

Implementasi Metode Recurrent Neural Network Untuk Prediksi Kejang Pada Penderita Epilepsi Berdasarkan Data Electroencephalogram

Raisya Febiyane¹, Yulison Harry Chrisnanto², Gunawan Abdillah^{3,*}

Fakultas Sains Dan Informatika, Program Studi Informatika, Universitas Jenderal Achmad Yani, Cimahi, Indonesia

Email: ¹raisyaafff@gmail.com, ²yhc@if.unjani.ac.id @gmail.com, ^{3,*}gunawanabdillah03@gmail.com

Email Penulis Korespondensi: yhc@if.unjani.ac.id

Submitted 23-05-2025; Accepted 30-06-2025; Published 30-06-2025

Abstrak

Epilepsi adalah gangguan neurologis kronis yang menyebabkan penderita mengalami kejang berulang. Kejang merupakan salah satu gejala utama epilepsi, yang memerlukan penanganan medis dan pengawasan yang ketat. Tantangan utama dalam penanganan epilepsi adalah sulitnya memprediksi kapan kejang akan terjadi. Electroencephalogram (EEG) dapat mendeteksi kejang karena mengandung informasi fisiologis aktivitas saraf otak. Penelitian ini bertujuan untuk memprediksi kejang dengan menggunakan metode Recurrent Neural Network berdasarkan data EEG. Deep Learning merupakan salah satu cabang dari Machine Learning yang menggunakan jaringan saraf tiruan untuk menyelesaikan masalah dengan dataset yang besar. Data yang digunakan pada penelitian ini dataset Epileptic Seizure Recognition yang diambil dari Kaggle. Terdiri dari atribut ID Pasien, 178 atribut numerik mewakili sinyal EEG, dan label y yang menunjukkan kondisi selama rekaman yang terdiri dari mata terbuka, mata tertutup, otak yang sehat, tempat tumor berada, dan aktivitas kejang. Model deep learning yang diuji merupakan Recurrent Neural Network (RNN) yang dirancang untuk mempelajari pola dalam data. Evaluasi kinerja dilakukan dengan menggunakan evaluasi matrik yang terdiri dari akurasi, presisi, recall dan F1-Score. Berdasarkan penerapan metode RNN dan pengujian menggunakan data EEG, diperoleh kondisi terbaik dengan arsitektur Long Short-Term Memory yang terdiri dari 3 layer dan parameter pelatihan yang optimal, menghasilkan akurasi prediksi kejang sebesar 98,6%. Hasil ini menunjukkan bahwa model mampu memprediksi kemungkinan terjadinya kejang secara efektif dan efisien.

Kata Kunci: Deep Learning; RNN; EEG; Kejang; Epilepsi

Abstract

Epilepsy is a chronic neurological disorder that causes patients to experience recurrent seizures. Seizures are one of the main symptoms of epilepsy, requiring medical treatment and close monitoring. A major challenge in epilepsy management is the difficulty in predicting when seizures will occur. Electroencephalogram (EEG) can detect seizures as it contains physiological information about brain neural activity. This study aims to predict seizures using a Recurrent Neural Network (RNN) method based on EEG data. Deep Learning is a branch of Machine Learning that uses artificial neural networks to solve problems involving large datasets. The data used in this research is the Epileptic Seizure Recognition dataset obtained from Kaggle. It consists of patient ID attributes, 178 numerical attributes representing EEG signals, and a label y indicating conditions during the recording, including eyes open, eyes closed, healthy brain, tumor location, and seizure activity. The deep learning model tested is a Recurrent Neural Network (RNN) designed to learn patterns in the data. Performance evaluation was conducted using metrics including accuracy, precision, recall, and F1-Score. Based on the application of the RNN method and testing using EEG data, the best condition was achieved with a three-layer Long Short-Term Memory architecture and optimal training parameters, resulting in a seizure prediction accuracy of 98.6%. This result demonstrates that the model is capable of effectively and efficiently predicting the likelihood of seizure occurrences.

Keywords: Deep Learning; RNN; EEG; Seizure; Epilepsy

1. PENDAHULUAN

Epilepsi merupakan penyakit otak kronis tidak menular yang dapat menurunkan kualitas hidup dan menimbulkan biaya yang besar [1]. Epilepsi juga dikenal dengan sebutan ayan yang ditandai dengan kejang yang sering berulang tanpa adanya pemicu yang jelas dan dapat menyebabkan kehilangan kesadaran [1]. Mekanisme yang mendasari kejang epilepsi adalah aktivitas saraf yang berlebihan dan abnormal pada korteks otak. Mengapa hal ini terjadi pada sebagian besar kasus epilepsi tidak jelas, dalam beberapa kasus hal ini dapat disebabkan oleh cedera otak, stroke tumor otak, infeksi otak, atau cacat lahir melalui proses yang dikenal sebagai epileptogenesis [2]. diperkirakan sekitar 50 juta orang di dunia menderita epilepsi dan 80% diantaranya tinggal di negara berkembang [3].

Seiring berkembang pesatnya teknologi di bidang medis, berbagai metode telah dikembangkan untuk mendeteksi dan memprediksi kejadian kejang dari data EEG pasien epilepsi. Salah satunya adalah *Deep Learning*, yaitu jenis machine learning yang menggunakan model artificial neural network. Salah satu kelebihan deep learning dibandingkan dengan machine learning lainnya adalah kemampuannya yang lebih baik dalam menganalisis data yang tidak terstruktur, seperti suara dan gambar digital. Contoh masalah yang dapat diatasi dengan model deep learning meliputi klasifikasi, regresi, dan pengelompokan data [4]. Salah satu arsitektur Deep Learning adalah Recurrent Neural Network (RNN), RNN merupakan algoritma machine learning yang sangat efektif untuk memproses data dalam bentuk time series dan dikenal memiliki tingkat akurasi tinggi dalam prediksi. [5].

Beberapa penelitian terdahulu yang menjadi referensi dalam penelitian ini melibatkan lima studi terkait prediksi kejang, masing – masing menggunakan metode dan hasil yang berbeda.

Penelitian pertama oleh K. Alalayah, E. Senan, H. Atlam et al. (2023) menggunakan beberapa metode seperti Discrete Wavelet Transform (DWT), Principal Component Analysis (PCA), t-Distributed Stochastic Neighbor Embedding (t-SNE), K-Means, XGBoost, KNN, Decision Trees, Random Forest dan Multi-Layer Perceptron (MLP). Hasil Penelitian menunjukkan model yang menggunakan kombinasi DWT dan PCA dengan Random Forest mencapai akurasi 97.96%, sedangkan pengklasifikasi Random Forest dengan DWT dan t-SNE dengan mencapai akurasi 98.09%. Sebagai perbandingan pengklasifikasi MLP dengan PCA dan K-Means mencapai akurasi tertinggi sebesar 98.98%. Namun penelitian ini fokus pada berbagai teknik ekstraksi fitur dan reduksi dimensi [6]. Perbedaan utama dari penelitian ini adalah penggunaan model Recurrent Neural Network (RNN) untuk meningkatkan kemampuan dalam mengenali pola data time series pada EEG.

Penelitian kedua oleh V. Gabeff et al. (2021) dengan menggunakan metode Deep Learning dengan model Convolutional Neural Networks (CNN) untuk deteksi kejang epilepsi secara daring dari sinyal EEG, hasil penelitian menunjukkan bahwa model mencapai F1-Score sebesar 0.873 dan berhasil mendeteksi 90% kejadian kejang [7]. Perbedaan utama dari penelitian ini adalah bahwa penelitian saat ini menggunakan deep learning (RNN) untuk memprediksi kejang dari sinyal EEG.

Penelitian ketiga oleh S. Mounika dan S. R. Reeja (2024) menerapkan metode XGBoost dengan teknik SMOTE untuk deteksi kejang epilepsi. Hasil penelitian menunjukkan bahwa metode tersebut berhasil mencapai Tingkat akurasi 98,7%. Namun, penelitian ini hanya menggunakan data dari 14 pasien saja [8]. Sementara itu, penelitian ini menggunakan model Recurrent Neural Network (RNN) yang lebih cocok untuk data time series dan diuji pada dataset yang lebih besar.

Penelitian keempat oleh Mulaab (2021) mengembangkan pendekatan deteksi epilepsi berbasis seleksi fitur Information Gain dan pengklasifikasi Ensemble Random Forest. Hasilnya menunjukkan bahwa dengan pemilihan fitur tertentu dari sinyal EEG pada beberapa subjek pasien, model ini mampu mencapai akurasi di atas 0.99. Namun, penelitian ini terbatas pada penggunaan dataset yang relatif kecil [9]. Perbedaan utama dari penelitian ini adalah menggunakan Recurrent Neural Network (RNN) yang tidak memerlukan pemilihan fitur manual dan mampu mengenali pola temporal secara otomatis dari data EEG.

Penelitian kelima oleh O. Nurdian, A. Faqih (2024) mengoptimalkan model Backpropagation untuk mendeteksi kejang epilepsi dengan memanfaatkan data sinyal EEG yang telah diproses menggunakan Discrete Cosine Transform (DCT). Hasil penelitian menunjukkan bahwa model Backpropagation mencapai akurasi optimal sebesar 91,15% pada 100 epoch dan meningkat menjadi 93,05% pada 200 epoch. Meskipun akurasi meningkat dengan bertambahnya epoch, waktu komputasi yang lebih lama menimbulkan risiko overfitting [10].

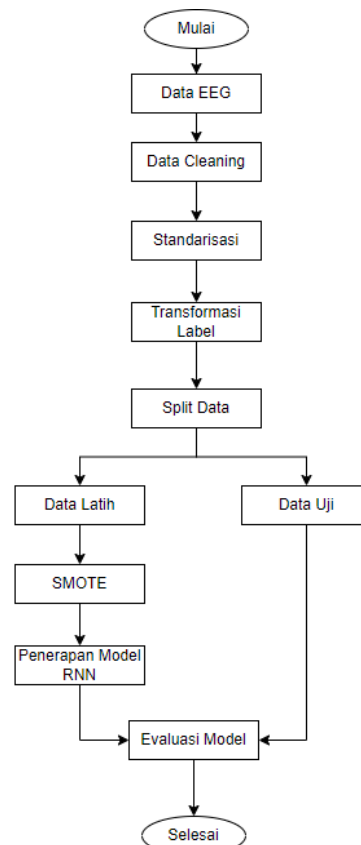
Berbeda dengan penelitian terdahulu, studi ini menggunakan Recurrent Neural Network (RNN) untuk memprediksi kejang epilepsi berdasarkan data sinyal EEG. Model RNN dipilih karena data sinyal EEG berbentuk time series, dan kelebihan RNN adalah kemampuannya menangkap informasi pada data time series melalui banyak lapisan jaringan yang memungkinkan pemodelan pola temporal menjadi lebih baik sehingga menghasilkan akurasi prediksi yang tinggi [11]. Dengan mempertimbangkan kelebihan dan kekurangan RNN, penelitian ini akan fokus pada penggunaan RNN untuk memprediksi kejadian kejang epilepsi berdasarkan data EEG.

Oleh karena itu, tujuan dari penelitian ini adalah untuk mengimplementasikan model Recurrent Neural Network (RNN) dalam memprediksi kejang epilepsi berdasarkan data sinyal EEG. Penelitian ini akan mengimplementasikan arsitektur RNN pada dataset EEG *Epileptic Seizure Recognition* yang tersedia di situs kaggle, mengevaluasi performa model menggunakan metrik akurasi, precision, recall, dan F1-score. Dengan memanfaatkan karakteristik RNN yang mampu mengenali pola temporal pada data time series, diharapkan model ini dapat memberikan prediksi yang akurat dalam membedakan kondisi kejang dan tidak kejang.

2. METODOLOGI PENELITIAN

2.1 Tahapan Penelitian

Berikut ini merupakan gambar diagram alur pada tahap penelitian ini adalah sebagai berikut:



Gambar 1. Tahapan Penelitian

Berikut adalah penjelasan rinci untuk setiap tahapan dalam penelitian ini:

a. Persiapan Data

Data yang digunakan dalam penelitian ini diambil dari website Kaggle.com. Data yang diperoleh dari website tersebut berformat CSV dan terdapat rekaman aktivitas otak (EEG). Data direkam selama 23,6 detik dan dipecah menjadi 4097 titik data. Data ini memiliki 11.500 baris data, di mana setiap baris menggambarkan sinyal EEG pada waktu tertentu dan terdiri dari 178 fitur yang merekam aktivitas otak pada detik tersebut. Kolom terakhir berisi label yang menunjukkan kondisi saat rekaman dilakukan. Label ini terdiri dari lima kategori, yaitu label 1 Kejang, label 2 Rekaman EEG dari lokasi tumor, label 3 Rekaman dari bagian otak yang sehat, label 4 Mata tertutup saat direkam, dan label 5 Mata terbuka saat direkam. Untuk memberikan gambaran mengenai struktur awal data EEG, ditampilkan pada Tabel 1 berikut:

Tabel 1. Data EEG

unnamed	X1	X2	X3	...	X176	X177	X178	y
X21.V1.791	135	190	229	...	-116	-83	-51	4
X15.V1.924	386	382	356	...	154	143	129	1
X8.V1.1	-32	-39	-47	...	-35	-35	-36	5
...	...	...	...	...	...	...	...	...
X8.V1.28	14	6	-13	...	-2	-1	-8	5
X10.V1.932	-40	-25	-9	...	68	59	55	3
X16.V1.210	29	41	57	...	-2	2	20	4

Berdasarkan Tabel 1 di atas, dapat dilihat bahwa struktur data terdiri dari beberapa komponen utama. Kolom pertama unnamed berisi id untuk setiap rekaman EEG yang mengikuti format seperti X21.V1.791, X15.V1.924, dan seterusnya. Kolom X1 hingga X178 merepresentasikan 178 fitur sinyal EEG yang merekam aktivitas listrik otak pada setiap titik waktu, dengan nilai-nilai yang dapat berupa angka positif maupun negatif sesuai dengan karakteristik sinyal EEG. Kolom terakhir 'y' merupakan variabel target yang berisi label dengan nilai 1 hingga 5, dimana setiap angka merepresentasikan kondisi spesifik saat rekaman EEG dilakukan.

b. Data Cleaning

Pada tahap ini dilakukan beberapa proses seperti identifikasi dan penanganan missing value, Menghapus kolom yang tidak digunakan, penerapan metode normalisasi min-max untuk mengubah nilai – nilai dalam dataset agar berada dalam rentang tertentu [12].

c. Transformasi Label

Transformasi data adalah merupakan proses mengubah data ke dalam bentuk yang sesuai dengan kebutuhan untuk siap diproses [13]. Pada data awal yang memiliki beberapa kelas seperti 1, 2, 3, 4, dan 5 diubah menjadi dua kelas saja, yaitu 0 untuk tidak kejang dan 1 untuk kejang.

d. Split Data

Pada tahap ini data dibagi menjadi dua yaitu data training dan data testing dengan tujuan untuk melatih serta menguji model. Data penelitian ini, proporsi data training dan data testing adalah 80%:20%.

e. SMOTE

Pada tahap ini dilakukan SMOTE untuk mengatasi ketidakseimbangan data. SMOTE (Synthetic Minority Over-sampling Technique) merupakan teknik oversampling yang digunakan untuk mengatasi ketidakseimbangan kelas dalam data dengan menyeimbangkan jumlah antara kelas mayoritas dan minoritas [14].

f. Model RNN

Penelitian ini menggunakan model Recurrent Neural Network (RNN) yang terdiri dari tiga lapisan secara berurutan dengan jumlah unit masing-masing 64, 32, dan 16. Setiap lapisan diikuti dengan lapisan Dropout untuk mengurangi risiko overfitting, masing-masing sebesar 25%, 25%, dan 50%. Model ini memiliki total 32.465 parameter yang dapat dilatih. Proses pelatihan model direncanakan hingga maksimum 100 epoch namun dapat berhenti lebih awal jika tidak terjadi peningkatan pada validation loss selama 10 epoch berturut-turut (early stopping).

g. Evaluasi Model

Model yang dihasilkan dari pelatihan menggunakan metode RNN dievaluasi dengan menggunakan Beberapa metrik yang digunakan untuk mengevaluasi model adalah sebagai berikut:

1. Confusion Matrix: yang memberikan gambaran yang jelas tentang seberapa baik model memprediksi kejang dan tidak kejang. Struktur umum confusion matrix yang digunakan dapat dilihat pada Tabel 2 dibawah ini.

Tabel 2. Confusion Matrix

		Predicition Class	
		+	-
Actual Class	+	True Positive (TP)	False Negative (FN)
	-	False Positive (FP)	True Negative (TN)

Keterangan:

True Positive (TP) : Jumlah data positif yang terdeteksi dengan benar.

True Negative (TN) : Jumlah data negatif yang terdeteksi dengan benar.

False Positive (FP) : Jumlah data negatif yang terdeteksi sebagai data positif.

False Negative (FN) : Jumlah data positif yang terdeteksi sebagai data negatif.

2. Akurasi: Akurasi merupakan ukuran utama untuk menilai seberapa efektif sebuah model dalam menghasilkan prediksi yang tepat. Ukuran ini menunjukkan seberapa banyak prediksi yang benar, baik berupa True Positives maupun True Negatives, dari seluruh sampel yang diuji [15].

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

3. Precision: Precision merupakan sebuah metrik dalam model klasifikasi yang digunakan untuk mengukur sejauh mana prediksi yang dihasilkan oleh model tersebut tepat atau akurat [16].

$$Precision = \frac{TP}{TP+FP} \quad (2)$$

4. Recall: Recall merupakan rasio antara jumlah prediksi positif yang benar dengan total data yang sebenarnya positif [17].

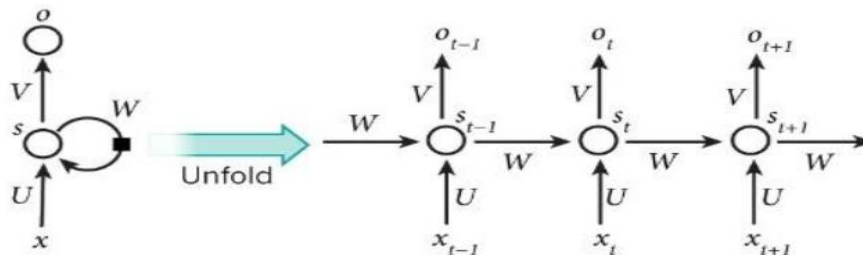
$$Recall = \frac{TP}{TP+FN} \quad (3)$$

5. F1-Score: f1-score memberikan gambaran tentang kemampuan model dengan menggabungkan rata-rata precision dan recall yang telah diberi bobot secara seimbang.

$$F1\ Score = \frac{2(Recall+Precision)}{(Recall+Precision)} \quad (4)$$

2.2 Metode RNN

Metode RNN digunakan dalam penelitian ini untuk memprediksi kejang berdasarkan data EEG. Recurrent Neural Network (RNN) merupakan algoritma dalam deep learning yang dapat mengolah data berurutan atau time series [18]. Pada dasarnya Recurrent Neural Network (RNN) memiliki parameter yang sama dengan jaringan neural biasa. Perbedaannya terletak pada cara kerjanya, dimana RNN menggunakan hidden layer sebelumnya sebagai input untuk langkah berikutnya. Hidden layer tersebut menyimpan informasi hasil ekstraksi fitur sebelumnya, sehingga saat digunakan kembali, ia menyimpan memory (ingatan) dari input sebelumnya [19].



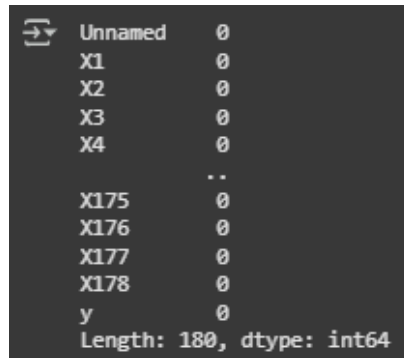
Gambar 2. Arsitektur RNN

Pada gambar 2 menunjukkan proses awal algoritma RNN sebelum masuk (unrolled) kedalam jaringan penuh (full network). Setelah itu, memperlihatkan bentuk RNN yang telah dibuka (full network) sehingga alur prosesnya tampak lebih rinci. Tanda panah biru menunjukkan hidden layer, dan simbol (st) menunjukkan hidden state pada setiap waktu (t). Hidden state ini berfungsi sebagai “memori” jaringan untuk menyimpan hasil perhitungan sebelumnya. Sementara itu, Ot merupakan output pada waktu (t) yang digunakan untuk memprediksi data berikutnya. [20].

3. HASIL DAN PEMBAHASAN

3.2 Data Cleaning

Proses Data Cleaning atau pembersihan data, yaitu proses menghilangkan noise dan data yang tidak digunakan. Data yang diperoleh masih terdapat nilai-nilai kosong atau atribut yang tidak dibutuhkan dalam proses analisis. Oleh karena itu, dilakukan proses pembersihan untuk menghilangkan bagian-bagian yang tidak diperlukan agar data menjadi lebih rapi dan terstruktur. Berikut merupakan hasil proses data cleaning pada gambar 3.



```

Unnamed      0
X1            0
X2            0
X3            0
X4            0
...          ..
X175          0
X176          0
X177          0
X178          0
y             0
Length: 180, dtype: int64

```

Gambar 3. Hasil Data Cleaning

Gambar 3 menunjukkan bahwa proses pembersihan data telah berhasil dilakukan. Berdasarkan hasil pembersihan, tidak ditemukan adanya nilai kosong atau *null* dalam dataset, sehingga tidak diperlukan penanganan data hilang. Data yang digunakan sudah tidak ada nilai *null* atau kosong, sehingga siap untuk tahap pemrosesan selanjutnya.

3.3 Standarisasi

Standarisasi merupakan teknik yang sering diterapkan sebagai bagian dari persiapan data untuk pembelajaran mesin, tujuannya adalah untuk mengubah nilai pada kolom numerik ke dalam skala standar, yaitu dengan mengatur nilai – nilai tersebut agar memiliki rata – rata 0 dan standar deviasi 1, tanpa menghilangkan perbedaan antar nilai. Pada tahap ini dilakukan standarisasi dengan menggunakan metode *StandardScaler* dengan rumus:

$$Z = \frac{x_i - \bar{X}}{\sigma} = \frac{135 - 11.5814}{165.6191} = \frac{146.5814}{165.6191} = 0.88505130159 \quad (5)$$

Dimana:

$\bar{X}$ = Rata – rata nilai sampel

σ = Standar Deviasi

Setelah proses standarisasi dilakukan pada seluruh dataset, diperoleh data dengan skala yang sama. Hasil dari proses standarisasi ini dapat dilihat pada Tabel 3 di bawah ini.

Tabel 3. Hasil Standarisasi Data

Unnamed	X1	X2	X3	...	X177	X178	y
X21.V1.791	0.885051	1.209929	1.462764	..	-0.433290	-0.235399	4
X15.V1.924	2.400577	2.366190	2.239441	..	0.954241	0.856537	1
X8.V1.1	-0.123287	-0.169154	-0.225131	...	-0.138593	-0.144405	5
...	...	...	...	...	...	...	...
X8.V1.28	0.154459	0.101845	-0.017202	...	0.070150	0.025452	5
X10.V1.932	-0.171590	-0.084843	0.007260	...	0.438521	0.407630	3
X16.V1.210	0.245028	0.312622	0.410887	...	0.088569	0.195309	4

Berdasarkan Tabel 3 di atas, dapat diamati bahwa setelah proses standarisasi, nilai-nilai pada setiap fitur X1 hingga X178 telah ditransformasi ke dalam skala yang seragam. Nilai-nilai yang ditampilkan berkisar di sekitar 0 dengan variasi yang relatif kecil, yang menunjukkan bahwa proses standarisasi telah berhasil diterapkan. Sebagai contoh, pada sampel X15.V1.924, nilai fitur X1 sebesar 2.400577 menunjukkan bahwa nilai asli fitur tersebut berada sekitar 2.4 standar deviasi di atas rata-rata. Sebaliknya, nilai negatif seperti -0.433290 pada fitur X177 untuk sampel X21.V1.791 menunjukkan bahwa nilai asli berada di bawah rata-rata. Kolom y tetap mempertahankan nilai aslinya 1 hingga 5 karena merupakan variabel target yang tidak perlu distandarisasi.

3.4 Transformasi Label

Pada tahap ini, dilakukan transformasi label. Data awal terdapat 5 label, yaitu label 1 untuk kondisi Kejang, label 2 untuk Rekaman EEG dari lokasi tumor, label 3 untuk Rekaman dari bagian otak yang sehat, label 4 untuk Mata tertutup saat direkam, dan label 5 untuk Mata terbuka saat direkam. Karena fokus penelitian ini adalah untuk memprediksi terjadinya kejang, maka dilakukan penyederhanaan kategori label. Label 1 yang merepresentasikan kondisi kejang dipertahankan sebagai kelas kejang, sedangkan label 2 hingga 5 yang merepresentasikan kondisi non-kejang digabungkan dan diubah menjadi label 0 sebagai kelas tidak kejang. Transformasi ini mengubah permasalahan dari multi-class classification dengan 5 kelas menjadi binary classification dengan 2 kelas. Hasil dari proses transformasi label tersebut dapat dilihat pada Tabel 4 berikut ini, yang menunjukkan struktur data setelah dilakukan perubahan kategori label.

Tabel 4. Hasil Transformasi Label

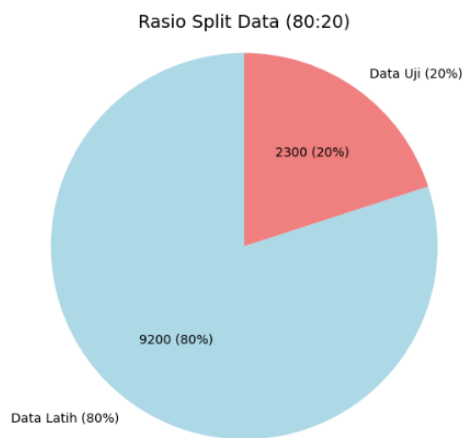
Unnamed	X1	X2	X3	...	X177	X178	y
---------	----	----	----	-----	------	------	---

X21.V1.791	0.885051	1.209929	1.462764	..	-0.433290	-0.235399	0
X15.V1.924	2.400577	2.366190	2.239441	..	0.954241	0.856537	1
X8.V1.1	-0.123287	-0.169154	-0.225131	...	-0.138593	-0.144405	0
...	...	...	...	...	...	...	...
X8.V1.28	0.154459	0.101845	-0.017202	...	0.070150	0.025452	0
X10.V1.932	-0.171590	-0.084843	0.007260	...	0.438521	0.407630	0
X16.V1.210	0.245028	0.312622	0.410887	...	0.088569	0.195309	0

Berdasarkan Tabel 4 di atas, dapat dilihat bahwa kolom 'y' merupakan hasil transformasi label yang telah diubah menjadi nilai biner, dimana nilai 0 menunjukkan kondisi tidak kejang dan nilai 1 menunjukkan kondisi kejang. Transformasi ini memungkinkan model untuk fokus dalam mengenali dan memprediksi kejadian kejang.

3.5 Split Data

Pembagian data dalam penelitian ini dilakukan dengan proporsi 80% untuk data latih dan 20% untuk data uji. Pembagian ini dilakukan menggunakan fungsi `train_test_split` dari library Scikit-Learn. Pada gambar 4 memperlihatkan visualisasi dari hasil pembagian data, termasuk jumlah total data yang masuk ke dalam data uji dan data latih.

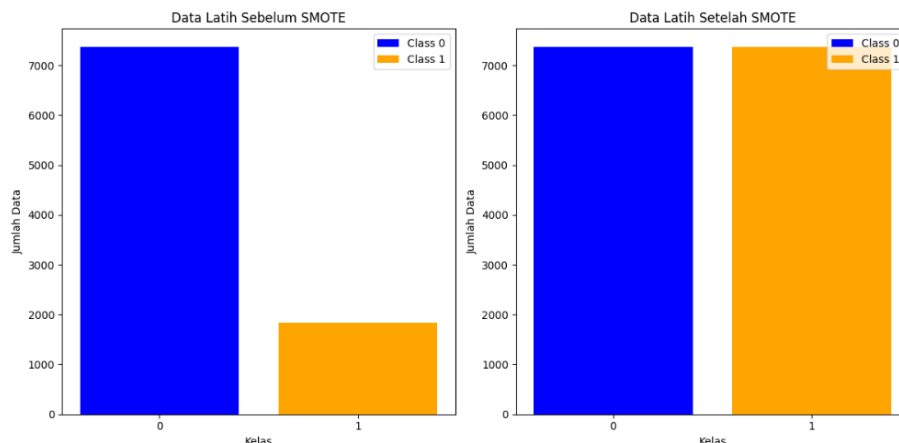


Gambar 4. Hasil Split Data

Gambar 4 menunjukkan bahwa dari seluruh data yang tersedia, sebanyak 9200 data atau 80% digunakan untuk melatih model, sedangkan sisanya yaitu 2300 data atau 20% digunakan sebagai data uji untuk mengevaluasi performa model.

3.6 SMOTE

Pada tahap ini dilakukan penanganan ketidakseimbangan kelas pada data latih dengan menggunakan metode SMOTE (Synthetic Minority Over-sampling Technique) untuk menyeimbangkan jumlah data antar kelas. Sebelum dilakukan SMOTE jumlah data latih untuk kelas 0 sebanyak 7360, sedangkan kelas 1 hanya 1840. Setelah proses SMOTE diterapkan, jumlah data pada kedua kelas menjadi seimbang, yaitu masing-masing sebanyak 7360 data.



Gambar 5. Hasil SMOTE

Gambar 5 menunjukkan perbandingan distribusi kelas pada dataset sebelum dan sesudah penerapan SMOTE. Bagian kiri gambar memperlihatkan kondisi awal dataset yang tidak seimbang dengan dominasi kelas 0 sebanyak 7360

dibandingkan kelas 1 sebanyak 1840 data. Bagian kanan gambar menunjukkan hasil setelah penerapan SMOTE, di mana kedua kelas memiliki jumlah sampel yang sama, yaitu 7360 data untuk masing-masing kelas. Berikut merupakan hasil SMOTE pada gambar 5.

3.7 RNN

Arsitektur model RNN dalam penelitian ini dirancang untuk memprediksi kejang berdasarkan data EEG. Model RNN dibangun dengan tiga lapisan arsitektur LSTM yang masing - masing memiliki 64, 32, dan 16 unit. Setiap lapisan diikuti dengan layer Dropout dengan persentasi 0,25 pada dua lapisan pertama dan 0,5 pada lapisan terakhir. Pada akhir jaringan, digunakan lapisan Dense dengan satu neuron untuk menghasilkan output klasifikasi biner. Total parameter model sebanyak 32,465, seluruhnya dapat dilatih (trainable). pada output layer digunakan sigmoid karena model ditujukan untuk klasifikasi biner. Optimizer yang digunakan adalah Adam dengan loss function berupa Binary Crossentropy. Pelatihan direncanakan hingga 100 epoch, namun dapat berhenti lebih awal jika validation loss tidak membaik selama 10 epoch melalui mekanisme early stopping. Untuk memberikan gambaran yang lebih jelas mengenai arsitektur model RNN yang digunakan, berikut disajikan detail struktur lapisan beserta parameter-parameter yang terlibat dalam setiap komponen model.

Tabel 5. Model RNN

Layer (Type)	Output Shape	Param #
lstm (LSTM)	(None, 178, 64)	16,896
dropout (Dropout)	(None, 178, 64)	0
lstm_1 (LSTM)	(None, 178, 32)	12,416
dropout_1 (Dropout)	(None, 178, 32)	0
lstm_2 (LSTM)	(None, 16)	3,136
dropout_2 (Dropout)	(None, 16)	0
dense (Dense)	(None, 1)	17

Total Parameter: 32,465 (126.82 KB)

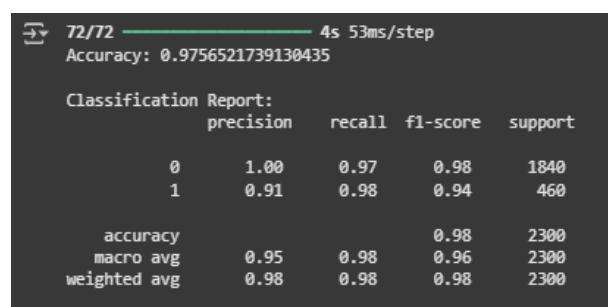
Trainable Parameter: 32,465 (126.82 KB)

Non-Trainable Parameter: 0 (0.00 B)

Tabel 5 menunjukkan bahwa model RNN dibangun menggunakan tiga lapisan utama LSTM yang memiliki jumlah unit berbeda-beda. Lapisan pertama memiliki 64 unit dan 16.896 parameter, lalu dilanjutkan dengan dropout untuk mencegah overfitting. Lapisan kedua memiliki 32 unit dengan 12.416 parameter, juga disertai dropout. Lapisan ketiga memiliki 16 unit dan 3.136 parameter, serta dropout terakhir. Di akhir model, terdapat lapisan dense dengan 1 neuron dan 17 parameter yang digunakan untuk menghasilkan output berupa dua kelas (kejang dan tidak kejang). Nilai None pada output shape menandakan model bisa menerima jumlah data input yang bervariasi. Angka 178 menunjukkan panjang sekuens EEG tiap sampel, sementara ukuran unit 64, 32, dan 16 menggambarkan tingkat kompleksitas representasi fitur pada tiap lapisan LSTM.

3.8 Evaluasi Model

Evaluasi model dilakukan untuk mengetahui sejauh mana kemampuan model dalam memprediksi kondisi kejang dan tidak kejang berdasarkan sinyal EEG. Metode evaluasi yang digunakan mencakup akurasi, precision, recall, f1-score, dan confusion matrix. Gambar 6 menunjukkan hasil evaluasi model dalam melakukan prediksi. Model berhasil mencapai akurasi sebesar 97,56%. Hasil evaluasi menunjukkan dengan precision, recall, dan f1-score Kelas 0 mencapai precision 1.00, recall 0.97, f1-score 0.98, sedangkan kelas 1 memiliki precision 0.91, recall 0.98 dan f1-score 0.94. Hal ini menunjukkan bahwa model mampu melakukan prediksi dengan baik terhadap salah satu kelas. Hasil evaluasi model dapat dilihat pada Gambar 6 berikut.

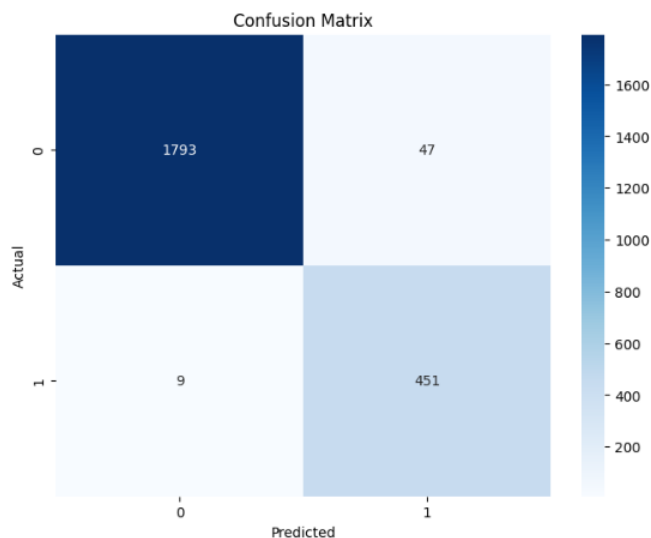


72/72 4s 53ms/step				
Accuracy: 0.9756521739130435				
Classification Report:				
	precision	recall	f1-score	support
0	1.00	0.97	0.98	1840
1	0.91	0.98	0.94	460
accuracy			0.98	2300
macro avg	0.95	0.98	0.96	2300
weighted avg	0.98	0.98	0.98	2300

Gambar 6. Hasil Evaluasi Model

Selanjutnya, untuk memberikan gambaran mengenai kinerja model dalam melakukan prediksi kelas, digunakan confusion matrix yang divisualisasikan pada Gambar 7. Model berhasil memprediksi sebagian besar data dengan benar, yaitu sebanyak 1793 data kelas 0 berhasil diprediksi dengan benar sebagai kelas 0 (*True Negative*) dan 451 data kelas 1

berhasil diprediksi dengan benar sebagai kelas 1 (*True Positive*). Jumlah kesalahan prediksi sebanyak 47 data kelas 0 yang terprediksi sebagai kelas 1 (*false positive*), dan 9 data kelas 1 yang terprediksi sebagai kelas 0 (*false negative*).



Gambar 7. Hasil Confusion Matrix

4. KESIMPULAN

Penelitian ini telah berhasil mengimplementasikan model Recurrent Neural Network (RNN) untuk memprediksi kejang epilepsi berdasarkan data sinyal EEG dengan hasil yang sangat memuaskan. Model RNN yang dikembangkan terdiri dari tiga lapisan LSTM dengan arsitektur 64, 32, dan 16 unit neuron, dilengkapi dengan layer Dropout untuk mencegah overfitting, dan memiliki total 32.465 parameter yang dapat dilatih. Hasil evaluasi model menunjukkan kinerja yang sangat baik dengan akurasi mencapai 97,56%. Secara detail, model mampu mencapai precision 1,00, recall 0,97, dan F1-score 0,98 untuk kelas 0 (tidak kejang), serta precision 0,91, recall 0,98, dan F1-score 0,94 untuk kelas 1 (kejang). Hasil confusion matrix menunjukkan bahwa dari 2.300 data uji, model berhasil memprediksi 1793 data kelas 0 dengan benar (*True Negative*) dan 451 data kelas 1 dengan benar (*True Positive*), dengan hanya 47 kesalahan prediksi *false positive* dan 9 kesalahan prediksi *false negative*.

REFERENCES

- [1] V. M. P. Ade Rahayu, Achmad Fauzi, "Diagnosa Penyakit Epilepsi Menggunakan Metode Bayes," vol. 2, no. 4, 2024.
- [2] A. Abdullah and A. Ikhsani, "Epilepsi Bangkitan Umum Tonik-Klonik Pada Pasien Laki - Laki Berusia 22 Tahun : Laporan Kasus," *J. Kesehat. Tambusai*, vol. 2, no. 3, pp. 262–266, 2021, doi: 10.31004/jkt.v2i3.2375.
- [3] K. N. Fatmi, D. Roshinta, L. Dewi, and M. In'am Ilmiawan, "The Relation of Duration of Epilepsy, Seizure Frequency and AED Adherence With Cognitive Function in Epilepsy Patients," *J. Nas. Ilmu Kesehat.*, vol. 4, no. 2621–6507, p. 52, 2022.
- [4] M. I. Girindra, "Pengenalan Pola Sinyal Eeg Berbasis Fitur Iaf Dan Metode Deep Learning Untuk Deteksi Epilepsi," 2021.
- [5] A. Hanafiah, Y. Arta, H. O. Nasution, and Y. D. Lestari, "Penerapan Metode Recurrent Neural Network dengan Pendekatan Long Short-Term Memory (LSTM) Untuk Prediksi Harga Saham," *Bull. Comput. Sci. Res.*, vol. 4, no. 1, pp. 27–33, 2023, doi: 10.47065/bulletincsr.v4i1.321.
- [6] K. M. Alalayah, E. M. Senan, H. F. Atlam, I. A. Ahmed, and H. S. A. Shatnawi, "Effective Early Detection of Epileptic Seizures through EEG Signals Using Classification Algorithms Based on t-Distributed Stochastic Neighbor Embedding and K-Means," *Diagnostics*, vol. 13, no. 11, pp. 1–24, 2023, doi: 10.3390/diagnostics13111957.
- [7] V. Gabeff *et al.*, "Interpreting deep learning models for epileptic seizure detection on EEG signals," *Artif. Intell. Med.*, vol. 117, p. 102084, 2021, doi: 10.1016/j.artmed.2021.102084.
- [8] S. Mounika and S. R. Reaja, "Forecasting Epileptic Seizures Using XGBoost Methodology and EEG Signals," *EAI Endorsed Trans. Pervasive Heal. Technol.*, vol. 10, pp. 1–8, 2024, doi: 10.4108/eetpht.10.5569.
- [9] Mulaab, "Deteksi Kejang Epilepsy Dengan Menggunakan Pemilihan Fitur Information Gain Dan Pembelajaran Ensemble Random Forest," vol. 9, no. 2, pp. 77–83, 2021.
- [10] O. Nurdiawan and A. Faqih, "Optimisasi Model Backpropagation untuk Meningkatkan Deteksi Kejang Epilepsi pada Sinyal Electroencephalogram," vol. 9, no. 2, pp. 151–160, 2024.
- [11] A. A. Ghazi, A. Aprianti, A. D. P. Dimas, and R. Fauzi, "Analisis Prediksi Data Kasus Covid-19 di Provinsi Lampung Menggunakan Recurrent Neural Network (RNN)," *Indones. J. Appl. Math.*, vol. 2, no. 1, p. 25, 2022, doi: 10.35472/indojam.v2i1.763.
- [12] A. P. Putra and A. M. Harahap, "Implementasi Time Series Forecasting dengan Algoritma LSTM untuk Pemantauan dan Prediksi Produktivitas Kelapa Sawit Berdasarkan Hasil Panen," vol. 12, no. 2, pp. 46–56, 2025, doi: 10.30865/jurikom.v12i2.8495.
- [13] M. Guntara, "Komparasi Kinerja Label-Encoding dengan One-Hot-Encoding pada Algoritma K-Nearest Neighbor menggunakan Himpunan Data Campuran Performance Comparison of Label-Encoding with One-Hot-Encoding Methode on K-Nearest

- Neighbor Algorithm with Mixed Type Data Set,” no. 2, pp. 352–360, 2025, doi: 10.26798/jiko.v9i2.1605.
- [14] M. P. Pulungan, A. Purnomo, and A. Kurniasih, “Penerapan SMOTE untuk Mengatasi Imbalance Class dalam Klasifikasi Kepribadian MBTI Menggunakan Naive Bayes Classifier,” *J. Teknol. Inf. dan Ilmu Komput.*, vol. 10, no. 7, pp. 1493–1502, 2023, doi: 10.25126/jtiik.1077989.
 - [15] A. Pramudyantoro, E. Utami, and D. Ariatmanto, “Penggabungan K-Nearest Neighbors dan Lightgbm Untuk prediksi Diabetes Pada Dataset Pima Indians: Menggunakan pendekatan Exploratory Data Analysis,” *JUPI (Jurnal Ilm. Penelit. dan Pembelajaran Inform.)*, vol. 9, no. 3, pp. 1133–1144, 2024, [Online]. Available: <https://jurnal.stkipgritlungagung.ac.id/index.php/jupi/article/view/4966/2114>
 - [16] S. Farisi and S. Hadi, “Analisis Sentimen menggunakan Recurrent Neural Network Terkait Isu Anies Baswedan Sebagai Calon Presiden 2024,” *eProceedings ...*, vol. 10, no. 2, p. 1682, 2023, [Online]. Available: <https://openlibrarypublications.telkomuniversity.ac.id/index.php/engineering/article/view/19992%0Ahttps://openlibrarypublications.telkomuniversity.ac.id/index.php/engineering/article/view/19992/19357>
 - [17] W. I. Rahayu, C. Prianto, and E. A. Novia, “Perbandingan Algoritma K-Means dan Naive Bayes untuk Memprediksi Prioritas Pembayaran Tagihan Rumah Sakit Berdasarkan Tingkat Kepentingan pada PT. Pertamina (Persero),” *J. Tek. Inform.*, vol. 13, no. 2, pp. 1–8, 2021.
 - [18] I. Farisi, J. Shadiq, W. Priyadi, D. Maulana, and F. Sonia, “Penerapan Model Recurrent Neural Network (RNN) untuk Prediksi Curah Hujan Berbasis Data Historis,” vol. 9, no. 2, pp. 217–226, 2024.
 - [19] S. S. Nurashila, F. Hamami, and T. F. Kusumasari, “Perbandingan Kinerja Algoritma Recurrent Neural Network (Rnn) Dan Long Short-Term Memory (Lstm): Studi Kasus Prediksi Kemacetan Lalu Lintas Jaringan Pt Xyz,” *JUPI (Jurnal Ilm. Penelit. dan Pembelajaran Inform.)*, vol. 8, no. 3, pp. 864–877, 2023, doi: 10.29100/jupi.v8i3.3961.
 - [20] D. Prediksi and B. Jaringan, “Kinerja algoritma backpropagation dan rnn dalam prediksi beban jaringan,” vol. 7, pp. 1044–1053, 2024, doi: 10.37600/tekinkom.v7i2.1586.