

# PENERAPAN *MACHINE LEARNING* EXTREME GRADIENT BOOSTING DALAM KLASIFIKASI POTENSI TSUNAMI BERDASARKAN DATA GEMPA BUMI

I Gede Jayadi Kurniarwan<sup>1</sup>, Candra Dewi<sup>2</sup>, Muh Arif Rahman<sup>3</sup>

Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya

Email: <sup>1</sup>jayadi@student.ub.ac.id, <sup>2</sup>dewi\_candra@ub.ac.id, <sup>3</sup>m\_arif@ub.ac.id

## Abstrak

Tsunami merupakan bencana alam yang berdampak besar, khususnya di wilayah rawan seperti Indonesia. Penelitian ini bertujuan mengklasifikasikan potensi tsunami berdasarkan data gempa bumi menggunakan algoritma Extreme Gradient Boosting (XGBoost). Dataset mencakup data gempa dari 1900 hingga 2023 dengan total 1.376 data dan mengandung ketidakseimbangan kelas. Untuk mengatasi masalah ini, teknik Synthetic Minority Over-sampling Technique (SMOTE) diterapkan. Model dikembangkan dengan menyetel *hyperparameter* seperti *max\_depth*, *learning\_rate*, *gamma*, *min\_child\_weight*, *colsample\_bytree*, dan *subsample* menggunakan Grid Search. Hasil pengujian menunjukkan bahwa kombinasi *hyperparameter* terbaik menghasilkan *accuracy* 85%, dengan *recall*, *precision*, dan F1-score masing-masing sebesar 0.74. Setelah penerapan SMOTE, *accuracy* menurun menjadi 84%, tetapi *recall* meningkat menjadi 0.79 dengan *precision* 0.71 dan F1-score 0.75. Evaluasi menggunakan metrik *accuracy*, *precision*, *recall*, dan F1-score menunjukkan bahwa penerapan SMOTE memberikan prediksi yang lebih baik untuk kelas tsunami. Pendekatan ini diharapkan berkontribusi dalam mitigasi bencana tsunami melalui sistem deteksi dini berbasis data gempa bumi.

**Kata kunci:** tsunami, xgboost, SMOTE, grid-search, klasifikasi

## Abstract

*Tsunamis are natural disasters with significant impacts, especially in vulnerable areas such as Indonesia. This study aims to classify tsunami potential based on earthquake data using the Extreme Gradient Boosting (XGBoost) algorithm. The dataset includes earthquake data from 1900 to 2023, comprising a total of 1,376 entries with class imbalance. To address this issue, the Synthetic Minority Over-sampling Technique (SMOTE) was applied. The model was developed by tuning hyperparameters such as max\_depth, learning\_rate, gamma, min\_child\_weight, colsample\_bytree, and subsample using Grid Search. The testing results showed that the best hyperparameter combination achieved 85% accuracy, with recall, precision, and F1-score each at 0.74. After applying SMOTE, the accuracy decreased to 84%, but recall improved to 0.79, with precision at 0.71 and F1-score at 0.75. Evaluation using metrics such as accuracy, precision, recall, and F1-score demonstrated that SMOTE application provides better predictions for the tsunami class. This approach is expected to contribute to tsunami disaster mitigation through an early detection system based on earthquake data.*

**Keywords:** tsunami, xgboost, SMOTE, grid-search, classification

## 1. PENDAHULUAN

Tsunami adalah salah satu bencana alam yang memiliki dampak besar terhadap lingkungan dan kehidupan manusia. Tsunami dapat disebabkan oleh berbagai faktor, seperti gempa bumi bawah laut, pergeseran lempeng tektonik, aktivitas vulkanik, hingga tanah longsor. Gelombang tsunami yang tinggi dan bergerak cepat ke daratan dapat menyebabkan

kerusakan parah dan korban jiwa. Indonesia, sebagai negara yang berada di wilayah Cincin Api Pasifik, memiliki risiko tinggi terhadap tsunami karena seringnya terjadi gempa bumi akibat aktivitas tektonik di wilayah tersebut. Oleh karena itu, penting untuk mengembangkan sistem yang dapat membantu mitigasi bencana melalui deteksi dini potensi tsunami (Reid & Mooney, 2023).

Dalam bidang teknologi, *machine learning*

menjadi salah satu solusi yang dapat diandalkan untuk klasifikasi terjadinya bencana alam. Algoritma *machine learning* dapat digunakan untuk mempelajari pola pada data gempa bumi dan menghasilkan prediksi terkait potensi tsunami (Novianty, et al., 2019). Salah satu algoritma yang dapat digunakan adalah Extreme Gradient Boosting (XGBoost). XGBoost memiliki kinerja yang sangat baik dalam tugas klasifikasi, seperti yang telah ditunjukkan dalam berbagai penelitian. Penelitian oleh Yusri et al. (2022) membuktikan bahwa XGBoost lebih unggul dibandingkan algoritma lain, seperti Support Vector Machine (SVM), dengan *accuracy* hingga 94% pada tugas klasifikasi kualitas air (Hasnol Yusri, et al., 2022).

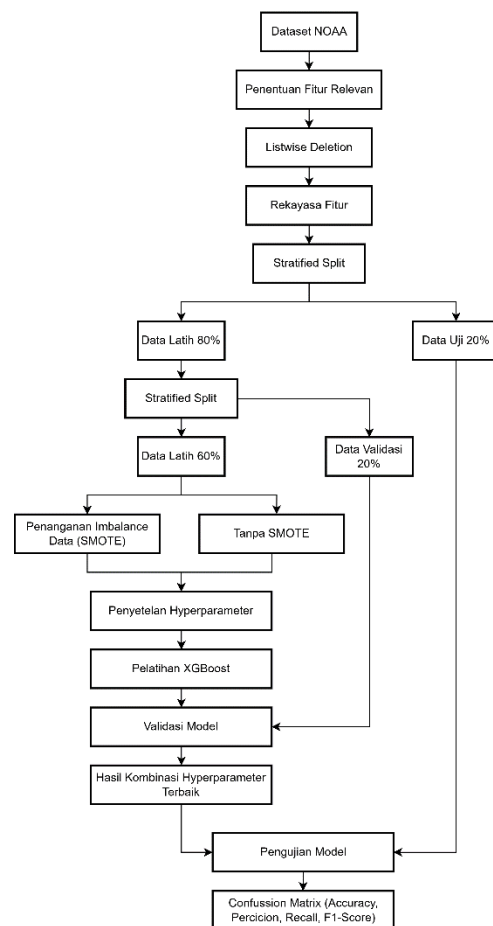
Namun dalam penelitian ini, dataset yang digunakan tidak seimbang, yaitu data kejadian tsunami lebih sedikit dibandingkan dengan data non-tsunami. Ketidakseimbangan data ini dapat memengaruhi kinerja model pembelajaran mesin. Untuk mengatasi masalah ini, digunakan teknik Synthetic Minority Over-sampling Technique (SMOTE) yang bertujuan menyeimbangkan jumlah data pada setiap kelas. SMOTE telah terbukti efektif meningkatkan kinerja model dalam penelitian sebelumnya (Han, et al., 2024).

Selain menangani ketidakseimbangan data, kinerja XGBoost juga dipengaruhi oleh penyetelan *hyperparameter* seperti *max\_depth*, *learning\_rate*, *gamma*, *min\_child\_weight*, dan *subsample*. Penyetelan *hyperparameter* yang optimal dapat meningkatkan *accuracy* dan stabilitas model secara signifikan. Sebagai contoh, penelitian Priyana et al. (2024) menunjukkan bahwa tuning *hyperparameter* mampu meningkatkan *accuracy* model XGBoost dari 81% menjadi 85% (Priyana, et al., 2024).

Penelitian ini menggabungkan teknik SMOTE dan penyetelan *hyperparameter* pada algoritma XGBoost untuk meningkatkan *accuracy* klasifikasi potensi tsunami. Dengan pendekatan ini, diharapkan hasil prediksi dapat mendukung mitigasi bencana tsunami di Indonesia, sehingga upaya evakuasi dapat dilakukan lebih cepat dan tepat waktu. dapat secara signifikan meningkatkan kinerja model (Priyana, et al., 2024).

## 2. METODE

Alur dari penelitian ini ditunjukkan pada Gambar 1.



Gambar 1. Alur Penelitian

### 2.1 Dataset

Dataset yang digunakan adalah data gempa bumi dari 1900 hingga 2023 yang diambil dari situs NOAA (<https://www.ngdc.noaa.gov>). Dataset ini terdiri dari fitur-fitur seperti:

- *Focal Depth*: Kedalaman pusat gempa.
- *Mag*: Magnitudo utama gempa yang menggambarkan kekuatan gempa berdasarkan energi yang dilepaskan.
- *Ms*: Magnitudo gempa bumi yang diukur berdasarkan gelombang permukaan.
- *Mb*: Magnitudo yang diukur dari gelombang dalam yang bergerak melalui interior bumi.
- *Tsu*: Kelas target “tsunami” atau “non-tsunami”.

### 2.2 Pra-pemrosesan data

Data gempa bumi yang telah dikumpulkan akan melalui beberapa tahap sebelum dilanjutkan ke pelatihan model.

#### 2.2.1 Penanganan Missing Value

Dalam penelitian ini, metode listwise deletion digunakan untuk menangani *missing*

*value* dalam dataset gempa bumi. Listwise deletion adalah menghapus seluruh baris data jika terdapat setidaknya satu nilai yang hilang dalam salah satu kolom. Dengan menghapus data yang hilang, penelitian ini memastikan bahwa data yang digunakan untuk pelatihan berasal dari kejadian nyata (Pepinsky, 2018).

### 2.2.2 Rekayasa Fitur

Rekayasa fitur adalah proses menciptakan atau memodifikasi fitur dalam dataset untuk meningkatkan kinerja model *machine learning*. Proses ini bertujuan untuk mengambil informasi dari data mentah dan merepresentasikannya dalam bentuk yang lebih relevan (Sun & Yang, 2019). Pada dataset gempa bumi, terdapat fitur seperti lokasi gempa dalam satuan koordinat *latitude* dan *longitude*. Sehingga dapat membuat dua fitur baru dibuat berdasarkan data lokasi gempa:

- Jarak ke garis pantai: Dihitung menggunakan dataset jarak global dari NASA Ocean Biology Processing Group.
- Lokasi gempa (laut atau daratan): Ditentukan menggunakan API Isitwater.

### 2.2.3 Pembagian Data

Metode *stratified split* diterapkan untuk memastikan distribusi kelas tetap proporsional dalam *subset* data latih, validasi, dan uji. *Dataset* dibagi menjadi 60% data latih, 20% data validasi, dan 20% data uji, dengan menjaga keseimbangan distribusi kelas di setiap *subset*.

### 2.2.4 Penanganan Ketidakseimbangan Data

*Dataset* yang digunakan tidak seimbang yaitu jumlah data kelas tsunami lebih sedikit daripada non-tsunami. Terdapat beberapa teknik untuk mengatasi masalah ini seperti Borderline SMOTE, Random DownSampling, Random Upsampling, SMOTE, ADASYN. Namun dalam penelitian ini menerapkan SMOTE pada data latih karena telah menunjukkan kinerja terbaik pada penelitian sebelumnya (Han, et al., 2024). SMOTE adalah metode untuk membuat sampel sintetis pada kelas minoritas dengan cara interpolasi antara sampel minoritas yang ada.

## 2.3 Pelatihan Model

Proses pelatihan ini menggunakan model Extreme Gradient Boosting (XGboost) dengan dua tahap yaitu data SMOTE dan data tanpa SMOTE. Setiap proses akan melalui penyetalan

*hyperparameter* Xgboost menggunakan grid search.

### 2.3.1 Xgboost

XGBoost bekerja dengan membangun model *ensemble* yang terdiri dari banyak pohon keputusan yang digabungkan secara bertahap untuk memperbaiki kesalahan prediksi dari model sebelumnya. Setiap pohon dalam *ensemble* ini berkontribusi dalam mengurangi error secara bertahap menggunakan pendekatan gradient boosting (Yulianti, et al., 2022). Proses ini melibatkan beberapa langkah berikut:

1. Dalam kasus klasifikasi biner, prediksi awal diinisialisasi dengan nilai probabilitas tetap. XGBoost menginisialisasi prediksi ini dengan nilai 0,5 untuk setiap *instance*.
2. Perhitungan gradien dan *hessian* didapatkan dari turunan pertama dan kedua terhadap prediksi model dari log loss pada Persamaan (1).

$$L(y_i, p_i) = -[y_i \log(p_i) + (-y_i) \log(-p_i)] \quad (1)$$

Rumus gradien dan *hessian* dituliskan dalam Persamaan (2) dan (3).

$$g_i = \hat{y}_i - y_i \quad (2)$$

$$h_i = \hat{y}_i \times (1 - \hat{y}_i) \quad (3)$$

3. Untuk menemukan split terbaik, XGBoost menggunakan gain, yang merupakan ukuran peningkatan kinerja model. Rumus gain dituliskan dalam Persamaan (4).

$$Gain = \frac{1}{2} \left[ \frac{(\sum_{i \in L} g_i)^2}{\sum_{i \in L} h_i + \lambda} + \frac{(\sum_{i \in R} g_i)^2}{\sum_{i \in R} h_i + \lambda} - \frac{(\sum_{i \in I} g_i)^2}{\sum_{i \in I} h_i + \lambda} \right] - \gamma \quad (4)$$

4. Setelah pohon dibangun, setiap *node* daun (*leaf*) diberikan nilai yang menunjukkan bagaimana pembaruan prediksi harus dilakukan. Rumus *leaf node* dituliskan dalam Persamaan (5).

$$Leaf\ value = - \frac{\sum g}{\sum h + \lambda} \quad (5)$$

5. Setelah nilai daun dihitung, prediksi diperbarui pada setiap iterasi. Rumus pembaruan prediksi dituliskan dalam Persamaan (6).

$$new\ \hat{y} = \hat{y} + (\alpha \times leaf\ value) \quad (6)$$

6. XGBoost menggunakan fungsi sigmoid untuk mengubah output mentah dari model menjadi probabilitas antara 0 dan 1. Rumus sigmoid dituliskan dalam Persamaan (7).

$$sigmoid(z) = \frac{1}{1 + e^{-new\ \hat{y}}} \quad (7)$$

### 2.3.2 Hyperparameter

XGBoost adalah algoritma machine learning yang memiliki berbagai *hyperparameter* yang dapat disesuaikan untuk mengoptimalkan kinerja model. Beberapa *hyperparameter* utama yang berpengaruh dalam algoritma ini yaitu (Chimphlee & Chimphlee, 2024)

- *Gamma*: Mengontrol kompleksitas model dengan menambahkan penalti pada pembuatan *node* baru di pohon keputusan.
- *Min\_child\_weight*: Menentukan total hessian minimum yang diperlukan untuk melakukan split pada sebuah *node*.
- *Learning\_rate*: Mengatur laju pembelajaran dalam pembentukan sebuah pohon.
- *Subsample*: Menentukan proporsi data latih yang digunakan untuk membangun setiap pohon.
- *Colsample\_bytree*: Menentukan proporsi fitur yang digunakan untuk membangun setiap pohon keputusan.
- *Max\_depth*: Mengatur kedalaman maksimum dari setiap pohon keputusan.

### 2.3.3 Grid Search

Grid Search adalah metode pencarian yang dilakukan secara menyeluruh untuk menentukan kombinasi nilai-nilai optimal dari *hyperparameter* dalam model pembelajaran mesin. Proses Grid Search akan mencoba seluruh kombinasi dari nilai *hyperparameter* yang mungkin digunakan dalam model. Setelah semua kombinasi disusun, metode ini akan menerapkan setiap kombinasi pada pelatihan model, melakukan *fitting* pada data validasi, dan mengembalikan kombinasi *hyperparameter* terbaik yang menghasilkan kinerja model yang paling optimal (Montesinos-López, et al., 2022). Kombinasi *hyperparameter* yang digunakan ditampilkan pada Tabel 1.

Tabel 1. Kombinasi *Hyperparameter*

No	<i>Hyperparameter</i>	Rentang Nilai
1	<i>learning_rate</i>	[0,3, 0,4, 0,5, 0,6, 0,7, 0,8, 0,9]
2	<i>max_depth</i>	[5, 6, 7, 8, 9, 10]
3	<i>Gamma</i>	[0, 0,001, 0,01, 0,1, 0,5, 1]
4	<i>colsample_bytree</i>	[0,6, 0,7, 0,8, 0,9, 1,0]
5	<i>subsample</i>	[0,6, 0,7, 0,8, 0,9, 1,0]
6	<i>min_child_weight</i>	[0,1,2,3,4]

### 2.4 Evaluasi Model

Hasil kombinasi *hyperparameter* terbaik akan diuji menggunakan data uji untuk melihat kinerja model dalam mengklasifikasikan gempa bumi yang berpotensi tsunami atau tidak berpotensi tsunami. Evaluasi dilakukan dengan menggunakan metrik evaluasi. Terdapat 4 matrix utama yaitu (Cao, et al., 2023)

- *Accuracy*: menjelaskan tingkat kemampuan model dalam memprediksi dengan benar jumlah sampel yang tepat di antara semua sampel yang ada
- *Precision*: mengukur tingkat keakuratan model klasifikasi dalam memprediksi hasil positif.
- *Recall*: mengukur kemampuan model klasifikasi dalam menemukan semua hasil positif yang sebenarnya.
- F1-score: merupakan harmonisasi dari dua metrik, yaitu *precision* dan *recall*.

### 3. HASIL DAN ANALISIS

Pada bagian ini akan menampilkan hasil pelatihan dan pengujian model XGBoost melalui penyetelan *hyperparameter* dan penerapan metode SMOTE untuk mengatasi ketidakseimbangan data. Hasil yang disajikan berupa confusion matrix, metrix evaluasi, gambar grafik pengaruh *hyperparameter*, dan grafik perbandingan model.

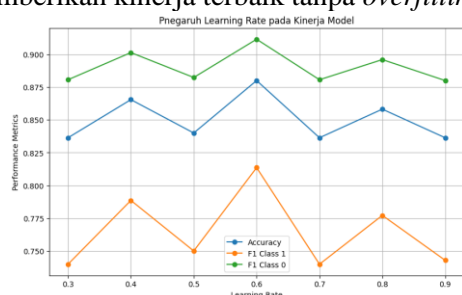
Proses penyetelan *hyperparameter*, terdapat 31500 kombinasi menggunakan nilai yang telah ditetapkan pada Tabel 1. Proses ini menggunakan data latih tanpa melalui proses SMOTE dan hasil kombinasi *hyperparameter* terbaik yang diperoleh adalah sebagai berikut: *max\_depth* = 8, *learning\_rate* = 0,6, *gamma* = 0, *colsample\_bytree* = 0,8, *subsample* = 1, dan *min\_child\_weight* = 2. Kombinasi ini menghasilkan kinerja terbaik pada data validasi dengan *accuracy* sebesar 88% dan F1-score kelas 1 = 0,9115 dan F1-score kelas 0 = 0,8136.

Hasil penyetelan *hyperparameter* dengan data latih yang melalui proses SMOTE adalah sebagai berikut *max\_depth* = 7, *learning\_rate* = 0,6, *gamma* = 0,001, *colsample\_bytree* = 0,6, *subsample* = 0,7, dan *min\_child\_weight* = 0. Kombinasi ini menghasilkan kinerja terbaik pada data validasi dengan *accuracy* sebesar 86,18% dan F1-score kelas 1 = 0,7791 dan F1-score kelas 0 = 0,8995.



### 3.1 Pengaruh *Hyperparameter Learning\_rate*

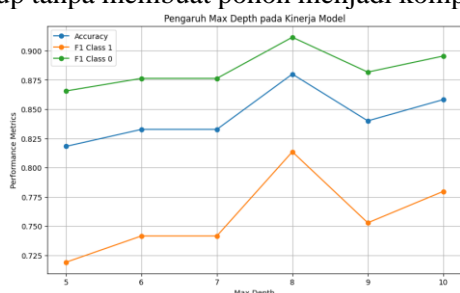
Grafik pada Gambar 2 menunjukkan pengaruh *learning\_rate* terhadap kinerja model XGBoost berdasarkan *accuracy* dan F1-score untuk kelas tsunami (kelas 1) dan non-tsunami (kelas 0). Nilai optimal tercapai pada 0,6, dengan *accuracy* 88,00%, F1-score kelas tsunami 0,8136, dan F1-score kelas non-tsunami 0,9115, menunjukkan keseimbangan antara kecepatan belajar dan generalisasi. Penurunan ke 0,4 mengurangi *accuracy* menjadi 86,55%, sedangkan peningkatan ke 0,7 menurunkan *accuracy* menjadi 83,64%. Nilai optimal 0,6 memberikan kinerja terbaik tanpa *overfitting*.



Gambar 2 Grafik Pengaruh *Learning\_rate*

### 3.2 Pengaruh *Hyperparameter Max\_depth*

Grafik pada Gambar 3 menunjukkan bahwa *max\_depth* optimal adalah 8, dengan *accuracy* 88,00%, F1-score kelas tsunami 0,8136, dan F1-score kelas non-tsunami 0,9115. *Max\_depth* lebih kecil 5 menurunkan *accuracy* menjadi 81,82%, sementara nilai lebih besar seperti 9 menurunkan generalisasi dengan *accuracy* 84,00%. Nilai optimal 8 menangkap pola yang cukup tanpa membuat pohon menjadi kompleks.

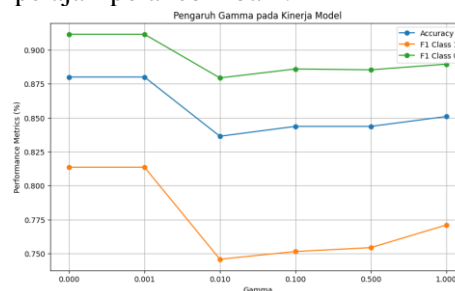


Gambar 3. Grafik Pengaruh *Max\_depth*

### 3.3 Pengaruh *Hyperparameter Gamma*

Grafik pada Gambar 4 menunjukkan bahwa gamma optimal pada 0 dan 0,001, dengan *accuracy* 88,00%, F1-score kelas tsunami 0,8136, dan F1-score kelas non-tsunami 0,9115. Gamma terlalu besar yaitu diatas 0,01 menurunkan *accuracy* hingga 83,64%. Nilai rendah seperti 0 dan 0,001 memberikan

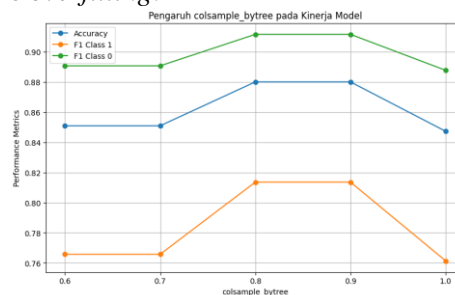
fleksibilitas lebih ketika model membentuk *node* baru, sehingga pohon Xgboost dapat mempelajari pola lebih baik.



Gambar 4. Grafik Pengaruh Gamma

### 3.4 Pengaruh *Hyperparameter Colsample\_bytree*

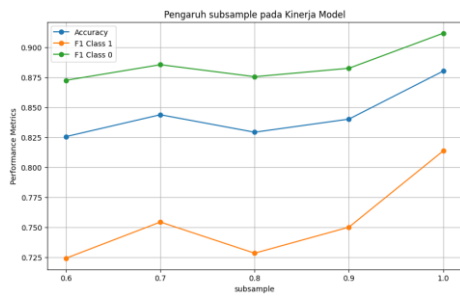
Grafik pada Gambar 5 menunjukkan bahwa *colsample\_bytree* optimal pada 0,8 dan 0,9, dengan *accuracy* 88,00%, F1-score kelas tsunami 0,8136, dan F1-score kelas non-tsunami 0,9115. Nilai lebih kecil 0,6 atau lebih besar 1 menurunkan *accuracy* menjadi 85,09% dan 84,73%. Ketika terlalu sedikit fitur yang digunakan, model bisa kehilangan informasi penting dari fitur yang tidak terpilih. Sebaliknya, ketika seluruh fitur digunakan setiap kali pembentukan pohon, maka dapat meningkatkan risiko *overfitting*.



Gambar 5. Grafik Pengaruh *Colsample\_bytree*

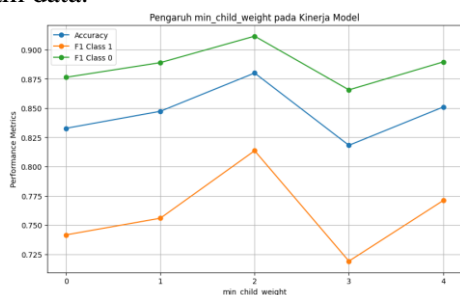
### 3.5 Pengaruh *Hyperparameter Subsample*

Grafik pada Gambar 6 menunjukkan bahwa *subsample* optimal pada 1,0, dengan *accuracy* 88,00%, F1-score kelas tsunami 0,8136, dan F1-score kelas non-tsunami 0,9115. Ketika *subsample* diturunkan, *accuracy* juga ikut menurun secara bertahap dengan nilai paling rendah yaitu *accuracy* 82,55%. Jika *subsample* terlalu kecil, otomatis jumlah data yang digunakan dalam pembentukan pohon semakin sedikit dan model dapat kehilangan informasi penting dari data yang tidak terpilih, sehingga kinerja menurun.

Gambar 6. Grafik Pengaruh *Subsample*

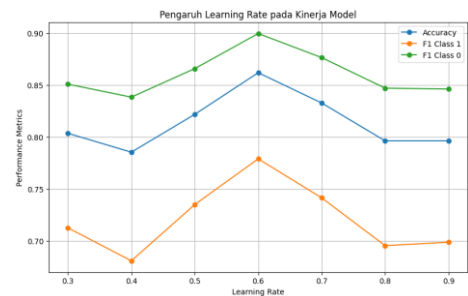
### 3.6 Pengaruh Hyperparameter *Min\_child\_weight*

Grafik pada Gambar 7 menunjukkan bahwa *min\_child\_weight* optimal pada 2, dengan *accuracy* 88,00%, F1-score kelas tsunami 0,8136, dan F1-score kelas non-tsunami 0,9115. Nilai terlalu kecil dan besar dapat menurunkan hasil *accuracy* hingga 81,82%. Hal ini menunjukkan bahwa pembatasan yang terlalu tinggi atau rendah dapat membuat model mengurangi fleksibilitas atau mempelajari *noise* dalam data.

Gambar 7. Grafik Pengaruh *Min\_child\_weight*

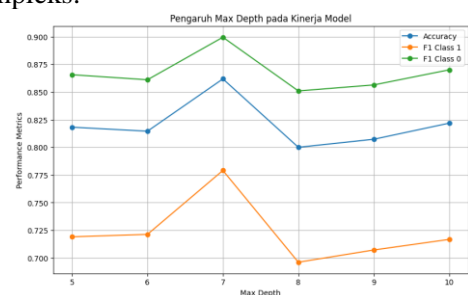
### 3.7 Pengaruh Hyperparameter *Learning\_rate* (SMOTE)

Grafik pada Gambar 8 menunjukkan bahwa *learning\_rate* memengaruhi kinerja model dengan SMOTE. Nilai optimal 0,6 memberikan *accuracy* 86,18%, F1-score kelas tsunami 0,7790, dan F1-score kelas non-tsunami 0,8995. Nilai lebih kecil 0,4 atau besar 0,9 menurunkan *accuracy* masing-masing menjadi 78,55% dan 79,64%, menunjukkan bahwa nilai 0,6 menghasilkan keseimbangan terbaik antara kecepatan belajar tanpa melewati solusi optimal.

Gambar 8. Grafik Pengaruh *Learning\_rate* (SMOTE)

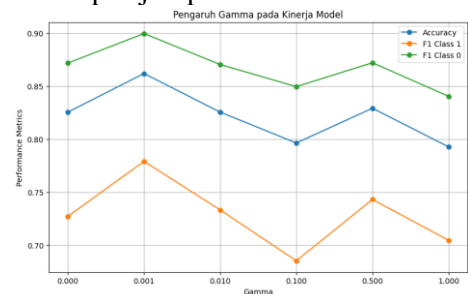
### 3.8 Pengaruh Hyperparameter *Max\_depth* (SMOTE)

Grafik pada Gambar 9 menunjukkan bahwa *max\_depth* optimal pada 7, dengan *accuracy* 86,18%, F1-score kelas tsunami 0,7790, dan F1-score kelas non-tsunami 0,8995. *Max\_depth* lebih kecil 6 atau besar 8 menurunkan *accuracy* dan generalisasi model. *Max\_depth* yang kecil membentuk pohon yang sederhana. Sebaliknya, *max\_depth* besar menyebabkan pohon terlalu kompleks.

Gambar 9. Grafik Pengaruh *Max\_depth* (SMOTE)

### 3.9 Pengaruh Hyperparameter *Gamma* (SMOTE)

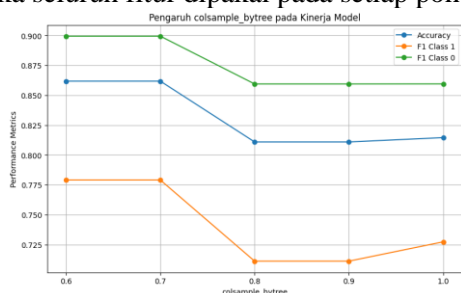
Grafik pada Gambar 10 menunjukkan bahwa *gamma* optimal pada 0,001, dengan *accuracy* 86,18%, F1-score kelas tsunami 0,7790, dan F1-score kelas non-tsunami 0,8995. Nilai lebih besar menyebabkan pembatasan yang ketat sehingga model kehilangan fleksibilitas untuk mempelajari pola data.

Gambar 10. Grafik Pengaruh *Gamma* (SMOTE)

### 3.10 Pengaruh Hyperparameter

### *Colsample\_bytree* (SMOTE)

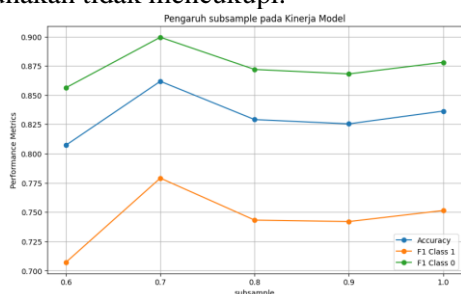
Grafik pada Gambar 11 menunjukkan bahwa *colsample\_bytree* optimal pada 0,6 dan 0,7, dengan *accuracy* 86,18%, F1-score kelas tsunami 0,7790, dan F1-score kelas non-tsunami 0,8995. Nilai lebih besar seperti 0,8 dan 1 menurunkan hasil *accuracy* yang disebabkan risiko *overfitting* dan kurangnya generalisasi ketika seluruh fitur dipakai pada setiap pohon.



Gambar 11. Grafik Pengaruh *Colsample\_bytree* (SMOTE)

### 3.11 Pengaruh Hyperparameter Subsample (SMOTE)

Grafik pada Gambar 12 menunjukkan bahwa *subsample* optimal pada 0,7, dengan *accuracy* 86,18%, F1-score kelas tsunami 0,7791, dan F1-score kelas non-tsunami 0,8995. *Subsample* yang terlalu kecil atau besar seperti 0,6 dan 1 menurunkan *accuracy*. Dengan nilai optimal, subset data yang berbeda untuk setiap pohon dapat mencegah *overfitting* dan menciptakan variasi antar pohon. Namun, jika nilainya terlalu rendah, model mungkin kehilangan pola penting karena data latih yang digunakan tidak mencukupi.

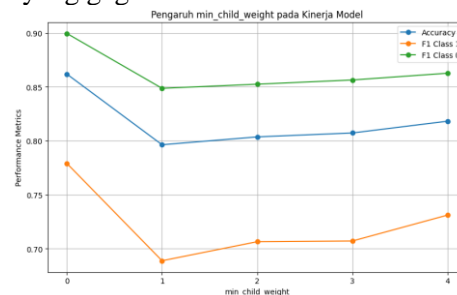


Gambar 12. Grafik Pengaruh *Subsample* (SMOTE)

### 3.12 Pengaruh Hyperparameter Min\_child\_weight (SMOTE)

Grafik pada Gambar 13 menunjukkan bahwa *min\_child\_weight* optimal pada 0, dengan *accuracy* 86,18%, F1-score kelas tsunami 0,7791, dan F1-score kelas non-tsunami 0,8995. Nilai lebih besar dari 1 hingga 4 menghasilkan kinerja lebih rendah pada seluruh metrik

evaluasi. Dengan nilai *min\_child\_weight* yang tinggi, mencegah model membuat pohon dengan leaf yang mengandung data sedikit. Sehingga pada dataset kecil dapat menyebabkan banyak *node* yang gagal dibentuk.



Gambar 13. Grafik Pengaruh *Min\_child\_weight* (SMOTE)

### 3.13 Pengujian Hyperparameter Terbaik

Tabel 2. Kinerja Sebelum SMOTE

Kelas	Precision	Recall	F1-Score
0	0.89	0.89	0.89
1	0.74	0.74	0.74
Accuracy			85%

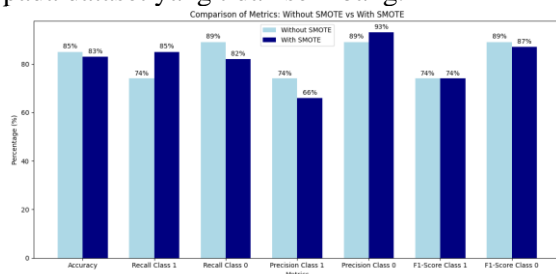
Tabel 3. Kinerja Setelah SMOTE

Kelas	Precision	Recall	F1-Score
0	0.93	0.82	0.87
1	0.66	0.85	0.74
Accuracy			83%

Penerapan SMOTE memberikan dampak signifikan pada kinerja model, seperti terlihat pada Gambar 8 yang membandingkan metrik evaluasi sebelum dan sesudah penerapan SMOTE. Pada hasil Tabel 2 dan 3, sebelum penerapan SMOTE model memiliki *accuracy* sebesar 85%, yang menurun menjadi 83% setelah penerapannya. Penurunan ini mengindikasikan adanya *trade-off* dalam *accuracy* keseluruhan untuk meningkatkan kemampuan model dalam mengenali kelas minoritas. Hal ini tercermin dari peningkatan *recall* kelas tsunami (kelas 1) dari 0,74 menjadi 0,85, menunjukkan bahwa SMOTE efektif dalam membantu model mendeteksi lebih banyak kejadian tsunami yang sebelumnya tidak teridentifikasi. Namun, *precision* kelas tsunami menurun dari 0,74 menjadi 0,66, akibat data sintetis yang dihasilkan SMOTE hanya mereplikasi pola yang ada tanpa menambahkan informasi baru.

Untuk kelas non-tsunami, *recall* menurun dari 0,89 menjadi 0,82, sementara *precision*

meningkat dari 0,89 menjadi 0,93. Ini menunjukkan bahwa model menjadi lebih fokus pada kelas tsunami, dengan sedikit penurunan kinerja pada kelas mayoritas. Secara keseluruhan, penerapan SMOTE meningkatkan *recall* kelas tsunami, yang sangat penting dalam mitigasi bencana. Peningkatan *recall* membantu meminimalkan *false negatives*, yaitu kejadian tsunami yang tidak terdeteksi, sehingga memenuhi tujuan utama SMOTE dalam mengurangi bias model terhadap kelas mayoritas pada dataset yang tidak seimbang.



Gambar 14. Grafik Perbandingan Kinerja Sebelum dan Setelah SMOTE

#### 4. KESIMPULAN

Hasil penelitian menunjukkan bahwa penyetelan *hyperparameter* secara optimal melalui grid search memiliki pengaruh besar terhadap kinerja model Extreme Gradient Boosting dalam klasifikasi potensi tsunami, dengan *accuracy* terbaik sebesar 88% pada data validasi dan 85% pada data uji. Selain itu, penerapan teknik SMOTE berhasil meningkatkan *recall* kelas tsunami dari 0.74 menjadi 0.85, meskipun *accuracy* keseluruhan menurun dari 85% menjadi 83%. Hal ini membuktikan bahwa SMOTE efektif dalam mengatasi ketidakseimbangan data, memungkinkan model untuk lebih mengenali kejadian tsunami dengan baik meskipun terdapat kompromi pada *accuracy* keseluruhan.

#### 5. DAFTAR PUSTAKA

- Cao, Q. et al., 2023. Comparative study of neonatal brain injury fetuses using machine learning methods for perinatal data. *Computer Methods and Programs in Biomedicine*, Volume 240, p. 107701 .
- Chimphlee, W. & Chimphlee, S., 2024. Hyperparameters optimization XGBoost for network intrusion detection using CSE-CIC-IDS 2018 dataset. *IAES International Journal of Artificial Intelligence (IJ-AI)*, Volume 13, pp. 817- 826.
- Han, Y., Wei, Z. & Huang, G., 2024. An imbalance data quality monitoring based on SMOTE-XGBOOST supported by edge computing. *Scientific Reports*, 14(1).
- Hasnol Yusri, H. I. et al., 2022. *Water Quality Classification Using SVM And XGBoost Method*. Shah Alam, IEEE 13th Control and System Graduate Research Colloquium (ICSGRC).
- Montesinos-López, O., Montesinos, A. & Crossa, J., 2022. *Multivariate Statistical Machine Learning Methods for Genomic Prediction*. Cham: Springer.
- Novianty, A. et al., 2019. Tsunami Potential Identification based on Seismic Features using KNN Algorithm. *2019 IEEE 7th Conference on Systems, Process and Control (ICSPC)*, pp. 155-160.
- Pepinsky, T. B., 2018. A Note on Listwise Deletion versus Multiple Imputation. *Political Analysis*, Volume 26, p. 480–488.
- Priyana, I. et al., 2024. *Predictive Boosting for Employee Retention with SMOTE and XGBoost Hyperparameter Tuning*. Surakarta, IEEE.
- Reid, J. A. & Mooney, W. D., 2023. Tsunami Occurrence 1900–2020: A Global Review, with Examples from Indonesia. *Pure and Applied Geophysics*, 180(5), pp. 1549-1571.
- Sun, Y. & Yang, G., 2019. *Feature Engineering for Search Advertising Recognition*. Qingdao, Nanjing, IEEE (Institute of Electrical and Electronics Engineers).
- Yulianti, S., Soesanto, O. & Sukmawaty, Y., 2022. Penerapan Metode Extreme Gradient Boosting (XGBOOST) pada Klasifikasi Nasabah Kartu Kredit. *Journal of Mathematics Theory and Application*, 4(1), pp. 21-26.