JNATIA Volume 4, Nomor 2, Februari 2026 p-ISSN: 2986-3929
Jurnal Nasional Teknologi Informasi dan Aplikasinya e-ISSN: 3032-1948

Analisis Sentimen Terhadap Ulasan Aplikasi Info BMKG Menggunakan *Logistic Regression* dan XGBoost

I Made Rovan Puja Wardana[a1], Gst. Ayu Vida Mastrika Giri[a2]

[a]Program Studi Informatika, Fakultas Matematika dan Ilmu Pengetahuan Alam
Universitas Udayana
Jln. Raya Kampus UNUD, Bukit Jimbaran, Kuta Selatan, Badung, Bali, Indonesia
[1]wardana.2308561011@student.unud.ac.id
[2]vida@unud.ac.id

Abstract

Public opinion through user reviews plays an important role in evaluating the quality of digital public services, especially in mobile-based government applications. This study aims to classify sentiment from 5,000 user reviews of the Info BMKG application using machine learning techniques. The text data underwent pre-processing, including lowercasing, cleaning, normalization, stopword removal, and stemming to improve data quality. Feature extraction was performed using Term Frequency–Inverse Document Frequency (TF-IDF), while class imbalance was handled through the application of the Synthetic Minority Oversampling Technique (SMOTE). Two classification models were tested: Logistic Regression and Extreme Gradient Boosting (XGBoost). Tuning of hyperparameters was conducted through Grid Search combined with five-fold cross-validation to find the optimal model configuration. Evaluation results show that Logistic Regression achieved the best performance with 86.4% accuracy and a weighted F1-score of 0.86, while XGBoost achieved 85.7% accuracy. Both models demonstrated consistent ability in detecting positive and negative sentiment. The findings suggest that Logistic Regression is a suitable and reliable approach for sentiment analysis in short-text reviews within digital public service applications.

Keywords: *Sentiment Analysis, Logistic Regression, XGBoost, TF-IDF, SMOTE*

1. Pendahuluan

Letak geografis Indonesia serta dampak perubahan iklim menjadikannya sebagai salah satu negara dengan tingkat kerentanan tinggi terhadap bencana alam. Kondisi ini menuntut upaya serius untuk meningkatkan kesiapsiagaan masyarakat dalam menghadapi kemungkinan bencana yang bisa terjadi kapan saja [1]. Mengacu pada data dari Data Informasi Bencana Indonesia (DIBI), tercatat sebanyak 19.861 kejadian bencana terjadi di Indonesia selama periode 2017 hingga 2022 [2]. Oleh karena itu, peningkatan kesadaran akan risiko dan dampak bencana menjadi sangat penting. Salah satu pendekatan yang efektif dalam membangun kesadaran publik adalah melalui edukasi dan diseminasi informasi menggunakan berbagai saluran media.

Badan Meteorologi, Klimatologi, dan Geofisika (BMKG) merupakan instansi resmi pemerintah Indonesia yang memiliki tugas dalam pengamatan, pengolahan, dan penyebaran informasi terkait fenomena meteorologi, klimatologi, dan geofisika. Sebagai bagian dari upaya meningkatkan aksesibilitas informasi kepada masyarakat, BMKG mengembangkan aplikasi digital resmi yang bernama “Info BMKG”. Aplikasi ini dirancang untuk menyampaikan informasi seperti perkiraan cuaca, gempa bumi, kualitas udara, dan peringatan dini bencana secara cepat dan akurat [3].

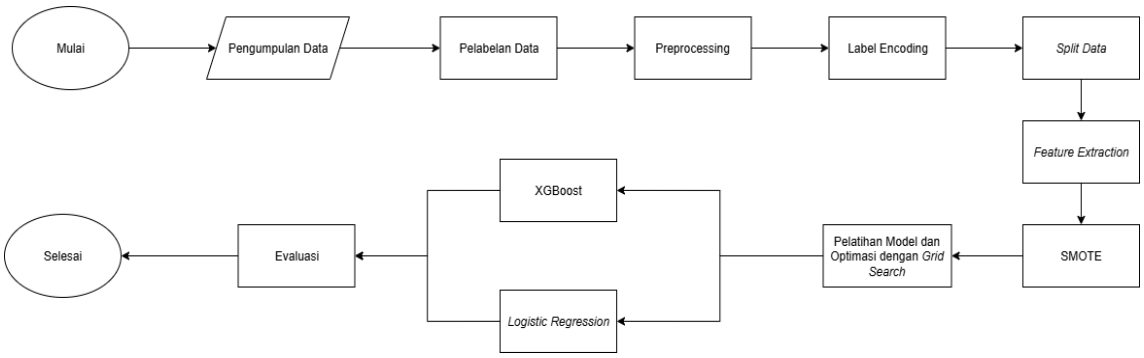
Seiring dengan meningkatnya penggunaan aplikasi di kalangan masyarakat, ulasan dan penilaian pengguna yang tersedia di platform seperti Google Play Store menjadi indikator penting dalam mengevaluasi kepuasan serta efektivitas aplikasi tersebut. Ulasan ini seringkali mencerminkan pengalaman, persepsi, dan ekspektasi pengguna terhadap layanan yang disediakan [4]. Untuk memahami opini publik yang terkandung dalam ulasan tersebut secara objektif, analisis sentimen menjadi metode yang tepat untuk mengidentifikasi kecenderungan

sikap atau perasaan pengguna terhadap suatu aplikasi. Melalui analisis sentimen, dapat diperoleh wawasan yang lebih mendalam mengenai aspek layanan yang perlu ditingkatkan, dipertahankan, atau disesuaikan dengan kebutuhan pengguna.

Sejumlah penelitian terdahulu telah menggunakan berbagai algoritma *machine learning* dalam analisis sentimen. Penelitian pada [5] membandingkan kinerja algoritma Naïve Bayes dan Logistic Regression dalam melakukan klasifikasi sentimen dan mendapatkan hasil akurasi algoritma *Logistic Regression* lebih tinggi dibandingkan *Naïve Bayes* secara berurutan 95% dan 91%, dimana hasil ini didapatkan setelah SMOTE digunakan untuk menangani masalah distribusi data yang tidak seimbang. Adapun pada penelitian [6] menggunakan algoritma XGBoost pada analisis sentimen PPKM di media sosial twitter dan menunjukkan hasil akurasi 85,27%. Berdasarkan temuan tersebut, kedua algoritma ini dipilih untuk dibandingkan dalam penelitian ini guna menentukan metode yang sesuai untuk menganalisis sentimen ulasan aplikasi Info BMKG.

2. Metode Penelitian

2.1. Alur Penelitian



Gambar 1. Alur Penelitian

Pada penelitian ini, dimulai dengan mengumpulkan data ulasan aplikasi Info BMKG. Data ulasan yang didapat kemudian diberi label secara manual untuk menentukan kategori sentimen antara *positive* atau *negative*. Data yang telah diberi label akan melalui tahap *pre-processing* untuk dibersihkan datanya agar siap digunakan dalam analisis. Setelah itu dilakukan label encoding untuk mengubah kategori sentimen menjadi format numerik. Langkah berikutnya adalah membagi data menjadi data latih (*train*) dan data uji (*test*), kemudian mengekstrak fitur dengan metode TF-IDF untuk mendapatkan bobot dari setiap kata. Teknik SMOTE diterapkan untuk menangani ketidakseimbangan jumlah data antar kelas sentimen. Dengan data yang sudah seimbang, dilakukan penelitian menggunakan algoritma *Logistic Regression* dan XGBoost. Evaluasi model dilakukan menggunakan metrik *Accuracy, Precision, Recall, F1-Score*, dan *Confusion Matrix* untuk mengukur performa klasifikasi sentimen.

2.2. Pengumpulan Data

Data yang digunakan dalam penelitian ini merupakan data sekunder yang diperoleh dari platform Kaggle, yang berupa ulasan pengguna terhadap aplikasi Info BMKG. Dari dataset tersebut, sebanyak 5000 data ulasan yang digunakan pada penelitian ini. Setiap ulasan diberi label secara manual dan diklasifikasikan ke dalam dua kategori sentimen yaitu *positive* dan *negative*, serta proses pelabelan dilakukan menggunakan Label Studio.

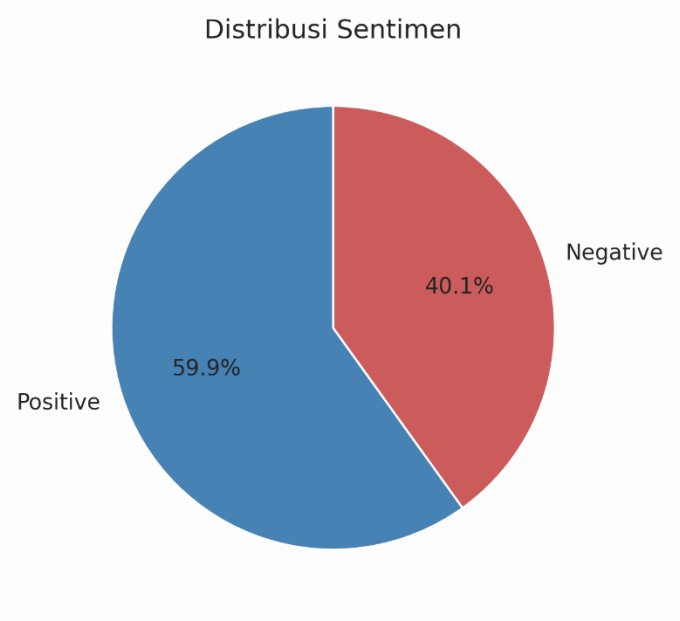
Table 1. Contoh Data Ulasan

Content	Score	Sentiment
Overall aplikasinya sudah bagus, tetapi tolong dong min untuk notifikasi gempa di munculkan saat gempa yang letak lokasinya dekat dengan lokasi	2	Negative

p-ISSN: 2986-3929
e-ISSN: 3032-1948

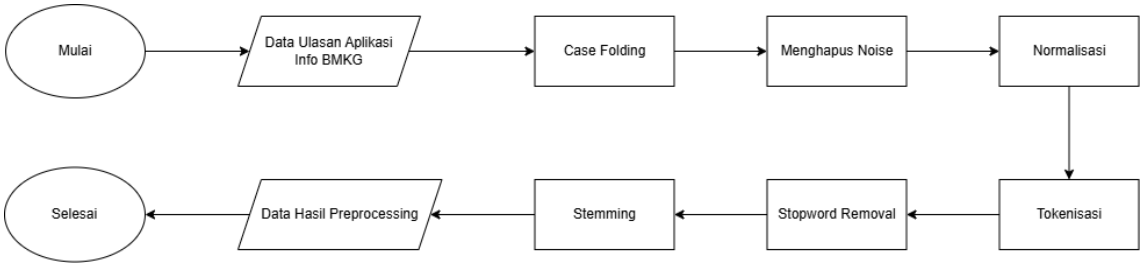
Content	Score	Sentiment
user jangan munculkan setiap ada gempa yang jarak lokasinya jauh dari user. rada risih soalnya.		
Tampilan informasi sudah cukup bagus. Saran: bagian dashboard dibuat untuk user bisa save beberapa lokasi prakiraan cuaca dan dapat ditampilkan secara bersamaan. Terimakasih	5	Positive

Distribusi sentimen dalam dataset menunjukkan bahwa ulasan *positive* memiliki proporsi lebih besar dibandingkan ulasan *negative*. Secara keseluruhan, sebanyak 59,9% ulasan bersifat *positive*, sementara 40,1% bersifat *negative* sebagaimana ditunjukkan pada Gambar 2.



Gambar 2. Distribusi Sentimen pada Dataset

2.3. *Pre-processing*



Gambar 3. Alur Data *Pre-processing*

Data *pre-processing* dilakukan dengan melalui beberapa tahapan untuk membersihkan dan menyiapkan teks ulasan agar siap dianalisis. Dimulai dengan *case folding*, dengan mengubah seluruh huruf menjadi huruf kecil. Selanjutnya, dilakukan pembersihan teks untuk menghapus *mention*, hashtag, URL, emoji, angka, tanda baca, serta karakter yang tidak relevan. Setelah itu, tahap normalisasi dilakukan dengan mengganti kata tidak baku menjadi bentuk bakunya berdasarkan kamus normalisasi yang telah dibuat. Proses dilanjutkan dengan tokenisasi, yaitu memecah teks menjadi daftar kata (token), diikuti oleh *stopword removal*, yang menggunakan kombinasi daftar *stopword* Bahasa Indonesia standar dan *stopword* kustom yang disesuaikan dengan konteks data. Terakhir, dilakukan *stemming* menggunakan stemmer Bahasa Indonesia

untuk mengubah setiap token menjadi bentuk dasarnya. Hasil akhir dari semua tahapan ini disimpan dalam file CSV sebagai data siap olah.

2.4. Label Encoding

Label encoding merupakan metode yang digunakan untuk mengubah data kategorikal berbentuk teks menjadi representasi dalam bentuk angka, dengan cara memberikan angka unik untuk setiap kategori [7]. Metode ini berguna untuk mengonversi label seperti *positive* dan *negative* menjadi representasi numerik yang dapat dikenali oleh algoritma. Pada penelitian ini, *label encoding* diterapkan pada kolom *sentiment* dengan tujuan untuk mengubah nilai *positive* dan *negative* menjadi numerik, sehingga data dapat diproses oleh algoritma *Logistic Regression* dan XGBoost yang memerlukan target dalam format numerik.

2.5. Feature Extraction

Pada penelitian ini menggunakan *feature extraction* TF-IDF (*Term Frequency–Inverse Document Frequency*). TF-IDF merupakan metode yang mengubah kata-kata dalam teks menjadi representasi numerik berbentuk vector, dengan tujuan memberikan bobot pada setiap kata berdasarkan tingkat kepentingannya dalam suatu dokumen [8]. Bobot tersebut menunjukkan tingkat kepentingan sebuah kata dalam satu dokumen dibandingkan dengan semua dokumen dalam dataset. Perhitungan TF-IDF terdiri dari dua komponen utama, yaitu TF (*Term Frequency*) yang menghitung seberapa sering sebuah kata muncul dalam satu dokumen, dengan rumus:

$$tf_{t,d} = \frac{n_{t,d}}{Total\ number\ of\ term\ in\ document} \quad (1)$$

Dan IDF (*Inverse Document Frequency*) yang mengukur seberapa jarang atau umum suatu kata dalam keseluruhan dokumen, dirumuskan sebagai :

$$idf_t = \log\left(\frac{Number\ of\ document}{Total\ number\ of\ term\ in\ document}\right) \quad (2)$$

Kedua nilai ini kemudian dikalikan untuk mendapatkan skor akhir TDF-IDF dari suatu kata terhadap dokumen:

$$tfidf_{t,d} = tf_{t,d} \times idf_t \quad (3)$$

Dengan demikian, TF-IDF memberikan nilai bobot yang lebih tinggi bagi kata-kata yang sering muncul dalam suatu dokumen tetapi jarang muncul di dokumen lain, sehingga kata tersebut dianggap lebih representatif atau penting dalam konteks dokumen tersebut.

2.6. SMOTE

SMOTE (*Synthetic Minority Oversampling Technique*) merupakan metode *resampling* yang digunakan untuk menyeimbangkan distribusi kelas dalam dataset dengan cara menghasilkan data sintetis dari kelas minoritas. Teknik ini menambahkan contoh baru tanpa mengurangi data yang sudah ada, sehingga tidak menyebabkan kehilangan informasi. Dengan memperbesar representasi kelas minoritas, SMOTE membantu model klasifikasi untuk lebih memahami pola pada kelas tersebut dan berpotensi meningkatkan akurasi prediksi pada kelas minoritas [9].

2.7. Grid Search

Grid Search merupakan tahapan yang digunakan untuk menentukan kombinasi *hyperparameter* terbaik dalam suatu model *machine learning*. Teknik ini bekerja dengan mengevaluasi berbagai kemungkinan nilai parameter yang telah ditentukan sebelumnya, kemudian menguji setiap kombinasi secara menyeluruh. Seluruh konfigurasi yang terbentuk akan dinilai berdasarkan performa model menggunakan metrik evaluasi tertentu, biasanya melalui proses validasi silang (*cross-validation*). Dengan cara ini, *Grid Search* membantu menemukan pengaturan parameter

yang paling optimal untuk meningkatkan kinerja model. Pendekatan ini sering diterapkan dalam proses tuning pada berbagai algoritma pembelajaran mesin [10].

2.8. Logistic Regression

Logistic Regression merupakan metode statistik yang digunakan untuk menganalisis hubungan antara satu atau lebih variabel bebas dengan variabel target kategorikal. Teknik ini umum digunakan dalam tugas klasifikasi biner, seperti menentukan apakah sebuah kalimat bernilai *positive* atau *negative* [5]. Logistic Regression menggunakan fungsi sigmoid untuk menghasilkan *output* dalam rentang 0 hingga 1, yang kemudian dikategorikan menjadi dua kelas. Persamaan fungsi sigmoid dalam model ini dituliskan sebagai berikut:

$$P(Y) = \frac{1}{1+e^{-(b_0+b_1x_1+b_2x_2+\cdots+b_kx_k)}} \quad (4)$$

Rumus tersebut menunjukkan bahwa probabilitas keluaran P(Y) berada pada skala biner, yaitu hanya bernilai mendekati 0 atau 1. Pelatihan model dilakukan dengan tujuan menemukan nilai koefisien terbaik yang mampu meminimalkan selisih antara probabilitas prediksi model dan nilai aktual pada data latih. Setelah proses pelatihan selesai, model dapat digunakan untuk memperkirakan probabilitas suatu hasil berdasarkan nilai-nilai prediktor dari data yang belum pernah dilihat sebelumnya.

2.9. XGBoost

XGBoost merupakan salah satu bentuk penerapan *ensemble learning* yang mengimplementasikan metode *gradient boosting* secara efisien dan cepat, terutama saat digunakan pada dataset berskala besar dan kompleks [11]. Proses pelatihannya dilakukan secara bertahap, di mana setiap pohon keputusan yang dibangun bertujuan untuk memperbaiki kesalahan dari iterasi sebelumnya. Model ini dioptimalkan dengan meminimalkan fungsi loss melalui pendekatan gradien. Untuk klasifikasi biner, XGBoost menggunakan *binary logistic loss* yang dirumuskan sebagai berikut:

$$L = -\sum_{i=1}^{n}\left(y_i log(p_i) + (1-y_i) log(1-p_i)\right) \quad (5)$$

Sementara itu, prediksi akhir untuk setiap sampel dihitung sebagai penjumlahan dari output semua pohon yang terbentuk hingga iterasi ke-K, sebagaimana ditunjukkan dalam persamaan:

$$y_i = \sum_{k=1}^{K} f_k(x_i) \quad (6)$$

2.10. Confusion Matrix

Confusion matrix merupakan alat evaluasi penting dalam machine learning untuk menilai kinerja model klasifikasi, khususnya dalam kasus dua kelas. Matriks ini menunjukkan jumlah prediksi yang benar dan salah, yang terbagi menjadi empat komponen utama: *true positive* (TP), *false positive* (FP), *false negative* (FN), dan *true negative* (TN). Dari matriks ini dapat dihitung beberapa metrik evaluasi, seperti akurasi yang mengukur proporsi prediksi benar terhadap total prediksi, *precision* yang menunjukkan ketepatan prediksi *positive*, *recall* yang menggambarkan kemampuan model dalam mendeteksi data *positive* secara benar, serta *F1-score* yang merupakan rata-rata harmonis dari *precision* dan *recall*. Rumus perhitungan masing-masing metrik adalah sebagai berikut:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (7)$$

$$Precision = \frac{TP}{TP+FP} \quad (8)$$

$$Recall = \frac{TP}{TP+FN} \quad (9)$$

$$F1 - Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (10)$$

3. Hasil dan Diskusi

3.1. *Pre-processing* Data

Sebanyak 5000 data ulasan pengguna diproses menggunakan Google Colab dengan bahasa pemrograman Python untuk melalui tahapan *data pre-processing*. Proses dimulai dengan *case folding* untuk menyeragamkan huruf menjadi kecil, lalu pembersihan teks dari karakter tidak relevan seperti URL, emoji, angka, dan tanda baca. Selanjutnya dilakukan normalisasi kata, tokenisasi, penghapusan *stopword* dengan daftar Bahasa Indonesia dan kustom, serta *stemming* untuk mengubah kata menjadi bentuk dasarnya. Contoh hasil *pre-processing* dapat dilihat pada Tabel 2. Setelah tahap *pre-processing* selesai, data kemudian dibagi menjadi dua bagian, yaitu 80% untuk data latih (*training*) dan 20% untuk data uji (*testing*), yang digunakan dalam proses pelatihan dan evaluasi model.

Tabel 2. Contoh Data *Pre-processing*

	Content	**Sentiment**
Data Sebelum *Pre-processing*	Aplikasinya mengalami peningkatan pesat, baik dari tampilan maupun fiturnya	Positive
Case Folding	aplikasinya mengalami peningkatan pesat, baik dari tampilan maupun fiturnya	Positive
Menghapus Noise	aplikasinya mengalami peningkatan pesat baik dari tampilan maupun fiturnya	Positive
Normalisasi	aplikasinya mengalami peningkatan pesat baik dari tampilan maupun fiturnya	Positive
Tokenisasi	['aplikasinya', 'mengalami', 'peningkatan', 'pesat', 'baik', 'dari', 'tampilan', 'maupun', 'fiturnya']	Positive
Stopword Removal	['aplikasinya', 'mengalami', 'peningkatan', 'pesat', 'tampilan', 'fiturnya']	Positive
Stemming	['aplikasi', 'alami', 'tingkat', 'pesat', 'tampil', 'fiturnya']	Positive

3.2. Feature Extraction dengan TF-IDF

Setelah data melewati tahap *pre-processing*, langkah berikutnya adalah melakukan ekstraksi fitur menggunakan metode TF-IDF (*Term Frequency–Inverse Document Frequency*). Metode ini mengubah data teks menjadi representasi numerik dengan mempertimbangkan tidak hanya frekuensi kemunculan kata, tetapi juga seberapa penting kata tersebut dalam keseluruhan dokumen. Proses ini dilakukan menggunakan TfidfVectorizer dari library *scikit-learn*, yang membangun vektor berdasarkan bobot masing-masing kata. Semakin sering kata muncul di satu dokumen dan semakin jarang di dokumen lain, maka bobotnya akan semakin tinggi. Berdasarkan Gambar 4, didapatkan hasil matriks TF-IDF dengan bentuk (3997, 3504), yang menunjukkan bahwa terdapat 3997 data ulasan dan 3504 kata unik yang digunakan sebagai fitur. Representasi ini selanjutnya digunakan sebagai input dalam pelatihan model klasifikasi.

p-ISSN: 2986-3929
e-ISSN: 3032-1948

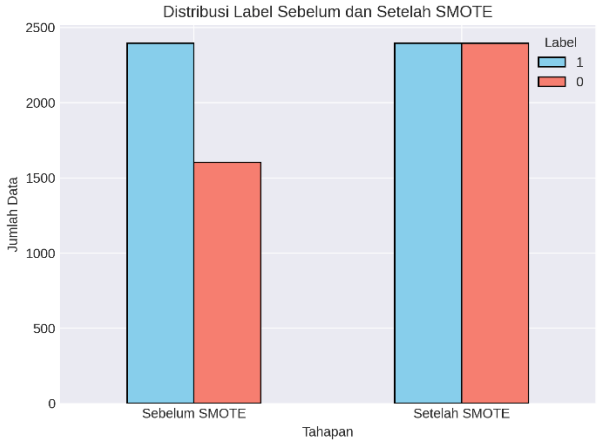
```
tfidf = TfidfVectorizer()
X_train_tfidf = tfidf.fit_transform(X_train)
X_test_tfidf = tfidf.transform(X_test)
print(f"TF-IDF features extracted. Shape: {X_train_tfidf.shape}")
```

```
TF-IDF features extracted. Shape: (3997, 3504)
```

Gambar 4. Implementasi TF-IDF

3.3. SMOTE

Distribusi label pada data mengalami ketidakseimbangan sebelum dilakukan proses SMOTE, di mana kelas *positive* (1) berjumlah 2395 dan kelas *negative* (0) hanya 1602. Setelah diterapkan metode SMOTE, jumlah data pada kelas minoritas ditingkatkan sehingga kedua kelas memiliki jumlah yang seimbang, yaitu masing-masing 2395 data. Diagram pada Gambar 5 memperlihatkan bagaimana SMOTE menyamakan distribusi kelas untuk membantu proses pelatihan model.



Gambar 5. Distribusi Data Sebelum dan Sesudah SMOTE

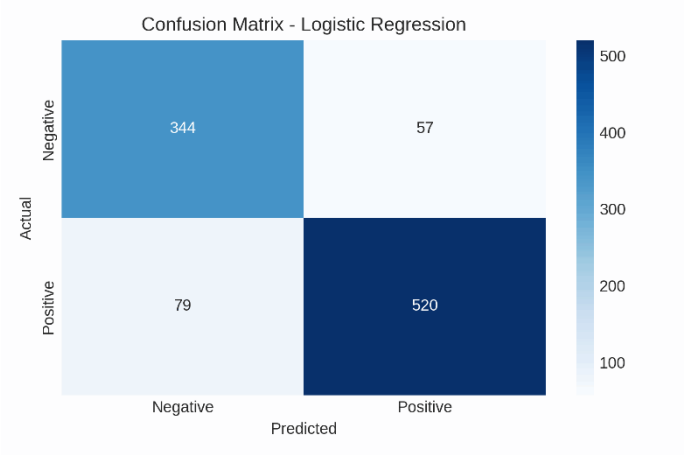
3.4. Pelatihan Model

Pada tahap pelatihan model, dilakukan optimasi *hyperparameter* menggunakan teknik *Grid Search Cross Validation* untuk mendapatkan kombinasi parameter terbaik pada setiap algoritma. Dataset dibagi dengan rasio 80:20, menghasilkan 3997 data untuk *training* dan 1000 data untuk *testing*. Proses ini menggunakan data *training* yang telah melalui tahap *pre-processing* dan *resampling* menggunakan SMOTE untuk mengatasi ketidakseimbangan kelas. *Grid Search* CV dilakukan dengan *5-fold cross validation* dan menggunakan metrik akurasi sebagai *scoring parameter* untuk menentukan kombinasi *hyperparameter* yang optimal. Setelah parameter terbaik diperoleh, model diuji pada data *testing* untuk melihat seberapa baik performanya dalam melakukan prediksi pada data yang belum digunakan saat pelatihan.

a. Logistic Regression

Algoritma *Logistic Regression* dipilih sebagai model *baseline* karena sederhana, efektif untuk klasifikasi teks, dan mudah diinterpretasikan. Proses pelatihan dilakukan dengan pencarian *hyperparameter* menggunakan *Grid Search*, di mana parameter yang diuji meliputi nilai regularisasi C [0.01, 0.1, 1, 10, 100], penalti l2, dan pilihan solver ‘liblinear’ serta ‘saga’. Hasil pencarian menunjukkan bahwa kombinasi parameter terbaik adalah C = 10, penalty = 'l2', dan solver = 'saga', yang memberikan akurasi sebesar 86,4% pada data uji.

p-ISSN: 2986-3929
e-ISSN: 3032-1948

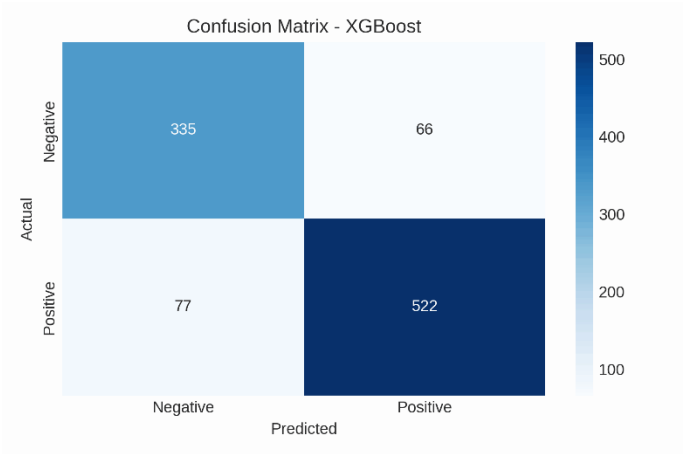


Gambar 6. Confusion Matrix Logistic Regression

Model ini menunjukkan performa yang cukup seimbang antara kedua kelas, dengan *precision* untuk kelas *negative* sebesar 0,81 dan *positive* 0,90, serta *recall* masing-masing 0,86 dan 0,87. Nilai *f1-score* mencapai 0,86 yang menunjukkan keseimbangan antara presisi dan sensitivitas model terhadap data uji. Berdasarkan *confusion matrix*, model berhasil memprediksi dengan benar 344 data kelas *negative* dan 520 data kelas *positive*, sementara terdapat kesalahan prediksi sebanyak 57 data *negative* yang diprediksi sebagai *positive* dan 79 data *positive* yang diprediksi sebagai *negative*. Hasil ini mengindikasikan bahwa model sedikit lebih condong dalam memprediksi kelas *positive*, tetapi secara keseluruhan tetap menunjukkan performa klasifikasi yang baik dan seimbang.

b. XGBoost

XGBoost (*Extreme Gradient Boosting*) diterapkan sebagai salah satu algoritma klasifikasi yang memanfaatkan teknik boosting untuk membentuk prediksi yang lebih akurat dengan menggabungkan sejumlah *decision tree*. Proses pelatihan model dilakukan dengan optimasi *hyperparameter* menggunakan *Grid Search*, dengan parameter yang diuji meliputi n_estimators [100, 200], max_depth [3, 5, 7], learning_rate [0.01, 0.1, 0.2], dan subsample [0.8, 1.0]. Kombinasi parameter terbaik yang diperoleh adalah learning_rate = 0.1, max_depth = 7, n_estimators = 200, dan subsample = 1.0, yang menghasilkan akurasi sebesar 85,7% pada data uji.



Gambar 7. Confusion Matrix XGBoost

Hasil evaluasi menunjukkan *precision* sebesar 0,81 untuk kelas *negative* dan 0,89 untuk kelas *positive*, dengan *recall* masing-masing 0,84 dan 0,87, menandakan bahwa model mampu mengidentifikasi kedua kelas dengan cukup baik, serta nilai *f1-score* berada pada nilai 0,86. Berdasarkan *confusion matrix*, terdapat 335 prediksi benar untuk kelas *negative* dan 522 prediksi benar untuk kelas *positive*, serta kesalahan prediksi sebanyak 66 data *negative* dan 77 data *positives*. Distribusi *error* yang seimbang ini mencerminkan bahwa model bekerja dalam membedakan kedua kelas sentimen.

4. Kesimpulan

Hasil dari penelitian ini menunjukkan bahwa algoritma *Logistic Regression* mampu memberikan performa paling optimal dalam tugas klasifikasi sentimen terhadap ulasan pengguna aplikasi Info BMKG. Model ini berhasil mencapai akurasi sebesar 86,4% dan *f1-score* sebesar 0,86, menunjukkan kemampuan dalam mengenali kedua kelas sentimen secara seimbang. Sementara itu, model XGBoost memberikan hasil yang berbeda tipis dengan akurasi 85,7%, namun sedikit tertinggal dalam hal stabilitas prediksi pada data uji. Evaluasi melalui *confusion matrix* juga memperkuat bahwa *Logistic Regression* cenderung lebih akurat dalam membedakan sentimen *positive* dan *negative*. Selain itu, penerapan metode TF-IDF sebagai representasi fitur dan SMOTE untuk penyeimbangan kelas menjadi faktor yang berkontribusi pada peningkatan performa model. Dengan demikian, dapat disimpulkan bahwa *Logistic Regression* merupakan pendekatan yang efektif dan andal untuk diterapkan dalam analisis sentimen pada aplikasi layanan publik, khususnya dengan data teks pendek seperti ulasan pengguna. Pendekatan ini memiliki ruang untuk pengembangan penelitian lebih lanjut, termasuk penggunaan *deep learning* atau penggunaan jenis data lain.

Daftar Pustaka

[1] A. P. Meidina and A. Murfi, “Environmental communication and disaster mitigation by mobile application and website,” *Jurnal Kajian Komunikasi*, vol. 12, no. 1, pp. 60–79, Jun. 2024, doi: 10.24198/jkk. v12i1.53996.

[2] I. Nabilla Audy, T. Nur Padilah, and B. Nurina Sari, “Pengelompokan Daerah Rawan Bencana Alam Di Jawa Barat Menggunakan Algoritma Fuzzy C-Means,” *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 7, no. 4, pp. 2799–2803, Jan. 2024, doi: 10.36040/jati.v7i4.7205.

[3] Benny Hartanto, Ningrum Astriawati, Supartini, and Damar Kuncoro Yekti, “Pencarian dan Pemanfaatan Informasi Data Badan Meteorologi, Klimatologi, dan Geofisika (BMKG),” *INSOLOGI: Jurnal Sains dan Teknologi*, vol. 1, no. 5, pp. 553–564, Oct. 2022, doi: 10.55123/insologi.v1i5.906.

[4] M. Iqrom, M. Afdal, R. Novita, M. Rahmawita, and T. Khairil Ahsyar, “Sentiment Analysis Of Gojek, Grab, And Maxim Applications Using Support Vector Machine Algorithm,” *Jurnal Inovtek Polbeng*, vol. 10, no. 1, p. 2025, Mar. 2025.

[5] B. Ramadhani and R. R. Suryono, “Komparasi Algoritma Naïve Bayes dan Logistic Regression Untuk Analisis Sentimen Metaverse,” *Jurnal Media Informatika Budidarma*, vol. 8, no. 2, p. 714, Apr. 2024, doi: 10.30865/mib.v8i2.7458.

[6] I. P. A. P. Widiarta, R. Dwiyansaputra, and A. Aranta, “Analisis Sentimen Masyarakat Terhadap Kebijakan Penerapan Ppkm Di Media Sosial Twitter Dengan Menggunakan Metode Xgboost,” *Jurnal Teknologi Informasi, Komputer, dan Aplikasinya (JTIKA)*, vol. 5, no. 2, pp. 154–163, Sep. 2023, doi: 10.29303/jtika.v5i2.342.

[7] C. Herdian, A. Kamila, and I. G. Agung Musa Budidarma, “Studi Kasus Feature Engineering Untuk Data Teks: Perbandingan Label Encoding dan One-Hot Encoding Pada Metode Linear Regresi,” *Technologia : Jurnal Ilmiah*, vol. 15, no. 1, p. 93, Jan. 2024, doi: 10.31602/tji.v15i1.13457.

[8] K. Tri Putra, M. Amin Hariyadi, and C. Crysdian, “Perbandingan Feature Extraction Tf-Idf Dan Bow Untuk Analisis Sentimen Berbasis SVM,” *Jurnal Cahaya Mandalika*, no. 2, pp. 1449–1463, Nov. 2023.

[9] K. Pramayasa, I. M. D. Maysanjaya, and I. G. A. A. D. Indradewi, “Analisis Sentimen Program Mbkm Pada Media Sosial Twitter Menggunakan KNN Dan SMOTE,” *SINTECH*

(Science and Information Technology) Journal, vol. 6, no. 2, pp. 89–98, Aug. 2023, doi: 10.31598/sintechjournal. v6i2.1372.

[10] M. Bayu Prayoga, N. Cahyono, S. Subektiningsih, and K. Kamarudin, "Penerapan Grid Search Untuk Optimasi Model Machine Learning Dalam Klasifikasi Sentimen Komentar Youtube," *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 9, no. 3, pp. 3817–3824, May 2025, doi: 10.36040/jati.v9i3.13375.

[11] L. G. A. Putri, S. A. Wicaksono, and B. Rahayudi, "Analisis Klasifikasi Spam Email Menggunakan Metode Extreme Gradient Boosting (XGBoost)," *2022 International Conference on Computer Communication and Informatics (ICCCI)*, vol. 9, no. 2, Feb. 2025, Accessed: Jul. 03, 2025. [Online]. Available: https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/14475