

# Detection of Epilepsy Through EEG Signals Using the DWT and Extreme Gradient Boosting Methods

## [Deteksi Penyakit Epilepsi Melalui Sinyal EEG Menggunakan Metode DWT dan Extreme Gradient Boosting]

Erlina Agustin<sup>1)</sup>, Ade Eviyanti<sup>2)</sup>, Nuril Lutvi Azizah<sup>3)</sup>

<sup>1,2,3)</sup>Program Studi Informatika, Universitas Muhammadiyah Sidoarjo, Indonesia

\*Email Penulis Korespondensi: 191080200167@umsida.ac.id

**Abstract.** Epilepsy is a disorder of the central nervous system due to excessive patterns of electrical activity in the brain. This disease causes patients to experience repeated seizures in one or all parts of the body. Therefore, epilepsy must be detected early so that the patient immediately gets the right treatment so that the condition does not get worse. This study proposes the detection of epilepsy using the Discrete Wavelet Transform method for feature extraction and Extreme Gradient Boosting for classification. Detection results are classified into two classes, namely seizures and non-seizures. The EEG recording data used came from CHIB MIT Hospital Boston which was obtained online. This study uses four comparisons of the percentage of training data and test data as well as tuning parameters which are processed by Randomized Search Cross Validation. The combination of these methods produces the highest accuracy, namely an accuracy value of 81%.

**Keywords** - Discrete Wavelet Transform; EEG; Epilepsy; Extreme Gradient Boosting, Classification

**Abstrak.** Epilepsi adalah gangguan pada sistem saraf pusat akibat pola aktivitas listrik yang berlebihan di otak. Penyakit ini menyebabkan penderitanya mengalami kejang berulang pada satu atau seluruh bagian tubuh. Oleh karena itu, epilepsi harus dideteksi sejak dini agar penderita segera mendapatkan penanganan yang tepat agar kondisinya tidak semakin parah. Penelitian ini mengusulkan deteksi epilepsi menggunakan metode Discrete Wavelet Transform untuk ekstraksi ciri dan Extreme Gradient Boosting untuk klasifikasi. Hasil deteksi diklasifikasikan menjadi dua kelas, yaitu kejang dan non kejang. Data rekaman EEG yang digunakan berasal dari CHIB MIT Hospital Boston yang diperoleh secara online. Penelitian ini menggunakan empat perbandingan persentase data latih dan data uji serta parameter tuning yang diolah dengan Randomized Search Cross Validation. Kombinasi metode tersebut menghasilkan akurasi tertinggi yaitu nilai akurasi sebesar 81%.

**Kata Kunci** - Discrete Wavelet Transform; EEG; Epilepsi; Extreme Gradient Boosting; Klasifikasi

## I. PENDAHULUAN

Epilepsi merupakan salah satu penyakit kronik dengan angka kejadian tinggi khususnya di negara berkembang. Penyakit ini dapat mengganggu kualitas hidup dan membutuhkan biaya cukup banyak [1]. Pada seseorang yang mengidap epilepsi, kejang dapat terjadi lebih dari satu kali atau berulang-ulang di waktu yang sama atau berbeda bahkan bisa menyebabkan kejang saat tidur. Hal ini terjadi karena adanya perubahan fase tubuh dari bangun tidur yang memicu aktivitas tidak normal pada otak [2]. Ciri utama yang tampak adalah penderita mengalami kejang dan kehilangan kesadaran. Penyebab secara pasti dari kelainan ini belum diketahui, namun sebagian besar dari para penderita memiliki riwayat epilepsi dari keluarga sehingga dapat disebut dengan penyakit akibat genetik. Selain itu epilepsi juga bisa disebabkan oleh penyakit lain yang menyerang otak. Penyakit ini dapat terjadi pada manusia di segala umur. Namun, gejalanya sangat terlihat pada anak-anak dan lansia di atas 65 tahun. Penyakit epilepsi secara medis tidak dapat disembuhkan. Obat-obatan hanya mampu mengontrol frekuensi dan tingkat keparahan [3]. Namun perlu diwaspadai, pada tingkat yang parah epilepsi dapat mengakibatkan kematian bagi penderitanya. Oleh karena itu penyakit epilepsi harus dideteksi sejak dini dan ditangani secara tepat agar penderita dapat menjalani kehidupan manusia secara normal.

Seiring dengan adanya perkembangan teknologi, saat ini manusia dimudahkan dengan segala hal salah satunya dalam dunia kedokteran. Pengaplikasian teknologi yang sangat berguna salah satunya adalah alat EEG. *Electroencephalogram* (EEG) merupakan alat yang dapat digunakan untuk memantau aktivitas listrik yang terjadi pada otak melalui beberapa elektroda yang diletakkan di kulit kepala [4]. EEG mengukur fluktuasi tegangan yang dihasilkan dari arus ionik yang beredar sepanjang neuron otak. Sinyal EEG telah digunakan secara luas di banyak bidang, seperti ilmu saraf, penelitian psikofisiologis, neurolinguistik dan banyak lagi [5]. EEG telah dieksplorasi dalam studi klinis termasuk gangguan otak, gangguan tidur, epilepsi dan disfungsi neurologis lainnya. Dalam pengaturan

eksperimental, sinyal EEG yang direkam memiliki bentuk yang kompleks, amplitude kecil dan tidak mempunyai pola yang baku. Oleh karena itu diperlukan metode untuk melakukan ekstraksi fitur.

DWT (*Discrete Wavelet Transform*) merupakan salah satu metode yang dapat digunakan untuk melakukan ekstraksi fitur pada sinyal EEG. Wavelet memiliki kemampuan untuk menganalisis sinyal satu dimensi dan multidimensi, terutama ketika sinyal tersebut mengandung informasi yang berbeda pada setiap titik waktu. Representasi wavelet adalah dekomposisi sinyal multi-skala di mana wavelet mengubah nilai menjadi seperangkat koefisien tertentu [6].

Untuk melakukan klasifikasi sinyal, maka diperlukan algoritma *machine learning* yang dianggap sebagai salah satu fitur kecerdasan buatan terpenting yang mendukung pengembangan sistem komputer [7]. Algoritma XGBoost merupakan salah satu algoritma yang paling populer dan paling banyak digunakan karena algoritma ini termasuk algoritma yang powerful. Pada dasarnya, algoritma ini sama dengan algoritma gradient boost hanya saja menggunakan beberapa proses tambahan sehingga lebih powerful. Proses tersebut adalah pemangkasan, newton boosting, dan parameter pengacakan ekstra. Proses pemangkasan atau penyusutan proporsional simpul daun digunakan untuk meningkatkan generalisasi model. proses newton boosting adalah proses untuk menyediakan rute langsung sehingga tidak memerlukan penurunan gradient. Proses pengacakan parameter bertujuan untuk mengurangi korelasi antar tree sehingga dapat meningkatkan kekuatan algoritma ensemble [8].

Beberapa penelitian mengenai deteksi penyakit epilepsi melalui sinyal EEG sudah pernah dilakukan sebelumnya. Diantaranya yaitu penelitian yang dilakukan oleh Ade Eviyanti, Hindarto hindarto, Sumarno dan Herlian Aliyasa Alamj Duddin dengan judul “Epilepsy detection system based on EEG record using neural network backpropagation method” pada tahun 2019. Penelitian ini menggunakan Discrete Cosinus Transform (DCT) untuk melakukan ekstraksi fitur dari sinyal EEG. Lalu untuk melakukan identifikasi sinyal EEG, penelitian ini menggunakan Backpropagation Neural Network [9].

Penelitian yang dilakukan Mulaab dengan judul “Deteksi Kejang *Epilepsy* Dengan Menggunakan Pemilihan Fitur *Information Gain* Dan Pembelajaran *Ensemble Random Forest*” pada tahun 2021. Dataset yang digunakan dalam penelitian ini diperoleh dari CHB-MIT Scalp EEG Database. Tujuan dari penelitian ini adalah untuk melakukan deteksi epilepsi dengan menggunakan *information gain* untuk ekstraksi fitur dan *ensemble random forest* untuk melakukan klasifikasi. Gabungan dari dua metode tersebut berhasil mendapatkan nilai akurasi sebesar 99% [10].

Penelitian yang dilakukan Ade Eviyanti, Hindarto hindarto, M Abror pada tahun 2020 dengan judul “Peningkatan Ekstraksi Ciri Sinyal Epilepsi Menggunakan Teknik Sampling”. Metode yang digunakan pada penelitian ini yaitu metode Teknik sampling yang digunakan untuk menentukan ciri sinyal. Selanjutnya untuk klasifikasi menggunakan metode KNN. Data yang digunakan adalah data sinyal EEG yang terdiri dari lima set data dan berasal dari data publik. Pada saat proses Teknik sampling, nilai yang diambil adalah nilai rata-rata dan standart deviasi. Penelitian ini menghasilkan tingkat akurasi sebesar 100% [11].

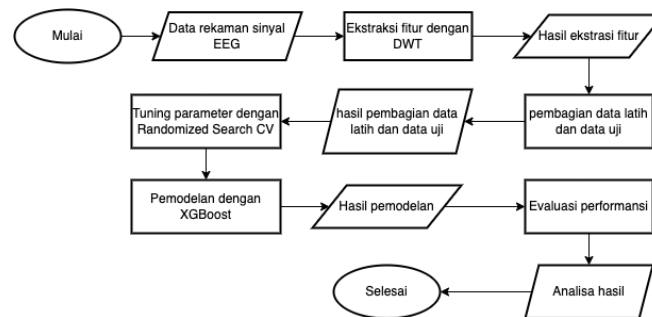
Penelitian yang dilakukan oleh Siswandari Noertjahjani dengan judul “Analisis Tipe Wavelet Coiflets 1 Dan Coiflets 5 Untuk Deteksi Penyakit Epilepsi” pada tahun 2022. Penelitian ini bertujuan untuk membedakan pola sinyal EEG epilepsi dan normal menggunakan mother wavelet coiflets 1 dan coiflets 5 dengan sinyal EEG pada elektroda F3 dan Fz. Hasil korelasi terbesar sinyal aprosimasinya mendekati 85% dengan menggunakan coiflets 1 dan menunjukkan bahwa sistem prediksi epilepsi berbasis SVM yang diusulkan dapat mencapai prediksi yang kuat dengan akurasi 90% dan sensitivitasnya sekitar 91% [12]. Adapun penelitian yang dilakukan oleh Yuna Sugianela, Qonita Luthfia Sutino, and Darlis Herumurti dengan judul “EEG Classification For Epilepsy Based On Wavelet Packet Decomposition And Random Forest” pada tahun 2018. Pada penelitian ini peneliti melakukan ekstraksi fitur menggunakan DWT dengan dekomposisi sebanyak 5 level. Kemudian peneliti melakukan klasifikasi menggunakan random forest [13].

Berdasarkan latar belakang dan penjelasan yang sudah dijabarkan di atas, sehingga penelitian ini akan melakukan deteksi penyakit epilepsi melalui sinyal EEG menggunakan metode XGBoost. Harapan dari penelitian ini yaitu dapat bermanfaat bagi komunitas medis dapat dengan mudah mendiagnosa apakah seseorang menderita epilepsia tau tidak. Karena semakin dini penyakit itu diketahui, semakin cepat pula diobati agar penderita dapat hidup seperti orang normal.

## II. METODE

### A. Tahapan Penelitian

Tahapan penelitian merupakan gambaran umum terkait alur penelitian yang akan dilakukan dalam pengerjaan penelitian ini dari awal hingga akhir. Tahapan yang dilakukan dalam penelitian ini dapat dipaparkan melalui diagram alir seperti pada Gambar 1. Penelitian ini akan dikerjakan menggunakan Bahasa pemrograman python.



Gambar 1 Gambaran Tahapan Penelitian

## B. Data

Data yang digunakan pada penelitian ini adalah data rekaman sinyal multi channel EEG dari pasien epilepsi dan normal yang diunduh dari website milik CHB MIT Hospital Boston. Data yang digunakan terdiri dari 21 channel dan 25590 rekaman. Adapun ke-21 channel tersebut yaitu Fp1, Fp2, F3, F4, C3, C4, P3, P4, O1, O2, F7, F8, T3, T4, T5, T6, Cz, Fz, Pz, A1, A2. Sebelum dilakukan klasifikasi, dilakukan ekstraksi fitur terlebih dahulu menggunakan metode DWT.

## C. Ekstraksi fitur dengan DWT

DWT digunakan untuk dekomposisi dengan teknik penyaringan. Sinyal melewati filter frekuensi dan skala yang berbeda. Ada dua jenis filter yaitu filter highpass dan filter lowpass. Persamaan untuk kedua filter dapat ditemukan pada persamaan 2.3 dan 2.4 di bawah ini.

$$Y_{high}[k] = \sum_n X[n]h[2k - n] \quad (1)$$

$$Y_{low}[k] = \sum_n X[n]g[2k - n] \quad (2)$$

Dimana  $Y_{high}$  merupakan hasil dari *highpass filter* atau disebut dengan Detail (D) dan  $Y_{low}$  merupakan hasil dari *lowpass filter* atau *Approximation* (A). proses dekomposisi ini diulang berkali-kali pada *approximation* sesuai dengan jumlah level yang dikehendaki dan menghasilkan D dan A baru. Pada penelitian ini data sinyal dibagi menjadi beberapa bagian kemudian setiap bagian akan didekomposisi menggunakan Discrete Wavelet Transform dengan jenis mother wavelet yang digunakan adalah Daubechies 4 dengan 5 tingkat dekomposisi. Hasil dekomposisi akan menghasilkan fitur yang diperoleh dari perhitungan nilai  $cD\_Energy$ ,  $cA\_Energy$ ,  $D\_Entropy$ ,  $A\_Entropy$ ,  $D\_mean$ ,  $A\_mean$ ,  $D\_std$ , dan  $A\_std$ . sehingga pada setiap channel menghasilkan 8 fitur. Seluruh fitur dijadikan dalam sebuah dataset. Maka didapatkan jumlah fitur yaitu  $8 \times 21$  sehingga totalnya yaitu 168 fitur.

## D. Klasifikasi dengan Algoritma XGBoost

Dataset hasil ekstraksi fitur selanjutnya dilakukan pembagian data latih dan data uji dengan prosentase sebagai berikut.

Tabel 1. Pembagian data latih dan data uji

Data Latih	Data Uji
90%	10%
80%	20%
75%	25%
70%	30%

Data latih digunakan untuk membangun model klasifikasi sedangkan data uji digunakan untuk menguji model. Proses pemodelan dilakukan dengan menggunakan metode Extreme Gradient Boosting (XGBoost). XGBoost merupakan salah satu teknik pembelajaran mesin untuk mengatasi masalah regresi dan klasifikasi berdasarkan Gradient Boosting Decision Tree (GBDT) [14]. Metode XGBoost merupakan pengembangan dari gradient boosting yang diusulkan oleh Dr. Tianqi Chen dari University of Washington pada tahun 2014 [15]. Konsep dasar dari algoritma ini adalah menyesuaikan parameter pembelajaran secara berulang untuk menurunkan loss function (mekanisme evaluasi atas model). XGBoost menggunakan model yang lebih teratur untuk membangun struktur pohon regresi, sehingga dapat memberikan kinerja yang lebih baik dan mampu mengurangi kompleksitas model untuk menghindari overfitting [16].

### E. Hyperparameter tuning dengan Randomized Search Cross Validation

pada metode machine learning, terdapat beberapa nilai parameter yang diperkirakan dapat meningkatkan kinerja model yang disebut dengan hyperparameter. Metode hyperparameter yang akan diaplikasikan adalah Randomized Search Cross Validation. Metode ini fungsinya sama dengan Grid Search CV. Namun bedanya adalah Randomized Search CV dapat digunakan untuk mencari parameter terbaik dalam waktu yang lebih cepat dibandingkan Grid Search CV. Teknik ini sangat berguna ketika parameter dan data yang dimiliki memiliki jumlah yang banyak [17]. Karena data pada penelitian ini memiliki 25590 rekaman, maka peneliti akan menggunakan Teknik Randomized Search CV untuk melakukan tuning parameter. Adapun parameter yang akan dilakukan tuning adalah sebagai berikut.

Tabel 2. Hyperparameter

Hyperparameter	Kegunaan
max_depth	Tingkat kedalaman pohon
learning_rate	Untuk mencegah model mengalami overfitting
n_estimators	Banyaknya pohon yang digunakan untuk proses klasifikasi
subsample	Rasio instance dari data latih
gamma	Menentukan pemangkasan dari node pada pohon yang dibuat. semakin besar gamma semakin konservatif model yang dibangun
colsample_bytree	Parameter untuk memilih banyak sample kolom yang akan digunakan
reg_alpha	Istilah regularisasi L1 pada bobot. Meningkatkan nilai ini akan membuat model lebih konservatif.
reg_lambda	Istilah regularisasi L2 pada bobot. Meningkatkan nilai ini akan membuat model lebih konservatif.

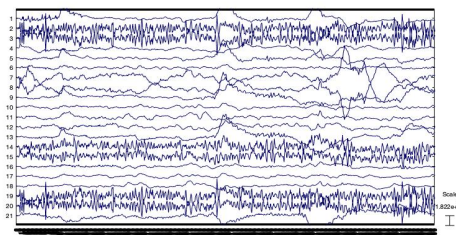
### F. Evaluasi

Tahapan ini digunakan untuk mengukur performa dari model *machine learning* yang telah dibuat. pada tahapan evaluasi, peneliti menggunakan *confusion matrix* untuk menganalisa performa model yang telah dibuat. *confusion matrix* merupakan pengujian performa untuk klasifikasi machine learning dimana hasil dari *confusion matrix* tersebut berupa dua kelas atau lebih. *Confusion matrix* merupakan tabel dengan 4 kombinasi berbeda dari nilai prediksi dan nilai aktual [18]. Selanjutnya dari *confusion matrix* tersebut dapat dihitung nilai akurasi, precision, recall dan F1 Score. Peneliti juga menggunakan Mean Absolute Error untuk mengetahui prosentase kesalahan rata-rata mutlak [19]. Mean Absolute Error merupakan dua diantara banyak metode yang digunakan untuk mengukur tingkat keakuratan suatu model peramalan [20].

## III. HASIL DAN PEMBAHASAN

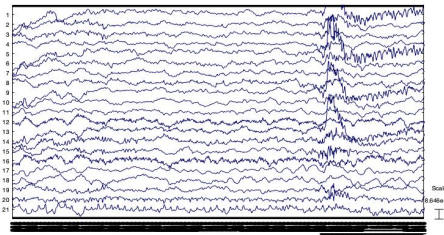
### A. Data Understanding

Proses yang dilakukan dalam tahap pemahaman data adalah pengumpulan dan pemahaman data. Data masukan dari penelitian ini berupa data hasil rekaman EEG pada pasien epilepsi dan orang normal yang didapatkan dari CHB MIT Hospital Boston dengan 12795 rekaman untuk data pasien epilepsi dan 12795 rekaman untuk data normal. Berikut merupakan hasil visualisasi sinyal penderita epilepsi dan orang normal.



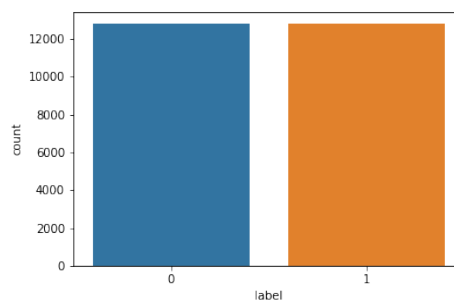
Gambar 2 Sinyal EEG Pasien Epilepsi

Hasil rekaman EEG dengan pola yang tidak normal menunjukkan adanya kelainan atau kondisi medis tertentu. Seperti pada gambar 2 yang menunjukkan pola sinyal yang lebih rapat dibandingkan gambar 3 yaitu sinyal EEG orang normal.



Gambar 3 Sinyal EEG orang normal

Adapun hasil rekaman EEG pada umumnya menunjukkan pola garis gelombang yang normal pula. Pada penelitian visualisasi sinyal EEG orang normal ditunjukkan pada gambar 3.



Gambar 4 Perbandingan jumlah rekaman

Dari visualisasi data pada gambar 4 dapat diketahui bahwa persebaran data seimbang antara data non kejang yang ditandai dengan label 0 dan data kejang yang ditandai dengan label 1. Data non kejang terdiri dari 12000 lebih rekaman dan data kejang memiliki jumlah rekaman yang sama yaitu terdiri 12000 rekaman.

## B. Ekstraksi Fitur

Data masukan yang telah dijelaskan pada sub bab 3.1 selanjutnya dilakukan ekstraksi fitur menggunakan metode DWT dengan jenis mother wavelet yang digunakan adalah Daubechies 4 dengan 5 tingkat dekomposisi. Hasil dekomposisi akan menghasilkan fitur yang diperoleh dari perhitungan nilai  $cD\_Energy$ ,  $cA\_Energy$ ,  $D\_Entropy$ ,  $A\_Entropy$ ,  $D\_mean$ ,  $A\_mean$ ,  $D\_std$ , dan  $A\_std$ . sehingga pada setiap channel menghasilkan 8 fitur. Kode sumber di bawah ini adalah fungsi yang akan digunakan untuk ekstraksi fitur

### Kode Sumber 1 fungsi ekstraksi fitur

```
def wavelet_avg_features(data,type_wav):

    coeffs = wavedec(data, type_wav,level=5)
    features = []
    cD_Energy = np.mean( [np.sum( np.square( coeffs[5] ) ),np.sum( np.square( coeffs[4] ) ),
                        np.sum( np.square( coeffs[3] ) ),np.sum( np.square( coeffs[2] ) ),
                        np.sum( np.square( coeffs[1] ) )] )
    cA_Energy = np.sum(np.square( coeffs[0] ) )
    D_Entropy = np.mean( [np.sum( np.square( coeffs[5] ) * np.log( np.square( coeffs[5] ) ) ),
                        np.sum( np.square( coeffs[4] ) * np.log( np.square( coeffs[4] ) ) ),
                        np.sum( np.square( coeffs[3] ) * np.log( np.square( coeffs[3] ) ) ),
                        np.sum( np.square( coeffs[2] ) * np.log( np.square( coeffs[2] ) ) ),
                        np.sum( np.square( coeffs[1] ) * np.log( np.square( coeffs[1] ) ) )] )

    A_Entropy = np.sum( np.square( coeffs[0] ) * np.log( np.square( coeffs[0] ) ) )
```

```

D_mean = np.mean( [np.mean( coeffs[5] ), np.mean( coeffs[4] ), np.mean( coeffs[3] ), np.mean( coeffs[2] ),
np.mean( coeffs[1] ) ] )
A_mean = np.mean( coeffs[0] )
D_std = np.mean( [np.std( coeffs[5] ), np.std( coeffs[4] ), np.std( coeffs[3] ), np.std( coeffs[2] ),
np.std( coeffs[1] ) ] )
A_std = np.std(coeffs[0])
features = [cD_Energy, cA_Energy, D_Entropy, A_Entropy, D_mean, A_mean, D_std, A_std]
return features

```

Kode Sumber 2 Proses Ekstraksi Fitur

```

chanel = []
df2 = pd.DataFrame()
df4 = pd.DataFrame()
df3 = pd.DataFrame()
ch = ['Fp1', 'Fp2', 'F3', 'F4', 'C3', 'C4', 'P3', 'P4', 'O1', 'O2', 'F7', 'F8',
      'T3', 'T4', 'T5', 'T6', 'Cz', 'Fz', 'Pz', 'A1', 'A2']
dict1 = dict()
dict2 = dict()
v = 0
df5 = pd.DataFrame()
df3 = pd.DataFrame()
data = pd.read_csv('data1.csv', names=cols)
k = int(data.shape[1]/21)
split_points = [i for i in range(0, data.shape[1], k)]
parts=[]
parts = [data.iloc[:, ind:ind + k] for ind in split_points]
for c in range(0,21):
    df = pd.DataFrame(columns=[ch[c]+'_cD_Energy', ch[c]+'_cA_Energy',
                              ch[c]+'_D_Entropy', ch[c]+'_A_Entropy',
                              ch[c]+'_D_mean', ch[c]+'_A_mean',
                              ch[c]+'_D_std', ch[c]+'_A_std'])
    # namefile = ch[c]
    chanel_data = parts[c]
    for i in range(0,25590):
        features = wavelet_avg_features(chanel_data.loc[i], 'db4')
        a_series = pd.Series(features, index = df.columns)
        df = df.append(a_series, ignore_index=True)
    df3 = pd.concat([df, df3], axis=1)
df4 = df4.append(df3, ignore_index=True)
df5 = df5.append(df4, ignore_index=True)
print(df5.shape)
save_to_csv(df5, 'data_extracted')

```

Function pada kode sumber 1 selanjutnya dimasukkan ke dalam kode sumber 2 dengan dilakukan looping sebanyak panjang dataset sehingga terbentuk dataset baru hasil ekstraksi fitur. Maka didapatkan jumlah fitur yaitu 8 x 21 sehingga totalnya yaitu 168 fitur. keluaran dari tahapan ini adalah dataset baru yang siap untuk dijadikan masukan pada tahap selanjutnya yaitu klasifikasi dengan XGBoost. Berikut merupakan contoh hasil ekstraksi fitur dengan value dari channel awal adalah 0,2715.

Tabel 3. Hasil Ekstraksi Fitur

Setelah ekstraksi fitur							
cD_Energy	cA_Energy	D_Entropy	A_Entropy	D_mean	A_mean	D_std	A_std
-0,1683	-0,2444	0,1786	-0,2270	-0,0008	-0,0053	-0,5988	-0,1869

### C. Dataset Splitting

Dataset hasil keluaran dari tahapan ekstraksi fitur selanjutnya dilakukan pembagian dataset menjadi data latih dan data uji. Data latih digunakan untuk melatih model sedangkan data uji digunakan untuk mengevaluasi model. Pada tahapan

dataset splitting ini, peneliti menggunakan fungsi dari library scikit learn yaitu `train_test_split`. Berikut merupakan kode sumber dari pembagian data.

**Kode Sumber 3** Pembagian data latih 90% dan data uji 10%

```
from sklearn.model_selection import train_test_split
X_train1, X_test1, y_train1, y_test1 = train_test_split(X, y, test_size = 0.1, stratify = y, random_state = 42)
```

**Kode Sumber 4** Pembagian data latih 80% dan data uji 20%

```
from sklearn.model_selection import train_test_split
X_train2, X_test2, y_train2, y_test2 = train_test_split(X, y, test_size = 0.2, stratify = y, random_state = 42)
```

*Kode Sumber 5* Pembagian data latih 75% dan data uji 25%

```
from sklearn.model_selection import train_test_split
X_train3, X_test3, y_train3, y_test3 = train_test_split(X, y, test_size = 0.25, stratify = y, random_state = 42)
```

*Kode Sumber 6* Pembagian data latih 70% dan data uji 30%

```
from sklearn.model_selection import train_test_split
X_train4, X_test4, y_train4, y_test4 = train_test_split(X, y, test_size = 0.30, stratify = y, random_state = 42)
```

Dari kode sumber di atas didapatkan jumlah data yang telah dibagi menjadi data latih dan data uji. Berikut merupakan hasil pembagian data.

*Tabel 4. Hasil Pembagian Data Latih dan Data Uji*

Prosentase data latih	Prosentase data uji	Jumlah rekaman data latih		Jumlah rekaman data uji	
		X train	y train	X test	y test
90%	10%	23031	23031	2559	2559
80%	20%	20472	5118	5118	5118
75%	25%	19192	6398	5118	6398
70%	30%	17913	7677	7677	7677

#### D. Klasifikasi dengan XGBoost

Setelah melalui tahapan pembagian data latih dan data uji, selanjutnya dilakukan tahapan modeling menggunakan algoritma XGBoost dengan parameter tuning yang disediakan oleh library `jcopml` dengan cara mengimport fungsi `random_search_params`. Berikut merupakan parameter tuning yang akan digunakan.

*Tabel 5. Hyperparameter Tuning*

Hyperparameter	Value
<code>max_depth</code>	Integer(low=1, high=10)
<code>learning_rate</code>	Real(low=-2, high=0, prior='log-uniform')
<code>n_estimators</code>	Integer(low=100, high=200)
<code>subsample</code>	Real(low=0.3, high=0.8, prior='uniform')
<code>gamma</code>	Integer(low=1, high=10)
<code>colsample_bytree</code>	Real(low=0.1, high=1, prior='uniform')
<code>reg_alpha</code>	Real(low=-3, high=1, prior='log-uniform')
<code>reg_lambda</code>	Real(low=-3, high=1, prior='log-uniform')

Berdasarkan parameter yang ada pada tabel 6 selanjutnya parameter tersebut dimasukkan ke dalam fungsi `Randomized Search Cross Validation` dengan `cross validation` adalah 5. `Randomized Search CV` merupakan fungsi dari library `scikit learn` yang berguna untuk mencari hyperparameter terbaik dari rangkaian parameter yang diberikan secara acak. Berikut merupakan kode sumber dari proses tuning dan modelling.

*Kode Sumber 7 Proses Tuning dan Modelling*

```
from sklearn.pipeline import Pipeline
from sklearn.impute import SimpleImputer
from sklearn.preprocessing import MinMaxScaler
from sklearn.compose import ColumnTransformer
from sklearn.model_selection import RandomizedSearchCV
from jcopml.tuning import random_search_params as rsp

numerical_pipeline = Pipeline([
    ('scaler', MinMaxScaler())
])
preprocessor = ColumnTransformer([
    ('numeric', numerical_pipeline, X_train1.columns)
])
pipeline = Pipeline([
    ('prep', preprocessor),
    ('algo', classifier)
])
model1_minmax = RandomizedSearchCV(pipeline, rsp.xgb_params, cv=5, n_jobs=-1, verbose=1)
model1_minmax.fit(X_train1, y_train1)
print(model1_minmax.score(X_train1,
y_train1),model1_minmax.best_score_,model1_minmax.score(X_test1, y_test1))

model2_minmax = RandomizedSearchCV(pipeline, rsp.xgb_params, cv=5, n_jobs=-1, verbose=1)
model2_minmax.fit(X_train2, y_train2)
print(model2_minmax.score(X_train2,
y_train2),model2_minmax.best_score_,model2_minmax.score(X_test2, y_test2))

model3_minmax = RandomizedSearchCV(pipeline, rsp.xgb_params, cv=5, n_jobs=-1, verbose=1)
model3_minmax.fit(X_train3, y_train3)
print(model3_minmax.score(X_train3,
y_train3),model3_minmax.best_score_,model1_minmax.score(X_test3, y_test3))

model4_minmax = RandomizedSearchCV(pipeline, rsp.xgb_params, cv=5, n_jobs=-1, verbose=1)
model4_minmax.fit(X_train4, y_train4)
print(model4_minmax.score(X_train4,
y_train4),model4_minmax.best_score_,model4_minmax.score(X_test4, y_test4))
```

**Tabel 6.** Hasil Tuning dan Modelling

Data Latih	Data Uji	Parameter Tuning		Skor Latih	Skor Uji
		Parameter	Value		
90%	10%	max_depth	6	88.65%	81.20%
		learning_rate	0.16		
		n_estimators	150		
		subsample	0.50		
		gamma	9		
		colsample_bytree	0.36		
		reg_alpha	0.06		
		reg_lambda	0.07		
80%	20%	max_depth	9	98.69%	85.15%
		learning_rate	0.21		
		n_estimators	106		
		subsample	0.67		
		gamma	4		
		colsample_bytree	0.99		
		reg_alpha	1.39		
		reg_lambda	0.004		
75%	25%	max_depth	6	92.40%	80.99%
		learning_rate	0.43		
		n_estimators	198		
		subsample	0.52		
		gamma	8		
		colsample_bytree	0.64		
		reg_alpha	0.21		
		reg_lambda	6.33		
70%	30%	max_depth	7	94.47%	82.93%
		learning_rate	0.07		
		n_estimators	171		
		subsample	0.69		
		gamma	5		
		colsample_bytree	0.72		
		reg_alpha	0.34		
		reg_lambda	0.003		

Berdasarkan tabel di atas dapat diketahui bahwa terdapat skor uji yang paling tinggi adalah 85,15% yang dihasilkan oleh prosentase data latih sebesar 80% dan data uji sebesar 20%. Namun dari hasil tersebut mengalami overfitting dengan gap antara skor uji dan skor latih yang cukup tinggi yaitu 13,54%. Secara keseluruhan hasil yang memiliki gap antara skor uji dan skor latih paling sedikit adalah skor uji 81,20% yang dihasilkan oleh prosentase data latih 90% dan data uji 10%.

#### E. Evaluasi

Setelah melalui tahapan klasifikasi, selanjutnya dilakukan evaluasi model yang bertujuan untuk mengukur kinerja dari model yang telah dibuat. peneliti menggunakan confusion matrix atau juga biasa disebut dengan error matrix. Pada dasarnya confusion matrix memberikan informasi perbandingan hasil klasifikasi yang dilakukan oleh model dengan hasil klasifikasi yang sebenarnya. Confusion matrix berbentuk tabel matriks yang menggambarkan kinerja model klasifikasi pada serangkaian data uji yang nilai sebenarnya diketahui. Berikut merupakan tabel confusion matrix dengan 4 kombinasi nilai prediksi dan nilai aktual yang berbeda.

*Tabel 7. Confusion Matrix*

	Positif	Negatif
Positif	True Positif (TP)	False Positif (FP)
Negatif	False Negatif (FN)	True Negatif (TN)

Terdapat 4 istilah yang direpresentasikan sebagai hasil proses klasifikasi pada confusion matrix. Keempat istilah tersebut adalah True Positive (TP), True Negative (TN), False Positive (FP) dan False Negative (FN). Agar lebih mudah memahaminya, peneliti menggunakan contoh kasus sederhana untuk memprediksi seorang pasien menderita epilepsi atau tidak.

- **True Positive (TP)**

Merupakan data positif yang diprediksi benar. Contohnya, pasien menderita epilepsi (class 1) dan dari model yang dibuat memprediksi pasien tersebut menderita epilepsi (class 1).

- **True Negative (TN)**

Merupakan data negatif yang diprediksi benar. Contohnya, pasien tidak menderita epilepsi (class 2) dan dari model yang dibuat memprediksi pasien tersebut tidak menderita epilepsi (class 2).

- **False Postive (FP) – Type I Error**

Merupakan data negatif namun diprediksi sebagai data positif. Contohnya, pasien tidak menderita epilepsi (class 2) tetapi dari model yang telah memprediksi pasien tersebut menderita epilepsi (class 1).

- **False Negative (FN) – Type II Error**

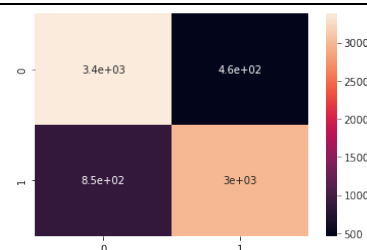
Merupakan data positif namun diprediksi sebagai data negatif. Contohnya, pasien menderita epilepsi (class 1) tetapi dari model yang dibuat memprediksi pasien tersebut tidak menderita epilepsi (class 2).

**Tabel 8.** Hasil Confusion Matrix

Data Latih	Data Uji	Confusion Matrix						
90%	10%	<table border="1"> <tr> <td>1.1e+03</td> <td>2.1e+02</td> </tr> <tr> <td>2.7e+02</td> <td>1e+03</td> </tr> <tr> <td>0</td> <td>1</td> </tr> </table>	1.1e+03	2.1e+02	2.7e+02	1e+03	0	1
1.1e+03	2.1e+02							
2.7e+02	1e+03							
0	1							
80%	20%	<table border="1"> <tr> <td>2.3e+03</td> <td>3e+02</td> </tr> <tr> <td>4.6e+02</td> <td>2.1e+03</td> </tr> <tr> <td>0</td> <td>1</td> </tr> </table>	2.3e+03	3e+02	4.6e+02	2.1e+03	0	1
2.3e+03	3e+02							
4.6e+02	2.1e+03							
0	1							
75%	25%	<table border="1"> <tr> <td>2.7e+03</td> <td>5.4e+02</td> </tr> <tr> <td>6.8e+02</td> <td>2.5e+03</td> </tr> <tr> <td>0</td> <td>1</td> </tr> </table>	2.7e+03	5.4e+02	6.8e+02	2.5e+03	0	1
2.7e+03	5.4e+02							
6.8e+02	2.5e+03							
0	1							

70%

30%



Berdasarkan tabel confusion matrix di atas, dapat dilakukan penghitungan nilai akurasi, precision dan recall. Nilai akurasi merupakan metode pengujian berdasarkan tingkat kedekatan antara nilai prediksi dengan nilai aktual. Dengan mengetahui jumlah data yang diklasifikasikan secara benar. Berikut rumus perhitungan akurasi.

$$Akurasi = \frac{TP + TN}{TP + FN + FP + TN} \times 100\% \quad (3)$$

Adapun precision merupakan metode pengujian dengan melakukan perbandingan jumlah informasi relevan yang didapatkan sistem dengan jumlah seluruh informasi yang terambil oleh sistem baik yang relevan maupun tidak. Berikut merupakan rumus perhitungan precision.

$$Precision = \frac{TP}{TP + FP} \quad (4)$$

Selanjutnya metode pengujian recall yaitu metode yang membandingkan jumlah informasi relevan yang didapatkan sistem dengan jumlah seluruh informasi relevan yang ada dalam koleksi informasi. Berikut merupakan rumus perhitungan recall.

$$Recall = \frac{TP}{TP + FN} \quad (5)$$

Adapun tahapan evaluasi selain confusion matrix yaitu mean absolute error dan F1 Score. Mean Absolute Error adalah rata-rata selisih mutlak nilai sebenarnya dengan nilai prediksi. MAE digunakan untuk mengukur keakuratan suatu model statistik dalam melakukan prediksi atau peramalan. Berikut merupakan rumus perhitungan MAE.

$$MAE = \frac{1}{n} \sum_{i=1}^n |A_i - F_i| \quad (6)$$

Dimana:

n adalah ukuran sampel

$A_i$  adalah nilai data aktual ke-i

$F_i$  adalah nilai data peralaman ke-i

Karena pada rumus MAE di atas terdapat tanda mutlak, maka nilai MAE akan selalu bernilai positif.

Berikut merupakan tabel hasil perhitungan akurasi, precision, recall, F1 Score dan Mean Absolute Error.

Tabel 9. Hasil Akurasi, Precision, Recall, F1 Score dan MAE

Data Latih	Data Uji	Akurasi	Precision	Recall	F1 Score	MAE
90%	10%	81%	83%	79%	81%	0.18
80%	20%	85%	87%	82%	85%	0.14
75%	25%	81%	82%	79%	81%	0.19
70%	30%	83%	87%	78%	82%	0.17

Pada tabel di atas dapat diketahui bahwa nilai akurasi yang paling tinggi adalah 85% dengan nilai precision 87%, recall 82%, F1 Score 85% dan MAE 0.14. namun skor uji dan skor latih memiliki gap yang cukup tinggi yaitu 13,52%. sehingga model tersebut mengalami overfitting. Adapun skor uji dan skor latih yang memiliki gap yang paling rendah adalah 88.65% untuk skor latih dan 81.20% untuk skor uji. Berdasarkan model tersebut didapatkan nilai akurasi sebesar 81% dengan precision 83%, recall 79%, F1 Score 81% dan Mean Absolute Error 0.18.

## VII. SIMPULAN

Penelitian ini telah berhasil melakukan ekstraksi fitur pada sinyal EEG dalam tahap preprocessing dengan menggunakan metode Discrete Wavelet Transform dan melakukan klasifikasi sinyal menggunakan salah satu algoritma machine learning yaitu algoritma Extreme Gradient Boosting. Hasil pengujian didapatkan kesimpulan bahwa nilai akurasi yang paling baik adalah 81% dengan precision 83%, recall 79%, F1 Score 81% dan Mean Absolute Error 0.18. hal ini dikarenakan gap antara skor latih dan skor uji lebih rendah daripada hasil model lainnya. Hasil ini didapatkan dari prosentase perbandingan data latih dan data uji sebesar 90% dan 10%. Adapun pada proses pemodelan, Randomized Search Cross Validation dengan cross validation sebanyak 5 kali dapat menghasilkan hyperparameter tuning terbaik yaitu learning rate adalah 0.16,  $n_{estimator}$  adalah 150, subsample adalah 0.50, gamma adalah 9, colsample bytree adalah 0.36, reg alpha adalah 0.06 dan reg lambda adalah 0.07. Namun hasil akurasi yang dihasilkan oleh model tersebut masih tergolong overfitting dikarenakan gap antara skor latih dan skor uji sebesar 7.45%. sehingga pada penelitian selanjutnya diharapkan dapat ditemukan improvisasi baik pada tahapan preprocessing dan processing yang sanggup mengatasi kelemahan pada penelitian ini serta dapat menerapkan metode machine learning yang lain untuk menjadi tolak ukur antara algoritma Extreme Gradient Boosting dengan algoritma machine learning yang lain.

## UCAPAN TERIMA KASIH

Puji dan syukur kami panjatkan kepada Tuhan Yang Maha Esa, karena atas berkat dan rahmat-Nya, kami dapat menyelesaikan penelitian ini. Peneliti juga mengucapkan terima kasih kepada prodi Informatika Universitas Muhammadiyah Sidoarjo yang telah memberikan dukungan maupun bimbingan selama penelitian ini dilakukan.

## REFERENSI

- [1] A. M. Gelgel and T. Sarongku, "PEMERIKSAAN PET DAN SPECT SCAN PADA BEDAH EPILEPSI," *Callosum Neurology*, vol. 3, no. 3, 2020, doi: 10.29342/cnj.v3i3.128.
- [2] N. L. Muzayyanah, S. Hapsara, and T. Wibowo, "Kejang Berulang dan Status Epileptikus pada Ensefalitis sebagai Faktor Risiko Epilepsi Pascaensefalitis," *Sari Pediatri*, vol. 15, no. 3, 2016, doi: 10.14238/sp15.3.2013.150-5.
- [3] M. N. Saefulloh, R. D. I. Astuti, W. Nurruhyuliawati, Y. Andriane, and M. K. Dewi, "Hubungan Lama Pengobatan dan Jenis Obat Anti Epilepsi dengan Derajat Depresi pada Pasien Epilepsi," *Jurnal Integrasi Kesehatan & Sains*, vol. 1, no. 2, 2019, doi: 10.29313/jiks.v1i2.4344.
- [4] N. K. C. Pratiwi, R. Magdalena, Y. N. Fuadah, S. Saidah, S. Rizal, and M. R. Isnaini, "Denoising Sinyal EEG dengan Algoritma Recursive Least Square dan Least Mean Square," *TELKA - Telekomunikasi, Elektronika, Komputasi dan Kontrol*, vol. 5, no. 2, 2019, doi: 10.15575/telka.v5n2.122-129.
- [5] J. Teye Brown and W. Zgallai, "Deep EEG: Deep learning in biomedical signal processing with EEG applications," in *Biomedical Signal Processing and Artificial Intelligence in Healthcare*, 2020. doi: 10.1016/b978-0-12-818946-7.00005-6.
- [6] A. al-Qerem, F. Kharbat, S. Nashwan, S. Ashraf, and khairi blaou, "General model for best feature extraction of EEG using discrete wavelet transform wavelet family and differential evolution," *Int J Distrib Sens Netw*, vol. 16, no. 3, 2020, doi: 10.1177/1550147720911009.
- [7] W. Apriliah, I. Kurniawan, M. Baydhowi, and T. Haryati, "Prediksi Kemungkinan Diabetes pada Tahap Awal Menggunakan Algoritma Klasifikasi Random Forest," *SISTEMASI*, vol. 10, no. 1, 2021, doi: 10.32520/stmsi.v10i1.1129.
- [8] A. M. Husein, F. R. Lubis, and M. K. Harahap, "Analisis Prediktif untuk Keputusan Bisnis : Peramalan Penjualan," *Data Sciences Indonesia (DSI)*, vol. 1, no. 1, 2021, doi: 10.47709/dsi.v1i1.1196.
- [9] A. Eviyanti, H. Hindarto, Sumarno, and H. A. A. Duddin, "Epilepsi detection system based on EEG record using neural network backpropagation method," in *Journal of Physics: Conference Series*, 2019, vol. 1381, no. 1. doi: 10.1088/1742-6596/1381/1/012037.
- [10] M. Mulaab, "DETEKSI KEJANG EPILEPSY DENGAN MENGGUNAKAN PEMILIHAN FITUR INFORMATIOAN GAIN DAN PEMBELAJARAN ENSEMBLE RANDOM FOREST," *Jurnal Simantec*, vol. 9, no. 2, 2021, doi: 10.21107/simantec.v9i2.11084.
- [11] A. Eviyanti, H. Hindarto, and M. Abror, "Peningkatan Ekstraksi Ciri Sinyal Epilepsi Menggunakan Teknik Sampling," *Telematika*, 2020, [Online]. Available: <https://ejournal.amikompurwokerto.ac.id/index.php/telematika/article/view/964>
- [12] S. Noertjahjani, "ANALISIS TIPE WAVELET COIFLETS 1 DAN COIFLETS 5 UNTUK DETEKSI PENYAKIT EPILEPSI," 2022.

- [13] Y. Sugianela, Q. L. Sutino, and D. Herumurti, "EEG CLASSIFICATION FOR EPILEPSY BASED ON WAVELET PACKET DECOMPOSITION AND RANDOM FOREST," *Jurnal Ilmu Komputer dan Informatika*, vol. 11, no. 1, 2018, doi: 10.21609/jiki.v11i1.549.
- [14] I. Muslim and K. Karo, "Implementasi Metode XGBoost dan Feature Importance untuk Klasifikasi pada Kebakaran Hutan dan Lahan," *Journal of Software Engineering, Information and Communication Technology*, vol. 1, no. 1, 2020.
- [15] E. H. Yulianti, O. Soesanto, and Y. Sukmawaty, "Penerapan Metode Extreme Gradient Boosting (XGBOOST) pada Klasifikasi Nasabah Kartu Kredit," *JOMTA Journal of Mathematics: Theory and Applications*, vol. 4, no. 1, 2022.
- [16] H. Sunata, "Komparasi Tujuh Algoritma Identifikasi Fraud ATM Pada PT. Bank Central Asia Tbk," *JATISI (Jurnal Teknik Informatika dan Sistem Informasi)*, vol. 7, no. 3, 2020, doi: 10.35957/jatisi.v7i3.471.
- [17] P. M. Kouate, "Machine Learning: GridSearchCV & RandomizedSearchCV," <https://towardsdatascience.com/machine-learning-gridsearchcv-randomizedsearchcv-d36b89231b10>, Sep. 11, 2020.
- [18] M. S. ANGGREANY, "Confusion Matrix," <https://socs.binus.ac.id/2020/11/01/confusion-matrix/>, Nov. 01, 2020.
- [19] R. Maulid, "Kriteria Jenis Teknik Analisis Data dalam Forecasting," <https://www.dqlab.id/kriteria-jenis-teknik-analisis-data-dalam-forecasting>, Jan. 14, 2022.
- [20] A. A. Suryanto, A. Muqtadir, and S. Artikel, "PENERAPAN METODE MEAN ABSOLUTE ERROR (MEA) DALAM ALGORITMA REGRESI LINEAR UNTUK PREDIKSI PRODUKSI PADI Info Artikel: ABSTRAK," no. 1, p. 11, 2019.

**Conflict of Interest Statement:**

*The author declares that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.*