	4	76	0	0	0	0	0	0	0
O	3	0	44	0	0	0	0	0	0
e	0	0	0	2	0	0	0	0	0
j	1	0	0	0	21	0	0	2	0
E	0	1	0	0	0	10	0	0	0
P	1	0	0	0	0	0	701	0	1
f	3	0	0	0	0	1	1	93	0
x	0	1	0	0	0	0	0	0	18

Proses akhir dari klasifikasi, membuat confusion matriks yang digunakan untuk menghitung rata-rata validasi data. Pada Tabel 6, Tabel 7 dan Tabel 8 dipaparkan dengan menggunakan model 5 hidden layer sebagai hasil terbaik. Berdasarkan tabel 6 terdapat kurang dari 1% dari data sinyal beat EKG yang tidak terklasifikasi, namun ini terjadi karena imbalance data, sehingga perlu menggunakan nilai F1 untuk menunjukkan data yang akurat. Pada (Tabel 5) Nilai F1 mencapai 95,88%, pada model terbaik. Pada beberapa penelitian yang menggunakan metode neural network. Pada penelitian [7] klasifikasi sinyal heartbeat EKG menggunakan ANN dengan accuracy 98.70%. Pada [9] menggunakan metode deep neural network accuracy 98.31%. Pada klasifikasi ini menunjukkan nilai terbaik mencapai 99,62%. Guna menunjukkan hasil baik untuk memvalidasi akurasi, telah dipaparkan model akurasi dan model loss (Gambar

5). Pada (Gambar 5) menunjukkan hasil testing yang meningkat seiring dengan hasil training. Di sisi lain, pada model loss, error menurun bersamaan dengan bertambahnya epoch. Hasil evaluasi menunjukkan perbedaan jumlah hidden layer sangat mempengaruhi hasil accuracy model evaluasi .



Gambar 5. Grafik Validasi Model Akurasi (a) dan Model Loss (b)

Panjang pendek nya fitur output mempengaruhi hasil testing dan validation pada proses klasifikasi DNN. Hasil ini ditampilkan pada (Gambar 5) dalam grafik model accuracy dan model loss. DNN dengan menggunakan input panjang fitur yang berbeda menunjukkan nilai model evaluasi yang berbeda, namun menghasilkan nilai pengukuran yang bervariasi. Pada model pengukuran 5 hidden layer memberikan nilai accuracy tertinggi mencapai 99,62% diikuti dengan model 3 hidden layer 99.59% yang terpaud beda 0.03%.

4. KESIMPULAN

Berdasarkan hasil pengukuran model evaluasi, bahwa model yang terbaik pada percobaan berikut dimiliki oleh percobaan model ke 3 dengan 5 hidden layer pada klasifikasi DNN. Hasil terbaik dipaparkan bahwa nilai akurasi mencapai 99,62%, Sensitivitas mencapai 97,18%, Sensitivitas mencapai 97,18%, Spesifisitas mencapai 99,74%, Presisi mencapai 95,40%, Nilai F1 mencapai 95,88% dan Rasio Error 0,37%.

5. SARAN

Perlu pengembangan penelitian menjalankan model keilmuan hasil training data yang harapannya dapat diaplikasikan pada software maupun hardware secara realtime.

UCAPAN TERIMA KASIH

Terima kasih atas Hibah Penelitian Sateks 2020 Universitas Sriwijaya yang telah mendanai penelitian dan Jurnal Jupiter yang telah membantu publikasi penelitian ini.

DAFTAR PUSTAKA

- [1] O. Castro-Lopez, D. E. Lopez-Barron, and I. F. Vega-Lopez, "Next-generation heartbeat classification with a column-store DBMS and UDFs," *J. Intell. Inf. Syst.*, Apr. 2019.
- [2] P. M. Kumar and U. Devi Gandhi, "A novel three-tier Internet of Things architecture with machine learning algorithm for early detection of heart diseases," *Comput. Electr. Eng.*, 2018.
- [3] M. Kachuee, S. Fazeli, and M. Sarrafzadeh, "ECG Heartbeat Classification: A Deep

- Transferable Representation,” in *2018 IEEE International Conference on Healthcare Informatics (ICHI)*, 2018, pp. 443–444.
- [4] W. N. Evans, R. J. Acherman, G. A. Mayman, R. C. Rollins, and K. T. Kip, “Simplified pediatric electrocardiogram interpretation,” *Clinical Pediatrics*. 2010.
- [5] N. Maglaveras, T. Stamkopoulos, K. Diamantaras, C. Pappas, and M. Strintzis, “ECG pattern recognition and classification using non-linear transformations and neural networks: A review,” in *International Journal of Medical Informatics*, 1998.
- [6] Q. Qin, J. Li, L. Zhang, Y. Yue, and C. Liu, “Combining Low-dimensional Wavelet Features and Support Vector Machine for Arrhythmia Beat Classification,” *Sci. Rep.*, vol. 7, no. 1, p. 6067, Dec. 2017.
- [7] E. Al-Masri, “Detecting ECG Heartbeat Abnormalities using Artificial Neural Networks,” in *2018 IEEE International Conference on Big Data (Big Data)*, 2018, pp. 5279–5281.
- [8] M. M. Al Rahhal, Y. Bazi, H. AlHichri, N. Alajlan, F. Melgani, and R. R. Yager, “Deep learning approach for active classification of electrocardiogram signals,” *Inf. Sci. (Ny)*., vol. 345, pp. 340–354, Jun. 2016.
- [9] J. H. Kim, S. Y. Seo, C. G. Song, and K. S. Kim, “Assessment of Electrocardiogram Rhythms by GoogLeNet Deep Neural Network Architecture,” *J. Healthc. Eng.*, 2019.
- [10] F. Firdaus, M. Anshori, S. P. Raflesia, A. Zarkasi, M. Afrina, and S. Nurmaini, “Deep Neural Network Structure to Improve Individual Performance based Author Classification,” *Comput. Eng. Appl. J.*, vol. 8, no. 1, pp. 77–83, Feb. 2019.
- [11] J. Masci, A. Giusti, D. Ciresan, G. Fricout, and J. Schmidhuber, “A fast learning algorithm for image segmentation with max-pooling convolutional networks,” in *2013 IEEE International Conference on Image Processing, ICIP 2013 - Proceedings*, 2013.
- [12] S. Nurmaini, R. U. Partan, W. Caesarendra, and T. Dewi, “applied sciences An Automated ECG Beat Classification System Using Deep Neural Networks with an Unsupervised Feature Extraction Technique.”
- [13] S. Kiranyaz, T. Ince, and M. Gabbouj, “Real-Time Patient-Specific ECG Classification by 1-D Convolutional Neural Networks,” *IEEE Trans. Biomed. Eng.*, vol. 63, no. 3, pp. 664–675, Mar. 2016.
- [14] G. B. Moody and R. G. Mark, “The impact of the MIT-BIH Arrhythmia Database,” *IEEE Eng. Med. Biol. Mag.*, vol. 20, no. 3, pp. 45–50, 2001.
- [15] R. G. Mark, P. S. Schluter, G. Moody, P. Devlin, and D. Chernoff, “ANNOTATED ECG DATABASE FOR EVALUATING ARRHYTHMIA DETECTORS.,” in *IEEE/Engineering in Medicine and Biology Society Annual Conference*, 1982.
- [16] S. Nurmaini, P. R. Umi, R. M. Naufal, and A. Gani, “Cardiac arrhythmias classification using Deep Neural Networks and principle component analysis algorithm,” *Int. J. Adv. Soft Comput. its Appl.*, 2018.
- [17] E. J. da S. Luz, W. R. Schwartz, G. Cámara-Chávez, and D. Menotti, “ECG-based heartbeat classification for arrhythmia detection: A survey,” *Comput. Methods Programs Biomed.*, vol. 127, pp. 144–164, Apr. 2016.
- [18] B. W. Putra, M. Fachrurrozi, M. R. Sanjaya, A. Muliawati, A. N. S. Mukti, and S. Nurmaini, “Abnormality Heartbeat Classification of ECG Signal Using Deep Neural Network and Autoencoder,” in *2019 International Conference on Informatics, Multimedia, Cyber and Information System (ICIMCIS)*, 2019, pp. 213–218.
- [19] A. E. Vincent and K. Sreekumar, “A survey on approaches for ECG signal analysis with focus to feature extraction and classification,” in *2017 International Conference on Inventive Communication and Computational Technologies (ICICCT)*, 2017, pp. 140–144.
- [20] I. Goodfellow, Y. Bengio, and A. Courville, “Deep Learning TOC,” *Deep Learn. B.*, 2017.
- [21] G. Sannino and G. De Pietro, “A deep learning approach for ECG-based heartbeat classification for arrhythmia detection,” *Futur. Gener. Comput. Syst.*, vol. 86, pp. 446–455, Sep. 2018.
- [22] S. Nurmaini, R. Umi Partan, and M. Naufal Rachmatullah, “Deep classifier on the electrocardiogram interpretation system,” *J. Phys. Conf. Ser.*, vol. 1246, p. 012030, 2019.