

# Klasifikasi Mata Terbuka dan Tertutup Berdasarkan Sinyal EEG Menggunakan *Artificial Neural Network* dan *Principal Component Analysis*

Abdurrahman Faqih<sup>1</sup>, Amanda Muchsin Chalik<sup>2</sup>, Bilal Abdul Qowy<sup>3</sup>, Faiz Hanafi<sup>4</sup>, Ahlijati Nuraminah<sup>5</sup>  
Program Studi Ilmu Komputer<sup>1,2,3,4,5</sup>  
Sekolah Tinggi Ilmu Manajemen dan Ilmu Komputer ESQ<sup>1,2,3,4,5</sup>  
JL TB Simatupang, Cilandak Timur, Kec. Ps. Minggu, Kota Jakarta Selatan 12560  
[a.faqih@students.esqbs.ac.id](mailto:a.faqih@students.esqbs.ac.id)<sup>1</sup>, [a.muchsin.chalik@students.esqbs.ac.id](mailto:a.muchsin.chalik@students.esqbs.ac.id)<sup>2</sup>, [b.abdul.q@students.esqbs.ac.id](mailto:b.abdul.q@students.esqbs.ac.id)<sup>3</sup>,  
[f.hanafi@students.esqbs.ac.id](mailto:f.hanafi@students.esqbs.ac.id)<sup>4</sup>, [ahlijati.nuraminah@esqbs.ac.id](mailto:ahlijati.nuraminah@esqbs.ac.id)<sup>5</sup>

**Abstrak.** Aktivitas otak manusia dapat direkam melalui biosinyal *electroencephalogram* (EEG). Dalam kondisi tertentu, otak manusia memberikan respon dengan mengeluarkan emosi tertentu yang ditampilkan pada wajah seperti mata terbuka ataupun tertutup. Untuk mengidentifikasi jenis respon emosi yang diberikan otak dapat dilakukan dengan cara mengklasifikasikan sinyal EEG dengan akurat. Oleh karena itu pada penelitian ini digunakan model berbasis *Artificial Neural Network* (ANN) dalam mengklasifikasikan mata tertutup dan mata terbuka. Dalam performansi pengujian, juga digunakan metode *Principal Component Analysis* yang berfungsi untuk menganalisis sekumpulan data multidimensi. Analisis menggunakan dataset *Eye State Classification* dari Kaggle. Hasil uji coba sampel pada 14980 data diperoleh akurasi sebesar 0.57 atau 57% dalam mengklasifikasi mata terbuka dan mata tertutup.

**Kata Kunci:** Elektroensefalogram, Klasifikasi, *Artificial Neural Network*, *Principal Component Analysis*.

## 1 Pendahuluan

Manusia adalah makhluk yang mempunyai perasaan atau bisa juga disebut dengan emosi [1]. Emosi menggambarkan suatu reaksi yang timbul terhadap suatu kejadian yang dialami oleh manusia dari lingkungan sekitarnya [2]. Reaksi yang ditimbulkan dari emosi dapat berupa perilaku negatif yang dapat mempengaruhi hubungan manusia dengan sekitarnya. Emosi dapat dikategorikan sebagai emosi negatif dan emosi positif. Contoh emosi negatif di antaranya kesedihan, bencian, kemarahan, perasaan malu, perasaan bersalah, dan putus asa. Sedangkan contoh emosi positif misalnya rasa bangga, rasa syukur, ketenangan, gembira dan terhibur [3].

Beberapa penelitian terdahulu menguji tentang bagaimana otak manusia bekerja, salah satunya mengenai respon otak manusia ketika dalam kondisi tertentu. Dalam kondisi tertentu, otak manusia memberikan respon dengan mengeluarkan emosi tertentu yang ditampilkan pada wajah seperti mata terbuka ataupun tertutup. Untuk mengidentifikasi jenis respon emosi yang diberikan otak dapat dilakukan dengan cara menganalisis sinyal yang ada pada tubuh manusia atau yang disebut dengan *biosignal*. Biosinyal merupakan sinyal yang berasal dari makhluk hidup yang timbul akibat adanya perbedaan potensial listrik [4].

Untuk mendeteksi sinyal pada tubuh manusia dapat dilakukan dalam berbagai alat antara lain *electroencephalogram* (EEG), *electrocardiogram* (ECG), *electrooculography* (EOG), dsb. Di antara berbagai sinyal tersebut, sinyal EEG paling umum dimanfaatkan untuk menangkap sinyal tubuh manusia. Rangsangan yang diterima otak yang berupa aktivitas listrik statis akan direkam pada EEG [5]. Karena kemampuan tersebut, EEG juga dapat dimanfaatkan untuk mendeteksi aktivitas abnormal pada tubuh manusia lainnya seperti gangguan tidur dan kematian otak melalui aplikasi sistem komputer atau *brain-computer interface*. Agar dapat digunakan sebagai *brain-computer interface*, sistem diharapkan dapat memberikan hasil jenis respon dengan cepat dan akurat khususnya untuk sistem yang bersifat *real time* [6].

Untuk dapat digunakan, sinyal EEG harus dapat dianalisis dengan memadai sehingga dapat mengklasifikasikan mata tertutup dan mata terbuka untuk mengetahui respon emosi dengan akurat. Terdapat berbagai metode yang dapat digunakan untuk mengklasifikasikan sinyal EEG, salah satunya adalah *Artificial Neural Network* (ANN). *Artificial Neural Network* merupakan metode *machine learning* yang memungkinkan untuk mengetahui hubungan antara data input dan output yang belum dipelajari sebelumnya [7]. Arsitektur *Artificial Neural Network* umumnya terdiri atas tiga sampai empat layer, dengan setiap layer memiliki beberapa neuron.

Lapisan-lapisan layer disebut sebagai dinamakan *Multilayer Perceptron* (MLP) di mana setiap neuron terhubung dengan layer akhir yang merupakan layer klasifikasi. Lapisan (layer) terdiri dari input layer, *hidden* layer dan output layer yang menyusun neuron-neuron [8].

Dalam performasi pengujian, juga digunakan metode *Principal Component Analysis* yang berfungsi untuk menganalisis sekumpulan data multidimensi. Cara kerja PCA dengan mereduksi dimensi data melalui pemilihan variabel-variabel untuk menggambarkan variabilitas data. *Principal Components* merupakan variabel-variabel baru yang dibentuk PCA untuk mengurangi dimensi. Konsep nilai eigen (*eigenvalue*) dan vektor eigen (*eigenvector*) diterapkan pada *Principal Components* [9]. Dengan demikian penelitian ini bertujuan untuk mengimplementasikan algoritma *Artificial Neural Network* dan *Principal Component Analysis* untuk mengklasifikasikan sinyal EEG pada mata tertutup dan mata terbuka yang dapat mengetahui emosi dari manusia dengan dataset bersumber pada website Kaggle.

## 2 Dasar Teori

### 2.1 Electroencephalograph

*Electroencephalograph* (EEG) merupakan proses perekaman yang terjadi pada otak manusia terhadap aktivitas listrik. Sejumlah elektroda akan ditempelkan pada bagian atas kepala untuk melakukan aktivitas perekaman. Penempatan kanal berupa Frontal (F), Parietal (P), Occipital (O) dan Temporal (T). Frontal untuk menangani pengontrolan, gerakan, dan kemampuan bicara. Sementara itu Parietal menangani rangsangan berupa sentuhan dan temperatur. Occipital terkait dengan rangsangan visual, sedangkan Temporal untuk rangsangan audio dan memori [10].

### 2.2 Artificial Neural Network

Cara berpikir manusia dapat dimodelkan ke dalam komputer melalui teknik *Artificial Neural Network* atau Jaringan Syaraf Tiruan (JST). Melalui JST, pengetahuan yang bersumber dari pengalaman akan disimpan dan disebarkan secara paralel sehingga siap untuk digunakan. Bobot koneksi antar sel syaraf yang ada (neuron) merepresentasikan pengetahuan yang telah didapat dalam JST. Arsitektur *Artificial Neural Network* umumnya terdiri atas tiga sampai empat layer, dengan setiap layer memiliki beberapa neuron. Lapisan (layer) terdiri dari input layer, *hidden* layer dan output layer yang menyusun neuron-neuron [3].

### 2.3 Analisis Sentimen

*Principal Component Analysis* (PCA) berfungsi untuk menganalisis sekumpulan data multidimensi. Cara kerja PCA dengan mereduksi dimensi data melalui pemilihan variabel-variabel untuk menggambarkan variabilitas data. *Principal Components* merupakan variabel-variabel baru yang dibentuk PCA untuk mengurangi dimensi. Konsep nilai eigen (*eigenvalue*) dan vektor eigen (*eigenvector*) diterapkan pada *Principal Components* [9].

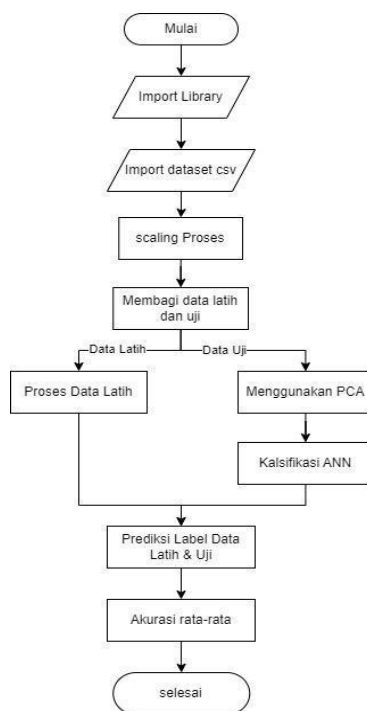
## 3 Metode Penelitian

### 3.1 Pengumpulan Data

Dataset yang digunakan pada penelitian ini sebanyak 14980 sampel yang terdiri dari 15 *feature*, di antaranya ada 14 atribut nilai real EEG dan 1 status keadaan mata (*eye detection*). Atribut nilai EEG terdiri atas 14 channel yaitu AF3, F7, F3, FC5, T7, P7, O1, O2, P8, T8, FC6, F4, F8, AF4 dan 1 status *eye detection* yang berisikan nilai integer 0 dan 1. 0 menandakan mata dalam keadaan terbuka dan 1 menandakan mata dalam keadaan tertutup. Seluruh dataset ini didapatkan dari platform Kaggle *Eye State Classification* EEG Dataset [11].

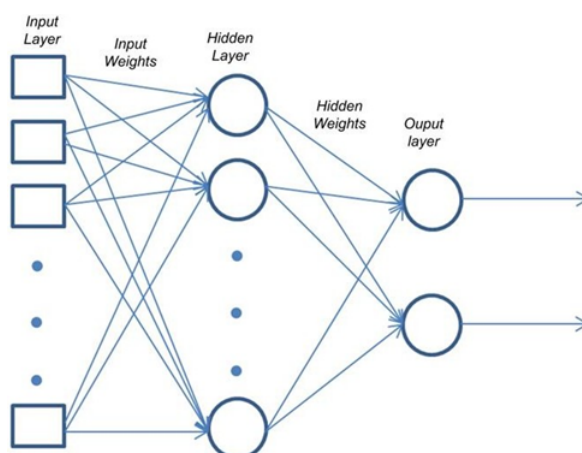
### 3.2 Pengolahan Data

Sistem klasifikasi mata terbuka dan mata tertutup ini dijalankan menggunakan bahasa pemrograman Python dengan format file .ipynb. Dengan demikian digunakan beberapa fitur tambahan dari python dalam pengklasifikasian sistem ini. Alur pengolahan data untuk sistem klasifikasi sinyal EEG menggunakan algoritma *Artificial Neural Network* dan *Principal Component Analysis* pada kondisi mata terbuka dan mata tertutup dapat dilihat pada Gambar 2.



**Gambar. 1.** Alur sistem klasifikasi

Implementasi sistem diawali dengan meng-import library yang dibutuhkan pada pemrosesan data, kemudian dilanjutkan dengan meng-iput dataset menggunakan format csv. Kemudian dilakukan proses *scaling* pada dataset yang sudah diinput ke dalam sistem. Proses dilanjutkan dengan pembagian dataset menjadi data latih dan data uji. Pada data latih dilakukan proses data latih, sedangkan pada data uji dilakukan uji sesungguhnya menggunakan algoritma ANN dan PCA. Setelah melalui pengujian data, dilakukan prediksi label data, yang dilanjutkan dengan perhitungan akurasi rata-rata. Gambar 2 adalah arsitektur dari ELM yang akan digunakan.



**Gambar. 2.** Penggambaran model dalam arsitektur *Neural Network*

Pada Gambar 2 terdapat tiga layer pada arsitektur neural network yaitu input layer, hidden layer dan output layer.

### 3.3 Parameter Performansi

Parameter performansi yang digunakan dalam perhitungan adalah akurasi sistem, *sensitivity*, dan *specificity* [12] dan waktu latih. Nilai akurasi yang tinggi menandakan performansi sistem yang baik. Sementara waktu latih yang rendah menunjukkan performa yang semakin baik. Untuk menghitung akurasi digunakan persamaan:

$$Akurasi = \frac{Jumlah\ klasifikasi\ yang\ benar}{Total\ data} * 100\% \quad (2)$$

Untuk menghitung *Sensitivity* digunakan persamaan: (kelas 1 adalah *True Positive*):

$$Sensitivity = \frac{True\ Positive}{True\ Positive + False\ Negative} * 100\% \quad (3)$$

Untuk menghitung *Specificity* digunakan persamaan: (kelas 1 adalah *True Negative*)

$$Specitivity = \frac{True\ Negative}{True\ Positive + False\ Negative} * 100\% \quad (4)$$

## 4 Hasil Penelitian

### 4.1 Persiapan *Library*

Persiapan *library* diawali dengan melakukan impor *package* pada baris kode python melalui *collab*. *Library* yang digunakan adalah *numpy*, *pandas*, dan *sklearn*. *Numpy* merupakan sebuah *library* untuk menyediakan perhitungan matematis pada program yang sedang dikembangkan, *library* ini bertindak sebagai *function mathematical*. *Pandas* adalah *library* yang berfungsi sebagai data manipulation dan analisis data yang berbentuk tabular, dalam penelitian ini digunakan data berupa file “eeg\_data.csv”. Sedangkan *sklearn* adalah *tools* paket yang berfungsi sebagai alat pendukung untuk melakukan perhitungan algoritma klasifikasi, regresi, *clustering*, dan *dimensionality reduction*. Beberapa fungsi secara spesifik dari *sklearn* diimpor seperti PCA (*Principal Component Analysis*) dan *MLPClassifier* (*a multi layer perceptron classifier*) yang merupakan salah satu algoritma *neural network*.

Selain ketiga *library* tadi, digunakan juga *StandardScaler*, yaitu sebuah alat bantu dari *sklearn* yang pada studi kasus ini digunakan untuk standarisasi fitur pada dataset. *train\_test\_split* adalah sebuah fungsi yang akan digunakan untuk membagi dataset menjadi data latih dan data , dan *classification\_report* sebuah fungsi untuk menunjukkan hasil rangkuman dari performa pada pengujian.

### 4.2 Persiapan Dataset

Setelah tahap persiapan *library* yang digunakan, tahap selanjutnya adalah dilakukan pemilihan data yang akan digunakan dalam pengujian. Data berupa file csv yang akan diimport ke dalam sistem dapat dilihat pada Tabel 1.

**Tabel 1.** Data sinyal EEG yang digunakan.

AF3	F7	F3	FC5	T7	P7	O1	O2	P8	T8	FC6	F4	F8	AF4	eye Detection
4329.	4009.2	4289.	4148.2	4350.		4096.	4641.0	4222.0	4238.4	4211.2	4280.5	4635.9	4393.	
23	3	23	1	26	4586.15	92	3	5	6	8	1	0	85	0
4324.	4004.6	4293.	4148.7	4342.		4097.	4638.9	4210.7	4226.6	4207.6	4279.4	4632.8	4384.	
62	2	85	2	05	4586.67	44	7	7	7	9	2	10	10	0
4327.	4006.6	4295.	4156.4	4336.		4096.	4630.2	4207.6	4222.0	4206.6	4282.0	4628.7	4389.	
69	7	38	1	92	4583.59	92	6	9	5	7	5	2	23	0
4328.	4011.7	4296.		4343.		4097.	4630.7	4217.4	4235.3	4210.7	4287.6	4632.3	4396.	
72	9	41	4155.9	59	4582.56	44	7	4	8	7	9	1	41	0
4326.	4011.7	4292.	4151.2	4347.		4095.	4627.6	4210.7	4244.1	4212.8	4288.2	4632.8	4398.	
15	9	31	8	69	4586.67	90	9	7	0	2	1	2	46	0

4321.	4004.6	4284.	4153.3	4345.		4093.	4616.9	4202.5	4232.8	4209.7	4281.0	4628.2	4389.	
03	2	1	3	64	4587.18	33	2	6	2	4	3	1	74	0
4319.	4001.0	4280.	4151.7	4343.		4089.		4212.3	4226.6	4201.0	4269.7	4625.1	4378.	
49	3	51	9	59	4584.62	74	4615.9	1	7	3	4	3	46	0
4325.	4006.6	4278.	4143.0	4344.		4087.	4614.8	4205.6	4230.2	4195.9	4266.6	4622.0	4380.	
64	7	46	8	10	4583.08	18	7	4	6	0	7	5	51	0
4326.	4010.7	4276.	4139.4	4345.		4091.	4608.2	4187.6	4229.7	4202.0	4273.8	4627.1	4389.	
15	7	41	9	13	4584.10	28	1	9	4	5	5	8	74	0

Dataset terdiri atas 15 *feature*, dengan 14 atribut nilai real EEG dan 1 status keadaan mata (*eye detection*). Atribut nilai real EEG tersebut adalah AF3, F7, F3, FC5, T7, P7, O1, O2, P8, T8, FC6, F4, F8, AF4 dan 1 status *eye detection* yang berisikan nilai integer 0 dan 1.

### 4.3 Proses *Scaling*

Proses *scaling* dilakukan untuk mendapatkan standarisasi data. Pada proses ini *StandardScaler* menggunakan  $x$  sebagai *mean* dan standar deviasi pada tiap fitur data, kemudian diaplikasikan pada  $X$  menggunakan transform yang mengembalikan data pada parameter  $x\_scaled$ . Standarisasi data ini bertujuan untuk memberikan hasil yang optimal pada data sebelum diolah melalui algoritma sehingga didapatkan data yang sempurna untuk diterapkan pada model. Kode program yang digunakan sebagai berikut:

```
# Scale the data using StandardScaler
Scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
```

Hasil yang didapatkan dari proses *scaling* berupa array data yang ditunjukkan pada Gambar 3.

**X\_scaled**

```
array([[ 0.00293429, -0.01170422,  0.56739848, ...,  0.03074472,
         0.01712667, -0.00383339 ],
       [ 0.00108436, -0.1120522 ,  0.6713903 , ...,  0.0061918 ,
         0.0145777 , -0.00548894],
       [ 0.00231631, -0.06742891,  0.70582915, ...,  0.06781481,
         0.01118458, -0.00461813],
       ...,
       [-0.0178483 , -0.41353148, -0.39058683, ..., -0.51230804,
        -0.01937827, -0.01306139],
       [-0.01496707, -0.39132867, -0.28682009, ..., -0.29012821,
        -0.01555481, -0.01114664],
       [-0.01373512, -0.26834254, -0.09054115, ..., -0.11729493,
        -0.01428032, -0.01114664]])
```

**Gambar. 3.** Hasil proses *scaling*

### 4.4 Pembagian Data Latih dan Data Uji

Dalam melakukan proses klasifikasi, data dibagi terlebih dahulu yakni data latih dan data uji. Data latih dipakai untuk membuat model klasifikasi, sedangkan data uji dipakai untuk menguji apakah model data latih sebelumnya dapat memberikan hasil yang baik dalam menganalisis data uji. Pembagian data ini berguna untuk mengetahui ketepatan metode dalam klasifikasi data. Presentase data latih dan data uji dapat dilihat pada Tabel 2.

**Tabel 2.** Pembagian data latih dan data uji

Keterangan	Data Latih	Data Uji	Total
Jumlah	11.984	2.996	14.980
Presentase	80%	20%	100%

Dari Tabel 2, total data dalam penelitian ini adalah 14.980 data, dengan komposisi 80% data latih sebanyak 11.984 data, dan 20% data uji sebanyak 2.996 data.

#### 4.5 Implementasi PCA

Pada tahap ini dilakukan implementasi PCA pada data dengan kode program berikut

```
# Use PCA to reduce the dimensionality of the data
pca = PCA(n_components = 0.95)
X_pca = pca.fit_transform(X_scaled)
```

Hasil yang diperoleh dari implementasi PCA dapat dilihat pada Gambar 4.

```
X_pca
array([[ 0.4178073 , -0.01306569, -0.28856727,  0.56742933, -0.43537135],
       [ 0.36421576, -0.07654914, -0.32675033,  0.31571214, -0.25580985],
       [ 0.32721677, -0.13621913, -0.35019806,  0.12514342, -0.10320847],
       ...,
       [-0.63609974, -0.11004379,  0.49994068, -0.53992691,  0.51492694],
       [-0.52682472, -0.10969886,  0.32483284, -0.26590071,  0.37905097],
       [-0.37329985, -0.19290387,  0.11959876,  0.09332011,  0.33594583]])
```

**Gambar. 4.** Hasil implementasi PCA

Dengan penggunaan PCA maka terjadi pengurangan dimensionality pada data `x_scaled`. Fungsi PCA pada kode tersebut adalah penggunaan `n_components` parameter, yang diset pada jumlah 0.95, artinya hasil dari `X_pca` array akan mengandung komponen dari 95% total dari variasi data.

#### 4.6 Pemodelan dan Klasifikasi Data

Pada tahap ini dilakukan pembuatan model, pengujian klasifikasi mata terbuka dan mata tertutup dan menghitung akurasi dari model yang dibuat. Tahap awal dilakukan dengan menginisialisasi ANN *classifier* menggunakan metode `MLPClassifier` dari `sklearn`. Kode program yang digunakan sebagai berikut:

```
# Initialize the ANN classifier with a larger maximum number of
iterations
clf = MLPClassifier(solver='lbfgs', alpha=1e-5, hidden_layer_sizes=(5,
2), random_state=1, max_iter=1000)
```

`MLPClassifier` merupakan sebuah parameter yang digunakan untuk mengkustomisasi *behaviour* pada model. Parameter `hidden_layer_size=(5, 2)` yang berarti pada setiap nomor secara spesifik pada hidden layer ANN, pada kasus ini artinya model memiliki dua hidden layers, dengan 5 dan 2 *neurons*. *Hidden layer size* 5 dan 2 terkait jumlah neuron dalam setiap *hidden layer*, yang pertama dengan 5 neuron dan yang kedua dengan 2 neuron. Pada kasus ini arsitektur ANN menggunakan *function* yang didefinisikan secara eksplisit dengan *hidden layer sizes*. Selain itu digunakan parameter `max_iter=1000` yang berarti mendefinisikan jumlah maksimum iterasi yang dijalankan. *Solver* yang digunakan adalah limited-memory Broyden–Fletcher–Goldfarb–Shanno `solver='lbfgs'` merupakan standar dari optimisasi ANN model.

Tahap berikutnya adalah melakukan *training* pada data latih menggunakan kode program berikut:

```
# Train the classifier on the training data
clf.fit(X_train, y_train)
# Print the number of epochs
print(f' Number of epochs: {clf.n_iter_}') # count by iteration epochs
```

Selanjutnya kode akan melakukan output pada hasil yang didapatkan pada perhitungan algoritma tersebut dengan jumlah hasil Number of epochs: 588 yang dapat dilihat pada Gambar 5.

```
# Print the number of epochs
print(f' Number of epochs: {clf.n_iter_}') # count by iteration epochs

Number of epochs: 588
```

**Gambar. 5.** Epoch berhenti pada angka 588

```
# Predict the labels for the test data
y_pred = clf.predict(X_test)
# Calculate the classification accuracy
accuracy = np.mean(y_pred == y_test)
print(f' Classification accuracy: {accuracy:.2f}')
# Print the classification report
print( classification_report(y_test, y_pred))
```

Terakhir baris kode tersebut akan memberikan hasil ringkasan output yang didapatkan dari perhitungan algoritma dengan *classification report* `print(classification_report(y_test, y_pred))` beserta hasil prediksinya keberhasilan akurasi sebesar 57% yang dapat dilihat pada Gambar 6.

```
Classification accuracy: 0.57
              precision    recall  f1-score   support

     0           0.57       0.71      0.64       1586
     1           0.56       0.40      0.47       1410

 accuracy                   0.57         2996
 macro avg                  0.57         2996
 weighted avg               0.57         2996
```

**Gambar. 6.** Report hasil klasifikasi

Pada Gambar 6 dapat dilihat bahwa untuk pengklasifikasian mata terbuka yang diinisialisasi dengan angka 0 mendapatkan presisi 57%, nilai *recall* 72%, *f1\_score* 64% dan nilai *support* 1586. Sedangkan untuk pengklasifikasian mata tertutup yang diinisialisasi dengan angka 0 mendapatkan presisi 56%, nilai *recall* 40%, *f1\_score* 46% dan nilai *support* 1410.

## 5 Kesimpulan

### 5.1 Kesimpulan

Kesimpulan yang dapat ditarik dari penelitian ini adalah:

1. Sistem klasifikasi sinyal EEG menggunakan ANN dan PCA pada pendeteksian mata terbuka dan mata tertutup mendapat akurasi sistem yang mencapai 57%. dengan nilai `random_state` 42

2. Untuk pengklasifikasian mata terbuka yang diinisialisasi dengan angka 1 mendapatkan presisi 57%, nilai recall 72%, f1\_score 64% dan nilai support 1586.
3. Untuk pengklasifikasian mata tertutup yang diinisialisasi dengan angka 0 mendapatkan presisi 56%, nilai recall 40%, f1\_score 46% dan nilai support 1410.

## 5.2 Saran

Pengembangan penelitian selanjutnya dapat dilakukan untuk mengurangi beban komputasi dengan menggunakan teknik reduksi dimensi dengan metode pemilihan variabel yang tepat untuk meningkatkan akurasi. Selain itu dapat menggabungkan metode pada penelitian ini dengan penelitian yang serupa antara lain penggunaan poligraf untuk mendeteksi tingkat kejujuran orang melalui *eyes state* gelombang sinyal EEG.

## Referensi

- [1] A. A. Saleh. (2018). *Buku Pengantar Psikologi*. Makassar: Penerbit Aksara Timur.
- [2] B. Sarasati dan O. Nurvia. (2021). Emosi Dalam Tulisan. vol. 14, no. 1, hlm. 40–48. doi: 10.30813/psibernetika.
- [3] A. Akbar dan A. Lawi. (2021). Implementasi Algoritma Deep Artificial Neural Network Menggunakan Mel Frequency Cepstrum Coefficient Untuk Klasifikasi Audio Emosi Manusia. *Konferensi Nasional Ilmu Komputer (KONIK) 2021*, [Daring]. Tersedia pada: <https://www.kaggle.com/ejlok1/toronto>.
- [4] A. J. P. L. Tobing, A. Bhawiyuga, dan D. P. Kartikasari. (2020). Implementasi Akuisisi Data Biosignal Manusia dengan Mekanisme Store-Forward Pada Perangkat Smartwatch Berbasis Wear Os Menggunakan Protokol Komunikasi Bluetooth Low Energy.
- [5] R. Anggara dan Y. Rahayu. (2020). Sistem Electroencephalogram (EEG) Untuk Analisis Sinyal Gelombang Otak Pada Pasien Depresi. *Jurnal Online Mahasiswa (JOM) Bidang Teknik dan Sains*.
- [6] E. C. Prakoso, U. N. Wisesty, dan . J. (2016). Klasifikasi Keadaan Mata Berdasarkan sinyal EEG menggunakan Extreme Learning Machines. *Indonesian Journal on Computing (Indo-JC)*, vol. 1, no. 2, hlm. 97. doi: 10.21108/INDOJC.2016.1.2.105.
- [7] W. Setiawan. (2020). Deep Learning Menggunakan Convolutional Neural Network. Malang: MNC.
- [8] E. S. Noviando, E. Ervianto, dan I. Yasri. (2016). Studi Penerapan ANN (Artificial Neural Network) Untuk Menghilangkan Harmonisa Pada Gedung Pusat Komputer.
- [9] A. Darmawan. (2019). Aplikasi Mobile Pengenalan Wajah Secara Real-Time Berbasis Principal Component Analysis. *Ubiquitous: Computers and its Applications Journal*, vol. 2, no. 1, hlm. 57–66.
- [10] Z. F. M. Ramli, S. T. M. . I. Wijayanto, dan S. T. M. . S. Hadiyoso. Deteksi Kondisi Konsentrasi Berdasarkan Sinyal Eeg Dengan Stimulasi Menghafal Al-Quran. *e-Proceeding Eng*, vol. 5, no. 3, hlm. 4683–4690.
- [11] O. Roesler. (2022). Kaggle Eye State Classification EEG Dataset.
- [12] D. Hindarto, R. Eko Indrajit, dan E. Dazki. (2022). Perbandingan Kinerja Akurasi Klasifikasi K-NN, NB Dan DT Pada APK Android. vol. 9, no. 1, hlm. 486–503.