JTERA (Jurnal Teknologi Rekayasa), Vol. 10, No. 2, Desember 2025, Hal. 95-104 p-ISSN 2548-737X
Terakreditasi "**Peringkat 3**" oleh Kemdiktisaintek, Nomor : 10/C/C3/DT.05.00/2025 e-ISSN 2548-8678
DOI: 10.31544/jtera.v10.i2.2025.95-104

Optimasi Hyperparameter Model Ensemble untuk Klasifikasi Sentimen Ulasan OVO

Annisa Himatul Chasanah[1], Harun Al Azies[2]
[1,2] Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Dian Nuswantoro, Semarang, Indonesia
[2]*Research Center for Quantum Computing and Materials Informatics*, Fakultas Ilmu Komputer,
Universitas Dian Nuswantoro, Semarang, Indonesia
harun.alazies@dsn.dinus.ac.id

Abstrak

Pertumbuhan layanan dompet digital di Indonesia mendorong meningkatnya jumlah ulasan pengguna yang mengandung opini penting terkait kualitas layanan. Analisis sentimen menjadi penting untuk memahami persepsi pengguna terhadap aplikasi OVO. Penelitian ini menganalisis 10.644 ulasan dari Google Playstore yang dikumpulkan melalui teknik web scraping. Ulasan tersebut diproses melalui tahapan *text preprocessing*, representasi fitur menggunakan Word2Vec, serta penyeimbangan kelas menggunakan *Synthetic Minority Oversampling Technique* (SMOTE). Tiga algoritma *ensemble learning* yaitu *Random Forest, XGBoost,* dan *LightGBM* diterapkan dan dioptimasi melalui *Grid Search* dan *Randomized Search*, dengan evaluasi menggunakan *10-Fold Cross-Validation* serta uji statistik *paired t-test*. Hasil menunjukkan bahwa meskipun *XGBoost* dan *LightGBM* memperoleh nilai *cross-validation* yang lebih tinggi, performa terbaik pada data uji dicapai oleh *Random Forest*. Model tersebut mencapai akurasi 89,90% dan ROC-AUC Macro 91,11% pada skema *Grid Search*, serta akurasi 89,76% dan ROC-AUC Macro 91,19% pada skema *Randomized Search*. Temuan ini menunjukkan bahwa *Random Forest* memiliki kemampuan generalisasi paling stabil terhadap data ulasan OVO dibandingkan dua model *boosting*. Penelitian ini memberikan kontribusi pada pengembangan analisis sentimen berbahasa Indonesia melalui integrasi Word2Vec, SMOTE, dan optimasi *hyperparameter*, serta membuka peluang eksplorasi lanjutan menggunakan *contextual embedding* dan teknik penyeimbangan data yang lebih adaptif.

Kata kunci: Analisis Sentimen, Word2Vec, SMOTE, Optimasi *Hyperparameter*, *Random Forest*

Abstract

The growth of digital wallet services in Indonesia has driven an increasing number of user reviews, containing valuable opinions on service quality. Sentiment analysis is crucial for understanding user perceptions of the OVO app. This study analysed 10,644 reviews from the Google Play Store collected through web scraping techniques. The reviews were processed through text preprocessing, feature representation using Word2Vec, and class balancing using the Synthetic Minority Oversampling Technique (SMOTE). Three ensemble learning algorithms, namely Random Forest, XGBoost, and LightGBM, were implemented and optimised through Grid Search and Randomised Search, with evaluation using 10-Fold Cross-Validation and paired t-test statistical tests. The results show that although XGBoost and LightGBM obtained higher cross-validation scores, the best performance on the test data was achieved by Random Forest. The model achieved 89.90% accuracy and 91.11% Macro ROC-AUC in the Grid Search scheme, and 89.76% accuracy and 91.19% Macro ROC-AUC in the Randomised Search scheme. These findings indicate that Random Forest has the most stable generalisation ability on OVO review data compared to the two boosting models. This research contributes to the development of Indonesian-language sentiment analysis by integrating Word2Vec, SMOTE, and hyperparameter optimisation. It opens up opportunities for further exploration using contextual embedding and more adaptive data balancing techniques.

Keywords: *Sentiment Analysis, Word2Vec, SMOTE, Hyperparameter Optimization, Random Forest*

I. PENDAHULUAN

Perkembangan teknologi informasi yang semakin pesat telah mendorong perubahan besar dalam cara

masyarakat bertransaksi dan mengelola keuangan. Kemajuan ini turut melahirkan inovasi *financial technology* (*fintech*) yang memadukan layanan keuangan dengan teknologi untuk mempermudah aktivitas pembayaran digital [1]. Menurut Bank Indonesia, nilai transaksi uang elektronik mencapai Rp19,2 triliun pada Februari 2021, meningkat 26,42% dibandingkan tahun sebelumnya, yang menunjukkan tingginya adopsi layanan pembayaran digital di Indonesia [2]. Salah satu bentuk *fintech* yang paling banyak digunakan adalah dompet digital (*e-wallet*) karena menawarkan kemudahan, efisiensi, dan fleksibilitas dalam bertransaksi. Di Indonesia, penggunaan *e-wallet* seperti OVO, DANA, dan GoPay terus meningkat seiring pergeseran masyarakat menuju sistem pembayaran nontunai [3]. Selain itu, survei tahun 2024 menunjukkan bahwa 96% masyarakat Indonesia telah menggunakan layanan *e-wallet* untuk berbagai kebutuhan sehari-hari, termasuk pembelian produk digital, transportasi, dan pembayaran tagihan [2]. OVO, sebagai salah satu platform populer, tidak hanya menawarkan layanan pembayaran, tetapi juga menyediakan berbagai fitur seperti *cashback*, promo, dan program loyalitas. Namun, tingginya jumlah pengguna tidak selalu diikuti dengan kualitas layanan yang memuaskan. Berbagai keluhan mengenai keterlambatan transaksi, gangguan sistem, dan kendala teknis banyak ditemukan pada ulasan pengguna di Google Playstore [4], sehingga perlu dilakukan analisis untuk memahami persepsi dan tingkat kepuasan pengguna secara lebih mendalam.

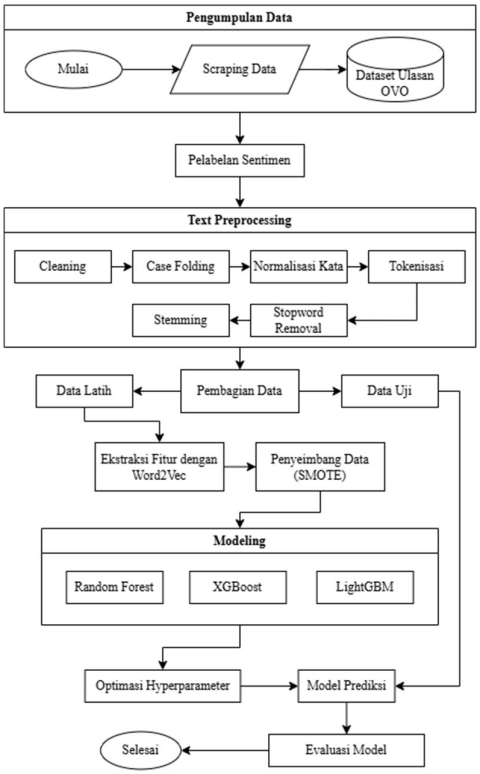
Ulasan pengguna pada platform digital merupakan sumber informasi penting untuk mengetahui persepsi dan emosi pengguna terhadap suatu layanan. Analisis sentimen menjadi metode yang banyak digunakan untuk mengekstraksi opini secara otomatis melalui pendekatan *Natural Language Processing* (NLP) [5]. Berbagai penelitian sebelumnya telah menerapkan metode berbeda-beda namun masih memiliki keterbatasan masing-masing. Penelitian yang menggunakan *K-Nearest Neighbor* (KNN) mencapai akurasi 84,86%, tetapi metode ini hanya mengandalkan jarak antarfitur tanpa representasi semantik serta tidak menggunakan evaluasi *cross-validation* sehingga stabilitas model belum teruji [6]. Penelitian berbasis LSTM dengan *word embedding* pada ulasan OVO dan LinkAja memperoleh akurasi 83%, namun dataset yang sangat tidak seimbang (91% negatif dan 9% positif) menimbulkan bias signifikan terhadap kelas mayoritas [7]. Penelitian lain menggunakan *Naïve Bayes* dan TF-IDF pada ulasan Shopeepay, Gopay, dan OVO dengan akurasi tertinggi 86%, namun pendekatan berbasis frekuensi kata tersebut tidak mampu menangkap kedekatan makna antar kata maupun nuansa emosional dalam teks [8]. Selain itu, penelitian pada domain berbeda seperti analisis sentimen aplikasi *JKN Mobile* menunjukkan bahwa penggunaan SVM yang dioptimasi *Grid Search* dan *Randomized Search* serta teknik SMOTE masih menghasilkan kinerja yang terbatas pada kelas minoritas [9]. Penelitian lain mengenai sentimen UKT menggunakan *Random Forest*, TF-IDF, SMOTE, dan *Grid Search* masih terbatas karena hanya mengevaluasi satu algoritma tanpa melibatkan perbandingan model ataupun strategi optimasi *hyperparameter* yang bervariasi [10], [11]. Berdasarkan studi tersebut, terlihat bahwa representasi fitur masih belum kontekstual, ketidakseimbangan kelas belum tertangani dengan baik, dan model klasifikasi yang digunakan belum mengevaluasi performa secara komprehensif melalui optimasi *hyperparameter* yang memadai. Selain itu, terdapat kesenjangan antara performa yang diharapkan dari model-model tersebut dan performa aktual yang ditunjukkan pada data nyata, sehingga diperlukan pendekatan yang mampu memberikan hasil yang lebih stabil dan konsisten.

Penelitian ini bertujuan untuk mengevaluasi dan membandingkan kinerja tiga algoritma *ensemble*, yaitu *Random Forest, XGBoost,* dan *LightGBM*, dalam klasifikasi sentimen ulasan pengguna OVO melalui integrasi Word2Vec, SMOTE, serta dua metode optimasi *hyperparameter*, yaitu *Grid Search* dan *Randomized Search*. Untuk menghasilkan representasi fitur yang lebih kontekstual, penelitian ini menggunakan Word2Vec yang mampu mempelajari hubungan semantik antarkata secara lebih mendalam. Ketidakseimbangan kelas ditangani menggunakan *Synthetic Minority Oversampling Technique* (SMOTE) agar proses pembelajaran tidak bias terhadap kelas mayoritas. Ketiga algoritma *ensemble* tersebut dipilih karena kemampuannya dalam menangani data berdimensi tinggi dan pola non-linear, sedangkan proses optimasi dilakukan melalui *Grid Search* dan *Randomized Search* yang divalidasi menggunakan *10-Fold Cross Validation* untuk memastikan evaluasi yang stabil dan andal. Secara teoretis, penelitian ini berkontribusi pada integrasi representasi semantik, penyeimbangan data, serta evaluasi dua pendekatan optimasi *hyperparameter* pada model *ensemble* dalam konteks analisis sentimen berbahasa Indonesia. Secara praktis, temuan penelitian ini diharapkan dapat membantu pengembang OVO dan pelaku industri *fintech* dalam memahami persepsi pengguna secara lebih akurat sehingga dapat mendukung pengambilan keputusan berbasis data. Penelitian ini memberikan kontribusi ilmiah berupa evaluasi komprehensif lintas-algoritma dan lintas-metode optimasi, yang

belum banyak dieksplorasi pada studi analisis sentimen berbahasa Indonesia sebelumnya.

II. METODE PENELITIAN

Metode penelitian ini disajikan melalui sebuah diagram alur yang memberikan gambaran umum mengenai proses penelitian secara terstruktur. Visualisasi tersebut membantu menjelaskan urutan tahapan yang dilakukan, sebagaimana ditunjukkan pada Gambar 1.



Gambar 1. Alur Penelitian

A. Pengumpulan Data

Dataset penelitian ini diperoleh dari ulasan pengguna aplikasi OVO di Google Play Store. Proses pengambilan data dilakukan menggunakan pustaka Python *Google-Play-Scraper* dengan rentang waktu Juni hingga Oktober 2025. Sebanyak 10.644 ulasan berhasil dikumpulkan, mencakup teks ulasan, skor penilaian (*rating*), tanggal ulasan, dan informasi pengguna. Data disimpan dalam format CSV sebagai dataset utama. Sebelum digunakan dalam tahap analisis, dilakukan pemeriksaan terhadap keberadaan *missing values* dan konsistensi data untuk memastikan kualitas dataset yang akan diolah pada tahap berikutnya.

B. Pelabelan Sentimen

Setelah data berhasil dikumpulkan, dilakukan proses pelabelan sentimen berdasarkan nilai *rating* yang diberikan oleh pengguna. Ulasan dengan rating 4–5 dikategorikan sebagai sentimen positif, sedangkan ulasan dengan rating 1–3 dikategorikan sebagai sentimen negatif. Pendekatan berbasis *rating* ini digunakan karena dianggap objektif dalam merepresentasikan persepsi pengguna terhadap layanan aplikasi. Hasil dari proses pelabelan menghasilkan dua kelas utama yang berfungsi sebagai variabel target (Y), sedangkan teks ulasan yang telah melalui tahap pra-pemrosesan berfungsi sebagai variabel fitur (X).

C. Preprocessing Data

Tahapan pra-pemrosesan bertujuan untuk membersihkan dan menstandarkan data teks agar dapat diolah dengan optimal oleh model pembelajaran mesin. Langkah-langkah yang dilakukan meliputi :

1) ***Cleaning*:** tahap menghapus tautan, tanda baca, angka, serta karakter non-alfabet agar teks menjadi bersih dan hanya memuat informasi yang relevan. Langkah ini dilakukan untuk mengurangi *noise* yang dapat mengganggu proses analisis fitur [12].
2) ***Case Folding:*** dilakukan dengan mengubah seluruh huruf menjadi huruf kecil (*lowercase*) guna menjaga konsistensi bentuk teks dan menghindari perbedaan makna akibat penggunaan huruf kapital [13].
3) ***Normalisasi Kata***: tahap ini dilakukan dengan menyelaraskan kata tidak baku, singkatan, dan bentuk bahasa informal ke bentuk baku menggunakan *kamus_singkatan.csv*. Tahap ini membantu menyamakan variasi penulisan kata agar model dapat mengenali konteks dengan lebih akurat.
4) ***Tokenisasi:*** proses memecah kalimat menjadi satuan kata (*token*) menggunakan pustaka NLTK. Dengan *tokenisasi*, setiap kata dapat diidentifikasi sebagai unit analisis tersendiri dalam teks.
5) ***Stopword Removal:*** dilakukan dengan menghapus kata-kata umum yang tidak memberikan kontribusi penting terhadap makna kalimat, seperti "yang", "dan", atau "di", namun tetap mempertahankan kata bermuatan sentimen seperti "tidak", "bagus", dan "lambat".
6) ***Stemming:*** tahap ini bertujuan mengubah setiap kata menjadi bentuk dasarnya dengan memanfaatkan pustaka Sastrawi *Stemmer Factory*. Dengan melakukan stemming, variasi morfologis dari kata-kata yang memiliki akar yang sama dapat diseragamkan, sehingga jumlah

fitur berkurang dan kualitas konsistensi data menjadi lebih baik [14].

D. Ekstraksi Fitur dengan Word2Vec

Word2Vec merupakan salah satu metode *word embedding* yang merepresentasikan teks ke dalam bentuk vektor berdimensi tinggi. Representasi ini dibangun dari pola kemunculan kata dalam korpus sehingga kata-kata dengan penggunaan yang mirip ditempatkan pada posisi yang berdekatan dalam ruang vektor[15].

Pada penelitian ini digunakan arsitektur *Skip-Gram* (sg=1), yaitu pendekatan yang mempelajari kata target untuk memprediksi kata-kata di sekitarnya [16]. Berbeda dengan *Continuous Bag of Words* (CBOW), *Skip-Gram* lebih efektif dalam mengolah kata dengan frekuensi rendah, sehingga cocok digunakan untuk teks ulasan yang memiliki variasi kosakata yang luas [17].

Model Word2Vec dilatih menggunakan pustaka Gensim dengan pengaturan parameter *vector_size* = 300, *window* = 5, *min_count* = 2, *epochs* = 50, dan *seed* = 42. Konfigurasi ini dipilih untuk menghasilkan representasi kata yang stabil sekaligus mampu menangkap hubungan semantik dalam ulasan pengguna. Parameter *window* sebesar lima memungkinkan model mempelajari konteks kata dalam rentang yang cukup luas, sedangkan nilai *min_count* = 2 memastikan bahwa hanya kata-kata yang muncul lebih dari satu kali yang digunakan dalam proses pelatihan. Hasil pelatihan menghasilkan *embedding* yang memetakan kata-kata dengan makna atau konteks serupa ke posisi vektor yang berdekatan dalam ruang fitur.

E. Pembagian Data dan Penyeimbang Data

Setelah representasi fitur teks dihasilkan melalui model Word2Vec, dataset dibagi menjadi dua subset, yaitu data latih (*training set*) dan data uji (*testing set*) dengan rasio 80:20 menggunakan metode *stratified sampling*. Pembagian ini dilakukan untuk memastikan proporsi label sentimen tetap seimbang di kedua subset, sehingga hasil pelatihan dan pengujian model menjadi lebih representatif. Rasio 80:20 dipilih karena memberikan keseimbangan antara jumlah data yang digunakan untuk melatih model dan data yang digunakan untuk mengevaluasi performanya.

Selanjutnya, dilakukan proses penyeimbangan kelas pada data latih menggunakan metode *Synthetic Minority Over-sampling Technique* (SMOTE). Ketidakseimbangan kelas terjadi karena jumlah ulasan negatif lebih banyak dibandingkan dengan ulasan positif, yang dapat menyebabkan model cenderung bias terhadap kelas mayoritas. Teknik ini bekerja dengan menghasilkan sampel sintetis pada kelas minoritas melalui pembentukan titik baru di antara data asli dan tetangga terdekatnya menggunakan pendekatan *k-nearest neighbors* [18]. Penerapan SMOTE bertujuan untuk memperoleh distribusi kelas yang lebih seimbang pada data latih sehingga model dapat mempelajari pola dari kedua kelas secara lebih optimal. Metode ini hanya diterapkan pada data latih untuk mencegah terjadinya *data leakage*, sehingga evaluasi pada data uji tetap objektif dan merefleksikan kinerja model yang sesungguhnya.

F. Modeling

Tahap modeling dilakukan dengan menguji tiga algoritma klasifikasi *Random Forest, XGBoost,* dan *LightGBM* untuk menentukan model dengan performa terbaik dalam mengklasifikasikan sentimen. Pemilihan ketiga algoritma tersebut didasarkan pada hasil studi sebelumnya yang menunjukkan bahwa *Random Forest, XGBoost,* dan *LightGBM* memiliki kemampuan unggul dalam menangani pola data yang bersifat non-linear dan berdimensi tinggi [19], [20], [21]. Selain itu, ketiga metode *ensemble* ini terbukti memberikan performa yang lebih stabil dan akurat pada berbagai penelitian berbasis representasi vektor, sehingga relevan digunakan untuk memodelkan sentimen pada ulasan pengguna OVO yang memiliki keragaman konteks dan struktur bahasa.

1) Random Forest

Random Forest merupakan algoritma *ensemble learning* berbasis *bagging* yang membangun sejumlah pohon keputusan dari sampel data yang dipilih secara acak. Pada proses pembentukannya, pemilihan baris (data) dan kolom (fitur) dilakukan secara acak untuk menghasilkan variasi model pada setiap pohon. Hasil prediksi dari seluruh pohon tersebut kemudian digabungkan melalui proses agregasi, sehingga meningkatkan akurasi dan stabilitas model dibandingkan dengan penggunaan satu pohon keputusan saja [22].

2) Extreme Gradient Boosting (XGBoost)

XGBoost merupakan algoritma *ensemble* berbasis *Gradient Boosting Decision Tree* (GBDT) yang dirancang untuk meningkatkan efisiensi komputasi baik dari sisi waktu maupun penggunaan memori. Pada proses pelatihannya, *XGBoost* membangun pohon keputusan secara bertahap, di mana setiap pohon berikutnya dilatih untuk memperbaiki kesalahan prediksi dari pohon sebelumnya. Mekanisme ini memungkinkan model untuk menggabungkan sejumlah pohon sederhana (*weak learners*) menjadi sebuah model prediksi yang lebih kuat dan lebih akurat. *XGBoost* juga memiliki kemampuan generalisasi yang baik pada data berdimensi tinggi serta menyediakan mekanisme

regularisasi untuk mengurangi risiko *overfitting* [23].

3) Light Gradient Boosting Machine (LightGBM)

LightGBM merupakan algoritma *gradient boosting* yang menggunakan pohon keputusan sebagai model dasarnya. Algoritma ini dirancang agar lebih cepat dan efisien, terutama pada data berukuran besar. Dua mekanisme utamanya adalah *Gradient-based One-Side Sampling* (GOSS), yang mempercepat proses pemilihan split dengan memprioritaskan sampel bergradien besar, serta *Exclusive Feature Bundling* (EFB), yang mengurangi jumlah fitur dengan menggabungkan fitur-fitur yang jarang muncul secara bersamaan. Selain itu, *LightGBM* menggunakan strategi pertumbuhan pohon secara *leaf-wise*, sehingga dapat menurunkan nilai kesalahan lebih cepat dibandingkan metode *level-wise*. Dengan pendekatan tersebut, *LightGBM* mampu memberikan kinerja yang lebih cepat dan akurat untuk tugas klasifikasi [24].

G. Optimasi Hyperparameter

Optimasi *hyperparameter* merupakan tahapan penting untuk memastikan model pembelajaran mesin beroperasi dengan konfigurasi yang paling efektif. Tahap ini dilakukan dengan menyesuaikan nilai parameter yang memengaruhi proses belajar model, sehingga kinerjanya dapat meningkat secara signifikan. Pada penelitian ini digunakan dua metode optimasi, yaitu *Grid Search* dan *Randomized Search*. *Grid Search* melakukan pencarian secara sistematis terhadap seluruh kombinasi parameter yang telah ditentukan untuk menemukan konfigurasi terbaik berdasarkan metrik evaluasi [25]. Metode ini memberikan hasil yang komprehensif, namun memerlukan waktu komputasi yang lebih tinggi ketika jumlah parameter semakin besar. Sebaliknya, *Randomized Search* menyeleksi sebagian kombinasi parameter secara acak, sehingga proses pencarian menjadi lebih efisien meskipun tidak selalu menjamin ditemukannya konfigurasi yang paling optimal [26]. Kedua pendekatan ini diterapkan untuk menguji efektivitas proses optimasi pada model *Random Forest, XGBoost,* dan *LightGBM*.

Proses optimasi tersebut dikombinasikan dengan *10-Fold Cross Validation* untuk memastikan evaluasi model lebih stabil dan tidak bergantung pada satu pembagian data. Setiap iterasi melibatkan pelatihan model pada sembilan bagian data dan pengujian pada satu bagian yang berbeda, hingga seluruh lipatan mendapatkan giliran sebagai data uji. Selain itu, digunakan *stratified splitting* agar distribusi kelas pada setiap lipatan tetap proporsional dengan data asli, sehingga proses validasi tetap representatif meskipun dataset memiliki ketidakseimbangan kelas [27].

Parameter yang digunakan dalam proses optimasi ditampilkan pada Tabel 1, Tabel 2, dan Tabel 3. Variasi nilai tersebut diuji melalui mekanisme tuning untuk menilai setiap konfigurasi secara sistematis, sehingga diperoleh model dengan performa paling optimal untuk tugas klasifikasi sentimen.

Tabel 1. Parameter Random Forest

Hyperparameter	Nilai
n_estimators	200, 400, 500, 600
max_depth	8, 10, 12
min_samples_split	2, 5, 10
min_samples_leaf	1, 2, 4
max_features	“sqrt”, “log2”
bootstrap	True, False

Tabel 2. Parameter XGboost

Hyperparameter	Nilai
n_estimators	200, 400, 600
max_depth	4, 6, 8
learning_rate	0.01, 0.05, 0.1
subsample	0.8, 0.9
colsample_bytree	0.8, 0.9
min_child_weight	1, 3, 5
reg_lambda	1, 3, 5

Tabel 3. Parameter LightGBM

Hyperparameter	Nilai
boosting_type	“gbdt”, “dart”
learning_rate	0.01, 0.05, 0.1
n_estimators	400, 600
max_depth	6, 8
num_leaves	40, 60, 80
subsample	0.8, 0.9
colsample_bytree	0.8, 0.9
reg_lambda	1, 3

H. Evaluasi Model

Setelah model selesai dilatih, tahap evaluasi dilakukan menggunakan data uji untuk menilai kemampuan model dalam mengklasifikasikan sentimen secara akurat. Metrik evaluasi yang digunakan meliputi *Accuracy*, *Precision*, *Recall*, *F1-Score*, dan ROC-AUC, dengan rumus sebagai berikut [28].

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

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

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

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

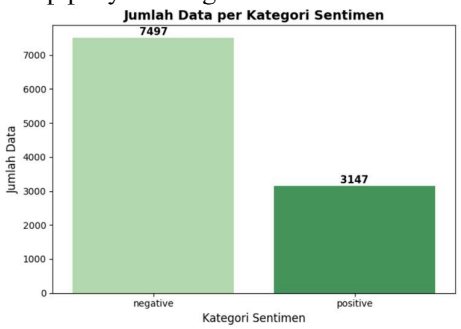
$$ROC - AUC = \int_0^1 TPR(FPR)d(FPR) \quad (5)$$

dengan TP (*True Positive*) adalah jumlah sampel positif yang diklasifikasikan dengan benar, TN (*True Negative*) adalah jumlah sampel negatif yang diklasifikasikan dengan benar, FP (*False Positive*) adalah sampel negatif yang salah diprediksi sebagai positif, dan FN (*False Negative*) adalah sampel positif yang salah diprediksi sebagai negatif.

III. HASIL DAN PEMBAHASAN

A. *EDA dan Preprocessing*

Analisis awal terhadap dataset menunjukkan adanya ketidakseimbangan kelas setelah proses pelabelan sentimen. Distribusi tersebut ditampilkan pada Gambar 2, di mana ulasan negatif mendominasi dengan jumlah 7.497 data, sementara ulasan positif berjumlah 3.147 data. Ketimpangan ini berpotensi memengaruhi performa model jika tidak ditangani pada tahap penyeimbangan data.



Gambar 2. Distribusi Sentimen Positif dan Negatif

Hasil preprocessing disajikan pada Tabel 4, yang memperlihatkan perubahan teks dari bentuk asli yang masih mengandung *noise* menjadi teks yang lebih bersih, terstandarisasi, dan telah melalui proses stemming. Transformasi tersebut menghasilkan representasi yang lebih konsisten, sehingga memudahkan tahap ekstraksi fitur.

Tabel 4. Hasil Preprocessing Teks Ulasan

Teks Asli	Teks Setelah Preprocessing
lemot banget pakai ovo mau transfer aja loding nya lama, padahal sinyal bagus	lambat sekali pakai ovo alih dana muat lama sinyal bagus

B. *Wordcloud*

Wordcloud pada Gambar 3 menampilkan kata-kata yang paling sering muncul dalam ulasan setelah preprocessing. Kata *"tidak"*, *"pakai"*, *"saldo"*, *"transaksi"*, dan *"masuk"* terlihat paling dominan, menunjukkan bahwa banyak ulasan terkait kendala penggunaan aplikasi, terutama proses transaksi dan saldo. Visualisasi ini memberikan gambaran umum mengenai topik yang paling sering dibahas pengguna sebelum dilakukan pemodelan sentimen.

Gambar 3. Wordcloud Ulasan Pengguna Setelah Preprocessing

C. *Ekstraksi Fitur dengan Word2Vec*

Hasil Word2Vec disajikan pada Gambar 4, yang menunjukkan daftar kata dengan kemiripan tertinggi terhadap kata acuan. Pola kemiripan ini menggambarkan bahwa model berhasil menangkap hubungan semantik dalam data ulasan.

```
[15]:
print("Kata terdekat dengan 'bagus':")
print(model_w2v.wv.most_similar("bagus", topn=5))

Kata terdekat dengan 'bagus':
[('sipp', 0.4534696042537689), ('luas', 0.4469189941883087), ('fasilitas', 0.4444698989391327),
('mantapp', 0.43665388226509094), ('baguss', 0.4357360005378723)]
```

Gambar 4. Kata Terdekat Berdasarkan Model Word2Vec

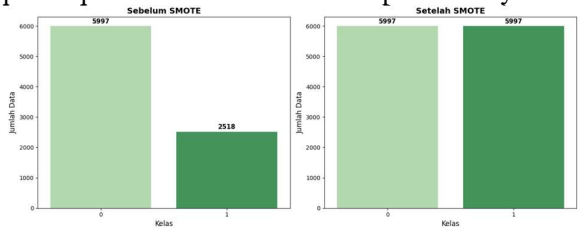
D. *Pembagian Data dan Penyeimbang Data*

Proporsi data setelah pembagian latih–uji ditampilkan pada Tabel 5. Pembagian menggunakan *stratified sampling* sehingga distribusi kelas pada kedua subset tetap mengikuti pola distribusi awal.

Tabel 5. Distribusi Data Latih dan Data Uji

Sentimen	Train	Test
Negatif	5997	1500
Positif	2518	629

Penerapan SMOTE menghasilkan distribusi kelas yang seimbang pada data latih, seperti ditunjukkan pada Gambar 5. Kondisi ini memastikan bahwa proses pelatihan tidak bias terhadap kelas mayoritas.



Gambar 5. Perbandingan Distribusi Kelas Sebelum dan Sesudah SMOTE

E. *Kinerja Model dan Optimasi Hyperparameter*

Bagian ini bertujuan untuk mengevaluasi efektivitas tiga algoritma *ensemble*, yaitu *Random Forest, XGBoost,* dan *LightGBM*, dalam tugas klasifikasi sentimen ulasan pengguna OVO. Fokus analisis diarahkan untuk menilai sejauh mana proses Optimasi *hyperparameter* menggunakan *Grid*

Search dan *Randomized Search* dapat meningkatkan kinerja model, serta mengidentifikasi algoritma dengan kemampuan generalisasi terbaik pada data uji. Evaluasi dilakukan melalui *10-Fold Cross-Validation* untuk melihat konsistensi performa, dilanjutkan dengan uji *paired t-test* guna menentukan signifikansi perbedaan antar model. Hasil akhir pada data uji digunakan sebagai dasar untuk menentukan model yang paling stabil dan akurat dalam memprediksi sentimen.

1) Hasil Evaluasi Cross-Validation (Grid Search)

Evaluasi awal dilakukan menggunakan *10-Fold Cross-Validation* terhadap ketiga algoritma setelah proses optimasi menggunakan *Grid Search*. Tahap ini bertujuan untuk mengamati stabilitas performa model pada setiap fold serta memastikan bahwa konfigurasi *hyperparameter* yang dihasilkan memberikan kinerja optimal. Nilai akurasi per-fold ditampilkan pada Tabel 6.

Tabel 6. Akurasi Per Fold – Grid Search

K-Fold	RF	XGB	LGBM
1	0.9008	0.9458	0.9350
2	0.9100	0.9433	0.9492
3	0.8917	0.9367	0.9367
4	0.9000	0.9408	0.9383
5	0.9141	0.9475	0.9399
6	0.9149	0.9466	0.9491
7	0.9149	0.9408	0.9458
8	0.8874	0.9216	0.9316
9	0.8941	0.9425	0.9366
10	0.8957	0.9374	0.9391

2) Hasil Evaluasi Cross-Validation (Random Search)

Berbeda dengan *Grid Search* yang menguji seluruh kombinasi *hyperparameter*, *Randomized Search* mengevaluasi sebagian kombinasi secara acak sehingga ruang pencarian menjadi lebih luas dengan waktu komputasi lebih efisien. Nilai akurasi per fold ditampilkan pada Tabel 7. Berdasarkan Tabel 7, terlihat bahwa performa *XGBoost* dan *LightGBM* konsisten lebih tinggi dibandingkan *Random Forest* pada sebagian besar fold, dengan rentang akurasi yang relatif stabil di atas 0.92. *LightGBM* menunjukkan nilai akurasi tertinggi pada beberapa fold, terutama pada fold ke-4 dan ke-7, yang mengindikasikan kemampuan model untuk menyesuaikan diri dengan variasi data latih hasil sampling *cross-validation*.

Sementara itu, *Random Forest* menampilkan akurasi yang lebih fluktuatif dan cenderung lebih rendah, meskipun masih berada pada rentang performa yang kompetitif.

Tabel 7. Akurasi Per Fold-Randomized Search

K-Fold	RF	XGB	LGBM
1	0.9042	0.9400	0.9400
2	0.9067	0.9450	0.9433
3	0.8850	0.9242	0.9292
4	0.8992	0.9458	0.9500
5	0.9124	0.9374	0.9374
6	0.9199	0.9450	0.9491
7	0.9016	0.9333	0.9874
8	0.8791	0.9258	0.9308
9	0.8899	0.9425	0.9391
10	0.8866	0.9229	0.9389

Keterangan: *RF = Random Forest, XGB = XGBoost, LGBM = LightGBM, GS = Grid Search, RS = Random Search.*

Pola ini sejalan dengan karakteristik model *boosting* yang umumnya lebih adaptif terhadap kompleksitas data, sehingga menghasilkan skor *cross-validation* yang lebih unggul dibandingkan metode berbasis *bagging* seperti *Random Forest*.

3) Uji Statistik Paired T-Test

Uji *paired t-test* digunakan untuk menilai signifikansi perbedaan akurasi antar model berdasarkan hasil *cross-validation* pada setiap fold, baik pada skema *Grid Search* maupun *Randomized Search*. Hasil pengujian pada Tabel 8 menunjukkan bahwa pada kedua skema, *Random Forest* memiliki perbedaan kinerja yang signifikan dibandingkan *XGBoost* dan *LightGBM* ($p < 0.05$), sehingga dapat disimpulkan bahwa performanya secara konsisten lebih rendah pada tahap pelatihan.

Tabel 8. Hasil Uji Paired T-Test Perbandingan Model

Perbandingan Model	T-Statistic	P-Value	Kesimpulan
RF → XGB (GS)	-16.5769	0.0000	Signifikan
RF → LGBM (GS)	-18.7301	0.0000	Signifikan
XGB → LGBM (GS)	0.0823	0.9362	Tidak signifikan
RF → XGB (RS)	-13.1012	0.0000	Signifikan
RF → LGBM (RS)	-8.5558	0.0000	Signifikan
XGB → LGBM (RS)	-1.5535	0.1547	Tidak signifikan

Sebaliknya, perbandingan antara *XGBoost* dan *LightGBM* menghasilkan p-value ≥ 0.05, yang menunjukkan bahwa kinerja keduanya tidak berbeda secara signifikan. Seluruh perbandingan yang melibatkan *Random Forest* menghasilkan p-value < 0.05, sehingga perbedaan akurasinya dibandingkan dua model *boosting* bersifat signifikan dan menunjukkan bahwa *Random Forest* memiliki performa pelatihan yang lebih rendah. Dengan

demikian, *XGBoost* dan *LightGBM* dapat dianggap setara secara statistik, meskipun *LightGBM* sedikit unggul pada rata-rata akurasinya.

4) Evaluasi Model Terbaik pada Data Test

Evaluasi pada data test dilakukan untuk menilai kemampuan generalisasi model setelah optimasi *hyperparameter* menggunakan *Grid Search* dan *Randomized Search*. Pada skema *Grid Search*, *Random Forest* menunjukkan performa paling baik dengan *accuracy* 89,9%, *precision* 90,14%, dan ROC-AUC Macro 91,11%. *XGBoost* memperoleh nilai ROC-AUC Macro sedikit lebih tinggi (91,12%), namun perbedaannya sangat kecil dan masih berada dalam kisaran performa yang sama secara praktis. Ringkasan performa ketiga model ditunjukkan pada Tabel 9.

Tabel 9. Performa Model Pada Data Test – Grid Search

Metrik	RF	XGB	LGBM
Accuracy	89.90%	89.81%	89.71%
Precision	90.14%	89.82%	89.71%
Recall	89.90%	89.81%	89.71%
F1-Score	89.48%	89.49%	89.39%
ROC-AUC Macro	91.11%	91.12%	90.62%

Pada skema *Randomized Search*, pola performa yang serupa kembali terlihat. *Random Forest* mempertahankan hasil terbaik dengan *accuracy* sebesar 89,76% dan nilai ROC-AUC Macro 91,19%, yang merupakan nilai tertinggi pada skema ini. Baik *XGBoost* maupun *LightGBM* menghasilkan metrik yang kompetitif, tetapi masih berada sedikit di bawah *Random Forest*, terutama pada *accuracy* dan *F1-score*. Ringkasan performa ketiga model pada skema *Randomized Search* ditampilkan pada Tabel 10.

Tabel 10. Performa Model Pada Data Test – Random Search

Metrik	RF	XGB	LGBM
Accuracy	89.76%	89.20%	89.38%
Precision	90.07%	89.15%	89.34%
Recall	89.76%	89.20%	89.38%
F1-Score	89.30%	88.86%	89.07%
ROC-AUC Macro	91.19%	90.69%	90.75%

Hasil pengujian pada data test menunjukkan bahwa *Random Forest* merupakan model dengan performa paling stabil pada kedua pendekatan optimasi. Walaupun *XGBoost* memperoleh nilai *cross-validation* tertinggi pada tahap pelatihan, performanya pada data test berada sedikit di bawah *Random Forest*.

Tabel 11. *Classification Report* Random Forest – Grid Search

	Precision	Recall	F1-Score
negatif	0.89	0.98	0.93
positif	0.93	0.72	0.81
accuracy			0.90

Perbedaan nilai *recall* pada Tabel 11 menunjukkan pola awal kesalahan prediksi model. *Recall* yang sangat tinggi pada kelas negatif (0.98) mengindikasikan bahwa sebagian besar ulasan negatif berhasil dikenali dengan baik. Sebaliknya, *recall* kelas positif yang lebih rendah (0.72) menunjukkan bahwa sejumlah ulasan positif masih sulit dipetakan secara akurat. Meskipun *precision* pada kelas positif relatif tinggi (0.93), variasi ekspresi dalam ulasan positif terutama ketika pujian bercampur dengan keluhan ringan menciptakan kompleksitas semantik yang membuat sebagian prediksi tidak tepat.

Tabel 12. Contoh Kesalahan Prediksi (*Failure Case*)

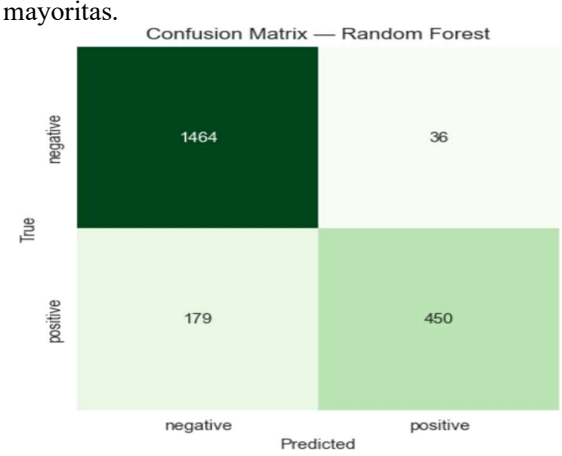
Teks Ulasan Setelah Preprocessing	Label Asli	Prediksi
"lambat sekali pakai ovo mau alih dana saja memuat nya lama padahal sinyal bagus"*	Negatif	Positif

Note: *) Kata bernada negatf lebih dominan secara statistik sehingga konteks positif tidak tertangkap oleh model.

Selanjutnya guna memahami sumber kesalahan tersebut, dilakukan peninjauan terhadap contoh konkret ulasan yang salah diklasifikasikan. Sebagai contoh, salah satu kesalahan prediksi ditunjukkan pada Tabel 12. Ulasan tersebut secara jelas mengungkapkan keluhan mengenai performa aplikasi yang lambat, sehingga seharusnya termasuk kategori negatif. Namun model memprediksinya sebagai positif. Kesalahan ini terutama dipengaruhi oleh keberadaan kata "bagus" yang secara semantik lebih kuat dalam *embedding* dan menggeser representasi kalimat ke arah sentimen positif, sementara ekspresi negatif seperti "lambat" dan "memuat lama" tidak terwakili secara proporsional dalam ruang vektor Word2Vec. Hal ini menunjukkan keterbatasan *embedding* berbasis konteks lokal dalam menangkap struktur makna yang lebih kompleks.

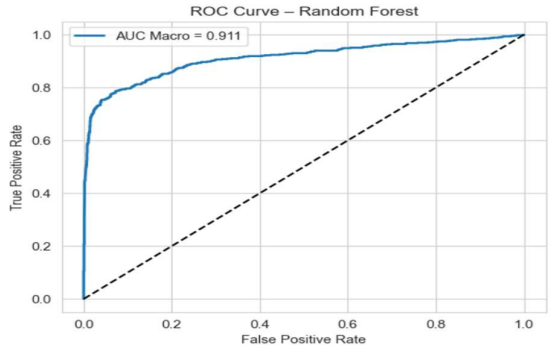
Kinerja *Random Forest* pada data uji juga terlihat jelas melalui visualisasi *confusion matrix* pada Gambar 6. Pada gambar tersebut, model mampu mengklasifikasikan ulasan negatif dengan sangat baik, ditunjukkan oleh nilai *True Negative* sebesar 1.464, sementara kesalahan prediksi terhadap kelas negatif relatif kecil, yaitu hanya 36 kasus yang diprediksi sebagai positif. Pada kelas positif, *Random Forest* juga menunjukkan performa yang konsisten

dengan nilai *True Positive* sebesar 450, meskipun masih terdapat 179 ulasan positif yang salah diklasifikasikan sebagai negatif. Distribusi ini menunjukkan bahwa model memiliki kecenderungan yang seimbang dalam mengenali kedua kelas, tanpa bias kuat terhadap kelas mayoritas.



Gambar 6. Confusion Matrix Model Random Forest - Grid Search

Kemampuan pemisahan kelas oleh model juga diperkuat oleh kurva ROC pada Gambar 7.



Gambar 7. Kurva ROC Model Random Forest - Grid Search

Kurva tersebut menunjukkan bahwa *Random Forest* memiliki nilai AUC Macro sebesar 0.911, mengindikasikan kemampuan diskriminatif yang sangat baik dalam membedakan sentimen positif dan negatif. Semakin mendekat kurva terhadap sisi kiri atas grafik, semakin tinggi kemampuan model dalam mengurangi kesalahan prediksi, dan pola kurva pada gambar tersebut menggambarkan performa yang stabil di seluruh rentang *threshold*.

IV. KESIMPULAN

Penelitian ini mengevaluasi pengaruh optimasi *hyperparameter* terhadap kinerja *Random Forest*, *XGBoost*, dan *LightGBM* dalam klasifikasi sentimen ulasan pengguna OVO. Hasil pengujian menunjukkan bahwa *Random Forest* memiliki kemampuan generalisasi paling baik, ditunjukkan oleh akurasi dan ROC-AUC Macro yang lebih stabil pada data uji dibandingkan dua model *boosting* yang meskipun unggul pada tahap *cross-validation*, mengalami penurunan performa pada data baru. Hasil ini menunjukkan bahwa model dengan performa pelatihan tertinggi tidak selalu memberikan generalisasi terbaik, serta bahwa *Random Forest* merupakan model yang paling sesuai untuk konteks dataset penelitian ini. Penelitian selanjutnya disarankan untuk mengeksplorasi embedding kontekstual seperti BERT, teknik penyeimbangan data yang lebih adaptif, serta perluasan dataset guna meningkatkan kemampuan model dalam memahami variasi bahasa informal pada ulasan aplikasi.

REFERENSI

[1] A. Lowell, A. Lowell, K. Candra, E. Indra, and S. Informasi, "Perbandingan Metode Support Vector Machine (SVM) Dan Naive Bayes Pada Analisis Sentimen Ulasan Aplikasi OVO," *Jurnal Media Informatika [Jumin]*, vol. 7, no. 1, pp. 125–133, 2025.

[2] S. A. Ghaffar and W. C. Setiawan, "Comparative Sentiment Analysis of Digital Wallet Applications in Indonesia Using Naïve Bayes," vol. 8, no. 2, pp. 55–66, 2025.

[3] M. J. Setiawan and V. R. S. Nastiti, "DANA App Sentiment Analysis: Comparison of XGBoost, SVM, and Extra Trees," *Jurnal Sisfokom (Sistem Informasi dan Komputer)*, vol. 13, no. 3, pp. 337–345, 2024, doi: 10.32736/sisfokom.v13i3.2239.

[4] M. Ramdan, A. Surya, and U. Hayati, "Analisis Sentimen Ulasan Pengguna Ovo Menggunakan Algoritma Naive Bayes Pada Google Play Store," *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 8, no. 3, pp. 2780–2786, 2024.

[5] A. B. Suleiman, E. D. Ajik, and S. A. Sule, "Leveraging Sentiment Analysis to Optimize Customer Experience in Digital Wallets: A Case of Opay Wallet," *Nigerian Journal of Physics*, vol. 33, no. 4, pp. 147–156, 2024, doi: 10.62292/njp.v33i4.2024.322.

[6] S. Masturoh, R. L. Pratiwi, M. R. R. Saelan, and U. Radiyah, "Application of the K-Nearest Neighbor (Knn) Algorithm in Sentiment Analysis of the Ovo E-Wallet Application," *JITK (Jurnal Ilmu Pengetahuan dan Teknologi Komputer)*, vol. 8, no. 2, pp. 78–83, 2023, doi: 10.33480/jitk.v8i2.3997.

[7] I. Kurniasari, A. A. Alfin, and E. Widodo, "Implementasi Long Short-Term Memory (LSTM) dan Word Embedding Model pada Analisis Sentimen Layanan Uang Elektronik Ovo dan Link Aja," *INFORMASI(Jurnal Informatika dan Sistem Informasi)*, vol. 15, no. 2, pp. 237–246, 2020.

[8] I. S. Widianto, Y. R. Ramadhan, Y. R. Ramadhan, M. A. Komara, and M. A. Komara, "Analisis Sentimen E-Wallet Gopay, Shopeepay, Dan Ovo

Menggunakan Algoritma Naive Bayes," *JITET(Jurnal Informatika dan Teknik Elektro Terapan)*, vol. 12, no. 3S1, 2024, doi: 10.23960/jitet.v12i3s1.5277.

[9] A. D. Rachmatsyah, T. Sugihartono, and K. Irfan, "Perbandingan Teknik Optimasi Grid Search dan Randomized Search dalam Meningkatkan Akurasi Metode Klasifikasi SVM Pada Sentimen Ulasan Pengguna Aplikasi JKN Mobile," vol. 8, pp. 13–22, 2025.

[10] A. Fauzi, A. H. Yunial, D. E. Saputro, and R. Saputra, "Optimalisasi Random Forest untuk Sentimen Bahasa Indonesia dengan GridSearch dan SMOTE," pp. 202–217, 2025.

[11] A. F. Zain, H. Al Azies, and I. K. Ananda, "Analisis Sentimen Ulasan Pengguna iPhone dengan Pendekatan Hibrida RoBERTa dan XGBoost," *Jurnal Algoritma*, vol. 22, no. 1, pp. 1039–1049, Jul. 2025, doi: 10.33364/ALGORITMA/V.22-1.2277.

[12] R. R. Putri and N. Cahyono, "Analisis Sentimen Komentar Masyarakat Terhadap Pelayanan Publik Pemerintah DKI Jakarta dengan Algoritma Super Vector Machine dan Naive Bayes," *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 8, no. 2, pp. 2363–2371, 2024.

[13] H. H. Mubaroroh, H. Yasin, and A. Rusgiyono, "Analisis sentimen data ulasan aplikasi ruangguru pada situs google play menggunakan algoritma naïve bayes classifier dengan normalisasi kata levenshtein distance," vol. 11, no. 1, pp. 257–266, 2022.

[14] Yusril, W. Fuadi, and Y. Afrillia, "Analisis sentimen review aplikasi stockbit di google play store dan x(twitter) menggunakan support vector machine," vol. 10, no. 2, pp. 1050–1062, 2025.

[15] J. Amalia, J. Pakpahan, M. Pakpahan, and Y. Panjaitan, "Model Klasifikasi Berita Palsu Menggunakan Bidirectional LSTM Dan Word2Vec Sebagai Vektorisasi," vol. 9, no. 4, pp. 3319–3331, 2022.

[16] P. Ayuningtyas and H. Tantyoko, "Perbandingan Metode Word2vec Model Skipgram pada Ulasan Aplikasi Linkaja menggunakan Algoritma Bidirectional LSTM dan Support Vector Machine Comparison of the Word2vec Skipgram Model Method Linkaja Application Review using Bidirectional LSTM Algorithm and Support Vector Machine," vol. 12, no. 1, pp. 189–196, 2024, doi: 10.26418/justin.v12i1.72530.

[17] S. K. Putri, A. Amalia, E. B. Nababan, and O. S. Sitompul, "Bahasa Indonesia pre-trained word vector generation using word2vec for computer and information technology field," pp. 0–10, doi: 10.1088/1742-6596/1898/1/012007.

[18] Badriyah, T. Chamidy, and Suhartono, "Application of SMOTE in Sentiment Analysis of MyXL User Reviews on Google Play Store," vol. 10, no. 1, pp. 74–86, 2025.

[19] R. Hornung and A. Boulesteix, "Interaction forests : Identifying and exploiting interpretable quantitative and qualitative interaction effects," *Computational Statistics and Data Analysis*, vol. 171, p. 107460, 2022, doi: 10.1016/j.csda.2022.107460.

[20] L. Zhang, X. Zhang, S. Gao, and X. Gu, "Revealing Nonlinear Relationships and Thresholds of Human Activities and Climate Change on Ecosystem Services in Anhui Province Based on the XGBoost – SHAP Model," pp. 1–22, 2025.

[21] L. Yang, Y. Peng, J. Chen, Y. Liu, and H. Yang, "Temporal variations in the non-linear relationships between metro ridership and the built environment : insights from interpretable machine learning using four-year data," 2025.

[22] J. Zhou, Y. Dai, M. Tao, M. Khandelwal, M. Zhao, and Q. Li, "Estimating the mean cutting force of conical picks using random forest with salp swarm algorithm," *Results in Engineering*, vol. 17, no. December 2022, p. 100892, 2023, doi: 10.1016/j.rineng.2023.100892.

[23] K. Afifah, I. N. Yulita, and I. Sarathan, "Sentiment Analysis on Telemedicine App Reviews using XGBoost Classifier," *2021 International Conference on Artificial Intelligence and Big Data Analytics*, pp. 22–27, 2021, doi: 10.1109/ICAIBDA53487.2021.9689735.

[24] D. D. Rufo, T. G. Debelee, A. Ibenthal, and W. G. Negera, "Diagnosis of Diabetes Mellitus Using Gradient Boosting Machine (LightGBM)," pp. 1–14, 2021.

[25] S. Mulyani and T. Arifin, "Prediksi Kelangsungan Hidup Pasien Gagal Jantung Menggunakan Pendekatan Machine Learning Dengan Optimasi Grid Searchcv," pp. 577–586, 2024.

[26] R. Reynaldi, I. Faisal, and K. Chiuloto, "Optimasi Hyperparameter Menggunakan RandomSearchCV dalam Upaya Peningkatan Akurasi Klasifikasi Pneumonia," vol. 1, no. 2, pp. 121–135, 2025.

[27] S. Ünalan, O. Günay, I. Akkurt, K. Gunoglu, and H. O. Tekin, "A comparative study on breast cancer classification with stratified shuffle split and K-fold cross validation via ensembled machine learning," *Journal of Radiation Research and Applied Sciences*, vol. 17, no. 4, p. 101080, 2024, doi: 10.1016/j.jrras.2024.101080.

[28] I. F. Rahman, H. Al Azies, and M. Akrom, "Deteksi Struktur Material Perovskit ABO3 Berbasis Machine Learning," *METIK JURNAL (AKREDITASI SINTA 3)*, vol. 9, no. 1, pp. 137–147, Jun. 2025, doi: 10.47002/METIK.V9I1.1036.