

## Implementasi *Extreme Learning Machine* dan *Fast Independent Component Analysis* untuk Klasifikasi Aritmia Berdasarkan Rekaman Elektrokardiogram

Aditya Septadaya<sup>1</sup>, Candra Dewi<sup>2</sup>, Bayu Rahayudi<sup>3</sup>

Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya  
Email: <sup>1</sup>aseptadaya@null.net, <sup>2</sup>dewi\_candra @ub.ac.id, <sup>3</sup>ubay1 @ub.ac.id

### Abstrak

Jenis aritmia dapat mengindikasikan letak gangguan dan penyebabnya. Cara mengetahui aritmia adalah dengan menggunakan strip elektrokardiogram (EKG). *Machine learning* dapat digunakan sebagai pendekatan untuk memudahkan identifikasi aritmia melalui EKG. *Extreme Learning Machine* (ELM) adalah salah satu *single-hidden layer feedforward neural networks* (SLFNs) yang dapat digunakan untuk klasifikasi aritmia dalam rangka membantu diagnosa medis. Untuk mengoptimalkan performa ELM, digunakan algoritme *Fast Independent Component Analysis* (FastICA) untuk *preprocessing* dan ekstraksi sinyal EKG. Dalam penelitian ini, dilakukan beberapa pengujian parameter untuk mengetahui dampaknya terhadap performa model klasifikasi. Data EKG diperoleh dari database aritmia yang dikelola oleh Massachusetts Institute of Technology-Beth Israel Hospital (MIT-BIH). Setiap data merupakan cuplikan EKG selama 3 detik dengan keseluruhan data sebanyak 210 dibagi ke dalam 6 kelas aritmia dan ritme normal. Hasil penelitian menunjukkan bahwa model klasifikasi mampu mencapai performa sempurna dengan akurasi, *precision-recall*, dan *F-1 score* sebesar 100% pada tahap pelatihan. Namun model klasifikasi mengalami *overfitting* pada tahap pengujian dengan rerata *matthew correlation coefficient* sekitar 0. *Overfitting* terjadi karena representasi fitur terlalu kompleks dan tidak sebanding dengan jumlah data yang ada. Hal ini mengakibatkan performa yang buruk pada pengujian ELM-FastICA untuk data yang belum dikenali.

**Kata kunci:** aritmia, sinyal elektrokardiogram, *extreme learning machine*, *fast independent component analysis*.

### Abstract

The type of arrhythmia can indicate the location of the disorder and its causes. The way to identify the arrhythmia is to use an electrocardiogram (ECG) strip. Machine learning can be used as an approach to assist identification of arrhythmias through an ECG. Extreme Learning Machine (ELM) is one single-hidden layer feedforward neural networks (SLFNs) that can be used for the classification of arrhythmias in order to assist medical diagnosis. To optimize ELM performance, Fast Independent Component Analysis (FastICA) algorithm is used for preprocessing and extracting ECG signals. In this study, several parameter tests were conducted to determine the impact on the performance of the classification model. ECG data obtained from the arrhythmia database managed by the Massachusetts Institute of Technology-Beth Israel Hospital (MIT-BIH). Each data is a 3 seconds ECG snippet with total of 210 data divided into 6 arrhythmia classes and normal rhythms. The results showed that the classification model was able to achieve perfect performance with accuracy, precision-recall, and F-1 score of 100% at the training stage. However, the classification model was experiencing overfitting at the testing stage with the mean of matthew correlation coefficient is approximately 0. Overfitting occurred because the feature representation is too complex and not proportional to the amount of available data. This resulted in poor performance in the ELM-FastICA test for data that was not yet recognized.

**Keywords:** arrhythmia, electrocardiogram signals, extreme learning machines, fast independent component analysis

## 1. PENDAHULUAN

Pada tahun 2010, stroke merupakan pembunuh nomor satu di Indonesia. Pada tahun 2011, satu dari tujuh orang penderita stroke meninggal dan 12 dari 1000 orang menderita stroke pada tahun 2014 (Prathivi, 2015). *Transient Ischemic Attack* (TIA) merupakan serangan stroke iskemik ringan. Statistik menunjukkan sekitar 30 persen dari orang yang mengalami TIA tanpa pengobatan yang tepat akan mengalami stroke yang lebih parah pada tahun yang sama (Stroke Association, 2016). Salah satu penanganan yang harus dilakukan paska TIA adalah melakukan rekam elektrokardiogram (EKG) (Pedersen et al., 2016). Hal ini untuk mengetahui ada tidaknya aritmia terutama *Atrial fibrillation* (Afib). Orang yang menderita Afib dan TIA lebih berisiko mengalami serangan stroke yang lebih parah jika tidak mendapatkan penanganan medis secara tepat (Sposato et al., 2015).

Afib adalah salah satu jenis aritmia pada bagian atrium jantung membuat jantung berdetak lebih cepat, sehingga membuat penderitanya berdebar. Afib menyebabkan terbentuknya emboli pada pembuluh darah. Pada sebagian besar emboli tersebut mengakibatkan stroke dan penyakit jantung (Hart et al., 2014). Sekitar 30 persen orang yang terkena TIA dan memiliki Afib juga mengalami serangan jantung pada tahun yang sama (Stroke Association, 2016).

Terdapat beberapa penelitian yang dilakukan untuk memanfaatkan *machine learning* dalam membantu diagnosa aritmia. Pada penelitian sebelumnya, pernah dilakukan klasifikasi aritmia menggunakan *Support Vector Machine - Particle Swarm Optimization*. Penelitian tersebut menghasilkan performa dengan rata rata akurasi 89,72% dengan menggunakan 40438 data detak jantung yang didapat dari 20 pasien (Melgani & Bazi, 2008).

Pada tahun 2014, Zekić-Sušac et al. (2014) melakukan perbandingan 4 metode dalam permasalahan klasifikasi dengan data berdimensi besar. Penelitian tersebut membuktikan bahwa *Support Vector Machine* (SVM) tidak selalu menjadi metode terbaik dalam permasalahan klasifikasi dengan parameter *sensitivity*, *specificity* dan *accuracy*. Pada data masukan berdimensi tinggi, *Artificial Neural Network* (ANN) menghasilkan performa yang paling baik jika dibandingkan dengan

CART *decision tree*, SVM dan K-Nearest Neighbor.

Permasalahan pada data rekam EKG adalah dimensi data yang besar dan memiliki kecenderungan bias yang tinggi akibat *mixed signal*. Pada *signal preprocessing*, terdapat “cocktail party problem” atau disebut dengan *blind signal separation* (BSS). Permasalahan ini terjadi ketika sinyal yang direkam tercampur dengan artefak dari sumber lain seperti artefak dari pernafasan dan pergerakan otot. *Noise* ini dapat mengganggu hasil klasifikasi dari rekaman EKG.

Pada tahun 2006 Huang, Zhu dan Siew (2006) memaparkan sebuah metode *single hidden layer feedforward network* (SLFNs) dalam penelitiannya dengan nama *Extreme Learning Machine* (ELM). Dalam beberapa literatur, ELM memiliki performa pembelajaran yang lebih baik dibandingkan dengan metode *backpropagation*, dan SVM dalam waktu yang signifikan lebih cepat (G. Bin Huang, 2015; G. Bin Huang, Chen, & Siew, 2014; G. Bin Huang et al., 2006).

Salah satu metode yang dapat digunakan untuk mengatasi masalah BSS adalah pendekatan statistik *independent component analysis* (ICA). Pada penelitian yang dilakukan oleh Kroupi et al. (2011), ICA juga digunakan untuk menyelesaikan permasalahan BSS pada data rekaman EEG. Pada penelitian tersebut *Fast Independent Component Analysis* (FastICA) mampu dengan baik memisahkan artefak dari sinyal EEG. Metode ini dapat digunakan untuk melengkapi ELM dalam mengolah data sinyal EKG.

Berdasarkan penjelasan yang telah dipaparkan, akan dilakukan penelitian implementasi ELM dan FastICA untuk klasifikasi aritmia berdasarkan rekaman EKG.

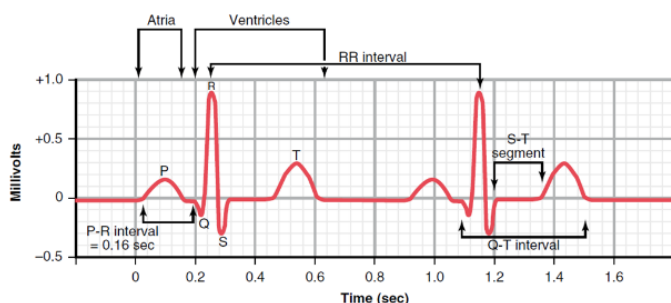
## 2. LANDASAN KEPUSTAKAAN

### 2.1 Elektrokardiogram dan Aritmia

Praktisi kesehatan mengidentifikasi aritmia dengan menganalisis strip EKG pasien yang telah direkam dalam kurun waktu tertentu. Identifikasi aritmia dilakukan dengan menganalisis komponen sinyal dari EKG. Beberapa komponen EKG pada Gambar 1 yang dianalisis dalam diagnosa aritmia dari strip EKG adalah (Thaler, 2015):

- a. Gelombang P
- b. Interval PR

- c. Kompleks QRS
- d. Segmen ST
- e. Gelombang T
- f. Interval QT



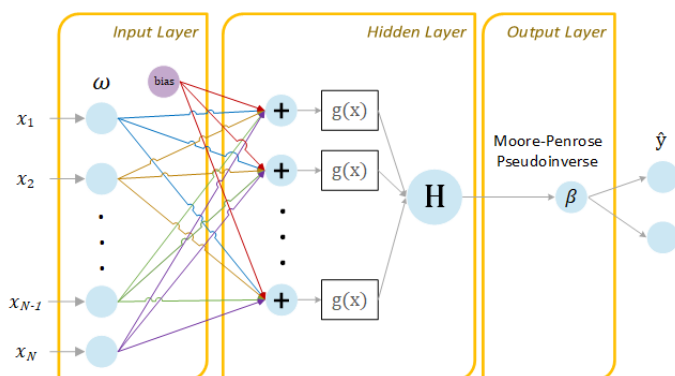
Gambar 1 Komponen sinyal EKG

Jenis aritmia dapat mengindikasikan letak gangguan serta penyebabnya. Gangguan ini pada dasarnya adalah gangguan impuls listrik yang terhantar dari saraf otak menuju titik-titik pacu jantung sehingga jantung tidak berdetak sebagaimana mestinya. Beberapa jenis aritmia diantaranya adalah:

- a. Atrial Fibrillation (Afib)
- b. Ventricular Tachycardia (VT)
- c. Premature Ventricular Contraction
- d. dll.

## 2.2 Extreme Learning Machine

Extreme Learning Machine (ELM) merupakan *single hidden layer feedforward network* (SLFN) yang dapat digunakan untuk permasalahan klasifikasi (G. Bin Huang et al., 2006). ELM menggunakan *least square solution* dari *pseudoinverse* sebagai solusi untuk estimasi fungsi dasar terhadap aproksimasi target. Arsitektur keseluruhan dari jaringan ELM diilustrasikan pada Gambar 2.



Gambar 2 Arsitektur Extreme Learning Machine

Fungsi dasar adalah proses komputasi

dimana data masukan dapat diaproksimasi menjadi nilai target. komputasi ini disebut dengan *hidden neuron*. Fungsi dasar atau *hidden neuron* dituliskan secara matematis pada Persamaan (1).

$$G(w_i, b_i, x) = g(w_i \cdot x_j + b_i) \quad (1)$$

dimana  $w_i \in R^n$  sebagai bobot masukan,  $b_i \in R$  sebagai bias dan  $j = 1, \dots, N$  sebagai banyak sampel.

Persamaan (1) adalah satu *hidden neuron* dari data masukan dan beberapa dari *hidden neuron* akan membentuk sebuah matriks *hidden layer*. *Hidden layer* pada dasarnya adalah beberapa persamaan linier yang terbentuk untuk memberikan aproksimasi dari data masukan menuju fungsi target. Persamaan *hidden layer* ditunjukkan pada persamaan (2).

$$\mathbf{H} = [G(w_i, b_i, x_j), \dots, G(w_L, b_L, x_j)] \quad (2)$$

*Hidden layer*  $\mathbf{H}$  membentuk beberapa persamaan linier sebagai fungsi dasar yang memetakan interval masukan ke fungsi target. Pemetaan tersebut dituliskan pada persamaan (3)

$$\mathbf{H}\beta = \mathbf{T} \quad (3)$$

dengan  $\beta$  sebagai *output weight* atau bobot hasil pembelajaran dan  $\mathbf{T}$  sebagai matriks target dari data latih.

Setelah matriks  $\mathbf{H}$  diperoleh, lakukan aktivasi neuron dengan fungsi aktivasi. Dalam penelitian ini digunakan fungsi aktivasi ELU (Clevert, Unterthiner, & Hochreiter, 2015). Fungsi aktivasi ELU ditunjukkan pada persamaan (4).

$$g(x) = \begin{cases} x & , x > 0 \\ \alpha(\exp(x) - 1) & , x \leq 0 \end{cases} \quad (4)$$

Hasil akhir dari pelatihan ELM adalah untuk mendapatkan *output weight*  $\beta$ . Nilai  $\beta$  yang tepat akan menghasilkan fungsi dasar yang optimal. ELM menggunakan *least square solution* untuk mendapatkan aproksimasi terbaik. Nilai  $\beta$  dapat diperoleh dengan mengalikan *pseudoinverse* dari matriks  $\mathbf{H}$  dengan matriks target data latih  $\mathbf{T}$  sebagaimana dituliskan pada persamaan (5)

$$\hat{\beta} = \mathbf{H}^\dagger \cdot \mathbf{T} \quad (5)$$

dimana  $\mathbf{H}^\dagger$  adalah matriks *pseudoinverse* dari  $\mathbf{H}$ .

Dengan memperoleh nilai  $\hat{\beta}$  terbaik, maka dapat dilakukan aproksimasi *output node* dengan persamaan (6)

$$\hat{\mathbf{Y}} = \mathbf{h}(\mathbf{x}) \cdot \hat{\beta} \quad (6)$$

Hasil  $\hat{\mathbf{Y}}$  masih perlu didefinisikan ke dalam label

kelas. Oleh karena itu diperlukan sebuah *decision function* terutama untuk kasus *multiclass classification*. *Decision function* tersebut dituliskan pada persamaan (7) (G.-B. Huang, Zhou, Ding, & Zhang, 2012).

$$\text{label}(x) = \arg \max_{i \in \{1, \dots, m\}} f_i(x) \quad (7)$$

### 2.3 Fast Independent Component Analysis

*Fast Independent Component Analysis* (FastICA) merupakan sebuah metode *signal preprocessing* yang digunakan untuk menyelesaikan permasalahan *blind source separation* (BSS) (Hyvärinen, Karhunen, & Oja, 2001). FastICA berupaya untuk mengestimasi *source signals* dengan hanya mengetahui *mixed signals* saja. Bentuk persamaan matematis dari BSS dituliskan pada persamaan (8).

$$\mathbf{x} = \mathbf{A}\mathbf{s} \quad (8)$$

Tujuan utama dari estimasi FastICA adalah untuk mengestimasi nilai  $\mathbf{A}$  sehingga dengan diketahuinya  $\mathbf{x}$  (*mixed signals*), nilai *source signals*  $\mathbf{s}$  dapat diestimasi. Berdasarkan persamaan (8), dengan mengetahui  $\mathbf{A}$ , maka *source signals*  $\mathbf{s}$  dapat diketahui. Hal ini dapat dihitung dengan persamaan (9).

$$\mathbf{s} = \mathbf{A}^{-1}\mathbf{x} \quad (9)$$

FastICA terdiri dari 2 tahap yaitu *preprocessing* dan de Korelasi. Tahap *preprocessing* meliputi *centering* dan *whitening*. Tahap de Korelasi adalah tahapan untuk mengestimasi nilai matriks  $\mathbf{w}$  dimana kemudian nilai  $\mathbf{w}$  yang konvergen dapat digunakan sebagai matriks  $\mathbf{A}$  pada persamaan (9).

*Centering* memusatkan data dengan mengurangi setiap elemen dari  $\mathbf{x}$  dengan reratanya sehingga  $\mathbf{x}$  memiliki *zero-mean*. *Whitening* dilakukan untuk mendapatkan  $\mathbf{x}$  dengan *unit-variance* (Hyvärinen et al., 2001). *Whitening* dapat menggunakan dekomposisi SVD yang dituliskan pada persamaan (10)

$$\mathbf{z} = (\mathbf{U}\mathbf{\Sigma}^{-1})^T \mathbf{x} \quad (10)$$

dimana  $\mathbf{U}$  adalah *unitary matrix* dengan elemen kolomnya berupa matriks *eigenvector* dari  $\mathbf{x}$ , dan Matriks  $\mathbf{\Sigma}$  sebagai matriks diagonal *eigenvalue* dari  $\mathbf{x}$ .

Hasil matriks *whitening*  $\mathbf{z}$  kemudian didekorelasi dengan FastICA. Terdapat 2 skema de Korelasi yaitu *parallel* dan *deflation*. Algoritme dari skema *parallel* yaitu:

1. lakukan *preprocessing* sehingga data menjadi *zero-mean* dan *unit-variance*

2. tentukan jumlah IC yang akan diestimasi sebagai  $m$
3. inialisasi vektor bobot awal  $\mathbf{w}_i$  secara acak dengan  $i = 1, \dots, m$
4. untuk setiap  $i = 1, \dots, m$ , hitung persamaan (11).

$$\mathbf{w}_i = E\{\mathbf{z}g(\mathbf{w}_i^T \mathbf{z})\} - E\{g'(\mathbf{w}_i^T \mathbf{z})\}\mathbf{w} \quad (11)$$

5. lakukan *symmetric orthogonalization* matriks  $\mathbf{W} = (\mathbf{w}_1, \dots, \mathbf{w}_m)^T$  dengan menggunakan persamaan (12).

$$\mathbf{W} = (\mathbf{W}\mathbf{W}^T)^{-\frac{1}{2}}\mathbf{W} \quad (12)$$

6. jika belum konvergen, ulangi dari tahap 5.

Algoritme dari skema *deflation* yaitu:

1. lakukan *preprocessing* sehingga data menjadi *zero-mean* dan *unit-variance*
2. tentukan jumlah IC yang akan diestimasi sebagai  $m$ , dan set  $p \leftarrow 1$
3. inialisasi vektor bobot awal  $\mathbf{w}_p$  secara acak
4. lakukan persamaan (13).

$$\mathbf{w}_p = E\{\mathbf{z}g(\mathbf{w}_p^T \mathbf{z})\} - E\{g'(\mathbf{w}_p^T \mathbf{z})\}\mathbf{w} \quad (13)$$

5. lakukan *orthogonalisasi* dengan persamaan (14):

$$\mathbf{w}_p = \mathbf{w}_p - \sum_{j=1}^{p-1} (\mathbf{w}_p^T \mathbf{w}_j) \mathbf{w}_j \quad (14)$$

6. normalisasikan  $\mathbf{w}_p$  dengan persamaan (15).

$$\mathbf{w}_p = \frac{\mathbf{w}_p}{\|\mathbf{w}_p\|} \quad (15)$$

7. jika  $\mathbf{w}_p$  belum konvergen, kembali ke tahap 4.
8. set  $p = p + 1$ . Jika  $p \leq m$ , kembali ke tahap 3.

Tahap terakhir dari FastICA adalah estimasi *source signals (independent component)* dengan menggunakan persamaan (16).

$$\mathbf{s} = \mathbf{W}(\mathbf{U}\mathbf{\Sigma}^{-1})^T \mathbf{x} \quad (16)$$

### 2.4 Confusion Matrix

*Confusion matrix* digunakan sebagai acuan metrik untuk mengukur performa model klasifikasi. Komponen dari *confusion matrix* dijelaskan pada Tabel 1. Metrik evaluasi yang digunakan untuk pelatihan dan pengujian adalah persamaan (17)-(19) yang diperoleh berdasarkan *confusion matrix*.



a. Akurasi

$$Accuracy = \frac{TP+TN}{Positive+Negative} \quad (17)$$

b. Skor F-1

$$F1 = \frac{2TP}{2TP+TN+FP+FN} \quad (18)$$

c. *Matthew correlation coefficient* (MCC)

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \quad (19)$$

Tabel 1 Komponen Tabel *Confusion Matrix*

		Ground Truth	
		Jantung Normal (positive)	Aritmia (negative)
Predicted condition	Jantung Normal	True Positive (TP)	False Positive (FP)
	Aritmia	False Negative (FN)	True Negative (TN)

Akurasi mengukur seberapa banyak sampel prediksi yang benar dari keseluruhan sampel. Skor F-1 merupakan nilai *harmonic mean* dari *precision* dan *recall*. *Precision* mengukur ketepatan sistem dalam mengenali suatu kelas, sedangkan *recall* mengukur seberapa sensitif sistem dalam mengenali suatu kelas. MCC adalah sebuah metrik yang mengukur performa model klasifikasi ke dalam skala [-1,1]. Apabila nilai koefisien bernilai 1, maka model klasifikasi dapat dikatakan bekerja dengan sempurna. Apabila MCC bernilai -1, maka MCC memahami sebuah pola, tetapi salah dalam mengenali label kelas. Jika MCC mendekati 0, maka model klasifikasi dapat dikatakan gagal dalam mengenali pola klasifikasi dari pembelajaran.

### 3. METODOLOGI

#### 3.1 Akuisisi Data

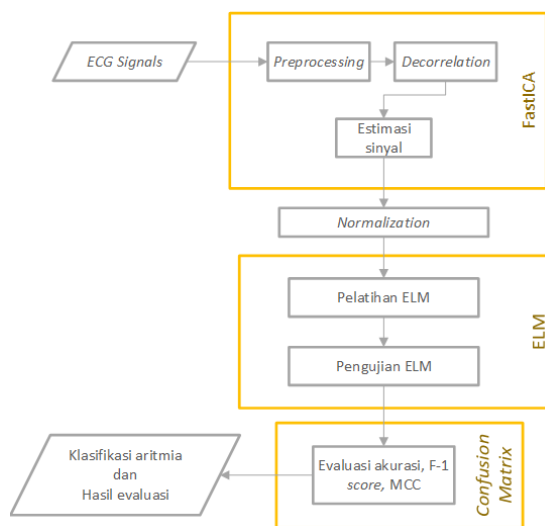
Data penelitian diperoleh dari database aritmia yang dikelola oleh Massachusetts Institute of Technology-Beth Israel Hospital (MIT- BIH) (Goldberger et al., 2000). Data merupakan cuplikan EKG selama 3 detik dari kanal MLII (limb lead II yang dimodifikasi) dan kanal V1. Sinyal dari masing-masing kanal diekstrak menjadi 2160 fitur.

Setiap data diklasifikasikan ke dalam 7 kelas yakni *normal sinus rhythm* (NSR), *atrial fibrillation* (Afib), *atrial flutter* (AFL), *ventricular tachycardia* (VT), *supraventricular tachycardia* (SVTA), *PVC bigeminy*, dan *PVC trigeminy*. Setiap label kelas memiliki jumlah sampel sebanyak 30 dari 210 total seluruh sampel.

#### 3.2 Model yang Diusulkan

Alur model yang ingin diimplementasikan ditunjukkan pada Gambar 3. Sinyal EKG diproses dengan FastICA untuk mendapatkan sinyal baru yang bebas dari *noise*. Tahapan pada FastICA terbagi menjadi 3 tahap yaitu *preprocessing*, *dekorelasi*, dan estimasi sinyal.

Sinyal hasil FastICA dinormalisasi dengan mengubah data ke dalam skala [-1,1]. Setelah dinormalisasi, data diproses oleh ELM. Dataset dipecah menjadi data latih dan data uji dengan *CV stratified shuffle split*. Data latih digunakan untuk mendapatkan bobot luaran pada tahap pelatihan. Bobot luaran digunakan pada pengujian untuk menentukan kelas data uji. Hasil dari klasifikasi kemudian dievaluasi dengan beberapa metrik. Luaran akhir adalah hasil klasifikasi keseluruhan data uji serta hasil evaluasinya.



Gambar 3 Model implementasi yang diusulkan

### 4. HASIL DAN PEMBAHASAN

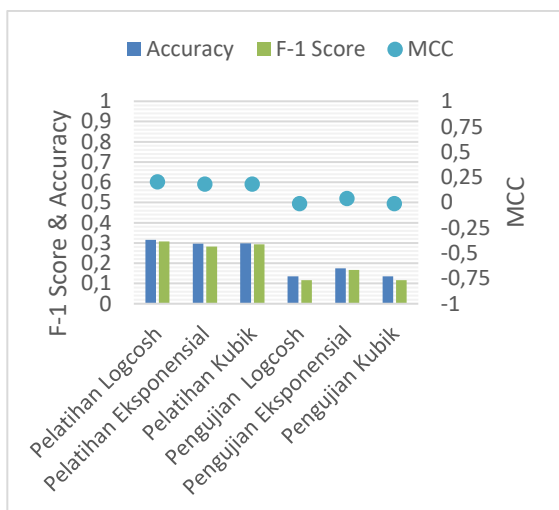
Pengujian dilakukan untuk menginvestigasi pengaruh beberapa parameter ELM dan FastICA terhadap performa model klasifikasi. Pengujian dilakukan dengan *cross-validation* (CV) *stratified shuffle split* dimana akan dibentuk 3 set hasil *stratified split* acak. *Stratified shuffle split* memastikan bahwa jumlah set uji memiliki

jumlah sampel yang sama untuk setiap kelasnya.

#### 4.1 Pengujian Fungsi Nonpolinomial FastICA

Pengujian fungsi nonpolinomial dilakukan dengan 3 skenario parameter yaitu fungsi logcos pada persamaan (21) dan (24), fungsi eksponensial pada persamaan (22) dan (25), dan fungsi kubik pada persamaan (23) dan (26). Setiap skenario pengujian dilakukan sebanyak 3 kali dengan CV *stratified shuffle split*. Parameter yang digunakan sebagai variabel kontrol adalah:

- jumlah hidden neuron = 10
- skema dekorelasi = parallel
- rasio data latih : data uji = 80 : 20



Gambar 4 Rerata skor F-1, akurasi, dan MCC pengujian fungsi nonpolinomial FastICA

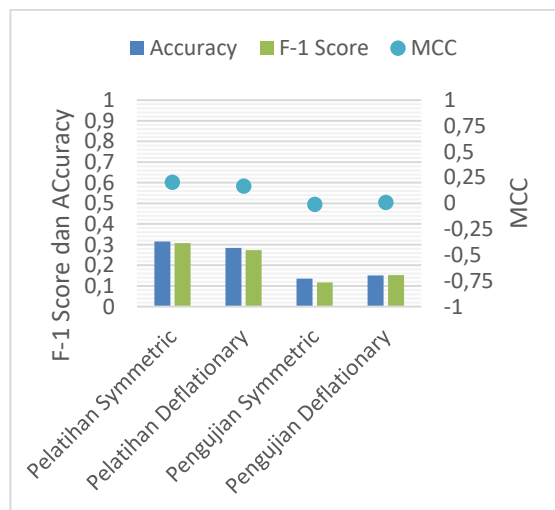
Gambar 4 adalah hasil pengujian fungsi nonpolinomial. Grafik tersebut adalah rerata dari CV *stratified shuffle split*. Dari grafik tersebut diperoleh hasil F-1 skor dan akurasi pelatihan 0,3 dan pengujian yang jauh lebih rendah dengan rerata 0,15. Selain itu, MCC juga sama-sama memiliki nilai mendekati 0. Hal ini mengindikasikan bahwa model klasifikasi tidak memahami sebuah pola; performa klasifikasinya menyerupai “*random guessing*”. Tidak ada perbedaan signifikan dari ketiga performa fungsi nonpolinomial. Dapat disimpulkan, bahwa pemilihan fungsi nonpolinomial tidak mempengaruhi performa ELM.

#### 4.2 Pengujian Skema Dekorelasi FastICA

Pengujian skema dekorelasi dibagi menjadi 2, yaitu skema *symmetric* dan skema *deflation*. Tujuan dari pengujian ini adalah mengetahui

pengaruh pemilihan skema dekorelasi pada FastICA terhadap performa klasifikasi. Parameter yang digunakan sebagai variabel kontrol adalah:

- jumlah *hidden neuron* = 10
- fungsi nonpolinomial = logcosh
- rasio data latih : data uji = 80 : 20



Gambar 5 Rerata skor F-1, akurasi, dan MCC pengujian skema dekorelasi FastICA

Gambar 5 adalah hasil pengujian skema dekorelasi. Grafik tersebut adalah rerata dari CV *stratified shuffle split*. Hasil ini tidak mengalami perubahan dari pengujian sebelumnya. Tidak ada perbedaan signifikan diantara kedua skema baik pada pelatihannya maupun pengujianya. Kedua skema sama-sama menghasilkan performa yang rendah. Dari grafik tersebut diperoleh hasil F-1 skor dan akurasi pelatihan 0,3 dan pengujian yang jauh lebih rendah dengan rerata 0,15. Selain itu, MCC juga sama-sama memiliki nilai mendekati 0. Data disimpulkan bahwa skema dekorelasi juga tidak memberikan pengaruh terhadap performa ELM.

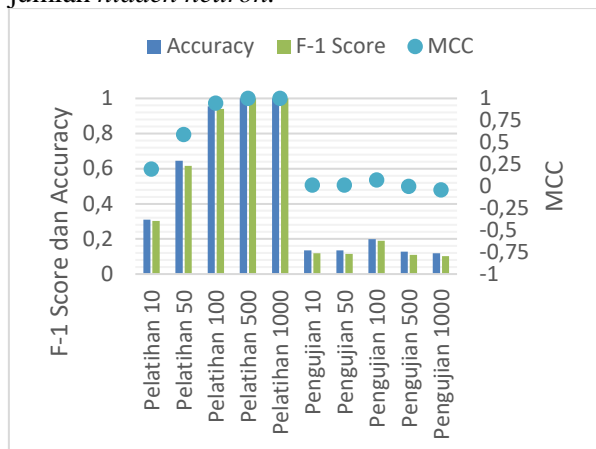
#### 4.3 Pengujian Jumlah Hidden Neuron ELM

Pengujian jumlah *hidden neuron* dilakukan dengan 5 skenario parameter, yaitu 10, 50, 100, 500 dan 1000. Pengujian dilakukan untuk mengetahui pengaruh jumlah *hidden neuron* terhadap performa klasifikasi. Parameter yang digunakan sebagai variabel kontrol adalah:

- fungsi nonpolinomial = logcosh
- skema dekorelasi = paralel
- rasio data latih : data uji = 80 : 20

Gambar 6 adalah hasil pengujian jumlah *hidden neuron*. Terdapat perubahan signifikan dari hasil pelatihan pada setiap skenario jumlah *hidden neuron*. Grafik menunjukkan adanya tren

peningkatan performa baik akurasi, skor F-1, maupun MCC seiring dengan meningkatnya jumlah *hidden neuron*.



Gambar 6 Rerata skor F-1, akurasi, dan MCC pengujian jumlah *hidden neuron* ELM

Pada jumlah *hidden neuron* 100, akurasi serta F-1 score sudah mencapai angka 0,9 dan MCC pada koefisien 0,98. Hal ini mengindikasikan bahwa performa yang sangat baik dalam pengenalan data latih mampu dicapai dengan mudah. Data menunjukkan hasil sempurna dapat dicapai dengan *hidden neuron* 500 ke atas. Namun hasil yang baik tersebut hanya berlaku pada pelatihan saja. Kenyataannya hasil pengujian tidak ada perubahan untuk setiap skenario, bahkan juga tidak ada perubahan signifikan jika dibandingkan dengan pengujian-pengujian sebelumnya. Hasil pengenalan data uji masih rendah dengan rerata skor F-1 dan akurasi tidak lebih dari 0,2 serta nilai MCC yang masih mendekati 0.

Dari hasil tersebut, performa klasifikasi akan semakin baik seiring dengan bertambahnya jumlah *hidden neuron*. Namun pada penelitian ini, hasil tersebut tidak dapat menjamin hasil pengujian yang baik. Faktanya, pelatihan yang baik tidak mempengaruhi apapun terhadap hasil pengujian. Dari fakta ini, dapat dikatakan bahwa model klasifikasi hanya mampu mengenali data dengan baik untuk data-data yang telah diajarkan pada tahap pelatihan tetapi tidak mampu mengenali data baru pada tahap pengujian dengan baik.

Dapat ditarik hipotesis jika model klasifikasi mengalami *overfitting*. *Overfitting* adalah fenomena dimana model klasifikasi “menghafal” pola dari data latih yang seharusnya adalah “mempelajari” sebuah pola yang lebih umum (Everitt & Skrondal, 2010). *Overfitting* ditandai dengan hasil pelatihan yang sangat bagus, tetapi memiliki performa yang

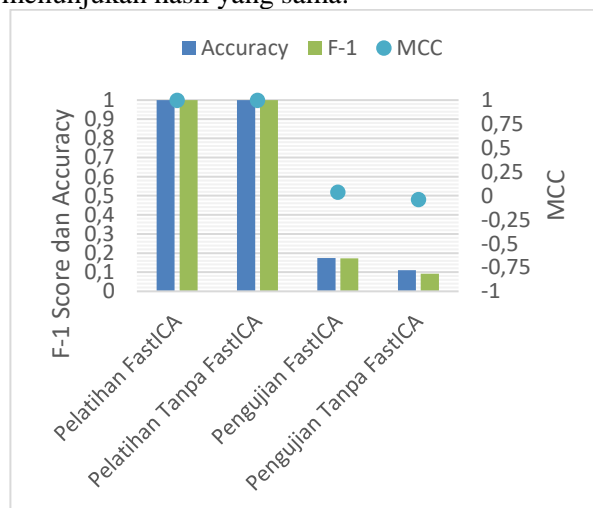
sangat buruk pada tahap pengenalan data uji. *Overfitting* dapat terjadi jika terlalu banyak parameter yang dapat dipahami oleh model dibandingkan dengan ketersediaan jumlah sampel data, sehingga model akan menghafal pola parameter pelatihan dimana seharusnya model klasifikasi memahami sebuah pola yang lebih umum untuk mengenali data uji yang belum pernah dikenali sebelumnya.

#### 4.4 Pengujian Perbandingan *Preprocessing*

Pengujian ini dilakukan untuk mengetahui perbedaan performa antara ELM dengan FastICA dan performa ELM tanpa *preprocessing* menggunakan FastICA. Parameter yang digunakan sebagai variabel kontrol adalah:

- fungsi nonpolinomial = logcosh
- skema dekorelasi = paralel
- jumlah *hidden neuron* = 500
- rasio data latih : data uji = 80 : 20

Berdasarkan hasil pada Gambar 7, model klasifikasi tidak mengalami perubahan dengan adanya *preprocessing* FastICA. Model klasifikasi masih bekerja seperti *random guessing* dengan nilai MCC mendekati 0. Skor F-1 juga tidak mengalami perubahan yang berarti jika dibandingkan dengan pengujian-pengujian sebelumnya. Hasil performa pengujian dan pelatihan antara skenario ELM dengan FastICA dan ELM tanpa FastICA menunjukkan hasil yang sama.



Gambar 7 Rerata skor F-1, akurasi, dan MCC pengujian perbandingan *preprocessing*

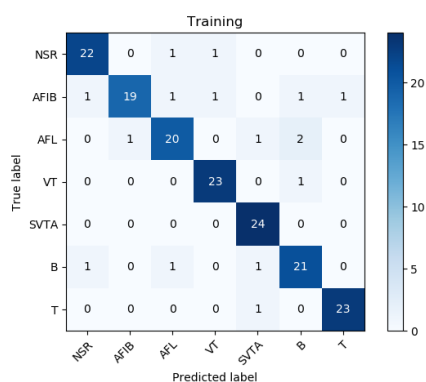
Dapat disimpulkan bahwa FastICA tidak benar-benar memberikan pengaruh pada performa ELM. Disisi lain, hal ini juga membuktikan bahwa hasil buruk yang dikerjakan oleh model klasifikasi pada pengujian

bukan dikarenakan adanya FastICA. Bukti bahwa model klasifikasi mengalami *overfitting* semakin kuat dengan fakta bahwa akurasi pelatihan yang tinggi dan hasil pengujian yang tetap buruk terjadi dengan ataupun tanpa adanya FastICA.

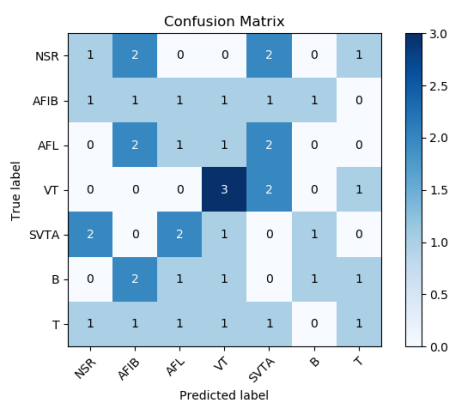
#### 4.5 Analisis hasil *overfitting*

##### 4.5.1 Analisis Confusion Matrix

Pada beberapa hasil pengujian, *overfitting* dapat dilihat dari hasil *confusion matrix*-nya. Contoh *overfitting* terjadi pada pengujian jumlah *hidden neuron* dengan skenario 100 *split-1*. Gambar 8 merupakan *confusion matrix* pelatihannya dan Gambar 9 merupakan *confusion matrix* pengujianya.



Gambar 8 Confusion matrix pelatihan 100 *hidden neuron split-1*



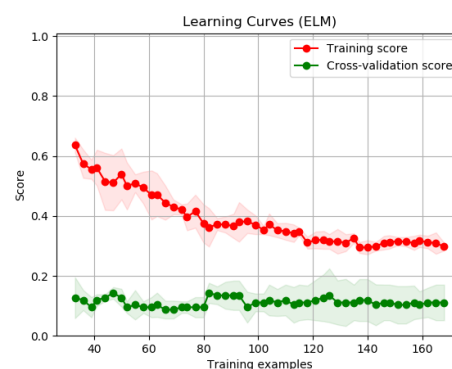
Gambar 9 Confusion matrix pengujian 100 *hidden neuron split-1*

Pada Gambar 8, model klasifikasi memiliki sensitivitas yang baik dalam mengenali setiap label kelas, serta sangat presisi dalam membedakan kelas satu dengan yang lain. Hal ini ditunjukkan dengan hasil *confusion matrix* terfokus pada diagonalnya. Semakin baik performa model klasifikasi, maka semakin

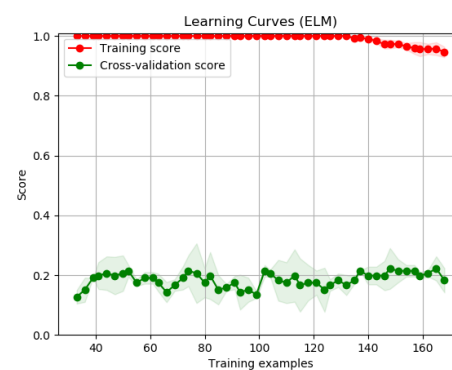
banyak nilai diagonal *confusion matrix*-nya. Disisi lain, hasil *confusion matrix* pengujianya pada Gambar 9 menunjukkan suatu pola acak dalam klasifikasi label kelas. model klasifikasi tidak dapat menentukan suatu data ke dalam labelnya dengan baik.

##### 4.5.2 Analisis Learning Curve

*Overfitting* dapat terjadi karena parameter data terlalu kompleks, dan jumlah data latih yang terlalu sedikit. Keduanya dapat mengakibatkan ketidakmampuan model klasifikasi memahami sebuah pola dari pelatihan untuk diterapkan pada data uji. *Overfitting* tampak ketika model klasifikasi mendapatkan hasil yang sangat baik dalam pelatihan, tetapi gagal dalam mengenali data baru yang tidak digunakan pada pelatihan.



Gambar 10 Learning curve pengujian 10 *hidden neuron*



Gambar 11 Learning curve pengujian 100 *hidden neuron*

Analisis ini dilakukan untuk menginvestigasi penyebab terjadinya *overfitting* dan pendekatan solusi saran yang dapat dilakukan untuk mengatasinya dengan melihat hasil *learning curve*-nya. Jika hasil validasi dan akurasi pelatihan konvergen pada suatu titik, maka dapat dikatakan *overfitting* terjadi karena kurangnya data latih.

Gambar 10 adalah *learning curve* dari

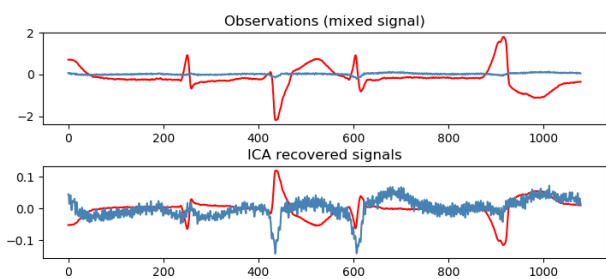


pengujian jumlah *hidden neuron* 10. Gambar 11 adalah *learning curve* dari pengujian jumlah *hidden neuron* 100. Terdapat peningkatan signifikan pada hasil akurasi pelatihannya. Namun pada hasil validasinya tidak ada perbedaan signifikan kecuali pada 100 hidden neuron, hasil validasinya memiliki standar deviasi yang lebih besar.

Berdasarkan kedua gambar tersebut, penambahan data latih tidak meningkatkan tren hasil validasi secara signifikan. Seharusnya semakin banyak data latih, semakin baik validasinya. Meskipun model klasifikasi dapat menghasilkan akurasi pelatihan sempurna, tetapi model klasifikasi masih terlalu sensitif terhadap data sinyal baru (pembelajaran yang diperoleh belum bersifat umum). Dapat disimpulkan bahwa kurangnya data latih bukanlah penyebab dari *overfitting*.

#### 4.5.3 Analisis Hasil Ekstraksi Fitur

Pada hasil pengujian perbandingan *preprocessing*, baik skenario dengan FastICA maupun tanpa FastICA memiliki hasil pelatihan yang bagus, serta pengujian yang sama-sama buruk. Hal ini dapat diartikan bahwa transformasi yang dilakukan FastICA tidak memberikan dampak apapun terhadap ELM. Jika dilihat pada contoh sampel sinyal *ventricular tachycardia* dan hasil transformasi FastICA-nya pada Gambar 12, hasil ekstraksinya benar-benar berbeda.



Gambar 12 Hasil komputasi FastICA untuk sampel vt200-2.csv

Hasil ekstraksi yang berbeda tersebut ternyata tidak membuat perbedaan sedikitpun antara ELM dengan FastICA maupun tanpa FastICA. Hal ini mengindikasikan bahwa *overfitting* yang terjadi pada model klasifikasi dikarenakan representasi fiturnya yang kurang baik. Dengan atau tanpa FastICA ternyata tidak berpengaruh terhadap ELM.

## 5. PENUTUP

Skema dekorrelasi, pemilihan fungsi

nonpolinomial, dan *preprocessing* dengan FastICA tidak mempengaruhi performa klasifikasi ELM secara signifikan. Tidak ada perubahan signifikan pada tren hasil pengukuran baik pada tahap pelatihan ataupun pengujian pada data baru.

Jumlah *hidden neuron* berpengaruh terhadap performa klasifikasi ELM secara keseluruhan. Semakin besar jumlah *hidden neuron*, semakin baik hasil performa pelatihan klasifikasi. Dalam penelitian ini, tidak dapat dipastikan pengaruhnya terhadap klasifikasi pengujian ELM terhadap data baru. Hal ini dikarenakan terjadinya *overfitting* pada model klasifikasi.

Permasalahan utama pada hasil penelitian ini adalah terjadinya *overfitting* pada model klasifikasi. *Overfitting* terjadi karena representasi fitur terlalu kompleks dan tidak sebanding dengan jumlah data yang ada. Hal ini mengakibatkan performa yang buruk pada pengujian ELM-FastICA untuk data yang belum dikenali.

## 6. DAFTAR PUSTAKA

- Clevert, D.-A., Unterthiner, T., & Hochreiter, S. (2015). Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs), 1–14. <https://doi.org/10.3233/978-1-61499-672-9-1760>
- Everitt, B. S., & Skrondal, A. (2010). *The Cambridge Dictionary of Statistic* (4th ed.). New York: Cambridge University Press.
- Hart, R. G., Diener, H. C., Coutts, S. B., Easton, J. D., Granger, C. B., O'Donnell, M. J., ... Connolly, S. J. (2014). Embolic strokes of undetermined source: The case for a new clinical construct. *The Lancet Neurology*, 13(4), 429–438. [https://doi.org/10.1016/S1474-4422\(13\)70310-7](https://doi.org/10.1016/S1474-4422(13)70310-7)
- Huang, G. Bin, Zhu, Q. Y., & Siew, C. K. (2006). Extreme learning machine: Theory and applications. *Neurocomputing*, 70(1–3), 489–501. <https://doi.org/10.1016/j.neucom.2005.12.126>
- Huang, G.-B., Zhou, H., Ding, X., & Zhang, R. (2012). Extreme learning machine for regression and multiclass classification. *IEEE Transactions on Systems, Man, and Cybernetics. Part B, Cybernetics*, 42(2),

- 513–529.  
<https://doi.org/10.1109/TSMCB.2011.2168604>
- Hyvärinen, A., Karhunen, J., & Oja, E. (2001). *Independent Component Analysis*. New York: John Wiley & Sons, Inc.
- Kroupi, E., Yazdani, A., Vesin, J. M., & Ebrahimi, T. (2011). Ocular artifact removal from EEG: A comparison of subspace projection and adaptive filtering methods. *European Signal Processing Conference*, (January), 1395–1399.
- Melgani, F., & Bazi, Y. (2008). Classification of electrocardiogram signals with support vector machines and particle swarm optimization. *IEEE Transactions on Information Technology in Biomedicine*, 12(5), 667–677.  
<https://doi.org/10.1109/TITB.2008.923147>
- Moore, P. (1972). The Moore-Penrose Pseudoinverse, *I*(1), 1–5.
- Pedersen, K. B., Chemnitz, A., Madsen, C., Sandgaard, N. C. F., Bak, S., & Brandes, A. (2016). Low Incidence of Atrial Fibrillation in Patients with Transient Ischemic Attack. *Cerebrovascular Diseases Extra*, 6(3), 140–149.  
<https://doi.org/10.1159/000451035>
- Prathivi, N. (2015). Detecting and dealing with strokes - Lifestyle - The Jakarta Post. Retrieved October 9, 2017, from <http://www.thejakartapost.com/news/2015/07/01/detecting-and-dealing-with-strokes.html>
- Serre, D. (2002). *Matrices: Theory and applications*. Springer (Vol. 216).  
<https://doi.org/10.1007/978-1-4419-7683-3>
- Sposato, L. A., Cipriano, L. E., Saposnik, G., Vargas, E. R., Riccio, P. M., & Hachinski, V. (2015). Diagnosis of atrial fibrillation after stroke and transient ischaemic attack: A systematic review and meta-analysis. *The Lancet Neurology*, 14(4), 377–387. [https://doi.org/10.1016/S1474-4422\(15\)70027-X](https://doi.org/10.1016/S1474-4422(15)70027-X)
- Stroke Association. (2016). Transient Ischemic Attack (TIA). Retrieved October 9, 2017, from <http://www.strokeassociation.org/STRO>
- KEORG/AboutStroke/TypesofStroke/TIA/Transient-Ischemic-Attack-TIA\_UCM\_492003\_SubHomePage.jsp
- Thaler, M. S. (2015). *The Only EKG Book You'll Ever Need* (8th Editio). Lippincott Williams & Wilkins. Retrieved from <http://content.wkhealth.com/linkback/openurl?sid=WKPTLP:landingpage&an=00006534-198911000-00025>
- Zekić-Sušac, M., Pfeifer, S., & Šarlija, N. (2014). A Comparison of Machine Learning Methods in a High-Dimensional Classification Problem. *Business Systems Research Journal*, 5(3), 82–96.  
<https://doi.org/10.2478/bsrj-2014-0021>