

MENGOPTIMALKAN KLASIFIKASI SINYAL *ELECTROENCEPHALOGRAPH* UNTUK MENDETEKSI KEJANG EPILEPSI DENGAN *BACKPROPAGATION NEURAL NETWORK*

Nurul Najwa Sabilla, Muhamad Fatchan, Ahmad Aguswin

Teknik Informatika, Universitas Pelita Bangsa

Jalan Inspeksi Kalimalang, Tegal Danas, No. 9, Kabupaten Bekasi, Indonesia

nurulnajwa.s03@mhs.pelitabangsa.ac.id

ABSTRAK

Epilepsi adalah gangguan neurologis yang memengaruhi lebih dari 50 juta orang di seluruh dunia, dengan sekitar 6 juta penderita di Eropa. Penyakit ini memiliki tingkat kesalahan diagnosis yang tinggi dan dapat menyebabkan gangguan psikologis akibat stigma sosial. Klasifikasi epilepsi berdasarkan *Electroencephalogram* (EEG) sangat penting dalam diagnosis, namun interpretasi hasil EEG seringkali sulit dan memerlukan keahlian khusus. Kesulitan dalam mendiagnosis epilepsi secara akurat akibat interpretasi EEG yang membutuhkan pengalaman dan waktu, hal ini memengaruhi efektivitas pengobatan. Penelitian ini bertujuan untuk mengoptimalkan klasifikasi kejang epilepsi secara akurat dengan menghasilkan performa tinggi. *Backpropagation Neural Network* adalah salah satu metode dari pembelajaran mendalam digunakan pada kasus memprediksi klasifikasi untuk menurunkan tingkat kesalahan dalam diagnosis dan pemantauan kejang epilepsi dan non-epilepsi. Hasil dari penelitian ini menghasilkan tingkat akurasi 98,63%. Pelatihan pada *split data* 80:20, *learning rate* 0.1, jumlah *neuron* pada *hidden layer* terdapat 9 *neuron*, dengan 100 *epoch* pelatihan. Saat dilatih menggunakan data target berhasil digunakan dalam 418 diprediksi sebagai epilepsi dan 1791 sebagai non-epilepsi.

Kata kunci : Kejang Epilepsi, Sinyal *Electroencephalogram*, *Backpropagation Neural Network*, *TensorFlow*

1. PENDAHULUAN

Epilepsi adalah penyakit otak yang serius melanda gangguan neurologis, bagi pascin dengan riwayat epilepsi di seluruh dunia. Berdasarkan kehidupan penderita epilepsi lebih dari 50 juta orang diseluruh dunia menderita. Dengan diperkirakan 125.000 angka kematian pertahun yang menjadi tantangan untuk melakukan pencegahan serta mengurangi beban kasus baru epilepsi [1].

Masalah kesehatan masyarakat yang signifikan terjadi di negara Jerman dan juga seluruh Eropa yang ditandai dengan masalah kompleks, manifestasi klinis yang beragam, dan dampak mendalam terhadap kualitas hidup pasien. Setengah dari 6 juta warga Eropa menderita epilepsi. Definisi baru dari *International League Against Epilepsy* (ILAE) membuat jumlah kasus epilepsi yang tercatat menjadi lebih banyak. Sekarang, seseorang yang mengalami satu kali kejang tanpa pemicu, tetapi memiliki kemungkinan kambuh lebih dari 60%, sudah dianggap sebagai penderita epilepsi. Karena kriteria ini lebih luas, jumlah orang yang didiagnosis epilepsi juga meningkat [2].

Salah satu aspek penting dalam memahami epilepsi adalah klasifikasi dan karakterisasi epilepsi. ILAE juga telah mengembangkan sistem klasifikasi yang mendefinisikan sindrom epilepsi berdasarkan kombinasi fitur klinis dan *Electroencephalogram* (EEG), serta faktor etiologis tertentu [3]. Etiologi epilepsi diklasifikasikan menjadi enam jenis: struktural, genetik, infeksi, metabolik, imunologis, dan terakhir etiologinya tidak diketahui. Setiap pasien

epilepsi mungkin mempunyai lebih dari satu etiologi serta penyebab epilepsi.

Pandangan masyarakat terhadap penyakit ini juga masih banyak yang keliru. Beberapa faktor yang menyebabkan kekeliruan tersebut diantaranya pemikiran awam seperti menganggap penyakit menular. Epilepsi ini dapat mengenai kelompok lanjut usia dan usia muda. Epilepsi tampak dalam bentuk kecenderungan yang berlangsung lama saat sedang kejang hingga hilang kesadaran. Selain itu, epilepsi dapat menyebabkan masalah psikologis yang signifikan, termasuk kecemasan dan depresi, yang sering kali diperburuk oleh stigma sosial [4].

Hubungan antara epilepsi dan EEG sangat penting dalam diagnosis dan manajemen penyakit ini. EEG digunakan untuk mendeteksi aktivitas listrik abnormal di otak, dengan menempatkan sensor (elektroda) di kulit kepala. Analisis EEG berperan dalam evaluasi prognosis dan efektivitas pengobatan pasien. Prosedur memperoleh sinyal EEG sangat mudah tidak memerlukan pembedahan. Namun, proses penafsiran untuk mengetahui orang dengan epilepsi (ODE) atau non-epilepsi membutuhkan waktu [5].

Pemula atau orang yang kurang berpengalaman bisa saja salah menafsirkan hasilnya. Oleh karena itu, metode pengenalan pola penting untuk mendeteksi kejang epilepsi. Hal ini dilakukan agar terdapat sistem yang membantu dokter spesialis neurologi dalam mendeteksi kejang secara otomatis. Deteksi kejang yang akurat penting untuk pengobatan yang efektif, karena diagnosis yang tertunda atau salah dapat menyebabkan peningkatan angka kesakitan.

Dalam penelitian dengan melihat permasalahan diatas akan melakukan pemodelan *deep learning* berjudul “Mengoptimalkan Klasifikasi Sinyal *Electroencephalogram* Untuk Mendeteksi Kejang Epilepsi Dengan *Backpropagation Neural Network*”. Model *Backpropagation* sebagai tahapan algoritma jaringan syaraf tiruan, memiliki keunggulan yaitu kemampuannya untuk bekerja tanpa memerlukan asumsi bahwa data harus mengikuti distribusi multivariat normal. Penelitian lain menunjukkan bahwa *Backpropagation* dapat memecahkan masalah pemodelan deret waktu nonlinier [6].

Diharapkan penelitian terbaru ini dapat mengoptimalkan klasifikasi input data sinyal EEG melalui peningkatan akurasi dan arsitektur model, sehingga mampu membedakan antara sinyal EEG normal dan sinyal yang menunjukkan kejang epilepsi dengan memiliki performa baik.

2. TINJAUAN PUSTAKA

2.1. Penelitian Terdahulu

Pelatihan jaringan saraf juga menjadi topik penting dalam teknologi masa depan, dengan banyak penelitian yang telah membahasnya. Pendekatan awal adalah menyederhanakan proses diagnosis menggunakan data EEG pasien. Berdasarkan studi, menurut Priti N. Bhagat, K. S.Ramesh, Dr.Venu Gopala Rao Matcha, dan S T Patil tahun 2020 dalam penelitian berjudul “*Robust Prior Stage Epileptic Seizure Diagnosis System using Resnet and Backpropagation Techniques*” menyatakan latar belakang yang ditulis seperti teknik visual dalam klasifikasi epilepsi saat ini membutuhkan waktu yang lama untuk pemeriksaan data EEG. Manusia dapat dengan mudah mengidentifikasi epilepsi tetapi diagnosisnya lambat. Pasien sering mengalami gejala tidak stabil, menyulitkan diagnosis, dan berisiko fatal. *Dataset* pelatihan terdiri dari 22 data pasien (5 laki-laki, 17 perempuan) dengan rentang usia balita hingga dewasa muda. Kombinasi teknik pembelajaran mendalam *Backpropagation* dan ResNet yang diusulkan berhasil mencapai akurasi klasifikasi sebesar 99,835% dan efisiensi sistem hingga 99,72% dengan menerapkan optimasi *Stochastic Gradient* [7].

Selain itu, metode jaringan saraf *Backpropagation* yang dikembangkan dengan pendekatan gradien konjugasi juga menunjukkan keberhasilan dalam analisis prediksi pasien COVID-19 di Selangor, Malaysia menurut Noor Amirah Binti Ajmal Khan, dan Siti Mahani Marjug pada tahun 2024 dalam penelitian berjudul “*Prediction Analysis Of Covid-19 In Selangor By Using Backpropagation Algorithm With Conjugate Gradient Method*”. Pasien yang terinfeksi COVID-19 rentan terhadap komplikasi serius hingga kematian, terutama bagi pasien yang memiliki gangguan paru-paru atau sistem kekebalan tubuh yang lemah. Oleh penelitian ini adalah untuk membuat model prediksi yang efektif untuk memproyeksikan jumlah kasus yang akan datang. Algoritma *Backpropagation* dapat mencapai akurasi 95% pada dataset validasi setelah 19

epoch dengan kecepatan pembelajaran 0,005 dan arsitektur 16-5-1, yang memiliki 16 neuron di lapisan input, 5 neuron di lapisan tersembunyi, dan 1 neuron di lapisan output [8].

Selanjutnya, mendukung solusi teknologi pertanian khusus, berhasil mengidentifikasi berbagai penyakit daun tanaman tomat dari penulis Anglita Wigina Putri dalam penelitian berjudul “Implementasi *Artificial Neural Network (ANN) Backpropagation* Untuk Klasifikasi Jenis Penyakit Pada Daun Tanaman Tomat”. Petani menghadapi kendala dalam budidaya tanaman tomat karena tanaman ini rentan terhadap serangan penyakit, sehingga mereka kesulitan membedakan berbagai jenis penyakit yang tampak serupa pada daun tomat. Model ini dievaluasi kinerjanya dengan *confusion matrix*, yang mencapai tingkat presisi terbaik sebesar 78% dan mampu mengidentifikasi data klasifikasi yang benar (*true positive*) dalam 117 kasus [9].

2.2. Data Mining

Data mining merupakan proses penggalian informasi dan pola yang bermanfaat dan pola yang bermanfaat dari data yang sangat besar [10]. *Data mining* menyimpan pola tersembunyi yang dimanfaatkan dalam evaluasi bisnis dan dapat diaplikasikan untuk memproyeksikan tren masa depan guna mendukung pengambilan keputusan atau merancang strategi dalam pengelolaan bisnis yang optimal, inovatif, dan fleksibel.

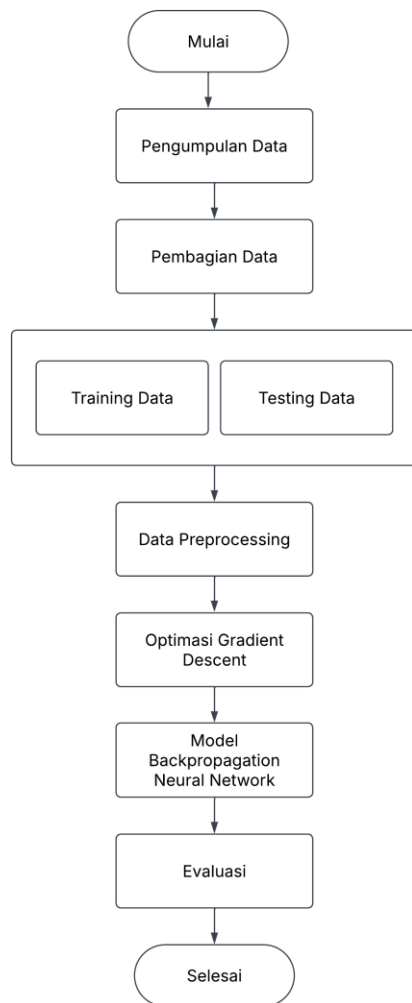
2.3. Backpropagation Neural Network

Salah satu teknik pembelajaran pada ANN *Multilayer Perceptron* adalah *Backpropagation*, yang melibatkan perhitungan dan propagasi balik dari error yang dihasilkan. Teknik ini menghasilkan bobot yang sesuai dengan pola data saat ini [11]. *Backpropagation* juga proses yang membantu menentukan gradien error terhadap nilai bobot yang ada, sehingga model dapat dilatih untuk menjadi model yang baik. Algoritma *Backpropagation* dilatih dalam tiga tahap. Ini adalah fase maju, fase mundur, dan fase ketiga memodifikasi bobot untuk menurunkan tingkat kesalahan.

3. METODE PENELITIAN

Bagian ini menguraikan tahapan analisis kejang epilepsi dengan *Backpropagation Neural Network*, seperti ditunjukkan pada Gambar 1.

Penelitian ini dipandu oleh tujuan berikut yang disusun berdasarkan metodologi *Cross Industry Standard Process for Data Mining (CRISP-DM)*. CRISP-DM sebagai pendekatan komprehensif dan standar untuk pengembangan data, yang terdiri dari lima tahapan yang digunakan oleh peneliti saat ini, yaitu *business understanding*, *data understanding*, *data preparation*, *modelling*, dan *evaluation*, berikut merupakan tahapan.



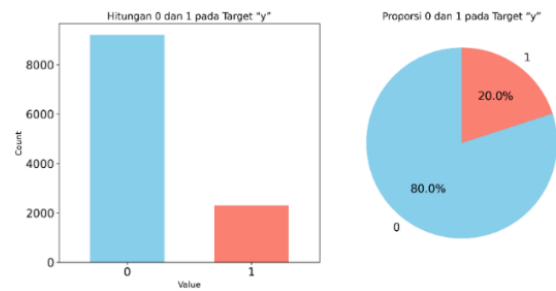
Gambar 1. Alur Penelitian Yang Dilakukan

a. *Business Understanding*

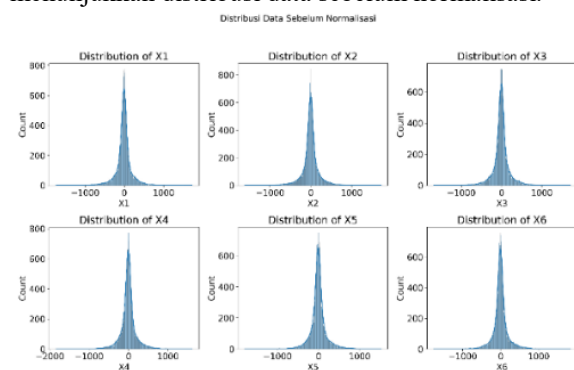
Peneliti memahami tentang masalah yang akan diselesaikan, yaitu analisis mengoptimalkan klasifikasi kejang epilepsi, merupakan bagian dari tahap ini. Dengan pencapaian metrik bisnis untuk menurunkan tingkat kesalahan dalam diagnosis dan pemantauan kejang epilepsi dan non-epilepsi, yang akan berdampak positif pada aspek bisnis kesehatan.

b. *Data Understanding*

Tahapan untuk melakukan investigasi pada *dataset* yang telah didapatkan dan *summarize* karakter utama pada berbagai karakteristik penting dari data untuk dipahami. Langkah yang dilakukan pada Gambar 2, data rekaman EEG adalah memeriksa asumsi terhadap proporsi kelas pada data target (y) untuk memastikan keseimbangan data.

Gambar 2. Visualisasi Proporsi Kelas Pada Data Target (y)

Selain itu, pada data fitur (x), dilakukan juga penyesuaian skala data. Karena data dengan angka yang besar mendominasi, sedangkan fitur dengan angka kecil terabaikan dan digunakan pada *Gradient Descent* nanti dengan tujuan menghitung jarak berdasarkan nilai fitur, jika skala fitur tidak seragam maka langkah gradien tidak stabil. Gambar 3, menunjukkan distribusi data sebelum normalisasi.



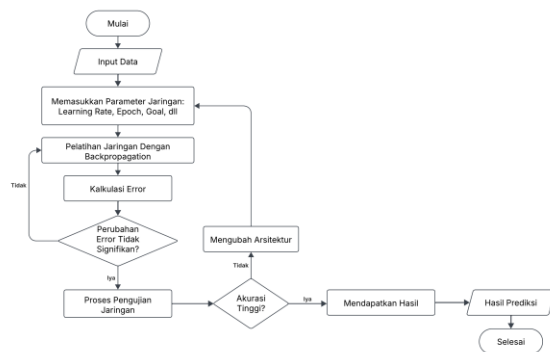
Gambar 3. Visualisasi Skala Data Fitur (X1-X6)

c. *Data Preparation*

Data Preparation seperti gerbang utama untuk membersihkan data. Setelah memperoleh berbagai informasi tentang data rekaman EEG, dilakukan berbagai perlakuan terhadap data numerik, termasuk membagi data dengan tiga rasio pembagian data, serta menerapkan metode persiapan (*treatment preparation*) yang berbeda pada *train data* maupun *test data*.

d. *Modeling*

Pada tahap ini, dilakukan pemilihan teknik *data mining* untuk mengolah data yang telah melalui proses persiapan. Teknik yang diterapkan adalah klasifikasi berbasis *deep learning* menggunakan algoritma *Backpropagation Neural Network* dalam lingkungan *Google Colab* dengan bahasa *Python*. Gambar 4, menyajikan alur pemodelan menggunakan *Backpropagation Neural Network*.



Gambar 4. Flowchart Backpropagation Neural Network

Tahapan proses perhitungan *Backpropagation* dimulai dengan memasukkan data yang akan diolah pada model. Setelah itu, parameter jaringan yang akan dilatih seperti *learning rate*, *epoch*, *goal*, dan arsitektur jaringan akan dilatih dan diuji. Proses pelatihan jaringan menggunakan metode *Backpropagation* dilakukan dengan tiga tahapan, yaitu *Feedforward* dari data masukan, *Backpropagation* menghitung error yang diperoleh, dan pembaruan bobot menggunakan algoritma optimasi *Stochastic Gradient Descent* (SGD) agar error semakin kecil. Setelah error dihitung, pelatihan akan terus dilakukan hingga perubahan error sudah tidak signifikan lagi. Selanjutnya, dilakukan pengujian jaringan untuk mengevaluasi performanya. Jika akurasi jaringan cukup tinggi, maka hasil pengujian dapat digunakan dalam proses prediksi. Namun, saat akurasi rendah arsitektur jaringan dapat disesuaikan dan pelatihan ulang dilakukan untuk meningkatkan performa model.

e. Evaluation

Evaluasi sebagai tempat parameter uji hasil pelatihan model yang telah dilakukan. Pada tahap ini, akan diputuskan apakah hasil dari pemodelan memenuhi tujuan teknis yang diharapkan. Salah satu fokus dalam penelitian ini dengan evaluasi untuk analisis kejang epilepsi menggunakan algoritma *Backpropagation Neural Network*. Pada penelitian ini, *error function* dengan perhitungan *Mean Square Error* (MSE) dapat dilihat pada persamaan (1).

$$j(\theta) = \frac{1}{2m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)})^2 \quad (1)$$

Penelitian ini dilakukan dengan beberapa pengujian model pola *Backpropagation Neural Network* yang terdiri dari tiga yaitu lapisan input, lapisan tersembunyi, dan lapisan *output* lalu dibagi kedalam tiga arsitektur berbeda yaitu (178-3-1), (178-6-1), dan (178-9-1), *learning rate* ($\alpha = 0.1$), ($\alpha = 0.01$), dan ($\alpha = 0.001$) dengan dilatih selama 100 *epoch* serta perbandingan data latih dan data uji yang digunakan dalam pengujian yakni 70:30, 80:20, dan 90:10. Berikut dapat dilihat dari Tabel 1, yaitu pembagian data untuk rasio penelitian.

Tabel 1. Pembagian Data

Pembagian Data	Training Data	Testing Data	Learning Rate
90:10	10350	1150	0.1
80:20	9200	2300	0.1
70:30	8050	3450	0.1
90:10	10350	1150	0.01
80:20	9200	2300	0.01
70:30	8050	3450	0.01
90:10	10350	1150	0.001
80:20	9200	2300	0.001
70:30	8050	3450	0.001

4. HASIL DAN PEMBAHASAN

4.1. Hasil Pengumpulan Data

Tabel 2. Dataset Sinyal EEG

NO	X1	X2	X3	X4	...	X176	X177	X178	y
1	135	190	229	223	...	-116	-83	-51	4
2	386	382	356	331	...	154	143	129	1
3	-32	-39	-47	-37	...	-35	-35	-36	5
4	-105	-101	-96	-92	...	-72	-69	-65	5
5	-9	-65	-98	-102	...	-83	-89	-73	5
...	...	...	...	...	...	...	...	...	...
11496	-22	-22	-23	-26	...	-37	-47	-48	2
11497	-47	-11	28	77	...	77	117	170	1
11498	14	6	-13	-16	...	-2	-1	-8	5
11499	-40	-25	-9	-12	...	68	59	55	3
11500	29	41	57	72	...	-2	2	20	4

Pada Tabel 2, merupakan tahapan pengumpulan data penelitian dengan menggunakan dataset EEG bersumber dari kaggle.com terdiri dari 11.500 sampel yang dikumpulkan dengan 179 titik data durasi 1 yang melibatkan 100 individu yang melakukan perekaman selama 23,6 detik. Variabel respons diberi label $y \in \{1,2,3,4,5\}$ di kolom terakhir, dan variabel penjelas $x \in \{x_1, \dots, x_{178}\}$. Data yang dikategorikan ke dalam kelas 1 adalah pasien dengan kejang epilepsi (kelas 1), dan data yang dikategorikan ke dalam kelas 2,3,4,5 adalah pasien yang tidak mengalami kejang epilepsi (kelas 2,3,4,5). Data ini akan melalui beberapa tahapan untuk dapat diproses ke tahapan pengujian.

4.2. Pembagian Data Latih dan Uji

Dataset dibagi kedalam tiga rasio. Pertama 90% data latih (X_{train}, y_{train}), yaitu 10.350 dan 10% data uji (X_{test}, y_{test}) yang terdiri dari 1.150 sample. Kedua, menjadi 80% data latih (X_{train}, y_{train}), yaitu 9.200 sampel, dan 20% data uji (X_{test}, y_{test}), yang terdiri dari 2.300 sampel. Ketiga, 70% data latih (X_{train}, y_{train}), yaitu 8050 sampel, dan 30% data uji (X_{test}, y_{test}), yang terdiri dari 3450 sampel. Setiap sampel memiliki 178 fitur dalam X_{train} dan X_{test} .

4.3. Data Preprocessing

Pada tahap ini meliputi proses pembersihan, pengolahan, dan pengelompokan data yang telah diperoleh. Pemisahan *preprocessing* antara *train data*

dan *test data* dilakukan untuk menghindari kebocoran data (*data leakage*), seperti kontaminasi antara data pelatihan dan data uji atau validasi. Kebocoran ini dapat terjadi ketika informasi dari data uji secara tidak sengaja digunakan saat membangun model, yang sering kali disebabkan oleh pemisahan atau *preprocessing* yang tidak tepat.

4.3.1. Training Data

Dalam hal ini, dilakukan dua proses *preparation*, yaitu *balancing data* menggunakan teknik SMOTE. Proses ini dilakukan karena jumlah data pasien pada kelas 1 (pasien dengan kejang epilepsi) lebih rendah dibandingkan kelas lainnya. Pada penelitian ini, digunakan parameter *sampling_strategy* dengan nilai '*minority*' pada metode SMOTE. Jumlah data hasil *resampling* akan mengikuti proporsi kelas mayoritas, yaitu kelas 0. Dengan perhitungan persamaan jarak (2). Dan, dilanjut menghitung jarak terdekat (3) berikut rumusnya.

$$dist = \sqrt{(X_1 - Y_1)^2 + (X_2 - Y_2)^2} \quad (2)$$

$$X_{syn} = X_i + (X_{knn} - X_i) \times \sigma \quad (3)$$

Tabel 3. Hasil Proses Penyeimbangan Kelas

Label Kelas (Rasio)	Jumlah Training Data
1 (90:10)	6440
0 (90:10)	6440
1 (80:20)	7360
0 (80:20)	7360
1 (70:30)	8280
0 (70:30)	8280

Pada Tabel 3, merupakan hasil proses penyeimbangan kelas dengan masing-masing rasio kelas. Selanjutnya, untuk normalisasi data, peneliti menggunakan *Z-score* dengan persamaan (4) berikut.

$$Z = \frac{x - \bar{x}}{\sigma} \quad (4)$$

Metode ini digunakan untuk mengubah data menjadi skala dengan rata-rata -3 hingga +3, sehingga membantu menjaga konsistensi data. Berikut perubahan pada skala data pada Tabel 4, yaitu Normalisasi *Training Data* dari rasio pembagian data 90:10.

Tabel 4. Normalisasi Training Data (90:10)

X1	X2	X3	...	X178
1.893567	2.262507	2.067268	...	-0.63911
0.077821	0.104718	0.088157	...	0.116269
1.452805	1.427080	1.582192	...	-1.44386
0.628773	0.696678	0.597487	...	0.096520
0.140103	0.066527	-0.008858	...	0.205137

Tabel 5, yaitu hasil normalisasi dari pembagian data sebesar 80:20.

Tabel 5. Normalisasi Training Data (80:20)

X1	X2	X3	...	X178
-0.797295	-1.397293	-1.743332	...	1.061480
-0.037059	0.023848	0.074202	...	0.383592
0.046322	0.048017	0.010515	...	0.254941
0.571130	0.719917	0.608194	...	0.007536
-0.321534	-0.329021	-0.322618	...	0.086706

Tabel 6, ini juga merupakan normalisasi dengan pembagian data 70:30.

Tabel 6. Normalisasi Training Data (70:30)

X1	X2	X3	...	X178
0.205671	0.160996	0.159167	...	0.326731
-0.051476	-0.085460	-0.138281	...	-0.03232
-0.255252	-0.491389	-0.635653	...	-0.47734
0.176560	0.214154	0.207929	...	0.286275
-0.264956	-0.244932	-0.152910	...	-0.06266

4.3.2. Testing Data

Pada *testing data* peneliti menggunakan hanya normalisasi data menggunakan *Z-score* seperti persamaan (4) di atas. Berikut Tabel 7, Normalisasi *Testing Data*.

Tabel 7. Normalisasi Testing Data (90:10)

X1	X2	X3	...	X178
0.010749	0.061753	0.054202	...	-0.145400
0.049076	0.023562	-0.023410	...	-0.051594
2.420564	3.040648	3.401228	...	-0.184897
-0.621648	-0.425182	-0.265949	...	-0.333011
-0.166514	-0.281966	-0.115575	...	0.259446

Tabel 8, dibawah ini merupakan normalisasi *testing data* dengan pembagian rasio data 80:20.

Tabel 8. Normalisasi Testing Data (80:20)

X1	X2	X3	...	X178
-0.115535	0.048017	0.093798	...	0.467709
-2.715051	-2.890942	-2.860307	...	-0.081529
1.022367	-2.605747	-5.015872	...	-5.717407
0.139512	0.144693	0.181980	...	0.165875
-2.160815	-2.059526	-2.022576	...	1.110961

Tabel 9, merupakan normalisasi akhir dari *testing data* dengan rasio penggunaan data 70:30.

Tabel 9. Normalisasi Testing Data (70:30)

X1	X2	X3	...	X178
1.510812	0.866538	0.320082	...	2.455764
0.166856	0.175494	0.168919	...	0.018249
-0.260104	-0.215937	-0.226053	...	-0.007036
0.152301	0.049849	0.051891	...	0.614985
-0.129105	-0.157947	-0.187043	...	-0.199206

4.4. Proses Feedforward dan Backpropagation

Proses *Feedforward* dilakukan untuk menghitung *output* dari jaringan berdasarkan input data, dan kemudian melakukan propagasi mundur (*Backward Propagation*) untuk menyesuaikan bobot pada setiap layer agar prediksi *output* semakin mendekati nilai target yang diinginkan. Untuk menentukan *output* dibantu dengan fungsi *sigmoid* sebagai fungsi yang menghasilkan output dalam rentang 0 hingga 1, yang menggambarkan probabilitas suatu kelas tertentu. Persamaan (5) *sigmoid* dapat digunakan untuk tugas klasifikasi biner.

$$\text{Output} = g(z) = \frac{e^z}{1 + e^z} = \frac{1}{1 + e^{-z}} \quad (5)$$

Pelatihan model dilakukan dengan *library TensorFlow Keras* yang tersedia. Berikut adalah kode sumber dari model yang dikembangkan.

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from tensorflow.keras.optimizers import SGD

# Buat arsitektur dengan 1 hidden layer dan 1 output
model = Sequential([
    Dense(3, input_shape=(X_train_clean.shape[1],),
        activation="sigmoid"),
    Dense(1, activation="sigmoid")
])

# Buat optimizer
opt = SGD(learning_rate=0.1)

# Compile model
model.compile(loss="binary_crossentropy",
    optimizer=opt,
    metrics=["accuracy"])
```

Model arsitektur diatas salah satu yang diteliti yaitu dengan pola menggunakan *neuron input layer* sebanyak 178 *neuron*, satu *hidden layer* dengan 3 *neuron*, dan *output layer* sebanyak 1 *neuron*.

Proses dalam model diatas, yaitu *Forward Propagation* di mana data input melewati *hidden layer* dan *output layer*, di mana setiap *neuron* melakukan operasi linier dan aktivasi *sigmoid*. Serta, *Backward Propagation* yaitu model dikompilasi menggunakan *loss function* "*binary_crossentropy*" dan *optimizer* SGD dengan *learning rate* sebesar 0.1, yang berperan dalam menghitung gradien error dan memperbarui bobot serta bias selama *training*.

4.5. Menghitung Perubahan Bobot dan Bias

Menghitung perubahan bobot dan bias sebagai pelatihan selama proses selama *training* menggunakan algoritma *Backpropagation Neural Network*. Tujuan utama dari tahap ini adalah untuk mengoptimalkan model sehingga dapat menghasilkan prediksi yang lebih akurat, baik pada data latih maupun data uji. Perhitungan perubahan bobot dan bias pada *hidden layer*. setiap unit pada *hidden layer* mengalami perubahan bobot dan bias berdasarkan propagasi error

dari layer di atasnya [12]. Nilai error yang diterima oleh setiap unit *hidden layer* dihitung dengan menjumlahkan delta input dari unit-unit pada lapisan sesudahnya dengan persamaan (6).

$$\delta_{inj} = \sum_{k=1}^m \delta_k w_{jk} \quad (6)$$

Hasil ini kemudian dikalikan dengan turunan fungsi aktivasi untuk mendapatkan informasi kesalahan pada *hidden layer* dengan persamaan (7).

$$\delta_j = \delta_{inj} f'(z_{inj}) \quad (7)$$

Selanjutnya, bobot pada *hidden layer* diperbarui menggunakan persamaan (8).

$$\nabla v_{ij} = a \delta_j x_i \quad (8)$$

Proses ini dilakukan secara iteratif hingga nilai error jaringan mencapai tingkat optimal, sehingga model dapat menghasilkan prediksi yang lebih akurat. Perhitungan perubahan bobot dan bias dalam *Keras* dilakukan secara otomatis selama pelatihan model. Berikut adalah kode untuk melatih model menggunakan *dataset* yang dilatih selama 100 *epoch*.

```
# Train model
model_hist = model.fit(X_train_clean, y_train,
    epochs = 100,
    verbose = 1)
```

Setelah model dilatih, prediksi *output* dapat dihasilkan berdasarkan input data dengan menggunakan *model.predict()* dari *Keras*, dimana argumen *X* merupakan data masukan dalam bentuk *array* atau *tensor*. Untuk penelitian ini sebagai klasifikasi dalam bentuk probabilitas

Penggunaan ambang batas (*threshold*) berperan dalam mengonversi hasil probabilitas dari *model.predict(X)* menjadi data diskrit. Jika probabilitas lebih besar dari *threshold* ($p > 0.5$), prediksi dikategorikan sebagai kelas positif (1). Sebaliknya, jika probabilitas lebih kecil atau sama dengan *threshold* ($p \leq 0.5$), prediksi dikategorikan sebagai kelas negatif (0).

4.6. Menguji Model Dengan Testing Data

Pengujian model yang sudah dilatih dengan proses propagasi maju akan dihitung tingkat evaluasi untuk memperkecil *false negative*, yang berarti meningkatkan nilai *recall*. Selain itu, kurva ROC-AUC digunakan sebagai metrik evaluasi, khususnya dalam kasus klasifikasi dengan data tidak seimbang, untuk mengukur kemampuan model dalam membedakan kelas positif dan kelas negatif.

4.7. Pembahasan dan Evaluasi

Hasil pengujian terdiri dari 27 skema pengujian model *Backpropagation Neural Network*, yang dibagi menjadi 3 arsitektur, yaitu (178-3-1), (178-6-1), dan (178-9-1), masing-masing dengan *learning rate* ($\alpha = 0.1$), ($\alpha = 0.01$), dan ($\alpha = 0.001$). Selain itu, data latih dan data uji yang digunakan dalam pengujian dibandingkan, yaitu 70:30, 80:20, dan 90:10. Hasil pengujian dapat dilihat pada Tabel 10-12 berikut:

Tabel 10. Arsitektur 178-3-1

Latih:Uji	Learning Rate	Akurasi (%)	Presisi (%)	Recall (%)	F1 (%)
90:10	0.1	92.95	77.09	92.17	83.93
80:20	0.1	94.34	94.22	94.50	94.36
70:30	0.1	94.74	94.76	94.70	94.73
90:10	0.01	92.28	92.90	91.60	92.24
80:20	0.01	93.29	93.79	92.73	93.26
70:30	0.01	92.75	93.55	91.83	92.68
90:10	0.001	64.98	67.07	58.89	62.40
80:20	0.001	69.10	74.31	46.14	56.93
70:30	0.001	72.83	80.18	60.67	69.07

Dari hasil pengujian pada Tabel 10, arsitektur model 178-3-1 menunjukkan hasil tertinggi dan terendah. Akurasi tertinggi yang diperoleh mencapai 94.74%, dengan nilai evaluasi lainnya yaitu presisi 94.76%, *recall* 94.70%, dan *F1-score* 94.73%. Hasil ini diperoleh dengan pembagian data latih 70:30 serta *learning rate* 0.1. Sementara itu, akurasi terendah yang diperoleh adalah 64.98%, dengan nilai evaluasi lainnya yaitu presisi 67.07%, *recall* 58.89%, dan *F1-score* 62.40%. Hasil ini diperoleh dengan pembagian data latih 90:10 serta *learning rate* 0.001.

Tabel 11. Arsitektur 178-6-1

Latih:Uji	Learning Rate	Akurasi (%)	Presisi (%)	Recall (%)	F1 (%)
90:10	0.1	97.10	96.81	97.41	97.10
80:20	0.1	97.53	97.67	97.39	97.53
70:30	0.1	97.90	97.95	97.84	97.89
90:10	0.01	94.57	95.25	93.82	94.53
80:20	0.01	95.44	95.96	94.88	95.42
70:30	0.01	98.39	98.24	98.56	98.40
90:10	0.001	77.54	89.32	62.55	73.58
80:20	0.001	71.13	78.82	57.79	66.68
70:30	0.001	73.59	83.24	59.08	69.11

Dari hasil pengujian pada Tabel 11, arsitektur model 178-6-1 menunjukkan hasil tertinggi dan terendah. Akurasi tertinggi yang menunjukkan performa terbaik, mencapai 98.24%, dengan nilai evaluasi lainnya yaitu presisi 98.24%, *recall* 98.56%, dan *F1-score* 98.40%. Hasil ini diperoleh dengan *learning rate* 0.01 serta rasio data 70:30. Sementara itu, hasil terendah diperoleh dengan akurasi 71.13%, serta nilai evaluasi lainnya yaitu presisi 78.82%, *recall* 57.79%, dan *F1-score* 66.68%. Hasil ini diperoleh dengan *learning rate* 0.001 serta rasio data 80:20.

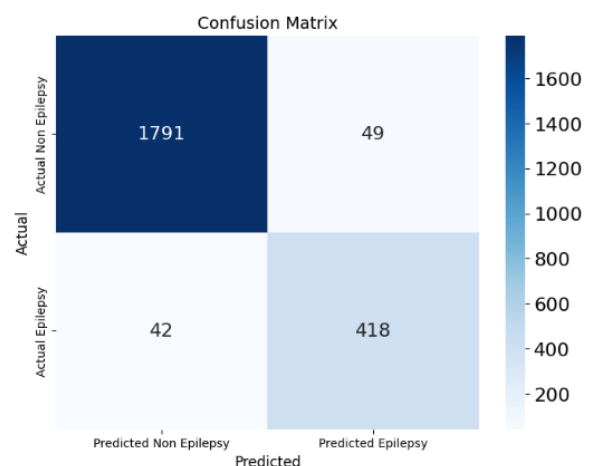
Tabel 12. Arsitektur 178-9-1

Latih:Uji	Learning Rate	Akurasi (%)	Presisi (%)	Recall (%)	F1 (%)
90:10	0.1	98.32	97.96	98.70	98.33
80:20	0.1	98.63	98.27	99.01	98.64
70:30	0.1	98.39	98.24	98.56	98.40
90:10	0.01	95.98	96.40	95.53	95.97
80:20	0.01	95.23	95.96	94.44	95.19
70:30	0.01	96.06	96.50	95.57	96.04
90:10	0.001	72.22	80.47	58.68	67.87
80:20	0.001	77.56	88.17	63.67	73.94
70:30	0.001	73.98	82.37	61.01	70.10

Untuk hasil pengujian pada Tabel 12, arsitektur 178-9-1 menunjukkan hasil tertinggi dan terendah. Akurasi tertinggi yang menunjukkan performa terbaik, mencapai 98.63%, dengan nilai evaluasi seperti presisi 98.27%, *recall* 99.01%, dan *F1-score* 98.64%. Hasil ini diperoleh dengan *learning rate* 0.1 serta rasio data 80:20. Selain itu, hasil terendah diperoleh dengan akurasi 72.22%, serta nilai evaluasi seperti presisi 80.47%, *recall* 58.68%, dan *F1-score* 67.87%. Hasil ini diperoleh dengan *learning rate* 0.001 dan rasio data 90:10.

Pengujian yang dilakukan terhadap 27 skema model *Backpropagation Neural Network* dengan berbagai arsitektur, yaitu (178-3-1), (178-6-1), dan (178-9-1) serta variasi *learning rate* (α) sebesar 0.1, 0.01, dan 0.001. Menunjukkan bahwa akurasi tertinggi sebesar 98.63% dicapai oleh model arsitektur (178-9-1). Model ini menggunakan *learning rate* 0.1 dengan rasio pembagian data latih dan data uji sebesar 80:20. Hasil evaluasi lainnya mencatat nilai presisi sebesar 98.27%, *recall* 99.01%, dan *F1-score* 98.64%. Di sisi lain, akurasi terendah sebesar 64.98% diperoleh dari arsitektur (178-3-1) dengan *learning rate* 0.001 serta pembagian data latih dan data uji sebesar 90:10. Evaluasi model ini menunjukkan nilai presisi 67.07%, *recall* 58.89%, dan *F1-score* 62.40%.

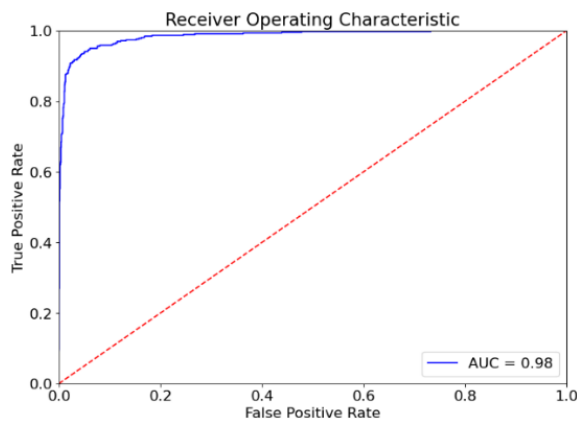
Secara keseluruhan, seluruh arsitektur memiliki rentang akurasi 98% hingga 73%, dengan satu pengecualian akurasi terendah yaitu 64.98% pada model (178-3-1) dengan parameter *learning rate* 0.001 dan pembagian data 90:10. Dibandingkan dengan skema lain, pengujian yang menggunakan rasio data 70:30 menghasilkan tingkat akurasi yang lebih stabil dan rata-rata lebih tinggi dibandingkan skema lain. Sementara itu, pengaruh *learning rate* terhadap akurasi akhir tidak menunjukkan pola yang konsisten, sehingga dapat disimpulkan bahwa variasi hasil lebih dipengaruhi oleh struktur arsitektur model.



Gambar 5. Confusion Matrix

Berdasarkan *confusion matrix* pada Gambar 5, merupakan performa model arsitektur terbaik (178-9-1) dengan pembagian data 80:20 dan *learning rate* 0.1. Ini bagian performa berdasarkan nilai aktual (y_{test})

dan prediksi (y_{pred_test}). Dapat dianalisa bahwa dari 2.300 sampel setelah dilatih dengan *train data* terdapat 1.791 diklasifikasi dengan benar sebagai non-epilepsi. Model salah memprediksi 49 sampel non-epilepsi sebagai penderita epilepsi. Lalu, model salah memprediksi 42 sampel epilepsi sebagai non-epilepsi. Namun, untuk model dengan benar dapat mengklasifikasi 418 sampel sebagai penyakit epilepsi.



Gambar 6. Kurva ROC-AUC

Untuk Gambar 6, dilengkapi dengan kurva ROC adalah sebuah grafik yang menggambarkan kemampuan sistem klasifikasi biner dalam melakukan diagnosis. Pada penelitian ini kurva ROC mendekati sudut kiri atas, menunjukkan model memiliki *False Positive Rate* (FPR) rendah dan *True Positive Rate* (TPR) tinggi.

Terdapat berbagai indikator penilaian dalam bidang klasifikasi, seperti akurasi dan *Area Under The Curve* (AUC). Pengujian AUC juga berfungsi untuk memberikan hasil yang lebih rinci dibandingkan dengan pengujian menggunakan akurasi. Hasil penelitian AUC mengukur seberapa baik model membedakan antara kelas positif dan negatif. 0.98 berarti model memiliki kemampuan 98% untuk membedakan antara kelas positif dan negatif. Kemampuan model untuk membedakan pada penelitian ini dalam kategori 0.90-1.00 yaitu *excellent classification*.

5. KESIMPULAN DAN SARAN

Berdasarkan tahapan dan alur kerja *data mining* untuk mendeteksi kejang epilepsi memiliki proses baik dari awal hingga akhir. Metode *Backpropagation Neural Network* telah berhasil digunakan, terdapat 1.791 yang diprediksi sebagai non-epilepsi dan 418 diprediksi sebagai epilepsi. Hasil implementasi algoritma *Backpropagation Neural Network* pada proses mengoptimalkan klasifikasi dan deteksi yang termasuk dalam epilepsi menggunakan data yang belum pernah dilihat oleh data latih. Diawali oleh, 9.200 data latih dan 2.300 data uji perbandingan data 80:20 memperoleh tingkat akurasi tinggi dibandingi dengan arsitektur yang sudah dipilih oleh peneliti, dengan hasil akurasi 98.63% dengan menggunakan parameter laju pembelajaran sebesar 0.1, jumlah

neuron pada *hidden layer* yaitu 9 *neuron*, jumlah *epoch* pelatihan 100. Hasil ini menunjukkan bahwa nilai prediksi kejang epilepsi mendekati nilai target yang diinginkan. Proses performa nilai target pada data uji menunjukkan model memiliki kemampuan 98% untuk membedakan antara kelas positif dan negatif. Oleh karena itu bagian penelitian ini termasuk dalam kategori *excellent classification* saat dilihat dari perhitungan AUC. Dan, menurunkan tingkat kesalahan dalam diagnosis dan pemantauan kejang epilepsi dan non-epilepsi. Selanjutnya, berdasarkan hasil yang diperoleh dari pengembangan model klasifikasi kejang epilepsi ini, disarankan agar penelitian ke depan dapat merancang mesin EEG yang mampu mengintegrasikan pemodelan ini dengan perangkat *Internet of Things* (IoT), seperti *Raspberry Pi* sehingga dapat diterapkan secara nyata dalam bidang kesehatan baik di rumah sakit umum maupun rumah sakit dengan multidisiplin ilmu.

DAFTAR PUSTAKA

- [1] Improving the lives of people with epilepsy: a technical brief. Geneva: World Health Organization; 2022. Licence: CC BY-NC-SA 3.0 IGO.
- [2] C. Linehan *et al.*, "Exploring the prevalence and profile of epilepsy across Europe using a standard retrospective chart review: Challenges and opportunities," *Epilepsia*, vol. 62, no. 11, pp. 2651–2666, Nov. 2021, doi: 10.1111/epi.17057.
- [3] S. M. Zuberi *et al.*, "ILAE Classification and Definition of Epilepsy Syndromes With Onset in Neonates and Infants: Position Statement by the ILAE Task Force on Nosology and Definitions," *Epilepsia*, vol. 63, no. 6, pp. 1349–1397, 2022, doi: 10.1111/epi.17239.
- [4] J. L. Hopp *et al.*, "Peri-Ictal Changes in Depression and Anxiety in Persons With Epileptic and Non-Epileptic Seizures," *Front Psychiatry*, vol. 13, 2022, doi: 10.3389/fpsyt.2022.912697.
- [5] H. Morris, P. W. Kaplan, and N. Kane, "Electroencephalography in encephalopathy and encephalitis," *Pract Neurol*, vol. 24, no. 1, pp. 2–10, 2024, doi: 10.1136/pn-2023-003798.
- [6] P. N. Napitupulu, A. R. Damanik, and Jessica Evonella Napitupulu, "Implementasi Algoritma Backpropagation Jaringan Syaraf Tiruan Untuk Prediksi Angka Harapan Hidup Di Kota Jambi," *JPILOM (Jurnal Penelitian Ilmu Komputer)*, vol. 1, no. 1, pp. 10–15, Sep. 2023.
- [7] P. N., "Robust Prior Stage Epileptic Seizure Diagnosis System using Resnet and Backpropagation Techniques," *International Journal of Emerging Trends in Engineering Research*, vol. 8, no. 5, pp. 2214–2222, May 2020, doi: 10.30534/ijeter/2020/120852020.
- [8] N. A. B. A. Khan and S. M. Marjugi, "Prediction Analysis Of Covid-19 In Selangor By Using Backpropagation Algorithm With Conjugate

- Gradient Method,” *Journal of Quality Measurement and Analysis*, vol. 20, no. 1, pp. 187–204, Mar. 2024, doi: 10.17576/jqma.2001.2024.14.
- [9] A. W. Putri, “Implementasi Artificial Neural Network (ANN) Backpropagation Untuk Klasifikasi Jenis Penyakit Pada Daun Tanaman Tomat,” *MATHunesa: Jurnal Ilmiah Matematika*, vol. 9, no. 2, pp. 344–350, 2021, doi: 10.26740/mathunesa.v9n2.p344-350.
- [10] M. Arhami, M. Kom, and S. T. Muhammad Nasir, *Data Mining-Algoritma dan Implementasi*. Penerbit Andi, 2020.
- [11] R. Saputra, S. Sunardiyo, A. Nugroho, and S. Subiyanto, “Implementasi Multilayer Perceptron Artificial Neural Network untuk Prediksi Konsumsi Energi Listrik PT PLN (Persero) UP3 Salatiga,” *Elektrika*, vol. 15, no. 2, p. 60, 2023, doi: 10.26623/elektrika.v15i2.6411.
- [12] I. N. Dita Suardinata, N. Widya Utami, and H. Syakh Alam, “Prediksi Potensi Kebangkrutan Pada Perusahaan Yang Terdaftar Di Bei Menggunakan Algoritma Backpropagation Neural Network,” *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 8, no. 1, pp. 748–755, 2024, doi: 10.36040/jati.v8i1.8826.