PERBANDINGAN ALGORITMA BOOSTING UNTUK KLASIFIKASI LINGKUNGAN TEKTONIK GEOKIMIA VULKANIK

Hans Santoso[1)] Sabrina Phalosa Phai[2)] Sarah Barbara[3)] Maryanto[4)]

[1) 2) 3) 4)] Teknik Informatika FTI Universitas Tarumanagara
Letjen S. Parman St No.1, Tomang, Grogol Petamburan,
West Jakarta City, Jakarta 11440
email : hans.535220129@stu.untar.ac.id [1)], sabrina.535220131@stu.untar.ac.id [2)], sarah.535220132@stu.untar.ac.id [3)]
maryanto.535220130@stu.untar.ac.id [4)]

ABSTRAK

Penelitian ini bertujuan untuk menentukan model machine learning paling efektif untuk klasifikasi lingkungan tektonik (tectonic setting) berdasarkan komposisi geokimia. Menggunakan dataset dari database GEOROC, tiga algoritma gradient boosting XGBoost, LightGBM, dan CatBoost diuji melalui beberapa skenario, termasuk pembagian data 70:30 dan 80:20. Model dengan kinerja terbaik kemudian dioptimalkan menggunakan Grid Search Cross-Validation (GridSearchCV). Hasil menunjukkan bahwa model LightGBM, setelah melalui proses hyperparameter tuning pada skenario 80:20, mencapai performa tertinggi dengan akurasi 80,87%. Analisis kepentingan fitur (feature importance) lebih lanjut mengidentifikasi bahwa Al_2O_3 (Aluminium Oksida), Na_2O (Natrium Oksida), dan FeOT (Besi Oksida Total) merupakan tiga prediktor paling signifikan. Studi ini membuktikan bahwa LightGBM adalah pendekatan yang superior dan andal untuk tugas klasifikasi geokimia otomatis.

Key words

CatBoost, Geokimia, LightGBM, Tektonik, XGBoost

1. Pendahuluan

Klasifikasi lingkungan tektonik (*tectonic setting)* merupakan salah satu langkah penting dalam ilmu geologi karena dapat digunakan untuk merekonstruksi proses geodinamika pembentukan batuan, mulai dari evolusi kerak bumi hingga pembentukan cadangan mineral [1]. Analisis geokimia sendiri telah terbukti menjadi alat yang ampuh untuk mengidentifikasi aktivitas tektonik, misalnya melalui karakteristik fluida panas bumi di zona patahan aktif [2]. Secara tradisional, para ahli geologi mengandalkan sebuah diagram diskriminasi berbasis indikator geokimia untuk melakukan klasifikasi tersebut [3]. Namun, di era *big data* dimana volume data yang sangat besar, metode konvensional tersebut menghadapi tantangan besar. Basis data geokimia global seperti GEOROC kini menyediakan ratusan ribu analisis sampel, sebuah volume data yang membuat analisis manual dengan diagram menjadi tidak efisien dan rentan terhadap ambiguitas interpretasi akibat tumpang tindih antar area geokimia [4].

Menjawab tantangan tersebut, *machine learning* telah muncul sebagai suatu pendekatan yang kuat untuk klasifikasi batuan secara otomatis dan objektif [5]. Kemampuan *machine learning* untuk mengenali pola kompleks dalam *dataset* multidimensi menjadikannya solusi ideal untuk analisis geokimia modern. Pendekatan berbasis data (*data-driven*) kini menjadi praktik umum untuk menyelesaikan masalah klasifikasi di ilmu geologi, mulai dari klasifikasi status erupsi gunung berapi, prediksi magnitudo gempa bumi [6], prediksi fasies sedimen [7], di mana hasilnya terbukti mampu melampaui metode-metode statistik konvensional.

Di antara berbagai algoritma yang ada, algoritma seperti LightGBM, XGBoost, dan CatBoost diakui sebagai model yang sering menjadi subjek studi perbandingan untuk menentukan metode paling superior dalam menyelesaikan tugas-tugas prediksi [8]. Keunggulan model-model ini tidak hanya terbukti dalam domain geologi [7], tetapi juga di bidang lain yang menuntut akurasi tinggi seperti prediksi risiko kredit di industri keuangan [9]. Meskipun studi komparatif semacam ini telah dilakukan untuk berbagai aplikasi geologi, perbandingan kinerja secara langsung antara ketiganya untuk klasifikasi lingkungan tektonik batuan vulkanik secara spesifik menggunakan *dataset* berskala besar masih menjadi area yang perlu dieksplorasi lebih dalam.

Oleh karena itu, penelitian ini bertujuan untuk melakukan perbandingan kinerja secara sistematis antara algoritma boosting yaitu LightGBM, XGBoost, dan CatBoost menggunakan data dari yang bersumber dari database GEOROC. Kinerja model akan dievaluasi berdasarkan metrik akurasi untuk mengidentifikasi algoritma mana yang menjadi terbaik. Hasil dari penelitian ini diharapkan dapat memberikan rekomendasi metode komputasi terbaik untuk analisis geokimia otomatis, yang dapat membantu para ahli geologi dalam proses interpretasi data yang lebih cepat dan objektif serta menjadi referensi penelitian di kemudian hari.

2. Metode Penelitian

2.1. Sumber Data

Penelitian ini menggunakan *dataset* geokimia batuan vulkanik yang diperoleh dari The GEOROC *Database* (*Geochemistry of Rocks of the Oceans and Continents*), yaitu basis data terbuka yang berisi hasil analisis kimia batuan beku (igneous) dan metamorf dari berbagai wilayah di dunia. GEOROC merupakan salah satu sumber data geokimia terpercaya dan lengkap, mencakup lebih dari 23.000 publikasi ilmiah, 697.000 sampel, dan 40 juta nilai data individual, yang tersedia secara bebas dibawah lisensi Creative Commons (CC BY-SA 4.0) [5].

Dataset spesifik yang digunakan bersumber dari kumpulan data oleh Qin et al. (2022) yang berjudul *"Global mantle clinopyroxene data (major and trace elements)"*, diterbitkan melalui GFZ Data Services [10]. Dataset tersebut diunduh dalam format Excel (.xlsx) dengan total 21.600 baris data mentah sebelum dilakukan pembersihan. Setiap entri mewakili satu sampel batuan vulkanik dengan informasi kandungan unsur utama seperti SiO_2, TiO_2, Al_2O_3, FeOt, MgO, CaO, Na_2O, Cr_2O_3, dan MnO (dalam persen berat/wt%), serta label lingkungan tektonik (*tectonic setting*) seperti *Intraplate Volcanics*, Ocean Island, *Convergent Margin*, *Continental Flood Basalt, Rift Volcanics, Archean Craton, Oceanic Plateau, Seamound , Complex Volcanic Settings,* dan *Submarine Ridge.* Untuk jumlah distribusi Kelas *Tectonic Setting* dapat dilihat pada Tabel 1 sebagai berikut.

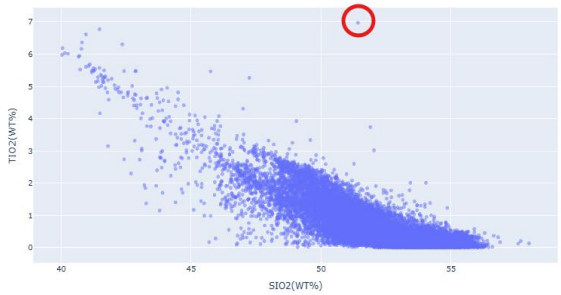
Tabel 1 Distribusi Kelas Tektonic Sebelum Penggabungan

Tectonic Setting	Jumlah
Intraplate Volcanics	11.371
Ocean Island	3.414
Convergent Margin	2.830
Continental Flood Basalt	1.640
Rift Volcanics	1.493
Archean Craton	253
Oceanic Plateu	193
Seamount	182
Complex Volcanic Settings	73
Submarine Ridge	6

2.2. Pra-pemrosesan Data

Tahap pra-pemrosesan data bertujuan untuk mentransformasi data mentah menjadi format yang bersih, terstruktur, dan siap untuk pemodelan *machine learning*. Proses ini meliputi pembersihan data, penanganan nilai hilang, penyederhanaan kelas, serta normalisasi fitur.

Langkah pertama adalah pembersihan data (*data cleaning*). Dari total 21.600 entri data mentah, semua baris yang tidak memiliki label *tectonic setting* dihapus. Selanjutnya, dilakukan seleksi fitur dengan hanya mempertahankan sembilan atribut kimia utama yaitu SiO_2, TiO_2, Al_2O_3, FeOT, MgO, CaO, Na_2O, Cr_2O_3, dan MnO. Data kosong (*missing values*) yang terdapat pada fitur-fitur tersebut diisi (diimputasi) menggunakan nilai median dari masing-masing kolom. Metode median dipilih karena lebih *robust* terhadap nilai ekstrim (*outliers*). Selain itu, satu entri data dengan nilai TIO2(WT%) yang sangat tinggi (dianggap *outlier*) juga dihapus untuk menjaga integritas statistik dataset. Outlier dapat terlihat pada Gambar 1.



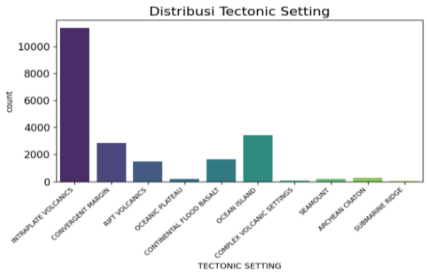
Gambar 1. *Outlier* pada Dataset

Langkah berikutnya adalah penyederhanaan kelas target untuk mengatasi ketidakseimbangan distribusi. Beberapa kelas dengan jumlah sampel yang sangat sedikit (kurang dari 1,5% dari total data), yaitu *Archean Craton, Oceanic Plateau, Seamount, Complex Volcanic Settings,* dan *Submarine Ridge*, digabungkan menjadi satu kelas baru berlabel **"***Others***"**. Langkah ini bertujuan agar model bisa lebih stabil saat pelatihan dan mencegah bias terhadap kelas-kelas yang dominan. Persebaran distribusi data per kelas tektonik secara setelah digabungkan dapat dilihat jumlah angka spesifiknya pada Tabel 2 sebagai berikut.

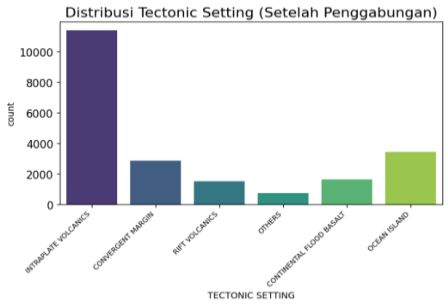
Tabel 2 Distribusi Kelas Tektonik Setelah Penggabungan

Tectonic Setting	Jumlah
Intraplate Volcanics	11.371
Ocean Island	3.414
Convergent Margin	2.830
Continental Flood Basalt	1.640
Rift Volcanics	1.493
Others	707

Kemudian untuk perbandingan distribusi kelas *Tectonic Setting* ketika sebelum dan sesudah penggabungan kelas minoritas menjadi kelas *Others* dapat dilihat pada Gambar 2 dan Gambar 3.



Gambar 2. Distribusi Jumlah Data per Kelas Tektonik



Gambar 3. Distribusi Data setelah Penggabungan

Diagram ini menunjukkan perbandingan jumlah sampel untuk setiap kategori lingkungan tektonik sebelum (Gambar 1) dan sesudah (Gambar 2) penggabungan kategori minor ke dalam kelas *Others.*

Setelah data bersih, dilakukan normalisasi fitur menggunakan *StandardScaler* dari scikit-learn agar setiap fitur memiliki skala yang seragam (rata-rata 0 dan standar deviasi 1). Terakhir, *dataset* dibagi menjadi data latih (*training set*) dan data uji (*testing set*) dengan dua skema proporsi, yaitu 70:30 dan 80:20. Pembagian ini dilakukan secara *stratified*, yang memastikan proporsi setiap kelas tektonik tetap terjaga baik di data latih maupun data uji, sehingga evaluasi model menjadi lebih objektif dan dapat diandalkan.

2.3. *Extreme Gradient Boost* (XGBoost)

XGBoost merupakan algoritma *gradient boosting* yang dirancang untuk efisiensi dan akurasi tinggi. Algoritma ini bekerja dengan membangun serangkaian model pohon keputusan (*decision tree*) secara sekuensial. Setiap pohon baru dibuat untuk memperbaiki kesalahan prediksi (*residual errors*) dari kombinasi pohon sebelumnya [11]**.** Proses ini berfokus pada contoh yang salah diklasifikasikan, menghasilkan model prediksi akhir yang kuat dan akurat [12].

Salah satu keunggulan utama yang membedakan XGBoost dari implementasi *gradient boosting* lainnya adalah mekanisme *regularisasi* yang terintegrasi secara canggih untuk mencegah *overfitting*. Teknik regularisasi ini secara efektif mengontrol kompleksitas model dengan memberikan *penalty* pada pohon yang tumbuh terlalu rumit, misalnya dengan membatasi jumlah daun atau kedalaman pohon [13]. Selain itu, XGBoost secara internal mengoptimalkan performa komputasi dengan menyederhanakan fungsi objektifnya. Hal ini memungkinkan penggabungan antara aspek prediktif (seberapa baik model cocok dengan data) dan aspek regularisasi (seberapa kompleks modelnya) dalam satu perhitungan yang efisien, sehingga tetap mempertahankan kecepatan komputasi yang optimal bahkan pada *dataset* berskala besar[13].

Struktur pohon keputusan pada XGBoost sendiri terdiri dari *node* bagian dalam yang merepresentasikan pengujian pada sebuah atribut (fitur), dan *leaf node* (daun) yang merepresentasikan skor akhir dari sebuah keputusan atau prediksi [14]**.** Secara lebih mendalam, proses pembelajaran aditif pada XGBoost tidak hanya berfokus pada residu sederhana, tetapi juga memanfaatkan turunan pertama (*Gradien*) dan kedua (*Hessian*) dari fungsi kerugian. Penggunaan informasi gradien orde kedua ini memungkinkan algoritma untuk melakukan optimasi yang lebih presisi dan cepat menuju titik minimal dari fungsi objektif. Hal ini memberikan keuntungan signifikan dibandingkan implementasi *gradient boosting* tradisional yang hanya menggunakan gradien orde pertama. Dengan demikian, XGBoost dapat secara efektif menemukan arah dan besaran perbaikan yang paling optimal pada setiap langkah, menghasilkan konvergensi yang lebih cepat dan model yang sering kali lebih akurat.

2.3.1. Fungsi Prediksi:

Hasil prediksi ($\hat{y}_i$) untuk sebuah data (x_i) adalah jumlah skor dari K pohon keputusan yang telah dibangun, seperti ditunjukkan pada persamaan berikut :

$$\hat{y}_i = \sum_{k=1}^{k} f_k(x_i), f_k \in F \qquad (1)$$

Dimana f_k adalah satu pohon keputusan independen. Untuk membangun model ini, XGBoost bekerja dengan cara meminimalkan sebuah fungsi objektif. Fungsi ini secara cerdas menyeimbangkan antara seberapa baik model cocok dengan data latih dan seberapa kompleks model tersebut.

2.3.2. Fungsi Objektif:

Tujuan fungsi ini adalah untuk mengukur apakah model tersebut cocok sebagai set data latih dan menentukan kompleksitas model. Fungsi objektif (Obj) yang diminimalkan oleh XGBoost terdiri dari dua komponen utama: fungsi kerugian (*loss function*) dan fungsi regularisasi (*regularization function*).

$$Obj(\theta) = L(\theta) + \Omega(\theta) \qquad (2)$$

Dimana $Obj(\theta)$ sebagai fungsi objektif, $L(\theta)$ sebagai pengukur selisih aktual dengan hasil prediksi, $\Omega(\theta)$ untuk mengontrol kompleksitas model.

2.3.3. Fungsi Regularisasi:

Komponen regularisasi inilah yang menjadi kunci kekuatan XGBoost dalam mengontrol *overfitting*. Fungsinya didefinisikan sebagai berikut:

$$\Omega(f_k) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2 \qquad (3)$$

Fungsi regularisasi ini mengukur kompleksitas model, di mana T adalah jumlah total daun (*leaves*) pada sebuah pohon dan w_j adalah skor (bobot) yang diberikan

pada setiap daun. Besarnya *penalty* dikontrol oleh parameter regularisasi γ dan λ . *Penalty* diberikan jika pohon menjadi terlalu kompleks dengan memiliki terlalu banyak daun, atau jika skor pada daun memiliki bobot yang terlalu ekstrim.

2.4. *Light Gradient Boosting Machine* (LightGBM)

LightGBM (*Light Gradient Boosting Machine*) adalah implementasi kerangka kerja *gradient boosting* yang dikembangkan oleh Microsoft, dirancang secara khusus untuk meningkatkan efisiensi dan kecepatan pelatihan pada dataset berukuran besar [15]. Seperti algoritma *boosting* lainnya, LightGBM bekerja dengan membangun model pohon keputusan (*decision tree*) secara bertahap, di mana setiap model baru (*weak learner*) ditambahkan untuk memperbaiki kesalahan prediksi dari kombinasi model sebelumnya [16]. Proses pembaruan model pada LightGBM mengikuti prinsip dasar *gradient boosting* dapat dilihat pada penjelasan berikut.

2.4.1. Fungsi Prediksi:

Fungsi prediksi pada setiap iterasi diperbarui dengan menambahkan pohon keputusan baru. Dalam persamaan ini, $F_m(x)$ adalah fungsi prediksi pada iterasi ke-m, $F_{m-1}(x)$ adalah fungsi dari iterasi sebelumnya, $h_m(x)$adalah pohon keputusan baru, η dan adalah *learning rate* yang mengontrol besarnya kontribusi dari pohon baru tersebut.

$$F_m(x) = F_{m-1}(x) + \eta \cdot h_m(x) \qquad (4)$$

2.4.2. Fungsi Objektif:

Seperti XGBoost, tujuan optimasi LightGBM adalah meminimalkan fungsi objektif yang terdiri dari dua komponen utama: fungsi kerugian (*loss function*) yang mengukur kesalahan prediksi, dan fungsi regularisasi yang mengontrol kompleksitas model untuk mencegah *overfitting* [17].

$$Obj = \sum_{i=1}^{n} L(yi, \hat{y}\imath) + \sum_{t=1}^{T} \Omega(ft) \qquad (5)$$

Dalam persamaan ini, $L(yi, \hat{y}\imath)$ adalah fungsi kerugian antara label aktual yi,dan prediksi model $\hat{y}\imath$ sedangkan $\Omega(ft)$ adalah fungsi regularisasi yang memberikan *penalty* pada kompleksitas pohon.

2.4.3. Inovasi Utama LightGBM:

Kecepatan dan efisiensi LightGBM yang superior berasal dari beberapa teknik inovatif yang membedakannya dari implementasi *boosting* lainnya. Dua yang paling signifikan yaitu yang pertama *Gradient-based One-Side Sampling* (GOSS) dimana berbeda dari metode yang menggunakan seluruh data iterasi, GOSS memilih sebagian data untuk membangun pohon keputusan. Teknik ini mempertahankan semua data yang sulit diprediksi (dengan nilai gradien tinggi) dan hanya mengambil sebagian kecil data acak dari sampel yang mudah diprediksi(dengan nilai gradien rendah). Pendekatan ini secara signifikan menurunkan beban komputasi tanpa kehilangan informasi penting dari distribusi data.

Kemudian yang kedua adalah *Exclusive Feature Bundlin*g (EFB) untuk mengatasi dataset dengan banyak fitur, EFB menjadi pendekatan yang efisien. Teknik ini mengurangi jumlah fitur dengan cara menggabungkan fitur-fitur yang bersifat *mutually exclusive* (jarang memiliki nilai bukan nol secara bersamaan) menjadi satu *bundle* fitur tunggal. Proses ini secara efektif mengurangi kompleksitas tanpa kehilangan banyak informasi, sehingga mempercepat proses pelatihan secara signifikan.

2.4.4. Struktur Pohon *leaf Wise*:

Berbeda dengan XGBoost yang membangun pohon secara *level-wise* (semua cabang pada satu level dikembangkan bersamaan), LightGBM menggunakan strategi *leaf-wise*. Strategi ini mampu menghasilkan model dengan akurasi lebih tinggi yang akan memberikan pengurangan *loss* terbesar. Meskipun pendekatan ini dapat menghasilkan pohon yang lebih akurat, ia berisiko menyebabkan *overfitting* pada dataset kecil, sehingga memerlukan pengaturan parameter seperti *max_depth* dengan cermat.

Berkat inovasi-inovasi tersebut, LightGBM menawarkan keunggulan berupa waktu pelatihan yang jauh lebih cepat dan konsumsi memori yang lebih rendah dibandingkan *boosting* lainnya, menjadikannya sangat cocok untuk analisis dataset geokimia berskala besar[16].

2.5. Categorical Boosting (CatBoost)

CatBoost (*Categorical Boosting*) adalah algoritma *gradient boosting* modern yang dikembangkan oleh Yandex. Algoritma ini dirancang secara spesifik untuk memberikan performa unggul dengan mengatasi dua tantangan utama dalam *machine learning* yaitu penanganan fitur kategorikal yang efisien dan pencegahan *overfitting* [18]. Seperti implementasi *gradient boosting* lainnya, CatBoost menggunakan pohon keputusan biner (*binary decision tree*) sebagai model dasar (*base learner*) yang dibangun secara sekuensial untuk memperbaiki kesalahan dari model sebelumnya [19].

Inovasi utama yang membedakan CatBoost adalah implementasi *ordered boosting*, sebuah variasi dari skema *gradient boosting* klasik yang secara signifikan lebih robust terhadap *overfitting*, terutama pada dataset berukuran kecil [18]. Selain itu, CatBoost memiliki metode internal untuk menentukan tingkat kepentingan fitur, di mana *Prediction Values Change* (PVC) menjadi metode default untuk mengukur kontribusi setiap fitur [20].

Secara fundamental, CatBoost tetap mengikuti prinsip dasar *gradient boosting*, di mana model dibangun secara bertahap dengan memperbarui fungsi prediksi pada

setiap iterasi untuk terus-menerus meminimalkan kesalahan.

2.5.1. Fungsi Prediksi:

CatBoost membangun model prediktifnya melalui proses yang bersifat aditif dan iteratif. Model tidak dibangun sekaligus, melainkan disempurnakan secara bertahap. Pada setiap iterasi, sebuah model dasar baru dalam hal ini, pohon keputusan ($h_m(x)$) dilatih secara spesifik untuk memperbaiki kesalahan atau residu yang ditinggalkan oleh kombinasi model dari semua iterasi sebelumnya. Pohon baru ini kemudian ditambahkan ke dalam *ensemble* dengan bobot tertentu yang diatur oleh *learning rate* ($\propto$).

Persamaan pembaruan model ini secara matematis merangkum proses tersebut. Dalam persamaan ini, $F_m(x)$ adalah fungsi prediksi akhir setelah iterasi ke-m, yang merupakan hasil dari fungsi prediksi sebelumnya, $F_{m-1}(x)$, ditambah dengan kontribusi dari pohon keputusan baru, $h_m(x)$. Parameter atau *learning rate* memainkan peran krusial dalam mengontrol seberapa besar dampak dari setiap pohon baru, di mana nilai yang lebih kecil membuat proses pembelajaran lebih lambat namun lebih *robust* terhadap *overfitting*.

$$F_m(x) = F_{m-1}(x) + \eta \cdot \propto h_m(x) \qquad (6)$$

2.5.2. Loss Function:

Tujuan utama dari setiap iterasi dalam algoritma CatBoost adalah untuk menemukan pohon Keputusan $h_m(x_i)$ yang paling optimal. Optimal dalam pembahasan ini berarti pohon yang, ketika ditambahkan ke model, mampu menghasilkan penurunan kesalahan prediksi yang paling signifikan. Proses pencarian ini dipandu oleh sebuah fungsi kerugian (*Loss Function*), yaitu sebuah fungsi matematis yang mengukur seberapa besar perbedaan antara nilai prediksi model dengan nilai target sebenarnya.

$$min \frac{1}{N} \sum_{i=1}^{N} \quad L(y_i, F_{m-1}(x_i) + h_m(x_i)) \qquad (7)$$

Tujuan optimasi CatBoost adalah untuk meminimalkan rata-rata dari fungsi kerugian (L) di seluruh sampel data pelatihan yang ada (N). Dalam rumus ini, adalah nilai target sebenarnya untuk sampel ke-i, sedangkan $\sum_{i=1}^{N} \quad F_{m-1}(x_i) + h_m(x_i)$ adalah nilai prediksi yang diperbarui setelah pohon baru ditambahkan. Dengan mencari $h_m(x_i)$ yang meminimalkan persamaan ini, algoritma secara efektif memastikan bahwa setiap pohon baru yang ditambahkan memberikan kontribusi paling besar dalam mengurangi kesalahan prediksi secara keseluruhan.

2.5.3. Ordered Target Statistic :

Fitur yang paling membedakan CatBoost adalah pendekatan inovatifnya dalam menangani fitur kategorikal. Alih-alih menggunakan teknik standar seperti *one-hot encoding*, CatBoost mengimplementasikan *ordered target statistics*. Metode ini menggantikan nilai fitur kategorikal dengan sebuah nilai numerik yang dihitung berdasarkan statistik dari nilai target (y). Untuk mencegah kebocoran target (*target leakage*), CatBoost menerapkan perhitungan ini pada permutasi acak dari dataset, sehingga perhitungan untuk satu sampel data hanya menggunakan informasi dari sampel-sampel yang muncul sebelumnya dalam urutan acak tersebut [18].

Dalam rumus ini, nilai pengganti x_{ik} untuk sebuah fitur dihitung berdasarkan jumlah nilai target (y_j) dari data sebelumnya yang memiliki kategori yang sama. Nilai ini kemudian distabilkan menggunakan sebuah *prior* atau parameter penghalus P dan nilai rata-rata target secara global A untuk memastikan representasi fitur yang stabil dan bebas bias [18].

$$value(x_{ik}) = \frac{\sum_{j=1}^{i=1} \quad [x_{jk} = x_{ik}] \cdot y_j + P \cdot A}{\sum_{j=1}^{i=1} \quad [x_{jk} = x_{ik}] + P} \qquad (8)$$

Dalam konteks penelitian ini, CatBoost diujikan untuk mengklasifikasi lingkungan tektonik. Meskipun seluruh fitur dalam *dataset* yang digunakan bersifat numerik, sehingga keunggulan utamanya dalam menangani fitur kategorikal tidak sepenuhnya bisa dimanfaatkan, algoritma ini tetap disertakan dalam perbandingan. Tujuannya adalah untuk mengevaluasi efisiensi dan stabilitas dari mekanisme *ordered boosting*-nya sebagai pembanding terhadap implementasi *boosting* konvensional seperti XGBoost dan LightGBM pada data numerik berskala besar.

2.6. Skenario Pengujian dan Evaluasi

Untuk mengukur dan memvalidasi kinerja model secara objektif, penelitian ini menerapkan metodologi eksperimen yang terstruktur. Proses ini dirancang untuk mengevaluasi model secara komprehensif, mulai dari pengujian awal hingga tahap optimasi akhir.

Langkah pertama dalam eksperimen adalah membagi *dataset* menjadi data latih dan data uji menggunakan dua skenario proporsi yang berbeda yaitu 70:30 dan 80:20. Pembagian ini dilakukan dengan teknik *stratified sampling* untuk memastikan bahwa distribusi kelas *tectonic setting* tetap terjaga secara proporsional di kedua set data, yang penting untuk konsistensi evaluasi.

Pada data latih ini, diterapkan *5-fold stratified cross-validation* sebagai strategi utama untuk evaluasi. Melalui teknik ini, model dilatih dan divalidasi sebanyak lima kali pada bagian data yang berbeda, dan hasil akhirnya dirata-ratakan. Pendekatan ini memberikan estimasi kinerja yang lebih stabil dan dapat diandalkan dibandingkan dengan pembagian data tunggal[21].

Metrik utama yang digunakan untuk mengevaluasi kinerja keseluruhan model adalah Akurasi (*Accuracy*), yang mengukur total persentase prediksi yang benar[13]. Untuk melengkapi evaluasi dan mendapatkan pemahaman yang lebih mendalam tentang karakteristik

performa model, metrik-metrik lain juga dianalisis, yaitu Presisi, *Recall*, dan *F1-Score* (dengan *weighted*). Metrik-metrik tambahan ini memberikan wawasan tentang bagaimana model menangani *trade-off* antara *false positive* dan *false negative* [22].

Terakhir, model yang menunjukkan performa paling unggul pada tahap evaluasi awal dipilih untuk melalui proses *hyperparameter tuning*. Proses ini dilakukan menggunakan metode *Grid Search Cross-Validation (GridSearchCV)*, yang secara sistematis menguji berbagai kombinasi parameter [23][24]. Proses pencarian ini dioptimalkan untuk menemukan konfigurasi yang menghasilkan skor Akurasi tertinggi selama *cross validation*. Model final dengan performa terbaik kemudian diuji untuk terakhir kalinya pada data uji data yang sepenuhnya baru dan belum pernah digunakan untuk mengukur kemampuan generalisasinya yang sesungguhnya.

3. Hasil dan Pembahasan

Pada bab ini, hasil dari skenario eksperimen dan analisis mendalam terhadap performa model-model *machine learning* XGBoost, LightGBM, dan CatBoost dalam tugas klasifikasi *tectonic setting* batuan vulkanik akan dibahas. Pembahasan mencakup perbandingan kinerja model, hasil optimasi *hyperparameter*, analisis kesalahan melalui *confusion matrix*, serta identifikasi fitur geokimia yang paling berpengaruh.

3.1. Perbandingan Kinerja Model Awal

Tahap evaluasi dilaksanakan untuk memberikan komparasi perbandigan untuk performa dari ketiga model *boosting* yang diuji. Komparasi ini dilakukan dengan menerapkan dua skenario jumlah rasio data yang berbeda, yakni dengan rasio 70:30 dan 80:20. Kinerja setiap model dievaluasi berdasarkan empat metrik standar yaitu Akurasi, Presisi, *Recall*, dan *F1-Score*. Untuk hasil dari kedua skenario pengujian tersebut disajikan secara rinci pada Tabel 3 dan Tabel 4 sebagai berikut

Tabel 3 Hasil Evaluasi Model pada Skenario 70:30

Algoritma	Akurasi	Precision	Recall	F1-score
LGBM	0,742	0,675	0,709	0,687
XGBoost	0,713	0,644	0,707	0,664
CatBoost	0,639	0,568	0,682	0,595

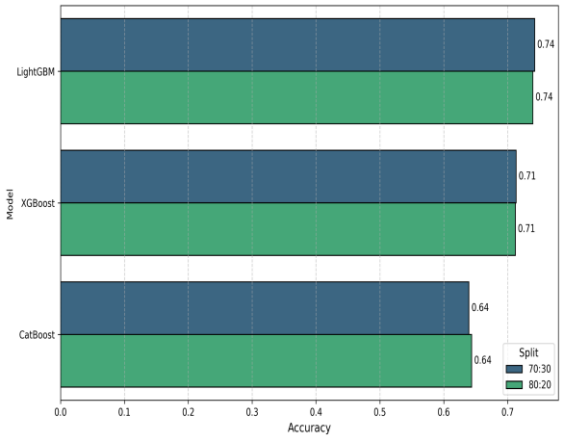
Tabel 4 Hasil Evaluasi Model pada Skenario 80:20

Algoritma	Akurasi	Precision	Recall	F1-score
LGBM	0,739	0,670	0,714	0,686
XGBoost	0,712	0,638	0,707	0,661
CatBoost	0,643	0,573	0,683	0,599

Berdasarkan hasil pada Tabel 3 dan 4, terlihat adanya pola yang konsisten di antara ketiga model. LightGBM secara konsisten memberikan performa terbaik pada kedua skenario, dengan akurasi sebesar 74,24% pada pembagian data 70:30 dan 73,97% pada pembagian 80:20. Keunggulan ini dapat diatribusikan pada strategi pertumbuhan pohon *leaf-wise* yang digunakan oleh LightGBM, yang memungkinkannya untuk lebih cepat mencapai konvergensi pada area kesalahan terbesar dalam data.

Model XGBoost menunjukkan kinerja yang sangat kompetitif dan menempati posisi kedua, mendekati performa LightGBM. Hal ini sejalan dengan reputasi XGBoost sebagai algoritma yang robust dan akurat. Sementara itu, CatBoost relatif tertinggal baik dari sisi akurasi maupun presisi. Meskipun CatBoost memiliki keunggulan dalam menangani fitur kategorikal, pada *dataset* ini yang seluruhnya bersifat numerik, keunggulan tersebut tidak dapat dimanfaatkan secara utuh, dan performanya sedikit di bawah dua implementasi *boosting* lainnya.

Selain akurasi, metrik F1-score juga memperlihatkan kecenderungan serupa, di mana LightGBM menunjukkan keseimbangan paling baik antara presisi dan recall. Ini mengindikasikan bahwa LightGBM tidak hanya akurat secara keseluruhan, tetapi juga lebih andal dalam mengidentifikasi setiap kelas secara seimbang. Perbandingan hasil tersebut selanjutnya disajikan pada Gambar 1 untuk memberikan ilustrasi yang lebih jelas mengenai perbedaan performa antar model.



Gambar 4. Perbandingan Akurasi Split

.

3.2. Optimasi dan Evaluasi Model Terbaik

Berdasarkan hasil perbandingan model, LightGBM menunjukkan performa paling unggul di antara ketiga algoritma yang diuji. Oleh karena itu, pada tahap selanjutnya dilakukan proses *hypertuning* terhadap LightGBM untuk mengoptimalkan parameter-parameter penting dan meningkatkan kinerja model secara keseluruhan.

Hypertuning dilakukan menggunakan SearchCV dengan 5 *fold cross validation* untuk memperoleh konfigurasi parameter terbaik pada model LightGBM. Pendekatan ini dipilih karena mampu mengevaluasi berbagai kombinasi parameter secara sistematis serta mengurangi potensi bias akibat pembagian data tertentu.

Tabel 5 menyajikan nilai-nilai parameter yang digunakan dalam proses pencarian konfigurasi optimal.

Tabel 5. Parameter GridSearchCV untuk Model LightGBM

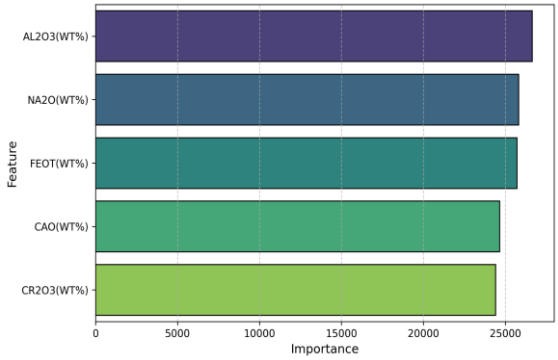
Parameter	Values
num leaves	[15, 31, 63]
max depth	[4, 6, 8]
learning rate	[0.01, 0.03, 0.05]
n estimators	[200, 400, 600]
subsample	[0.8, 1.0]
colsample bytree	[0.8, 1.0]

Setelah melalui proses *hypertuning*, diperoleh kombinasi parameter terbaik. Rincian nilai parameter final yang digunakan ditunjukkan pada Tabel 6.

Tabel 6. Parameter terbaik untuk Model LightGBM

Parameter	Values
num leaves	63
max depth	8
learning rate	0.05
n estimators	600
subsample	0.8
colsample bytree	0.8

Setelah diperoleh parameter optimal, dilakukan analisis lebih lanjut untuk mengetahui kontribusi masing-masing variabel terhadap performa model. Hal ini ditunjukkan melalui *feature importance* dari LightGBM yang disajikan pada Gambar 5.



Gambar 5. *feature Importance*

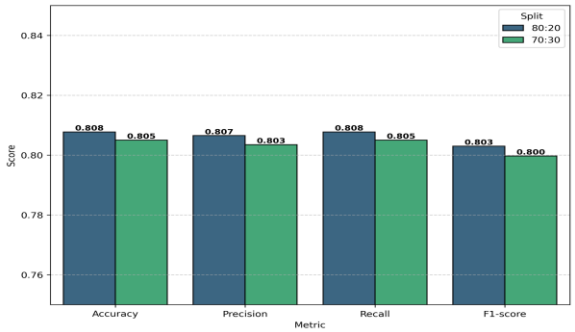
Analisis *feature importance* menunjukkan bahwa *AL2O3(WT%), NA2O(WT%),* dan *FEOT(WT%)* menjadi variabel paling dominan, dengan *CAO(WT%)* dan *CR2O3(WT%)* juga berperan penting dalam akurasi model LightGBM.

Setelah *hypertuning*, LightGBM dievaluasi pada pembagian data 80:20 dan 70:30. Hasilnya menunjukkan performa yang konsisten dengan nilai akurasi, presisi, *recall*, dan *F1-score* yang seimbang, sebagaimana ditunjukkan pada Tabel 7.

Tabel 7. Hasil Evaluasi Model LightGBM Setelah *Hypertuning*

Algoritma	Akurasi	Precision	Recall	F1-score
80:20	0.808	0.807	0.808	0.803
70:30	0.805	0.803	0.805	0.800

Model LightGBM setelah *hypertuning* menunjukkan performa yang sangat stabil pada kedua skenario. Pada *split* 80:20, diperoleh akurasi 80,8%, sebuah peningkatan substansial dari akurasi awal sebesar 73,9%. Hal ini membuktikan betapa krusialnya tahap optimasi *hyperparameter*. Nilai *precision, recall,* dan *F1-score* yang seimbang dan tinggi menunjukkan bahwa model tidak hanya akurat, tetapi juga andal dalam mengklasifikasikan setiap kelas. Performa yang relatif konsisten pada *split* 70:30 juga menandakan bahwa model yang telah dioptimalkan ini bersifat robust dan tidak terlalu sensitif terhadap variasi kecil pada data latih. Perbandingan metrik evaluasi sebelum dan sesudah *tuning* dapat divisualisasikan pada Gambar 6.



Gambar 6. Perbandingan Metrik Evaluasi LightGBM

4. Kesimpulan dan Saran

Berdasarkan perbandingan tiga algoritma *gradient boosting*, penelitian ini menyimpulkan bahwa LightGBM merupakan model yang paling unggul untuk tugas klasifikasi lingkungan tektonik batuan vulkanik. Proses optimasi menggunakan *Grid Search Cross-Validation* terbukti sangat efektif, di mana kinerja LightGBM berhasil ditingkatkan secara signifikan hingga mencapai akurasi 80,87%. Analisis *feature importance* juga mengidentifikasi bahwa komposisi Al_2O_3 (Aluminium Oksida), Na_2O (Natrium Oksida), dan FeOT (Besi Oksida Total) merupakan variabel geokimia yang paling berpengaruh dalam klasifikasi lingkungan teknik.

Meskipun metode ini menawarkan keunggulan berupa klasifikasi yang cepat dan objektif, penelitian ini juga menemukan adanya kekurangan, yaitu kesulitan model dalam membedakan kelas-kelas minoritas yang memiliki karakteristik geokimia tumpang tindih. Oleh karena itu, untuk pengembangan selanjutnya, disarankan agar penelitian mendatang menyertakan fitur tambahan seperti unsur jejak (*trace elements*), mengeksplorasi arsitektur model lain seperti *Deep Learning*, serta menerapkan teknik *resampling* seperti SMOTE untuk meningkatkan akurasi pada kelas-kelas yang sulit dibedakan.

REFERENSI

[1] K. Ueki, H. Hino, and T. Kuwatani, “An Introduction to SGTPPR: Sparse Geochemical Tectono-Magmatic Setting Probabilistic MembershiP DiscriminatoR,”

Geochemistry, Geophys. Geosystems, vol. 25, no. 2, 2024, doi: 10.1029/2023GC011237.

[2] Z. Yan, Yucong; Zhang, Zuocheng; Zhou, Xiaocheng; Wang, Guangcai; He, Miao; Tian, Jiao; Dong, Jinyuan; Li, Jingchao; Bai, Yunfei; Zeng, Zhaojun; Wang, Yuwen; Yao, Bingyu; Xing, Gaoyuan; Cui, Shihan; Shi, "Geochemical characteristics of hot springs in active fault zones within the northern Sichuan-Yunnan block: Geochemical evidence for tectonic activity," *J. Hydrol.*, vol. 09, no. 5, pp. 7352–7363, 2024.

[3] K. Itano and H. Sawada, "Revisiting the Geochemical Classification of Zircon Source Rocks Using a Machine Learning Approach," *Math. Geosci.*, vol. 56, no. 6, pp. 1139–1160, 2024, doi: 10.1007/s11004-023-10128-z.

[4] F. Liu, Bo; Zhai, Mingguo; Wen, "Developing a new geochemical database management system of 2 metamorphic rock in response to the big data era," *SSRN*, 2023, doi: http://dx.doi.org/10.2139/ssrn.4940952.

[5] J. Tamanna; Hezel, Dominik C.; Srivastava, Nishtha; Faber, "Using Machine Learning for automatic rock classification .," *Am. Mineral.*, pp. 1–37, 2025, doi: https://doi.org/10.2138/am-2025-9958.

[6] H. Utama *et al.*, "Integrasi Augmentasi Data dan Machine Learning dalam Prediksi Magnitudo Gempa Bumi," vol. 2, no. 3, pp. 97–108, 2025.

[7] M. Risha, P. Liu, and M. Risha, "Data-Driven Facies Prediction: A Comparative Study of Random Forest, XGBoost, SVM, CatBoost, and K-Means," pp. 0–3, 2025, [Online]. Available: https://eartharxiv.org/repository/view/9382/

[8] K. Ileri, "Comparative analysis of CatBoost, LightGBM, XGBoost, RF, and DT methods optimised with PSO to estimate the number of k-barriers for intrusion detection in wireless sensor networks," *Int. J. Mach. Learn. Cybern.*, vol. 16, no. 9, pp. 6937–6956, 2025, doi: 10.1007/s13042-025-02654-5.

[9] C. Yu, Y. Jin, Q. Xing, Y. Zhang, S. Guo, and S. Meng, "Advanced User Credit Risk Prediction Model Using LightGBM, XGBoost and Tabnet with SMOTEENN," *2024 IEEE 6th Int. Conf. Power, Intell. Comput. Syst. ICPICS 2024*, pp. 876–883, 2024, doi: 10.1109/ICPICS62053.2024.10796247.

[10] J. Qin, Ben; Fang, Huang; Shichun, Huang; Andre, Python; Yunfeng, Chen; ZhangZhou, "Global mantle clinopyroxene data (major and trace elements)," *GFZ Data Serv.*, 2022, doi: https://doi.org/10.5880/digis.e.2024.007.

[11] E. W. Nabila, Hanifah Afkar; Pamungkas, "PERBANDINGAN ALGORITMA MACHINE LEARNING: SVM, RANDOM FOREST, DAN XGBOOST UNTUK PREDIKSI STROKE," vol. 10, no. 2, pp. 1098–1110, 2025.

[12] R. Zizilia *et al.*, "Klasifikasi Penyakit Kanker Paru-Paru dengan Algoritma Extreme Gradient Boosting (XGBoost) dan Mutual Information sebagai Metode Feature Selection Lung Cancer Classification Using the Extreme Gradient Boosting (XGBoost) Algorithm and Mutual Informatio," vol. 14, pp. 2198–2214, 2025.

[13] Jan Melvin Ayu Soraya Dachi and Pardomuan Sitompul, "Analisis Perbandingan Algoritma XGBoost dan Algoritma Random Forest Ensemble Learning pada Klasifikasi Keputusan Kredit," *J. Ris. Rumpun Mat. Dan Ilmu Pengetah. Alam*, vol. 2, no. 2, pp. 87–103, 2023, doi: 10.55606/jurrimipa.v2i2.1470.

[14] I. M. Karo Karo, "Implementasi Metode XGBoost dan Feature Important untuk Klasifikasi pada Kebakaran Hutan dan Lahan," *J. Softw. Eng. Inf. Commun. Technol.*, vol. 1, no. 1, pp. 11–18, 2022, doi: 10.17509/seict.v1i1.29347.

[15] Y. I. Mahendra and R. E. Putra, "Penerapan Algoritma Gradient Boosted Decision Tree (GBDT) untuk Klasifikasi Serangan DDoS," *J. Informatics Comput. Sci.*, vol. 6, no. 01, pp. 158–166, 2024, doi: 10.26740/jinacs.v6n01.p158-166.

[16] A. Pramudyantoro, E. Utami, and D. Ariatmanto, "JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika) Journal homepage: https://jurnal.stkippgritulungagung.ac.id/index.php/jipi PENGGABUNGAN K-NEAREST NEIGHBORS DAN LIGHTGBM UNTUK PREDIKSI DIABETES PADA DATASET PIMA INDIANS: MENGGUNAKAN PENDEKATAN," vol. 9, no. 3, pp. 1133–1144, 2024, [Online]. Available: https://doi.org/10.29100/jipi.v9i3.4966

[17] P. Septiana Rizky, R. Haiban Hirzi, and U. Hidayaturrohman, "Perbandingan Metode LightGBM dan XGBoost dalam Menangani Data dengan Kelas Tidak Seimbang," *J Stat. J. Ilm. Teor. dan Apl. Stat.*, vol. 15, no. 2, pp. 228–236, 2022, doi: 10.36456/jstat.vol15.no2.a5548.

[18] J. T. Hancock and T. M. Khoshgoftaar, "CatBoost for big data: an interdisciplinary review," *J. Big Data*, vol. 7, no. 1, 2020, doi: 10.1186/s40537-020-00369-8.

[19] A. Febriansyah Istianto, A. Id Hadiana, and F. Rakhmat Umbara, "Prediksi Curah Hujan Menggunakan Metode Categorical Boosting (Catboost)," *JATI (Jurnal Mhs. Tek. Inform.*, vol. 7, no. 4, pp. 2930–2937, 2024, doi: 10.36040/jati.v7i4.7304.

[20] Y. Purbolingga, D. Marta, A. Rahmawatia, and B. Wajhi, "Perbandingan Algoritma CatBoost dan XGBoost dalam Klasifikasi Penyakit Jantung," *J. APTEK Vol. 15 No 2 126-133*, vol. 15, no. 2, pp. 126–133, 2023, [Online]. Available: http://journal.upp.ac.id/index.php/aptek/article/download/1930/1163/4970

[21] A. Ahmadi, S. S. Sharif, and Y. M. Banad, "A Comparative Study of Sampling Methods with Cross-Validation in the FedHome Framework," *IEEE Trans. Parallel Distrib. Syst.*, vol. 36, no. 3, pp. 570–579, 2025, doi: 10.1109/TPDS.2025.3526238.

[22] M. R. Salmanpour *et al.*, "Machine Learning Evaluation Metric Discrepancies Across Programming Languages and Their Components in Medical Imaging Domains: Need for Standardization," *IEEE Access*, vol. 13, no. February, pp. 47217–47229, 2025, doi: 10.1109/ACCESS.2025.3549702.

[23] Sugiarto *et al.*, "Optimizing The XGBoost Model with Grid Search Hyperparameter Tuning for Maximum Temperature Forecasting," *J. Appl. Data Sci.*, vol. 6, no. 4, pp. 2517–2529, 2025, doi: 10.47738/jads.v6i4.885.

[24] F. Budiman, "SVM-RBF parameters testing optimization using cross validation and grid search to improve multiclass classification," *Sci. Vis.*, vol. 11, no. 1, pp. 80–90, 2019, doi: 10.26583/sv.11.1.07.

Hans Santoso, saat ini sebagai mahasiswa program studi Teknik Informatika Universitas Tarumanagara angkatan 2022.

Sabrina Phalosa Phai, saat ini sebagai mahasiswa program studi Teknik Informatika Universitas Tarumanagara angkatan 2022.
Sarah Barbara, saat ini sebagai mahasiswa program studi Teknik Informatika Universitas Tarumanagara angkatan 2022.
Maryanto, saat ini sebagai mahasiswa program studi Teknik Informatika Universitas Tarumanagara angkatan 2022.