

Klasifikasi Keadaan Mata Berdasarkan sinyal EEG menggunakan *Extreme Learning Machines*

Ersa Christian Prakoso ^{#1}, Untari Novia Wisesty ^{*2}, Jondri ^{#3}

School of Computing, Telkom University

Jl. Telekomunikasi No. 01, Terusan Buah Batu, Bandung, Jawa Barat, Indonesia

¹ersachristian@students.telkomuniversity.ac.id

²untari.n.wisesty@gmail.com

Abstract

Electroencephalography signal or EEG signal is one of biosignals that currently being hot topic for research. EEG signal has many benefits including detection of epilepsy from patients, sleep disorders, or an input for a computer application. A kind of input that can be used based on EEG signal is eye state. However, An Classification with a suffice performance is needed for that purpose. Therefore, writer will do a research which will implement a new learning method for Artificial Neural Network called Extreme Learning Machine for classifying eye state based on EEG signal. Dataset that will be used for training and testing were donated by Oliver Roesler and combined with dataset from Repository University of California, Irvinel (UCI). There are 7 corpora which consist of EEG records of 4 differents people. Based on system testing, we can draw a conclusion that ELM classifier can be used to classify eye-state based on ELM with good performance. Tested system can achieve 97,95% accuracy and only need 0,81 second of training if the corpus is used separately. However, if all the corpora are used together as one big corpus, ELM classifier can only achieved 78,94% accuracy with 5,71 seconds training time.

Keywords : Classification, *Electroencephalography* (EEG), *Extreme Learning Machine* (ELM).

Abstrak

Electroencephalography atau sinyal EEG adalah salah satu biosignal yang marak menjadi topik penelitian saat ini. Sinyal EEG memiliki banyak manfaat seperti pendeteksian epilepsi, gangguan tidur, atau input dalam aplikasi komputer. Salah satu input yang dapat dideteksi berdasarkan sinyal EEG adalah keadaan mata. Namun untuk digunakan sebagai input dalam aplikasi diperlukan klasifikasi dengan performansi yang memadai. Oleh karena itu penulis akan dilakukan penelitian dimana salah satu metode pembelajaran Jaringan Syaraf Tiruan yaitu *Extreme Learning Machine* (ELM) akan diimplementasikan untuk mengklasifikasikan kondisi mata berdasarkan sinyal EEG. Dataset yang digunakan untuk melatih dan menguji model adalah dataset *eye-state* yang didonasikan oleh Oliver Roesler digabung dengan dataset yang berasal dari website *repository Universitas of California, Irvinel* (UCI) . Terdapat 7 *corpus* yang terdiri dari perekaman EEG yang dilakukan kepada 4 orang berbeda, lalu ditambahkan 1 *corpus* yang merupakan penggabungan seluruh *corpus* lain. Dari hasil pengujian yang dilakukan disimpulkan bahwa ELM dapat digunakan untuk klasifikasi keadaan mata dengan akurasi mencapai 97,95% dengan waktu latih hanya 0,81 detik jika masing-masing data digunakan secara terpisah, sedangkan penggabungan keseluruhan dataset hanya mencapai akurasi 78,94% dengan waktu latih 5,71 detik.

Kata Kunci : Klasifikasi, *Electroencephalography* (EEG), *Extreme Learning Machine* (ELM).

I. PENDAHULUAN

Akhir-akhir ini riset tentang bagaimana cara otak manusia bekerja semakin marak dan menjadi topik hangat untuk diperbincangkan dalam biomedik. Umumnya penelitian dilakukan dengan menganalisa sinyal yang dikeluarkan dari tubuh manusia, atau yang disebut biosignal. Berbagai metode dan alat diciptakan untuk menangkap sinyal yang ada dari tubuh manusia, antara lain adalah EEG, EOG, ECG, EMG, dan sebagainya.

Salah satu biosignal yang paling umum digunakan adalah EEG atau *Electroencephalography*. EEG paling umum digunakan sebagai metode untuk mendeteksi epilepsi. Namun dalam perkembangannya, EEG juga digunakan untuk mendeteksi berbagai hal seperti gangguan tidur, *brain death*, serta sebagai input dalam aplikasi komputer. Untuk digunakan sebagai *brain-computer interface*, selain diperlukan sistem klasifikasi dengan akurasi yang memadai, sistem harus dapat mengeluarkan hasil dengan cepat agar tidak *delay* antara *input* dengan respon yang akan dilakukan komputer terutama jika digunakan untuk sistem *real-time*.

Salah satu riset yang telah dilakukan adalah bagaimana perubahan EEG terhadap pergerakan mata manusia. Kesimpulan yang didapat adalah perbedaan terbesar dari kondisi saat mata tertutup dan terbuka adalah kekuatan dari kedua kondisi, kekuatan di mata saat mata tertutup adalah lebih besar daripada saat mata terbuka [1]. Hasil dari riset tersebut dapat digunakan untuk fitur dalam klasifikasi 2 status yang ada. Namun tidak banyak riset lanjutan yang mencoba menciptakan sistem klasifikasi untuk *eye-state* [2].

Salah satu riset yang telah dilakukan mencoba menggunakan berbagai metode yang ada untuk mengklasifikasikan kedua kondisi tersebut berdasarkan sinyal EEG yang didapat [2]. *K-star* dapat menghasilkan akurasi yang tinggi pada kasus ini yaitu hingga mencapai 97% tanpa perlu dilakukan *pre-processing* pada data EEG. Namun terdapat kelemahan yang cukup fatal dari *K-star* yaitu sangat lambat untuk mengeluarkan *output*, seperti halnya *instance-based learner* lain. Pada data ini diperlukan waktu hingga 38 menit untuk mendapatkan label hasil pengujian *K-star* dari keseluruhan data yang ada, sangat tidak bagus untuk digunakan dalam *brain-computer interface* yang bersifat *real-time*.

Jaringan Syaraf Tiruan (JST) adalah salah satu metode yang umum digunakan untuk kasus regresi maupun klasifikasi. Namun perkembangannya mengalami kemacetan. Pada tahun 2006, muncul konsep yang secara filosofis meniru cara belajar dari otak manusia, yaitu *Extreme Learning Machine* (ELM) [3]. Berbeda dengan JST dan *Deep Learning*, ELM menganggap neuron dari *hidden layer* tidak perlu di-*tuning*. Selain itu karena tidak perlu melakukan iterasi dalam data seperti *Backpropagation*, ELM dapat memangkas waktu training menjadi jauh lebih cepat [3]. *Deep Learning* sendiri telah diuji untuk kasus klasifikasi emosi berdasarkan sinyal EEG dan didapatkan performansi yang cukup baik hingga mencapai lebih dari 80% [4]. Riset lain telah dilakukan dan terbukti bahwa dalam kasus klasifikasi emosi menggunakan EEG, salah satu variasi ELM, yaitu GELM mampu menyaingi performansi dari *Support Vector Machine* dengan waktu training jauh lebih singkat. Selain itu karena modelnya hanya berupa perkalian matriks JST umumnya dapat mengeluarkan hasil yang cepat dari *input* yang ada.

Dalam penelitian kali ini, penulis ingin mengimplementasikan konsep *Extreme Learning Machine* untuk pengklasifikasian *eye-state* berbasis EEG. Penulis ingin membuktikan bahwa dengan waktu training yang jauh lebih cepat, ELM dapat memberikan performansi yang baik.

II. TINJAUAN PUSTAKA

A) *Electroencephalography*

Selama otak bekerja, neuron dalam otak akan menghasilkan suatu aktivitas elektrik tertentu. *Electroencephalography* adalah suatu kegiatan untuk merekam aktivitas elektrik spontan dari otak dalam periode tertentu [5]. Perekaman dilakukan menggunakan alat yang dinamakan *Electroencephalogram* (EEG). Penggunaan alat EEG dilakukan dengan menempelkan elektrode EEG ke bagian-bagian tertentu dari kepala.

Alat yang digunakan dalam perekaman adalah *Emotiv Epc Headset* yang dikembangkan oleh perusahaan EMOTIV dan dijual bebas untuk umum. Alat berikut memiliki daya sampling sebanyak 128 Hz yang berarti dalam 1 detik, headset dapat merekam hingga 128 nilai dari masing-masing *channel*. Rentang nilai yang dihasilkan dari tiap *channel* berkisar antara 0 hingga 9600. Seperangkat *Emotiv Epc Headset* dijual antara

US\$400 hingga US\$900 tergantung kelengkapannya. *Emotiv Epoc Headset* memiliki 14 channel yang penempatannya dapat dilihat pada gambar berikut:

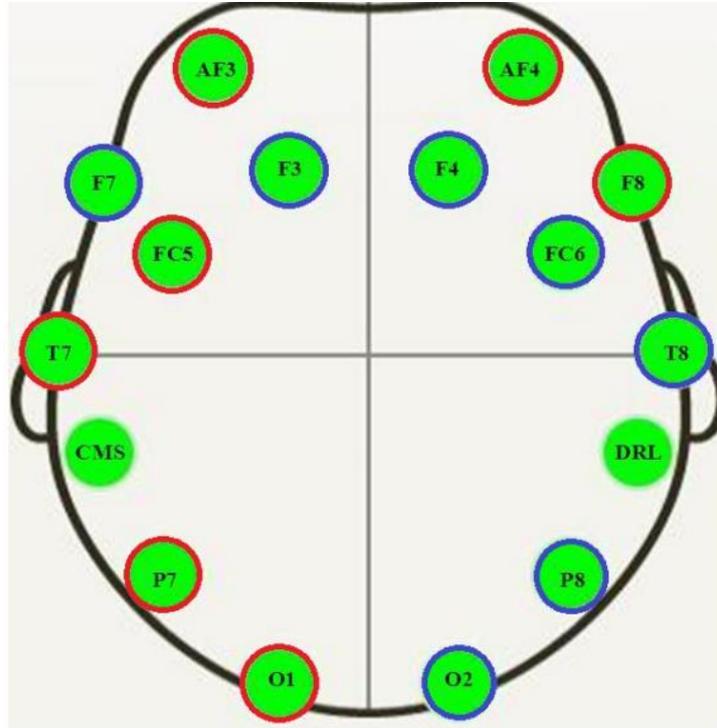


Fig. 1 Penempatan Elektrode pada dataset *Eye-state* [2]

Penelitian menunjukkan bahwa perbedaan terbesar dari keadaan saat mata tertutup dan terbuka adalah *power*. *Power* saat keadaan mata tertutup adalah lebih besar daripada saat mata terbuka [1]. Hal ini terlihat pada dataset *eye-state* yang dikeluarkan oleh Oliver Roesler [2]. Menurut dataset nilai *corpus* tersebut, amplitudo dari sensor F7, F3, O2, P8, T8, FC6, dan F4 menunjukkan nilai maksimum saat mata terbuka lebih besar daripada saat mata tertutup dan nilai minimumnya cenderung sama. Namun untuk sensor AF3, FC5, T7, P7, O1, F8, dan AF4 nilai minimum saat mata terbuka lebih besar daripada saat mata tertutup dan nilai maksimumnya cenderung tetap.

Ting Wang pada salah satu risetnya [10] melakukan ekstraksi fitur dari seluruh nilai secara temporal dan didapatkan *first-order feature* dan *second-order feature* untuk dicari rata-rata dan standar deviasinya sebagai fitur baru, dan terbukti dapat membantu akurasi dari sistem klasifikasi yang digunakan.

B) Artificial Neural Network

Artificial Neural Network atau Jaringan Syaraf Tiruan (JST) merupakan salah satu upaya manusia dalam memodelkan cara berpikir manusia kedalam komputer. JST dapat diartikan sebagai proses tersebar paralel yang sangat besar yang memiliki kecenderungan untuk menyimpan pengetahuan yang bersifat pengalaman dan membuatnya siap digunakan [6]. Dalam JST pengetahuan yang didapat dipresentasikan dalam bobot koneksi antar sel syaraf yang ada (*neuron*). Dalam setiap *neuron* terjadi perhitungan sebagai berikut:

$$y(x) = f(s(x)) \quad (1)$$

Dimana $s(x)$ adalah kombinasi linear dari *input* x , dengan bobot w dan bias b .

$$s(x) = \sum w_i x_i + b \quad (2)$$

Secara umum ANN terdiri atas 3 bagian *layer*, yaitu *input layer*, *hidden layer*, *output layer*. Jumlah dari *input layer*, dan *output layer* adalah 1, sedangkan *hidden layer* dapat terdiri dari 1 layer ataupun lebih tergantung dari arsitektur yang digunakan [5].

Salah satu jenis ANN yang paling simpel dan umum digunakan adalah *feedforward neural network*. Pada arsitektur ini, seluruh koneksi antar *neuron* terhubung ke *neuron* di *layer* berikutnya tanpa ada koneksi yang kembali ke *layer* sebelumnya.

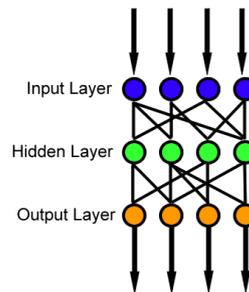


Fig. 2 Feedforward Neural Network [5]

Metode belajar paling umum yang digunakan untuk *feedforward neural network* adalah *Backpropagation*.

C) Extreme Learning Machines (ELM)

Extreme Learning Machines atau biasa disingkat ELM adalah algoritma *learning* yang awalnya dirancang untuk *Single Layer Feedforward Neural Network (SLFN)* untuk menggantikan *Backpropagation*. ELM menawarkan proses belajar yang ribuan kali lebih cepat dibanding *Backpropagation* namun dengan performansi generalisasi yang lebih baik [3].

Berbeda dengan *Backpropagation* yang melakukan iterasi dan mengubah nilai bobot koneksi antar seluruh neuron untuk meminilasi *error*, ELM membuktikan bahwa bobot dari *input layer* dan bias dari *hidden layer* dapat diberi nilai acak jika fungsi aktivasi di *hidden layer* terdiferensiasi.

Beberapa masalah dalam algoritma belajar BP yang dianggap perlu diperbaiki oleh ELM adalah [3]:

- 1) Parameter *Learning rate* dalam BP berpengaruh besar terhadap performansi sistem. Jika terlalu kecil, maka proses belajar akan berjalan sangat lambat, sedangkan jika terlalu besar, algoritma menjadi tidak stabil dan sulit mencapai konvergen.
- 2) BP cenderung terjebak dalam lokal minima.
- 3) Perlu metode khusus untuk mencegah *over-fit*.
- 4) *Gradient-based Learning* seperti BP sangat menghabiskan waktu untuk kebanyakan aplikasi.

Sebuah ELM dengan *Hidden node* dan fungsi aktivasi *g* dapat dimodelkan sebagai berikut:

$$\beta^T H = T, \quad (3)$$

$$\text{Dimana } H = \begin{bmatrix} g(w_1 \cdot x_1 + b_1) & \dots & g(w_1 \cdot x_N + b_1) \\ \dots & \dots & \dots \\ g(w_K \cdot x_1 + b_K) & \dots & g(w_K \cdot x_N + b_K) \end{bmatrix}_{K \times N} \quad (4)$$

$$\beta = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_k^T \end{bmatrix}_{K \times m}, T = [t_1, \dots, t_n]_{m \times N} \tag{5}$$

$w_j = (w_{j1}, w_{j2}, \dots, w_{jd})$ adalah *weight* input dari vektor yang menghubungkan *hidden node* ke- j dengan node input, dan $\beta_j = (\beta_{j1}, \beta_{j2}, \dots, \beta_{jm})^T$ adalah *weight* dari vektor yang menghubungkan *hidden node* ke- j dengan node output. B_j adalah bias dari node ke- j .

Guang-Bin Huang [3] mengungkapkan teori bahwa untuk *Single Layer Feed Forward Network* dengan fungsi aktivasi $g : \mathbb{R} \rightarrow \mathbb{R}$ dimana fungsi tersebut *infinitely differentiable*, untuk sampel (x_i, t_i) , dimana $x_i \in \mathbb{R}^n, t_i \in \mathbb{R}^m$ yang berbeda sejumlah N , untuk segala w_i dan b_i dipilih secara acak dalam interval \mathbb{R}^n dan \mathbb{R}^m maka dengan probabilitas 1, terdapat *output* dari matriks *hidden layer* H untuk SLFN dimana $\|H\beta - T\| = 0$.

Dari teori diatas maka terbukti bahwa bobot maupun bias dari *hidden layer* tidak perlu di-*tune* namun dapat dibangkitkan secara acak. Untuk nilai bobot dan bias tertentu maka untuk melatih SLFN secara sederhana dapat digambarkan sebagai mencari solusi *least-squares* dari sistem linear $H\beta = T$.

Jika jumlah *hidden node* sama dengan jumlah sampel yang berbeda dari data latih, maka matriks H adalah persegi dan dapat di-*inverse*, dan SLFN dapat memperkirakan sampel latih tersebut dengan *error* 0. Namun, untuk kebanyakan kasus, jumlah *hidden node* jauh lebih kecil daripada jumlah sampel data latih berbeda yang digunakan, maka H adalah matriks non-persegi dan solusi dengan *norm* terkecil dari sistem linear tersebut bisa didapat dari persamaan:

$$\beta = H^+ T \tag{6}$$

H^+ adalah *Moore-Penrose generalized inverse* dari matriks H .

Secara sederhana terdapat 3 langkah dalam proses latih ELM [3]:

Andaikan terdapat sebuah *training set* $\mathcal{S} = \{(x_i, t_i) | x_i \in \mathbb{R}^n, t_i \in \mathbb{R}^m, i = 1, \dots, n\}$, fungsi aktivasi $g(x)$ dan *hidden node* berjumlah \tilde{N} ,

1. Tetapkan nilai bobot *input* w_i dan *bias* $b_i, i = 1, \dots, \tilde{N}$.
2. Hitung matriks H *output* dari *hidden layer*.
3. Hitung bobot dari *output* β

$$\beta = H^+ T \tag{7}$$

dimana $T = [t_1, \dots, t_N]^T$

Orthogonal Projection dapat digunakan untuk menghitung *Moore-Penrose*, $H^+ = (H^T H)^{-1} H^T$ jika $H^T H$ *nonsingular*, atau $H^+ = H^T (H H^T)^{-1}$ jika $H H^T$ *nonsingular*. Menurut *ridge regression theory*, disarankan untuk menambahkan bilangan positif ke diagonal $H^T H$ atau $H H^T$ pada penghitungan bobot β sehingga model menjadi lebih stabil dan memiliki performansi generalisasi yang lebih baik [8, 14]. Sehingga penghitungan bobot β dapat ditulis sebagai berikut:

$$\beta = H^T \left(\frac{I}{C} + H H^T \right)^{-1} T \tag{8}$$

Atau,

$$\beta = \left(\frac{I}{C} + H^T H \right)^{-1} H^T T \tag{9}$$

dimana I adalah matriks identitas sebesar matriks persegi HH^T ataupun H^TH , dan C adalah parameter yang ditentukan oleh *user*. Untuk sistem dengan jumlah *hidden neuron* lebih sedikit dari jumlah baris data latih yang digunakan disarankan menggunakan persamaan (9) sedangkan untuk kasus sebaliknya disarankan menggunakan persamaan (8) [8, 14].

D) Moore-Penrose generalized inverse

Solusi dari sistem linear $Ax = y$, dimana A mungkin *singular* atau tidak persegi, dapat dibuat sederhana dengan menerapkan *Moore-Penrose generalized inverse* [3]. Matriks G dengan ordo $n \times m$ adalah *Moore-Penrose generalized inverse* untuk matriks A berordo $n \times m$ jika:

$$AGA = A, GAG = G, (AG)^T = AG, (GA)^T = GA. \quad (10)$$

Solusi *Moore-Penrose generalized inverse* bisa didapatkan menggunakan berbagai metode seperti *Single Value Decomposition* (SVD), dan *Orthogonal Projection*

E) Penulisan Terkait

Oliver Roesler [2] adalah yang pertama mengangkat kasus klasifikasi keadaan mata sekaligus membuat data yang digunakan penulis. Dataset EEG terdiri atas 15 atribut dimana 14 atribut adalah nilai dari masing-masing elektrode dan 1 atribut keadaan mata. Data disimpan secara terurut untuk memungkinkan analisa dependensi temporal. Menurut data berikut, amplitudo dari sensor F7, F3, O2, P8, T8, FC6, dan F4 menunjukkan nilai maksimum saat mata terbuka lebih besar daripada saat mata tertutup dan nilai minimumnya cenderung sama. Namun untuk sensor AF3, FC5, T7, P7, O1, F8, dan AF4 nilai minimum saat mata terbuka lebih besar daripada saat mata tertutup dan nilai maksimumnya cenderung tetap. Untuk eksperimen klasifikasi, digunakan *weka toolkit* dengan parameter standar. *Classifier* umum seperti *naive bayes*, ANN, SMO, dan *logistic regression* menghasilkan performansi yang buruk dengan *error* lebih dari 30%. Algoritma *Decision Tree* menghasilkan performansi lebih baik hingga sekitar 15%. Performansi terbaik didapat *instance-based learner* seperti KStar dengan *error* hanya mencapai 3,2%. Kelemahan dari *instance-based learner* ini adalah waktu *testing* yang lama. Untuk melakukan *testing* pada keseluruhan corpus pada paper terkait dibutuhkan waktu kurang lebih 38 menit sehingga perlu menjadi pertimbangan jika digunakan untuk *real-time system*.

Oliver Roesler melakukan riset lanjutan yang menguji pengaruh dari alat rekam EEG terhadap performansi sistem klasifikasi keadaan mata [9]. Terdapat 3 alat yang digunakan untuk perekaman EEG yaitu *MindWave*, *Emotiv EPOC*, dan *BrainAmp Standart*. Perbedaan terbesar dari alat-alat tersebut adalah jumlah *channel* dan *Sampling rate* masing-masing alat. *MindWave* memiliki hanya 1 *channel* namun dengan *sampling rate* cukup tinggi yaitu 512Hz, dan dijual umum dengan harga 80\$. *Emotiv EPOC* adalah alat rekam lain selain *MindWave* yang dijual umum, namun dengan harga yang lebih tinggi yaitu sekitar 700\$. *Emotiv EPOC* memiliki 14 *channel* dengan *sampling rate* 128Hz. Alat terakhir yaitu *BrainAmp Standart* tidak dijual untuk umum, karena harganya yang cukup mahal hingga mencapai 60000€. *BrainAmp Standart* memiliki 32 *channel* dengan *sampling rate* mencapai 1kHz. Dilakukan pengujian pengaruh alat masing-masing terhadap klasifikasi keadaan mata, namun khusus *BrainAmp* hanya digunakan 12 *channel* saja untuk menguji apakah pada jumlah *channel* yang sama, hasil yang didapat akan sama dengan alat lain atau tidak. Hasilnya, *BrainAmp* memiliki *error rate* yang paling tinggi. Hal ini tidak dikarenakan jumlah *channel* yang hanya 1, karena telah diuji klasifikasi menggunakan *Emotiv EPOC* dengan hanya 1 *channel* namun dapat menghasilkan akurasi yang jauh lebih baik, sedangkan *Error rate Emotiv EPOC* sendiri dengan menggunakan 14 *channel* hanya sedikit tertinggal dibawah *BrainAmp Standart* yang menggunakan 12 *channel* yaitu dari 11,1% menjadi 16,5%. Perlu diingat bahwa harga *Emotiv EPOC* berada jauh dibawah alat profesional *BrainAmp Standart* namun dengan performansi yang tidak terpaut jauh, sehingga *Emotiv EPOC* sangat cocok untuk digunakan sebagai *Brain-Computer Interface* yang dapat digunakan secara umum, hanya perlu dikembangkan sistem yang dapat menerjemahkan input *Emotiv EPOC*

Guang-Bin Huang [7] melakukan perbandingan antara *Extreme Learning Machine* (ELM) dengan *Support Vector Machine* (SVM) dan *Least Squares- Support Vector Machine* (LS-SVM). Masing-masing metode diuji dengan berbagai dataset untuk kasus *binary classification*, *multi-class classification*, dan *Regression*. Pada sebagian besar kasus ELM dapat menyaingi performansi SVM maupun LS-SVM dengan waktu *training* yang

lebih singkat. Contohnya untuk dataset *Wine*, ELM mampu mencapai akurasi 98.57 % dengan waktu latih 0.0206 detik dibanding SVM yang mencapai akurasi 98.37 dengan waktu latih 0.075 detik dan LS-SVM yang hanya mencapai akurasi 97.63 dengan waktu latih 1.82 detik. Contoh lain adalah pada dataset *Colon*, ELM mencapai akurasi 90.63% dibanding SVM dan LS-SVM dengan akurasi 84.38% dan 87.5%, dengan masing-masing waktu training 0.001 detik, 0.0462 detik, dan 0.0469 secara berurutan. Melihat waktu latih yang sangat singkat namun dengan akurasi yang lebih baik dari SVM, penulis melihat ELM sebagai alternatif yang dapat dicoba untuk *classifier* sinyal EEG.

Ting Wang [10] menggunakan dataset yang sama dengan yang digunakan pada paper 1. Metode yang digunakan sebagai *classifier* adalah *Neural Network* dengan menerapkan *Increasing Input Dimension (ITID)* dimana input dari learning akan bertambah seiring berjalannya proses *learning*. Pengujian dilakukan dengan 4 skenario. Skenario pertama adalah penggunaan *temporal feature extraction* untuk mendapatkan *first-order feature* dan *second-order feature*, penghitungan standar deviasi dan rata-rata sebagai fitur baru, lalu dilakukan pengurutan input untuk ITID. Pengurutan fitur input ini adalah proses khusus yang hanya dilakukan dalam sistem berbasis *Incremental Attribute Learning*. Pendekatan kedua adalah menggunakan *Temporal Feature Extraction*, lalu menghitung standar deviasi dan rata-rata sebagai fitur baru seperti skenario 1, namun dilakukan pengurutan ulang input dan mempertahankan urutan lama dari dataset. Skenario 3 menggunakan *Batch-training* dengan menggunakan fitur dari *second-order feature*. Skenario 4 adalah *Batch-training*, tanpa adanya *temporal feature extraction*, sehingga dataset digunakan secara langsung untuk training. Hasil dari tiap pendekatan secara terurut adalah *error* sebesar 27,4573%; 27,4793%; 29,5046%; 30,6328%.

Jia-Yi Zhu [11] memperkenalkan variasi baru dari ELM yaitu GELM dengan tujuan menyeimbangkan konsistensi properti dari data berdimensi besar, dan mendorong keluaran dari sampel di kelas yang sama menjadi semirip mungkin. Algoritma tersebut diterapkan dalam kasus klasifikasi emosi berdasarkan EEG dimana emosi dibagi menjadi 3 kelas yaitu positif, negatif, dan netral. Performansi keseluruhan dari GELM mampu menyaingi *Support Vector Machine (SVM)* dengan rata-rata 80,25% dan standar deviasi $\pm 11,92\%$ dibanding 76,62% dan standar deviasi $\pm 13,12\%$.

Yong Peng [12] memperkenalkan variasi lain yang baru untuk ELM dengan nama MRELM. GELM hanya memaksa output dari sampel kelas yang sama agar menjadi mirip. Namun dalam studi lebih lanjut, performansi *learning* dapat ditingkatkan jika struktur geometri dari data dieksploitasi. Algoritma tersebut diterapkan dalam kasus yang sama dengan paper nomor 3, yaitu klasifikasi emosi berdasarkan EEG dengan membagi kelas emosi menjadi 3, positif, negatif, dan netral. Secara keseluruhan performansi MRELM mampu menyaingi GELM maupun SVM dengan akurasi 81,01% dan standar deviasi $\pm 12,24\%$.

III. METODE PENELITIAN

A) Pengumpulan Data

Seluruh datasets yang akan digunakan didapatkan dari donasi oleh Oliver Roesler dan disatukan dengan dataset *eye-state* dari website *Repository University of California, Irvine (UCI)* dengan URL: <https://archive.ics.uci.edu/ml/datasets/EEG+Eye+State> yang dibuat oleh orang yang sama. Dataset terdiri atas 7 corpus dimana masing-masing *corpus* merupakan hasil perekaman selama kurang lebih 2 menit dari 4 orang yang berbeda. Masing-masing orang menghasilkan 2 *corpus* kecuali orang terakhir. Terdapat kamera yang memeriksa keadaan mata dari pelaku yang kemudian akan ditambahkan kedalam dataset sebagai target. Dataset *corpus* memiliki 15 *feature*, 14 atribut nilai *real* EEG dari *channel* AF3, F7, F3, FC5, T7, P7, O1, O2, P8, T8, FC6, F4, F8, AF4 dan 1 nilai integer target keadaan mata. 0 menandakan keadaan mata terbuka dan 1 menandakan keadaan mata tertutup. Seluruh keadaan saat mata belum sepenuhnya tertutup dikategorikan sebagai mata terbuka.

B) Alur sistem Keseluruhan

Berikut adalah *Flowchart* dari sistem klasifikasi keadaan mata berdasarkan EEG menggunakan *Extreme Learning Machine*:

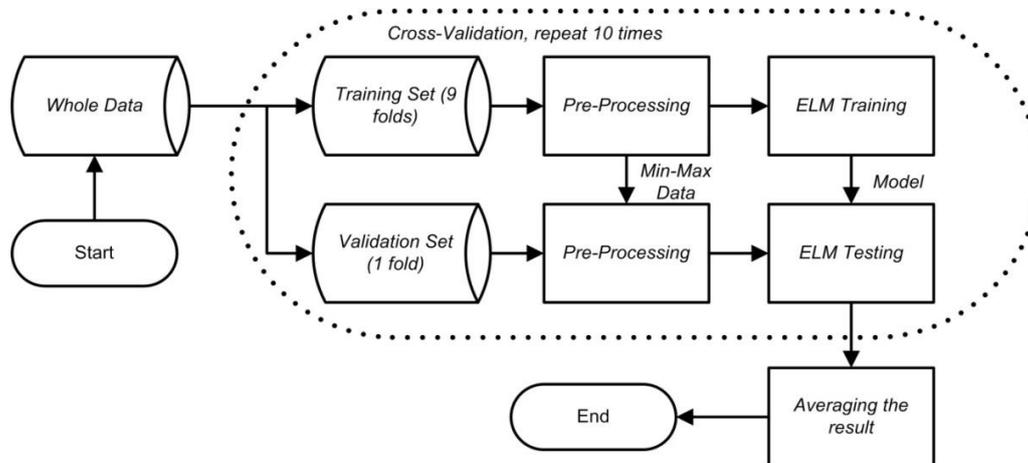


Fig. 3 Flowchart Pengujian Sistem

Secara umum terdapat 4 proses utama dari sistem klasifikasi yang dibuat yaitu: *Cross Validation*, *Preprocessing*, *Proses Training ELM*, dan klasifikasi dengan ELM untuk data uji.

1) *Data latih dan data Uji*

Dataset akan dibagi menjadi data latih dan data uji menggunakan teknik *10-fold cross validation*. Berikut adalah contoh baris fitur yang menjadi input untuk sistem serta target atau keluarannya:

TABLE I
 CONTOH INPUT DAN TARGET SISTEM

Input					Target
X1	X2	X3	X14	
4329.23	4009.23	4289.23	4393.85	0
4292.31	4005.64	4246.67	4345.13	1
4317.44	4010.77	4258.46	4376.41	0
....
4302.56	4006.67	4249.74	4375.9	0

2) *Pre-Processing*

Tahap *Pre-Processing* yang dilakukan adalah normalisasi data menjadi berkisar antara 0 hingga 1. Proses normalisasi dilakukan dengan persamaan berikut:

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (11)$$

Dimana X' adalah nilai baru fitur dalam *domain* normalisasi, X adalah nilai sebelum normalisasi, X_{min} adalah nilai terkecil dari seluruh data yang akan dinormalisasi, dan X_{max} adalah nilai terbesar dari seluruh data yang akan dinormalisasi

3) Sistem Klasifikasi ELM

Dalam kasus klasifikasi keadaan mata ini, akan digunakan *Extreme Machine Learning* (ELM) sebagai classifier. ELM adalah sebuah *Neural Network* dengan 1 *Hidden Layer* dengan menggunakan dengan algoritma *Least Square based* untuk proses latih [13]. Berikut adalah *flowchart* proses latih dari model ELM:

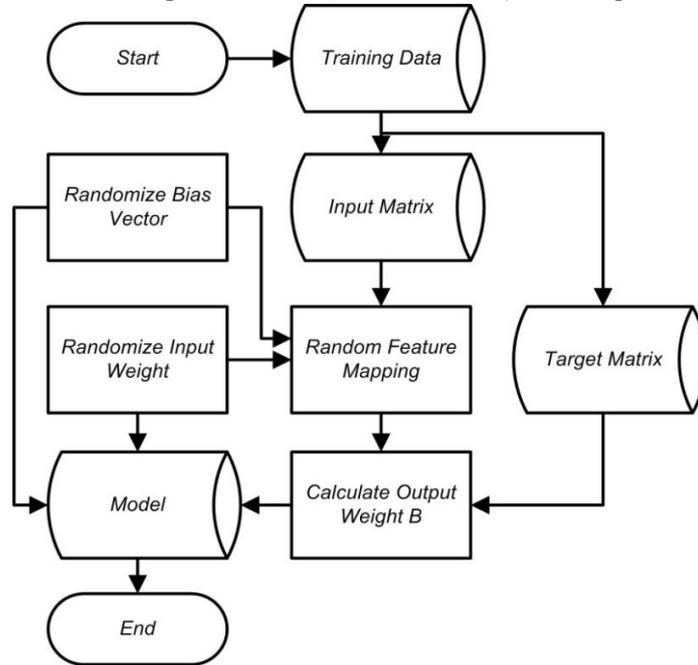


Fig. 4 Flowchart Proses Training ELM

Sebuah ELM dengan 1 *Hidden Layer* dan fungsi aktivasi g dapat dimodelkan sebagai berikut:

$$\beta^T H = T, \tag{12}$$

Dimana:

$$H = \begin{bmatrix} g(w_1 \cdot x_1 + b_1) & \dots & g(w_1 \cdot x_N + b_1) \\ \dots & \dots & \dots \\ g(w_K \cdot x_1 + b_K) & \dots & g(w_K \cdot x_N + b_K) \end{bmatrix}_{K \times N} \tag{13}$$

$$\beta = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_K^T \end{bmatrix}_{K \times m}, T = [t_1, \dots, t_n]_{m \times N} \tag{14}$$

$w_j = (w_{j1}, w_{j2}, \dots, w_{jd})$ adalah *weight* input dari vektor yang menghubungkan *hidden node* ke- j dengan node input, dan $\beta_j = (\beta_{j1}, \beta_{j2}, \dots, \beta_{jm})^T$ adalah *weight* dari vektor yang menghubungkan *hidden node* ke- j dengan node output. B_j adalah bias dari node ke- j .

Matriks Target akan menggunakan *one hot encoding* dimana kelas yang menjadi target akan diberi nilai 1, dan sisanya akan diberi nilai -1. *Weight* dari β dapat diestimasi [14]:

$$\beta = \arg_{\beta} \min \|\beta^T H - T\| = H^+ T = \left(\frac{I}{C} + H^T H \right)^{-1} H^T T \tag{15}$$

Jika digambarkan dalam bentuk *Neural Network*, berikut adalah arsitektur dari ELM yang akan digunakan:

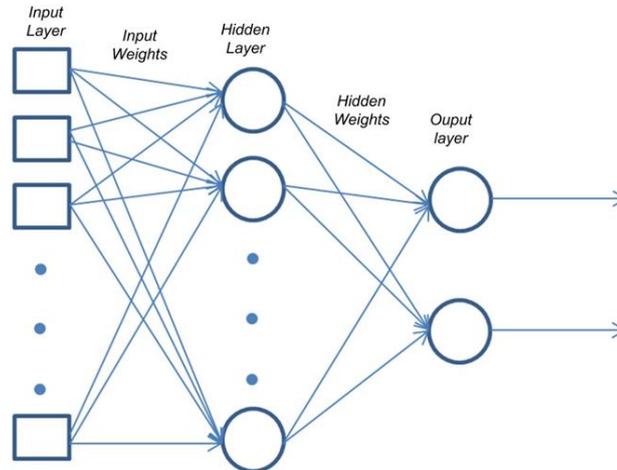


Fig. 5 Penggambaran Model dalam arsitektur NN

Nilai *weight* yang telah didapat kemudian dapat digunakan untuk klasifikasi dengan melakukan pengalian Matriks dari *Input Weight*, penambahan *Bias*, hingga *Hidden Weight*. Karena kasus keadaan mata merupakan *binary classification* dan digunakan 2 *output neuron*. Setiap *output neuron* mewakili masing-masing kelas, jika nilai yang keluar pada *output neuron 0* lebih besar dari *output neuron 1* maka baris data tersebut ditentukan sebagai kelas 0, begitu pula sebaliknya.

C) *Parameter Performansi*

Parameter performansi dari sistem klasifikasi ini adalah akurasi dari sistem, *sensitivity*, *specificity* dan waktu latih model. Semakin tinggi akurasi maka performansi sistem semakin baik sedangkan semakin rendah waktu latih, semakin baik. Akurasi dari sistem didapat dari persamaan:

$$Akurasi = \frac{Jumlah\ klasifikasi\ yang\ benar}{Total\ Data\ yang\ diklasifikasi} * 100\% \tag{16}$$

Sensitivity sistem didapat dari persamaan: (diasumsikan kelas 1 sebagai *True Positive*)

$$Sensitivity = \frac{True\ Positive}{True\ Positive + False\ Negative} * 100\% \tag{17}$$

Specificity sistem didapat dari persamaan: (diasumsikan kelas 1 sebagai *True Positive*)

$$Specificity = \frac{True\ Negative}{True\ Negative + False\ Positive} * 100\% \tag{18}$$

IV. HASIL DAN ANALISIS

A) Tujuan Pengujian

Tujuan dari pengujian yang akan dilakukan adalah sebagai berikut:

1. Menganalisis pengaruh jumlah *hidden neuron* terhadap akurasi dan waktu latih sistem.
2. Menganalisis pengaruh nilai C terhadap akurasi dan waktu latih sistem.
3. Menganalisis pengaruh fungsi aktivasi terhadap akurasi dan waktu latih sistem.
4. Menganalisis perbedaan tiap data saat diklasifikasikan dengan sistem.

B) Skenario Pengujian

Berikut adalah skenario parameter yang akan diuji untuk mengukur performansi dari sistem klasifikasi:

1. Jumlah *Hidden Neuron*
Sistem ELM hanya menggunakan 1 *hidden layer*, namun Jumlah *Hidden Neuron* mempengaruhi seberapa besar matriks dari *ELM feature Mapping*, yang secara teori dengan jumlah cukup besar dapat menghasilkan matriks *output layer* yang memiliki $error = 0$ terhadap data latih [3]. Diuji perubahan jumlah *neuron* dari 100 hingga 1000 dengan interval +100.
2. Nilai C
Sistem ELM menerapkan teori ridge regression dengan menambahkan konstanta $1/C$ ke diagonal *Random Feature Mapping*, yang akan meningkatkan kestabilan dari model yang dihasilkan saat proses latih [7]. Nilai C akan diuji dari 0,1 hingga $1E+17$ dengan interval $\times 10$ dan nilai 0. Nilai 0 berarti tidak ditambahkan nilai konstan apapun.
3. Fungsi Aktivasi
Hidden Neuron pada ELM dapat menggunakan berbagai jenis fungsi aktivasi asalkan merupakan *nonlinier piecewise continuous function*. Fungsi aktivasi yang berbeda akan memberikan matriks *ELM Feature Mapping* yang berbeda juga, sehingga dapat mempengaruhi tahap penyelesaian *least-square problem*. Pada pengujian diuji menggunakan fungsi sigmoid, dan fungsi radial basis.
4. Data latih dan data uji
Data yang akan digunakan adalah masing-masing corpus yang berjumlah 7 ditambah dengan corpus yang merupakan penggabungan dari keseluruhan corpus yang lain. Setiap orang memiliki ciri EEG yang berbeda, baik dari variasi, nilai maksimum, nilai minimum pada kondisi tertentu dsb. Oleh karena itu perlu diuji keefektifan model yang ada untuk orang yang berbeda.

Seluruh skenario akan dikombinasikan untuk melihat pengaruh parameter di tiap corpus yang berbeda, dan mendapatkan parameter terbaik untuk masing-masing corpus. Seluruh corpus akan diberlakukan 10-fold cross validation dan diulang sebanyak 5 kali dan dirata-ratakan. Selain itu untuk meminimalkan keacakan dari hasil yang dihasilkan sistem, pada setiap kombinasi parameter dengan jumlah *hidden neuron* yang sama akan menggunakan 5 matriks input weight yang sama yang telah dibangkitkan terlebih dahulu.

Berikut adalah spesifikasi masing-masing data yang akan digunakan dalam sistem:

C) Spesifikasi Data

Berikut adalah spesifikasi dari setiap *corpus* yang akan digunakan untuk proses latih dan uji pada sistem ELM yang ada:

TABLE II
SPESIFIKASI DATA YANG DIGUNAKAN

Data	Jumlah Baris Data	Persentase Kelas(%)	
		0	1
1-1	15475	49,03	50,97
1-2	15483	63,72	36,28
2-1	15483	56,20	43,80
2-2	15479	48,34	51,66

3-1	15473	17,76	82,24
3-2	15434	40,57	59,43
4	14976	55,11	44,89
Gabungan	107803	47,22	52,78

Terdapat data gabungan yang merupakan penggabungan dari keseluruhan data di atas. Data 1-1 menandakan *corpus* pertama oleh orang pertama, data 1-2 menandakan *corpus* kedua oleh orang pertama, dan seterusnya. *Corpus* 1, 2, dan 3 adalah *corpus* yang digunakan dalam penelitian Oliver Roesler [9], sedangkan *corpus* ke 4 adalah yang digunakan dalam Oliver Roesler [8]. Pada sebagian besar *corpus* persentase kelas yang ada seimbang, hanya pada *corpus* 3-1 kelas yang ada sangat tidak seimbang yaitu 17,76% untuk kelas 0 dan 82,24% untuk kelas 1. Nilai-nilai yang berada di luar batas yang memungkinkan di seluruh *corpus* dihapus, yang diakibatkan kesalahan pada alat perekaman yang digunakan.

D) Hasil Pengujian

Berikut adalah hasil pengujian dari tiap skenario yang ada:

1. Jumlah Hidden Neuron

Berikut adalah tabel hasil pengujian pengaruh jumlah *hidden neuron* terhadap sistem klasifikasi yang dibangun dengan mengambil akurasi dan waktu latih tertinggi pada tiap parameter:

TABLE III
 AKURASI SISTEM ELM PADA JUMLAH HIDDEN NEURON TERTENTU

Jumlah Hidden Neuron	Akurasi Sistem (%)							
	1-1	1-2	2-1	2-2	3-1	3-2	4	Gabungan
100	88,70	93,11	81,00	82,85	90,56	88,70	80,42	70,60
200	92,80	95,14	86,83	86,52	93,49	92,80	86,73	73,63
300	94,18	96,06	89,10	87,32	94,61	94,18	88,94	74,54
400	95,10	96,59	90,37	88,05	95,33	95,10	90,13	75,01
500	95,79	96,98	91,37	88,63	95,96	95,79	91,01	75,73
600	96,45	97,19	92,20	89,32	96,37	96,45	91,68	76,59
700	96,81	97,43	93,07	89,87	96,62	96,81	92,20	77,25
800	97,15	97,69	93,42	90,26	96,93	97,15	92,77	77,94
900	97,40	97,90	93,92	90,85	97,11	97,40	92,96	78,46
1000	97,60	97,95	94,07	91,03	97,32	97,60	93,34	78,94

TABLE IV
 SENSITIVITY SISTEM ELM PADA JUMLAH HIDDEN NEURON TERTENTU

Jumlah Hidden Neuron	Sensitivity (%)							
	1-1	1-2	2-1	2-2	3-1	3-2	4	Gabungan
100	86,42	87,74	69,11	82,56	99,64	93,21	72,27	91,83
200	90,87	92,23	78,54	85,92	99,33	95,50	81,76	89,19
300	92,73	93,75	81,96	86,68	99,20	96,04	85,12	87,60
400	93,10	94,65	84,14	87,27	99,14	96,40	86,82	86,74
500	93,38	95,16	86,00	87,72	99,09	96,94	87,99	85,91
600	93,55	95,47	87,65	88,52	99,07	97,49	88,90	85,48
700	93,53	95,94	89,16	89,03	99,03	97,71	89,64	85,01
800	93,83	96,23	89,85	89,43	99,04	97,97	90,43	84,59
900	93,95	96,51	90,80	89,98	99,07	98,20	90,57	84,43
1000	94,07	96,53	91,09	90,20	99,12	98,36	91,09	84,23

TABLE V
SPECIFICITY SISTEM ELM PADA JUMLAH HIDDEN NEURON TERTENTU

Jumlah Hidden Neuron	Specificity (%)							
	1-1	1-2	2-1	2-2	3-1	3-2	4	Gabungan
100	89,37	90,23	76,11	83,24	94,46	90,73	76,80	74,43
200	92,54	93,22	83,93	86,82	96,12	94,03	84,68	76,69
300	94,09	94,52	86,81	87,59	96,77	95,15	87,35	77,53
400	94,66	95,27	88,44	88,29	97,20	95,90	88,75	77,87
500	95,00	95,80	89,71	88,84	97,57	96,48	89,78	78,42
600	95,18	96,09	90,77	89,53	97,81	97,03	90,56	79,02
700	95,30	96,44	91,85	90,07	97,96	97,33	91,15	79,49
800	95,48	96,80	92,28	90,46	98,15	97,62	91,82	80,01
900	95,72	97,09	92,90	91,04	98,26	97,82	92,03	80,45
1000	95,79	97,15	93,08	91,22	98,38	97,99	92,46	80,82

TABLE VI
LAMA WAKTU LATIH SISTEM ELM PADA JUMLAH HIDDEN NEURON TERTENTU

Jumlah Hidden Neuron	Lama Waktu latih (s)							
	1-1	1-2	2-1	2-2	3-1	3-2	4	Gabungan
100	0,04	0,04	0,04	0,04	0,04	0,04	0,04	0,29
200	0,09	0,09	0,09	0,09	0,09	0,09	0,08	0,63
300	0,15	0,15	0,15	0,15	0,15	0,15	0,14	1,04
400	0,21	0,21	0,22	0,21	0,21	0,21	0,20	1,59
500	0,29	0,29	0,30	0,29	0,29	0,29	0,28	2,18
600	0,39	0,38	0,39	0,38	0,38	0,38	0,37	2,82
700	0,49	0,48	0,49	0,47	0,48	0,47	0,46	3,53
800	0,61	0,58	0,60	0,58	0,58	0,58	0,56	4,24
900	0,72	0,70	0,72	0,70	0,70	0,70	0,68	5,05
1000	0,86	0,83	0,86	0,83	0,82	0,82	0,80	5,98

Berdasarkan Tabel III, IV, dan V terlihat bahwa pada semakin besar jumlah hidden neuron yang digunakan pada sistem, maka akurasi, *Sensitivity*, dan *Specificity* sistem semakin meningkat. Pada data 3-1 *Sensitivity* pada jumlah *Hidden neuron* 100 paling tinggi diakibatkan kurang seimbangnya kelas dari *corpus* tersebut, dimana kelas 1 jauh lebih dominan. Peningkatan akurasi dari cenderung makin mengecil pada saat jumlah hidden neuron semakin besar. Contohnya pada data 1-1, terlihat bahwa peningkatan jumlah hidden neuron dari 100 ke 200 terjadi peningkatan akurasi dari 88,70% menuju 92,80%, namun pada jumlah hidden neuron 900 ke 1000, hanya terjadi peningkatan akurasi dari 97,40% menjadi 97,60%. Sedangkan untuk *sensitivity* dan *specificity*, dengan meningkatnya jumlah *hidden neuron*, keseimbangan antara kedua hal tersebut akan semakin baik, seperti pada data 3-1 dimana pada jumlah *hidden neuron* 100 *sensitivity* mencapai nilai tertinggi yaitu 99,64% namun *specificity* hanya mencapai 94,46%. Pada jumlah *hidden neuron* 1000, walaupun *sensitivity* mengalami penurunan kecil menjadi 99,12% namun *specificity* mengalami peningkatan drastis hingga 98,38% sehingga lebih baik dalam merekognisi kedua kelas. Selain itu pada tabel VI tampak bahwa jumlah hidden neuron sangat mempengaruhi seberapa lama waktu latih yang dibutuhkan oleh sistem ELM. Semakin besar jumlah hidden neuron maka waktu latih yang dibutuhkan semakin tinggi. Peningkatan waktu latih tidak terjadi secara linear seiring semakin banyak jumlah hidden neuron, namun secara eksponensial. Contohnya pada data 1-2, peningkatan jumlah hidden neuron dari 100 ke 200 hanya

meningkatkan waktu latih dari 0,04 detik menjadi 0,09 detik, namun pada jumlah hidden neuron 900 menuju 1000, terjadi peningkatan waktu latih dari 0,70 detik menjadi 0,83 detik.

2. Nilai C

Berikut adalah tabel hasil pengujian pengaruh jumlah *hidden neuron* terhadap sistem klasifikasi yang dibangun dengan mengambil akurasi dan waktu latih tertinggi pada tiap parameter:

TABLE VII
 AKURASI SISTEM ELM PADA NILAI C TERTENTU

Nilai C	Akurasi Sistem (%)							
	1-1	1-2	2-1	2-2	3-1	3-2	4	Gabungan
0	95,80	97,95	94,07	91,03	97,32	97,60	93,34	78,94
0,1	84,89	94,44	68,68	80,80	90,69	91,10	81,95	66,57
1	86,23	95,76	72,39	84,50	93,31	94,02	87,02	68,02
10	87,96	96,82	75,93	86,94	95,29	95,74	89,87	69,68
100	90,57	97,47	79,97	88,10	96,35	96,82	91,85	71,58
1000	92,37	97,82	84,10	89,29	96,88	97,30	92,92	73,37
10000	93,91	97,93	87,03	90,20	97,05	97,40	93,23	74,27
100000	94,75	97,94	89,19	90,66	97,08	97,40	93,20	75,09
1000000	95,22	97,95	90,87	90,71	97,08	97,44	93,21	76,06
10000000	95,63	97,95	92,30	90,72	97,27	97,60	93,31	77,27

TABLE VIII
 STANDAR DEVIASI AKURASI SISTEM ELM PADA NILAI C TERTENTU

Nilai C	Standar Deviasi (%)							
	1-1	1-2	2-1	2-2	3-1	3-2	4	Gabungan
0	3,29	0,43	4,84	0,82	0,58	0,66	0,77	2,49
0,1	0,93	0,74	1,68	1,18	0,82	1,02	1,22	0,56
1	0,86	0,64	2,11	1,11	0,77	0,92	1,16	0,55
10	0,83	0,57	2,72	1,01	0,71	0,84	1,05	0,58
100	0,82	0,50	3,04	0,96	0,68	0,75	0,95	0,58
1000	0,77	0,48	2,94	0,88	0,64	0,73	0,88	0,53
10000	0,73	0,45	2,60	0,84	0,62	0,70	0,85	0,54
100000	0,68	0,44	2,12	0,83	0,61	0,68	0,82	0,51
1000000	0,66	0,43	1,69	0,83	0,59	0,67	0,78	0,48
10000000	0,60	0,43	1,36	0,83	0,59	0,67	0,77	0,46

TABLE IX
 SENSITIVITY AKURASI SISTEM ELM PADA NILAI C TERTENTU

Nilai C	Sensitivity (%)							
	1-1	1-2	2-1	2-2	3-1	3-2	4	Gabungan
0	93,92	96,53	91,09	90,20	99,09	98,34	91,09	84,09
0,1	80,30	90,58	49,09	81,19	99,64	95,23	71,87	91,83
1	81,28	93,38	54,17	84,83	98,97	96,28	81,38	84,73
10	83,55	94,98	61,37	86,66	98,77	97,05	85,94	81,99
100	88,17	95,83	67,38	87,46	98,88	97,80	88,77	82,19
1000	90,67	96,34	73,38	88,67	99,00	98,11	90,42	82,58
10000	92,41	96,47	77,86	89,63	99,01	98,17	90,92	82,99
100000	93,29	96,50	81,58	90,02	98,98	98,17	90,99	83,63

1000000	93,68	96,53	84,48	89,98	99,05	98,29	91,03	83,90
10000000	94,07	96,53	87,32	89,96	99,12	98,36	91,02	84,08

TABLE X
SPECIFICITY AKURASI SISTEM ELM PADA NILAI C TERTENTU

Nilai C	Specificity (%)							
	1-1	1-2	2-1	2-2	3-1	3-2	4	Gabungan
0	95,79	97,15	93,08	91,22	98,38	97,98	92,46	80,82
0,1	84,33	92,19	57,83	81,37	94,57	92,71	78,13	73,31
1	85,74	94,10	63,11	84,95	96,04	95,03	84,90	72,67
10	87,60	95,59	68,96	87,26	97,18	96,44	88,39	73,86
100	90,50	96,48	74,56	88,36	97,81	97,34	90,71	75,33
1000	92,37	96,97	80,06	89,53	98,12	97,74	91,97	76,59
10000	93,92	97,12	83,97	90,43	98,22	97,82	92,33	77,30
100000	94,76	97,14	86,83	90,87	98,24	97,82	92,31	77,99
1000000	95,23	97,15	89,00	90,91	98,24	97,85	92,32	78,71
10000000	95,64	97,15	90,84	90,92	98,35	97,99	92,41	79,61

TABLE XI
LAMA WAKTU LATIH SISTEM ELM PADA NILAI C TERTENTU

Nilai C	Lama Waktu latih (s)							
	1-1	1-2	2-1	2-2	3-1	3-2	4	Gabungan
0	0,86	0,82	0,86	0,81	0,82	0,82	0,79	5,98
0,1	0,83	0,82	0,83	0,82	0,82	0,82	0,80	5,78
1	0,83	0,82	0,83	0,83	0,82	0,82	0,80	5,78
10	0,83	0,82	0,83	0,82	0,82	0,82	0,80	5,78
100	0,83	0,83	0,83	0,82	0,82	0,82	0,79	5,80
1000	0,83	0,82	0,83	0,82	0,82	0,82	0,79	5,78
10000	0,83	0,83	0,83	0,82	0,82	0,82	0,79	5,78
100000	0,83	0,82	0,83	0,82	0,82	0,82	0,79	5,78
1000000	0,83	0,82	0,83	0,82	0,82	0,82	0,80	5,78
10000000	0,83	0,82	0,83	0,82	0,82	0,82	0,80	5,80

Pada Tabel VII, IX, dan X terlihat semakin besar nilai C, yang berarti semakin kecil nilai konstan yang ditambahkan ke dalam diagonal dari ELM feature Mapping akurasi, *specificity*, dan *sensitivity* dari sistem cenderung semakin meningkat. Rata-rata akurasi terbaik sendiri didapatkan pada seluruh data kecuali data saat tidak ditambahkan nilai konstan apapun atau nilai I/C adalah 0 kecuali untuk data 3-2. Pada data 1-2 sendiri akurasi pada nilai C yang sangat besar yaitu 1E+06 dan 1E+07 dapat mencapai nilai akurasi yang hampir sama dengan saat nilai konstan sama dengan 0 yaitu 97,95%. Hal ini juga terjadi pada data 3-2 dimana akurasi sama-sama mencapai 97,60% pada nilai C 1E+07 dan saat nilai konstan diset sama dengan nol, dengan akurasi 1E+07 sedikit lebih tinggi dengan perbedaan 0,005%. Walaupun rata-rata akurasi yang didapat cukup tinggi pada *cross validation* namun tidak menambahkan nilai konstan sama sekali dapat berdampak pada stabilitas sistem yang berkurang. Pada satu data 1-1 Standar deviasi akurasi sistem yang menggunakan nilai konstan 0 memiliki standar deviasi yang cukup tinggi dibanding saat menggunakan nilai konstan yang sangat kecil sekalipun. Pada data 1-1 dengan nilai C 1E+07 hanya memiliki standar deviasi 0,6%, sedangkan saat konstanta diset menjadi 0, standar deviasi meningkat jauh hingga 3,29%. Hal ini juga terjadi pada 2-1 dan data gabungan. Oleh karena itu lebih baik menggunakan nilai konstan yang kecil. Selain didapatkan sistem dengan stabilitas lebih baik dengan standar deviasi yang lebih kecil, juga tidak menurunkan performansi sistem terlalu banyak dari segi akurasi, *sensitivity*, dan *specificity*, mengingat pada beberapa

kasus seperti pada data 3-2 *sensitivity* dan *specificity* tertinggi didapat pada saat nilai C 1E+07, lebih tinggi daripada saat tidak ditambahkan nilai konstan tertentu.

Untuk pengaruh waktu latih dari perubahan nilai C sendiri, terlihat pada tabel XI bahwa perubahannya tidak banyak mempengaruhi lama waktu latih yang dibutuhkan. Pada data yang terpisah seperti data 1-1 hingga 4, dibutuhkan waktu latih yang kurang lebih sama berkisar antara 0,82 hingga 0,83 detik yang kemungkinan dipengaruhi faktor lain dalam komputer saat eksperimen, sedangkan waktu latih dari data gabungan sendiri kurang lebih berada pada nilai 5,78 hingga 5,80. Namun pada saat nilai konstan diset sama dengan nol, kadang terjadi pelonjakan waktu latih seperti yang terjadi pada 1-1, 2-1 dan data gabungan. Pada data 1-1 dan 2-1 waktu latih menjadi 0,86 detik dibandingkan dengan waktu latih pada parameter yang lain yang berkisar antara 0,82 hingga 0,83 detik, sedangkan pada data gabungan waktu latih naik menjadi 5,98 detik dibanding pada parameter lain yang berkisar pada 5,78 hingga 5,80 detik.

3. Fungsi Aktivasi

Berikut adalah tabel hasil pengujian pengaruh fungsi aktivasi yang digunakan terhadap sistem klasifikasi yang dibangun dengan mengambil akurasi dan waktu latih tertinggi pada tiap parameter:

TABLE XII
 AKURASI SISTEM ELM DENGAN FUNGSI AKTIVASI TERTENTU

Fungsi Aktivasi	Akurasi Sistem (%)							
	1-1	1-2	2-1	2-2	3-1	3-2	4	Gabungan
Sigmoid	95,00	97,81	89,10	91,03	97,32	97,60	93,34	74,34
Radbas	95,80	97,95	94,07	90,72	97,09	97,41	93,23	78,94

TABLE XIII
 SENSITIVITY SISTEM ELM DENGAN FUNGSI AKTIVASI TERTENTU

Fungsi Aktivasi	Sensitivity (%)							
	1-1	1-2	2-1	2-2	3-1	3-2	4	Gabungan
Sigmoid	93,31	96,30	81,96	90,20	99,64	98,36	91,09	91,83
Radbas	94,07	96,53	91,09	90,02	99,01	98,17	91,03	84,71

TABLE XIV
 SPECIFICITY SISTEM ELM DENGAN FUNGSI AKTIVASI TERTENTU

Fungsi Aktivasi	Specificity (%)							
	1-1	1-2	2-1	2-2	3-1	3-2	4	Gabungan
Sigmoid	95,00	96,96	86,81	91,22	98,38	98,36	92,46	77,53
Radbas	95,79	97,15	93,08	90,92	98,24	97,83	92,33	80,82

TABLE XV
 LAMA WAKTU LATIH SISTEM ELM DENGAN FUNGSI AKTIVASI TERTENTU

Fungsi Aktivasi	Lama Waktu latih (s)							
	1-1	1-2	2-1	2-2	3-1	3-2	4	Gabungan
Sigmoid	0,82	0,83	0,86	0,83	0,82	0,82	0,80	5,98
Radbas	0,86	0,81	0,85	0,82	0,82	0,82	0,79	5,74

Tabel VII menunjukkan bahwa tiap fungsi aktivasi yang digunakan berpengaruh berbeda-beda pada tiap data yang digunakan. Terdapat sistem ELM yang menghasilkan akurasi lebih baik saat menggunakan fungsi aktivasi sigmoid, dan ada juga yang sebaliknya. Pada data 2-1 dan data gabungan bahkan didapatkan akurasi,

sensitivity, dan *specificity* yang berbeda cukup signifikan saat menggunakan kedua fungsi aktivasi yang berbeda yaitu 89,10%; 81,96%; 86,81% untuk sigmoid, dan 94,07%; 91,09%; 93,08% untuk radial basis. Begitu juga dengan waktu latih dari sistem, masing-masing fungsi aktivasi memberikan perilaku yang berbeda pada tiap-tiap data yang ada. Namun fungsi aktivasi sigmoid cenderung menghasilkan waktu latih yang lebih tinggi daripada radial basis. Walaupun pada beberapa data, sigmoid tercatat memiliki waktu latih yang lebih tinggi namun perbedaannya dengan waktu latih radial basis pada saat itu tidak terpaut jauh. Namun tidak ada korelasi antara lama waktu latih dari sistem dengan akurasi yang didapat, karena pada data 1-1, 1-2, 2-1 dan gabungan yang menggunakan sistem aktivasi radial basis membutuhkan waktu latih yang lebih rendah dibanding sigmoid namun didapat akurasi yang lebih baik. Sedangkan pada data lainnya, walaupun waktu latih dari radial basis lebih rendah, namun akurasi dari fungsi aktivasi sigmoid lebih baik dibanding menggunakan radial basis.

E) Analisis Keseluruhan

Berikut adalah tabel kombinasi parameter yang menghasilkan akurasi terbaik pada masing-masing data:

TABLE XVI
KOMBINASI PARAMETER TERBAIK UNTUK MASING-MASING DATA

Data	Fungsi Aktivasi	Jumlah Hidden Neuron	Nilai C	Akurasi (%)	Waktu Latih (s)
1-1	Radial Basis	1000	0	95,80	0,81
1-2	Radial Basis	1000	0	97,95	0,81
2-1	Radial Basis	1000	0	94,07	0,85
2-2	Sigmoid	1000	0	91,03	0,81
3-1	Sigmoid	1000	0	97,32	0,82
3-2	Sigmoid	1000	1E+07	97,60	0,82
4	Sigmoid	1000	0	93,34	0,79
Gabungan	Radial Basis	1000	0	78,94	5,71

Seperti terlihat pada Tabel XVI, saat masing-masing data dilatih dan diuji secara terpisah, sistem dengan kombinasi parameter yang tepat dapat melakukan klasifikasi dengan akurasi yang terhitung baik, yaitu diatas 90%. Sistem bahkan dapat mencapai akurasi diatas 95% untuk beberapa data. Hanya data 2-1, 2-2 dan 4 yang tidak mencapai batas 95%. Hal ini diduga karena setiap orang bisa saja dalam kondisi fisik maupun mental yang berbeda pada saat perekaman data. Orang 2 dan 4 terhitung lebih sulit diklasifikasi dibanding orang 1 dan 3 yang dapat mencapai akurasi lebih tinggi. Namun untuk data gabungan yang merupakan pencampuran dari keseluruhan data menjadi satu, akurasi dari sistem turun drastis hingga hanya dapat mencapai 78,80%. Hal ini menandakan bahwa sistem masih belum dapat digunakan untuk klasifikasi EEG umum secara langsung. Suatu model yang dibangun berdasarkan sinyal EEG satu orang tidak dapat digunakan untuk kasus klasifikasi dari sinyal EEG milik orang yang berbeda. Hal ini bisa diakibatkan pola sinyal EEG masing-masing orang yang berbeda-beda, mengingat tiap data memiliki nilai maksimum, minimum, dan rata-rata yang berbeda di tiap *channel*. Oleh karena itu perlu dilakukan suatu fitur ekstraksi untuk menyamakan ciri fitur dari kelas yang sama dari data milik orang yang berbeda. Dengan begitu walaupun secara mentah nilai sinyal masing-masing orang berbeda-beda, namun pada kelas yang sama bisa didapatkan suatu ciri yang sama.

Sistem klasifikasi ELM menggunakan metode *Random Feature Mapping* yang berarti pada setiap proses *training* dapat menghasilkan model yang benar-benar berbeda satu dengan lain walaupun menggunakan data dan parameter yang sama persis, karena sangat bergantung pada nilai *random* yang dihasilkan. Namun dari Standar Deviasi akurasi yang didapat dari masing-masing data, terlihat bahwa performansi yang didapat tidak terlalu jauh walaupun bergantung pada nilai acak. Pada nilai C sama dengan 0, atau tidak ditambahkan

konstanta sama sekali, kadang dapat berdampak stabilitas proses latih ELM yang kurang baik. Seperti yang terjadi data 1-1 dimana pada parameter C lain standar deviasi dari akurasi pengujian tidak sampai titik 1%, namun pada saat nilai C sama dengan 0, standar deviasi akurasi pengujian sistem meningkat hingga 4%. Peristiwa ini jarang terjadi, karena pada parameter nilai C sama dengan 0 untuk data lain tidak terjadi hal itu, namun dapat menjadi pertimbangan untuk selalu menggunakan parameter nilai C walaupun sekecil mungkin untuk mendapatkan stabilitas proses latih sistem yang lebih baik.

V. KESIMPULAN

Setelah melakukan perancangan, implementasi, pengujian dan analisis, maka kesimpulan yang dapat diambil dari penelitian ini adalah:

1. Sistem klasifikasi ELM dapat digunakan untuk melakukan klasifikasi keadaan mata berdasarkan sinyal EEG. Akurasi dinilai baik namun hanya jika setiap corpus dilatih dan diuji secara terpisah, terlihat dari akurasi sistem yang mencapai paling tinggi 97,95% untuk data terpisah, namun hanya dapat mencapai 78,94% untuk data gabungan.
2. Sistem klasifikasi ELM cocok digunakan untuk melakukan proses training secara real-time karena dalam 1 kali proses latih tidak memakan waktu yang lama, bahkan untuk data yang sangat banyak sekalipun. Untuk data terpisah waktu latih terlama yang dibutuhkan hanya 0,86 detik, sedangkan data gabungan hanya dibutuhkan waktu 5,98 detik.
3. Dengan semakin banyaknya jumlah *hidden neuron*, maka akurasi sistem akan semakin tinggi, terlihat dari akurasi terbaik selalu didapat pada jumlah hidden neuron 1000 pada seluruh data hingga 97,95%. Hanya saja perlu dipertimbangkan peningkatan waktu latih dari sistem yang akan digunakan.
4. Tidak menambahkan konstanta pada matriks ELM Feature Mapping menghasilkan sistem dengan akurasi terbaik pada hampir seluruh data kecuali pada data 3-2 dengan akurasi mencapai 97,95%, namun terkadang dapat berdampak pada stabilitas sistem ELM yang kurang baik, terlihat pada standar deviasi pengujian data 1-1 yang besar hingga mencapai 4%. Penggunaan parameter C cukup besar dapat menghasilkan sistem dengan akurasi yang tidak banyak menurun namun dengan stabilitas yang lebih baik seperti $1E+07$ yang dapat mencapai akurasi yang sama namun dengan standar deviasi yang jauh lebih kecil yaitu hanya sekitar 1%.
5. Fungsi aktivasi memiliki pengaruh masing-masing pada data yang berbeda, contohnya pada data 2-2 sigmoid dapat mencapai akurasi 91,03% namun radial basis hanya mencapai 90,72%, tetapi pada data 2-1 sigmoid hanya mencapai akurasi 89,10% dibanding radial basis yang mencapai 94,07%
6. Dengan didapatkannya model dengan akurasi yang baik dengan waktu latih yang cenderung cepat, algoritma ELM dapat dimanfaatkan untuk sebagai *Brain-computer Interface* dengan sebelumnya melakukan pelatihan berdasarkan sinyal EEG orang bersangkutan dengan waktu singkat untuk membangun model. Namun karena model dari 1 orang tidak bisa digunakan untuk orang yang berbeda, maka untuk setiap orang yang berbeda yang menggunakan sistem, perlu dilakukan proses latih ulang.

UCAPAN TERIMA KASIH

Penulis ingin berterima kasih kepada Oliver Roesler yang telah mendonasikan dataset *eye-state* yang digunakan penulis untuk penelitian ini, dan kepada Lab *Artificial Intelligence*, Lab *Computing*, dan Lab Praktikum Informatika Universitas Telkom yang telah meminjamkan komputer untuk uji coba sistem.

REFERENSI

- [1] Chen, L., L. Li, and L. Xiao. "Differences of EEG between Eyes-Open and Eyes-Closed States Based on Autoregressive Method." *Journal Of Electronic Science And Technology Of China*, 2009: 175-179.
- [2] Roesler, Oliver, dan David Suendermann. "A First Step towards Eye State Prediction Using EEG." in *Proceedings of International Conference on Applied Informatics for Health and Life Science*. Istanbul: Turkey, 2013.

- [3] Huang, Guang-Bin, Qin-Yu Zhu, dan Chee-Kheong Siew. "Extreme Learning Machine: Theory and applications." *Neurocomputing* 70, 2006: 489-501.
- [4] Zheng, Wei-Long, Jia-Yi Zhu, Yong Peng, dan Bao-Liang Lu. "EEG-based emotion classification using deep belief networks." *IEEE International Conference on Multimedia and Expo*. Chengdu: IEEE, 2014. 1-6.
- [5] Naibaho, Vania Cathlin Raissa. *Klasifikasi Emosi Melalui Sinyal EEG yang Dihasilkan Otak dengan Menggunakan Discrete Wavelet Transform dan Backpropagation Artificial Neural Network*. Undergraduate Thesis, Bandung: Telkom University, 2013.
- [6] Suyanto. *Artificial Intelligence*. Bandung: Informatika Bandung, 2009.
- [7] Huang, Guang-Bin, Hongming Zhou, Xiaojian Ding, dan Rui Zhang. "Extreme Learning Machine for Regression and Multiclass Classification." *IEEE Transaction on Systems, Man, and Cybernetics*, 2012: 513-529.
- [8] Tang, Jiexiong, Chenwei Deng, dan Guang-Bin Huang. "Extreme Learning Machine for Multilayer Perceptron." *IEEE Transactions on Neural Networks And Learning Systems*, 2013: 809 - 821.
- [9] Roesler, Oliver, Lucas Bader, Jan Forster, Yoshikatsu Hayashi, Stefan Hebler, dan David Suendermann. "Comparison of EEG Devices for Eye State Classification." *International Conference on Applied Informatics for Health and Life Sciences (AIHLS), Kusadasi, Turkey*, 2014.
- [10] Wang, Ting, Sheng-Uei Guan, Ka Lok Man, dan T. O. Ting. "Time Series Classification for EEG Eye State Identification based on Incremental Attribute Learning." *International Symposium on Computer, Consumer and Control*. Taichung: 2014, 2014. 158 - 161.
- [11] Zhu, Jia-yi, Wei-Long Zheng, Yong Peng, Ruo-Nan Duan, dan Bao-Liang Lu. "EEG-based Emotion Recognition using Discriminative Graph Regularized Extreme Learning Machine." *International Joint Conference on Neural Networks*, 2014: 525-532.
- [12] Peng, Yong, Jia-Yi Zhu, Wei-Long Zheng, dan Bao-Liang Lu. "EEG-based Emotion Recognition with Manifold Regularized Extreme Learning Machine." *36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 2014: 974-977.
- [13] Huang, Guang-Bin. "What are Extreme Learning Machines? Filling the Gap Between between Frank Rosenblatt's Dream and John von Neumann's Puzzle." *Cognitive Computation* 7, 2015: 263-278.
- [14] Huang, Gao, Shiji Song, Keyou You, dan Guang-Bin Huang. "Trends in extreme learning machines: A review." *Neural Networks*, 2015: 32-48.

