Klasifikasi Sentimen pada Dataset Ulasan Film menggunakan *Machine Learning* dan OpenAI *Text Embedding*

Azzam Abdurrahman[1], Moch. Arif Bijaksana[2], Kemas Muslim Lhaksmana[3]

[1,2,3]Fakultas Informatika, Universitas Telkom, Bandung
[1]azzamabdurrahman@student.telkomuniversity.ac.id,
[2]arifbijaksana@telkomuniversity.ac.id, [3]kemasmuslim@telkomuniversity.ac.id

Abstrak

Analisis sentimen pada ulasan film menjadi semakin penting seiring dengan meningkatnya volume data tekstual. Performa model *machine learning* untuk tugas ini sangat bergantung pada kualitas representasi teks yang digunakan. Penelitian ini bertujuan untuk mengevaluasi efektivitas model *embedding* teks kontekstual dari OpenAI, Text-embedding-3-large, untuk klasifikasi sentimen pada dataset *Movie Reviews*. Metodologi penelitian mencakup dua pendekatan klasifikasi: *supervised learning* menggunakan *Support Vector Machine* dan *Logistic Regression*, serta klasifikasi *zero-shot*. Performa Text-embedding-3-large dibandingkan secara langsung dengan model *embedding* statis Word2Vec pada dataset yang telah dibersihkan dan dataset asli. Hasil penelitian menunjukkan bahwa Text-embedding-3-large secara signifikan mengungguli Word2Vec, dengan peningkatan *F1-score* dari 78.01% menjadi 93.20%. Konfigurasi terbaik dicapai oleh kombinasi *Support Vector Machine* dengan *hyperparameter default* pada dataset yang tidak dibersihkan, yang mengindikasikan kemampuan model memanfaatkan informasi kontekstual dari tanda baca. Selain itu, pendekatan *zero-shot* menunjukkan kinerja yang cukup baik dengan *F1-score* 86.29%, yang membuktikan kapabilitas generalisasi model tanpa memerlukan data latih berlabel.

Kata kunci : klasifikasi sentimen, ulasan film, *machine learning*, openai, *embedding* teks, zero-shot.

Abstract

Sentiment analysis on film reviews is becoming increasingly important as the volume of textual data grows. The performance of machine learning models for this task is highly dependent on the quality of the text representation used. This research aims to evaluate the effectiveness of OpenAI's contextual text embedding model, Text-embedding-3-large, for sentiment classification on the Movie Reviews dataset. The research methodology includes two classification approaches: supervised learning using Support Vector Machine and Logistic Regression, as well as zero-shot classification. The performance of Text-embedding-3-large is directly compared with the Word2Vec static embedding model on both cleaned and uncleaned dataset. The results show that Text-embedding-3-large significantly outperforms Word2Vec, with an F1-score increase from 78.01% to 93.20%. The best configuration is achieved by the combination of Support Vector Machine with default hyperparameter on the uncleaned dataset, which indicates the model's ability to utilize contextual information from punctuation. Furthermore, the zero-shot approach shows quite good performance with an F1-score of 86.29%, which proves the model's generalization capabilities without requiring labeled training data.

Keywords: sentiment classification, movie reviews, machine learning, openai, text embedding, zero-shot

1. Pendahuluan

Latar Belakang

Kemajuan teknologi informasi dan komunikasi telah membawa perubahan besar dalam cara manusia berinteraksi serta mengakses informasi. Berbagai platform memungkinkan pengguna untuk mengekspresikan pendapat, pengalaman, dan memberikan ulasan mengenai berbagai topik. Di bidang perfilman, ulasan dan pendapat penonton berperan penting dalam mengukur respons masyarakat terhadap sebuah film, yang dapat membantu untuk memahami persepsi publik.

Dengan analisis sentimen, teks ulasan dapat diklasifikasikan ke dalam kategori polaritas seperti positif, negatif, atau netral [1] dengan memanfaatkan *machine learning* (ML). Performa model ML sangat bergantung pada kualitas representasi teks yang digunakan.

Meskipun kinerja model *deep learning* (DL) yang dapat mencapai hasil *state-of-the-art* dalam penelitian analisis sentimen [2], model ML masih relevan, terutama dalam skenario ketika sumber daya komputasi dan dataset yang terbatas [3, 4]. Selain itu, kinerja model ML dapat ditingkatkan secara signifikan dengan penggunaan teknik representasi teks yang efektif [5].

Salah satu teknik representasi teks adalah dengan *embedding* statis, seperti Word2Vec, di mana setiap kata

dipetakan ke satu vektor numerik, tetapi tanpa memperhitungkan konteks kalimatnya. Keterbatasan ini mendorong pengembangan *embedding* kontekstual, salah satunya adalah Text-embedding-3-large dari OpenAI [6]. Keunggulannya terletak pada kemampuannya menangkap nuansa semantik yang kompleks dan hubungan kontekstual dalam teks, yang penting untuk tugas analisis sentimen [7], dan terbukti unggul dalam berbagai tugas *Natural Language Processing* (NLP) [8].

Kecanggihan *embedding* Text-embedding-3-large ini mendukung dua pendekatan klasifikasi. Pertama, pendekatan *supervised learning*, di mana model ML tradisional seperti *Support Vector Machine* (SVM) dan *Logistic Regression* (LR) dilatih menggunakan data berlabel. Model-model ini dipilih karena kesederhanaan dan efisiensi komputasinya [9]. Kedua, pendekatan klasifikasi *zero-shot*.

Penelitian ini akan mengeksplorasi penggunaan model Text-embedding-3-large untuk klasifikasi sentimen, baik secara *supervised learning* dan juga melalui klasifikasi *zero-shot* pada dataset ulasan film *Movie Reviews* (MR) [10]. Meskipun sudah ada sejumlah penelitian yang menguji berbagai model dan *embedding* teks pada dataset MR [11, 12, 13, 14, 15], masih terdapat kesenjangan penelitian terkait bagaimana performa model *embedding* teks kontekstual dari OpenAI pada dataset tersebut ketika diintegrasikan dengan beragam pendekatan klasifikasi.

Topik dan Batasannya

Penelitian ini akan mengevaluasi bagaimana performa model *embedding* kontekstual Text-embedding-3-large untuk klasifikasi sentimen secara *supervised learning*. Selanjutnya, diukur sejauh mana peningkatan performa jika dibandingkan dengan *embedding* statis Word2Vec. Penelitian ini juga akan mengidentifikasi model ML yang paling sesuai untuk mengklasifikasikan sentimen menggunakan model Text-embedding-3-large. Terakhir, penelitian ini akan menguji bagaimana performa model Text-embedding-3-large dengan pendekatan klasifikasi *zero-shot*.

Untuk memastikan penelitian dapat diselesaikan dalam lingkup waktu yang tersedia, serta dengan mempertimbangkan sumber daya yang ada, maka ditetapkan beberapa batasan masalah. Pertama, penelitian ini hanya menggunakan dataset MR yang berbahasa Inggris.

Kedua, sebagai pembanding, penelitian ini menggunakan metode *embedding* statis Word2Vec. Model *embedding* kontekstual lain seperti BERT (*Bidirectional Encoder Representations from Transformers*) tidak diikutsertakan dalam perbandingan.

Dan yang terakhir, pendekatan *supervised learning* dalam penelitian ini terbatas pada penggunaan algoritma ML tradisional. Penelitian ini tidak melibatkan implementasi arsitektur model DL.

Tujuan

Penelitian ini bertujuan untuk memberikan informasi mengenai bagaimana model *embedding* Text-embedding-3-large dapat diimplementasikan pada kedua pendekatan klasifikasi untuk klasifikasi sentimen, dan seberapa besar peningkatan kinerja yang dapat dicapai.

Organisasi Tulisan

Setelah Pendahuluan, jurnal dilanjutkan dengan Bab 2 yang membahas landasan teori dan tinjauan pustaka. Bab 3 akan menjelaskan metodologi yang diimplementasikan. Bab 4 akan memaparkan hasil eksperimen yang disertai dengan analisis. Terakhir, jurnal ini ditutup dengan kesimpulan di Bab 5, dan memberikan saran untuk pengembangan di masa mendatang.

2. Studi Terkait

2.1. Penelitian dengan *Embedding* OpenAI

Giglietto [16] membandingkan kinerja model Text-embedding-3-large untuk *clustering* berita politik. Huang *et al.* [17] menggunakan model Text-embedding-3-large untuk analisis semantik melalui *hierarchical clustering*. Keraghel *et al.* [18], Korade *et al.* [19], dan Kheiri & Karimi [7] menunjukkan efektivitas model *embedding* OpenAI dalam tugas *clustering* dokumen, penilaian kemiripan kalimat, dan analisis sentimen. Ajroudi *et al.* [20] menerapkan beberapa versi model *embedding* sebagai ekstraktor fitur dari transkrip ucapan untuk deteksi penyakit Alzheimer. Lho *et al.* [21] menerapkan beberapa versi model yang sama untuk mendeteksi depresi dan risiko bunuh diri dari narasi pasien. Venkatesh & Raman [22] menggunakan model *embedding* OpenAI dalam *benchmark* untuk evaluasi pemahaman semantik teks.

2.2. Penelitian pada Dataset MR

Huang *et al.* [11] membandingkan *embedding* Word2Vec, GloVe, dan BERT menggunakan model *Convolutional Neural Network* (CNN) dan *Bidirectional Long Short-Term Memory* (BiLSTM). Deniz *et al.* [13] menunjukkan bahwa menyempurnakan *embedding* yang sudah ada dengan informasi kontekstual dan skor sentimen dari VADER dapat meningkatkan akurasi klasifikasi. Khasanah [15] mengeksplorasi model *embedding* FastText dengan arsitektur DL sederhana. Hasilnya, CNN dengan FastText mencapai akurasi 80%. Zulqarnain *et al.* [14] mengusulkan arsitektur *Two-State* GRU (TS-GRU) yang dilengkapi mekanisme atensi fitur,

menggunakan *embedding* GloVe, dan melaporkan akurasi 80.72%. Jin dan Zhao [12] menggunakan *embedding* BERT sebagai masukan untuk model BUGE (*Bert-based Unlinked Graph Embedding*) dan mencapai akurasi 80.44%.

2.3. Analisis Sentimen

Analisis sentimen adalah sebuah bidang yang berfokus pada pengidentifikasian emosi yang diekspresikan dalam teks. Tujuan utamanya adalah untuk mengkategorikan teks ke dalam sentimen positif, negatif, atau netral [1].

Di ranah media sosial, teknik ini dimanfaatkan untuk memahami opini publik mengenai isu atau figur tertentu [23]. Dalam dunia bisnis, perusahaan menggunakannya untuk menganalisis ulasan produk atau umpan balik pelanggan guna peningkatan kualitas dan kepuasan [3]. Dalam bidang politik, analisis sentimen dapat berfungsi sebagai alat untuk melacak sentimen pemilih terhadap kandidat atau kebijakan yang ada [24].

2.4. *Machine Learning* dalam Analisis Sentimen

Secara umum, terdapat dua pendekatan utama dalam melakukan analisis sentimen: pendekatan berbasis kamus dan pendekatan berbasis ML [25]. Pendekatan berbasis kamus mengandalkan leksikon yang telah diberi skor sentimen. Pendekatan berbasis ML memperlakukan analisis sentimen sebagai permasalahan klasifikasi teks, di mana sebuah model dilatih menggunakan data yang telah diberi label [26].

2.4.1. *Supervised Learning*

Dalam konteks pendekatan ML, teknik yang paling banyak digunakan dalam analisis sentimen adalah melalui teknik *supervised learning* atau pembelajaran terawasi, di mana model ML dilatih menggunakan dataset yang setiap datanya telah diberi label sentimen [26]. Model lalu belajar untuk memetakan fitur-fitur yang diekstraksi dari teks ke label sentimen yang sesuai.

2.4.2. *Zero-Shot*

Pendekatan ini memungkinkan klasifikasi sentimen pada dataset baru tanpa memerlukan data latih berlabel spesifik untuk tugas tersebut [27]. Teknik klasifikasi *zero-shot* memanfaatkan kemampuan model *Large Language Model* (LLM). Klasifikasi atau prediksi sentimen dapat dilakukan dengan mengukur kemiripan semantik antara representasi vektor teks masukan dengan representasi vektor dari deskripsi label kandidat [28].

2.5. Representasi Teks

Representasi teks adalah langkah untuk mengubah data tekstual menjadi format numerik yang dapat diproses oleh algoritma. Metode ekstraksi fitur tradisional seperti *Bag-of-Words* (BoW) dan *Term Frequency-Inverse Document Frequency* (TF-IDF) merepresentasikan teks sebagai vektor *sparse* berdasarkan frekuensi kemunculan kata [29]. Metode ini mengabaikan urutan kata dan hubungan semantik antar kata [13, 30].

2.5.1. *Text Embedding*

Embedding teks adalah representasi numerik dari teks dalam bentuk vektor berdimensi rendah [31]. Representasi ini menangkap makna semantik teks, sehingga teks yang memiliki makna serupa akan memiliki vektor yang berdekatan dalam ruang vektor. Model seperti Word2Vec [32] dan GloVe [33] menghasilkan *embedding* kata statis di mana setiap kata memiliki satu vektor tetap terlepas dari konteks di mana kata tersebut muncul [5, 31].

Perkembangan selanjutnya adalah *embedding* kontekstual yang dihasilkan oleh model bahasa berbasis *Transformer* [34] seperti BERT [35] dan berbasis *Bidirectional* LSTM seperti ELMo (*Embeddings from Language Models*) [36]. *Embedding* kontekstual menghasilkan representasi vektor untuk setiap kata dalam sebuah kalimat yang bergantung pada kata-kata lain dalam kalimat yang sama. Dengan demikian, *embedding* ini mampu mengatasi masalah polisemi [5]. Kemampuan untuk memahami konteks ini sangat penting dalam mendeteksi sentimen yang lebih halus seperti ironi dan sarkasme [37].

2.6. Word2Vec

Model ini menggunakan jaringan syaraf tiruan sederhana dengan dua arsitektur utama: *Continuous Bag-of-Words* (CBOW) dan *Skip-gram*. Arsitektur CBOW memprediksi kata target berdasarkan kata-kata di sekitarnya, sementara arsitektur *Skip-gram* memprediksi kata-kata konteks berdasarkan sebuah kata target.

Dalam penelitian ini, model *pre-trained* Word2Vec yang digunakan adalah *GoogleNews-vectors-negative300.bin*, yang merupakan model Word2Vec yang dilatih pada korpus Google News, yang berisi sekitar 100 miliar kata. Hasilnya adalah vektor 300 dimensi untuk sekitar 3 juta kata dan frasa dalam bahasa Inggris.

2.7. *Embedding* Teks OpenAI

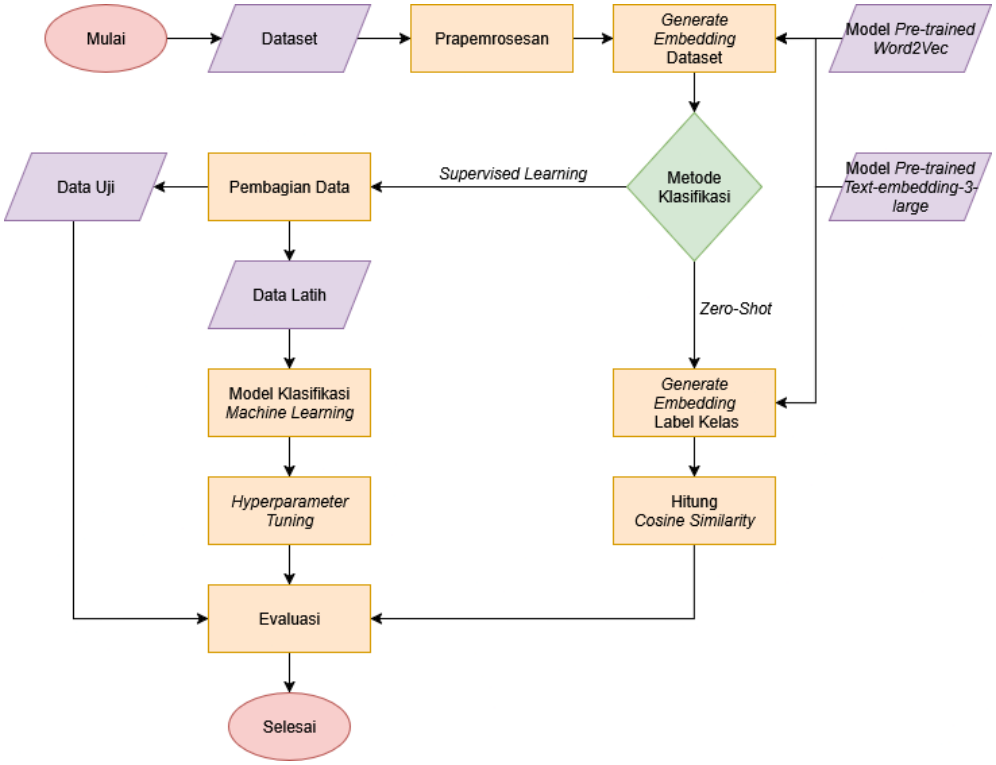
Kemajuan signifikan dalam bidang NLP dalam beberapa tahun terakhir terjadi dengan munculnya model LLM berbasis arsitektur *Transformer*, seperti seri GPT (*Generative Pre-trained Transformer*) dari OpenAI [38],

[39]. Model-model ini menunjukkan kemampuan luar biasa dalam memahami dan menghasilkan bahasa alami. Namun, OpenAI tidak mempublikasikan mengenai dataset yang digunakan untuk proses pelatihan model-model mereka.

OpenAI juga membuat beberapa model *embedding* teks: Text-embedding-3-small dan Text-embedding-3-large [6]. Arsitektur model *embedding* OpenAI melibatkan penggunaan *encoder Transformer* yang dilatih menggunakan *contrastive pre-training* pada data berpasangan *unsupervised* (seperti pasangan *docstring-code*) [40]. Model *embedding* dari OpenAI dapat dengan mudah digunakan melalui permintaan API, dengan biaya sebesar $0.13 per 1 juta token.

Pada *benchmark* MTEB (*Massive Text Embedding Benchmark*), Text-embedding-3-large mencapai hasil *state-of-the-art* untuk model dari OpenAI, dengan skor rata-rata 64.6%, melampaui skor Text-embedding-ada-002 sebesar 61.0% [8]. Secara bawaan, Text-embedding-3-large menghasilkan *embedding* dengan 3072 dimensi, dan Text-embedding-3-small dengan 1536 dimensi.

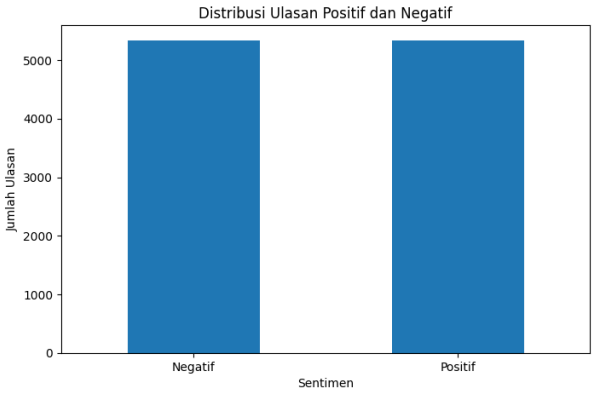
3. Sistem yang Dibangun



Gambar 1. Alur Sistem yang dibangun

3.1. Dataset *Movie Reviews*

Dataset ini terdiri dari 10.662 data ulasan film dengan distribusi data yang seimbang (Gambar 2) dari situs web *Rotten Tomatoes*. Setiap ulasan direpresentasikan sebagai satu kalimat. MR adalah dataset untuk klasifikasi sentimen biner. Meskipun tergolong dataset yang relatif kecil, dataset MR telah menjadi *benchmark* standar untuk evaluasi berbagai algoritma dan teknik klasifikasi sentimen [41].



Gambar 2. Distribusi Sentimen dalam Dataset

3.2. Prapemrosesan Dataset

Tahap prapemrosesan merupakan langkah penting dalam mempersiapkan data teks sebelum klasifikasi [42]. Dataset MR secara bawaan telah diproses dengan pemisahan tanda baca dan *lowercasing*. Prapemrosesan yang dilakukan meliputi beberapa langkah berikut:

1. *Exploratory Data Analysis*.
2. Penghapusan Karakter Non-Alfanumerik.
3. Penghapusan Spasi berlebih.

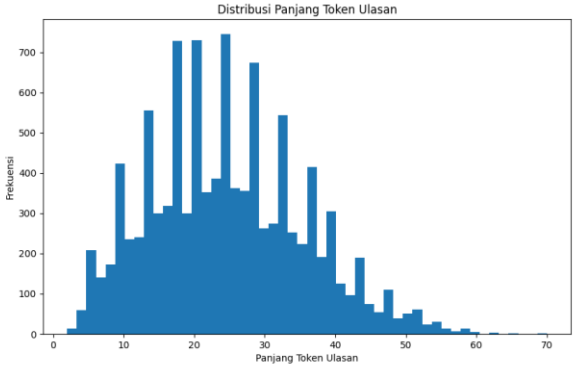
Saat menggunakan model *embedding* kontekstual, melakukan prapemrosesan minimal dapat menghasilkan performa yang lebih baik [43]. Ini berarti mempertahankan bentuk kata asli sebanyak mungkin [44].

3.3. *Generate Embedding*

Model Text-embedding-3-large, dan Word2Vec sebagai pembanding digunakan untuk menghasilkan *embedding*.

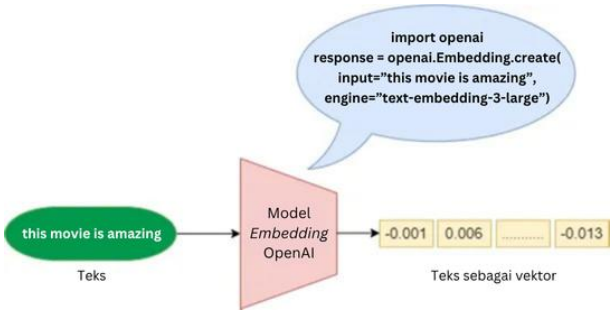
3.3.1. *Generate Embedding* Text-embedding-3-large

Setiap teks ulasan dipecah menjadi token menggunakan *tokenizer* dari pustaka. Selanjutnya, pengecekan jumlah token dalam dataset, tujuannya adalah untuk membatasi jumlah token yang digunakan sebelum pemanggilan API. Pembatasan ini dilakukan untuk efisiensi pemrosesan dan biaya, di mana hanya ulasan yang memenuhi batas token inilah yang kemudian diproses lebih lanjut.



Gambar 3. Distribusi Panjang Token Data

Setelah pengecekan token, total seluruh token dalam dataset MR adalah 263.277, estimasi biaya API untuk menghasilkan *embedding* seluruh token ini adalah Rp491.42. Setelah itu, sistem melakukan pemanggilan API melalui API *Key* dengan model Text-embedding-3-large untuk setiap ulasan. *Embedding* yang dihasilkan dari model ini memiliki dimensi sebesar 3072. Vektor *embedding* yang diperoleh lalu disimpan untuk digunakan dalam tahap klasifikasi.



Gambar 4. Proses *Generate Embedding* OpenAI

3.3.2. *Generate Embedding* Word2Vec

Model *Pre-trained* Word2Vec digunakan dengan pustaka Gensim. Pertama, setiap teks ulasan dipecah menjadi token kata menggunakan *tokenizer* dari pustaka NLTK. Kemudian, sistem akan mencoba mencari representasi vektor yang sesuai untuk setiap token dari model Word2Vec yang telah dimuat. Untuk mendapatkan keseluruhan representasi vektor dari sebuah ulasan, sistem merata-ratakan semua vektor kata yang berhasil ditemukan dalam ulasan. Kata-kata yang tidak terdapat dalam kamus model akan diabaikan dan tidak diikut sertakan. Vektor nol dengan dimensi 300 akan dihasilkan jika tidak ada satu pun kata dari keseluruhan ulasan yang dikenali oleh model.

3.4. Klasifikasi *Supervised Learning*

Sebelum pelatihan model, dilakukan proses pembagian data, lalu dua algoritma klasifikasi, SVM dan LR dipilih untuk membangun model klasifikasi sentimen. Kedua algoritma ini telah terbukti menunjukkan performa yang baik dalam berbagai tugas klasifikasi teks [45].

3.4.1. Pembagian Data

Dataset dibagi menjadi dua bagian: 80% untuk pelatihan (8529 data; 4265 sentimen positif, 4264 sentimen negatif), dan 20% untuk pengujian (2133 data; 1066 sentimen positif, 1067 sentimen negatif).

3.4.2. *Support Vector Machine*

Prinsip dari model SVM adalah menemukan *hyperplane* yang secara optimal memisahkan dua kelas data di dalam ruang fitur berdimensi tinggi yang dibentuk oleh vektor *embedding*. *Hyperplane* ini dapat direpresentasikan dengan persamaan:

$$w \cdot x + b = 0 \tag{3.1}$$

Di mana:

- w adalah vektor bobot yang normal terhadap *hyperplane*.
- x adalah vektor fitur input.
- b adalah *bias*.

Tujuan SVM adalah untuk memaksimalkan margin antara *hyperplane* dan titik data terdekat dari setiap kelas. Ini dicapai dengan menyelesaikan masalah optimisasi berikut:

$$\min_{w,b} \frac{1}{2}\|w\|^2 \tag{3.2}$$

Dengan batasan $y_i(w \cdot x_i + b) \geq 1$ untuk setiap titik data (x_i, y_i), di mana y_i adalah label kelas (+1 atau -1). Model ini dipilih karena sangat efektif untuk data berdimensi tinggi [41] dan kemampuannya klasifikasi biner [46].

3.4.3. *Logistic Regression*

LR adalah sebuah metode analisis regresi yang secara luas digunakan untuk tugas klasifikasi [26]. Inti dari LR adalah fungsi sigmoid, yang memetakan output dari kombinasi linear fitur-fitur *input* ke dalam rentang nilai antara 0 dan 1 [47]. Kombinasi linear dihitung sebagai:

$$z = w \cdot x + b \tag{3.3}$$

Di mana:

- w adalah vektor bobot.
- x adalah vektor fitur.
- b adalah *bias*.

Hasil z kemudian dimasukkan ke dalam fungsi sigmoid:

$$\sigma(z) = \frac{1}{1 + e^{-z}} \tag{3.4}$$

Nilai yang dihasilkan $\sigma(z)$ diinterpretasikan sebagai probabilitas ulasan tersebut memiliki sentimen positif.

3.5. *Hyperparameter Tuning*

Untuk mengoptimalkan performa model klasifikasi, dilakukan *hyperparameter tuning* menggunakan teknik RandomizedSearchCV dengan 50 kombinasi parameter yang diuji dengan *10-fold cross-validation* pada data pelatihan. Detail *hyperparameter* yang dioptimalkan untuk setiap model adalah sebagai berikut:

- SVM:
  - C: Kekuatan regularisasi. Distribusi nilai dipilih dari distribusi *uniform* antara 0.001 sampai 10.
  - Tol: Toleransi. Distribusi nilai dipilih dari distribusi *uniform* antara 1e-5 sampai 1e-2.
  - Max_iter: Batas maksimum iterasi yang diuji dengan nilai 1000, 5000, dan 10000.
- LR:

- C: Kekuatan regularisasi. Distribusi nilai dipilih dari distribusi *uniform* antara 0.001 sampai 10.
- Penalty: Tipe regularisasi yang digunakan (L1 atau L2).
- Tol: Toleransi. Distribusi nilai dipilih dari distribusi *uniform* antara 1e-5 sampai 1e-2.
- Max_iter: Batas maksimum iterasi yang diuji dengan nilai 100, 500, 1000, 5000 dan 10000.

3.6. Klasifikasi *Zero-Shot*

Langkah-langkah implementasi klasifikasi *zero-shot* dalam penelitian ini adalah sebagai berikut: (1) Mendefinisikan label untuk setiap kelas. (2) Menghasilkan vektor *embedding* untuk label ini menggunakan model Text-embedding-3-large. (3) Menghasilkan vektor *embedding* untuk setiap ulasan dalam dataset. (4) Mengukur *cosine similarity* antara vektor setiap ulasan dengan *embedding* setiap label kelas. Rumus untuk menghitung *cosine similarity* adalah sebagai berikut:

$$Cosine\_Similarity(A, B) = \frac{\boldsymbol{A} \cdot \boldsymbol{B}}{\|\boldsymbol{A}\|\|\boldsymbol{B}\|} \tag{3.5}$$

Di mana:

- A dan B adalah vektor *embedding* yang dibandingkan.
- A · B adalah *dot product* dari kedua vektor.
- ||A|| dan ||B|| adalah magnitudo atau norma Euclidean dari masing-masing vektor.

Untuk setiap ulasan, perhitungannya dilakukan dua kali. Pertama, vektor A adalah *embedding* dari ulasan film, dan vektor B adalah *embedding* dari label positif. Kedua, perhitungan diulang dengan vektor B sebagai *embedding* dari label negatif. Ulasan tersebut kemudian diklasifikasikan ke dalam kelas yang menghasilkan nilai *cosine similarity* tertinggi. Kinerja kemudian dievaluasi menggunakan label asli dataset sebagai *ground truth*.

3.7. Metrik Evaluasi

Untuk mengevaluasi kinerja kedua pendekatan, metrik evaluasi yang digunakan yaitu akurasi (3.6), presisi (3.7), *recall* (3.8), dan *F1-score* (3.9).

$$Akurasi = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.6}$$

$$Presisi = \frac{TP}{TP + FP} \tag{3.7}$$

$$Recall = \frac{TP}{TP + FN} \tag{3.8}$$

$$F1 - Score = 2 \times \frac{Presisi \times Recall}{Presisi + Recall} \tag{3.9}$$

Di mana TP, TN, FP, FN masing-masing adalah *True Positive, True Negative, False Positive,* dan *False Negative*.

4. Evaluasi

Penelitian ini dilakukan dengan dua skenario utama: *supervised learning* dan *zero-shot*. Setiap skenario diuji pada dua kondisi dataset yang berbeda: dataset yang telah melalui tahap prapemrosesan pembersihan (*cleaned*) dan dataset asli tanpa pembersihan (*uncleaned*).

1. Skenario Klasifikasi *Supervised Learning*: Model Word2Vec selanjutnya akan disebut sebagai W2V, dan model OpenAI Text-embedding-3-large selanjutnya akan disebut sebagai Embedding-3. Setiap model diuji dalam dua konfigurasi: sekali dengan *hyperparameter* bawaan (selanjutnya disebut sebagai *default*), dan sekali dengan *hyperparameter* yang dioptimalkan pada data latih. Model klasifikasi yang telah dioptimalkan akan disebut sebagai *tuned*.
2. Skenario Klasifikasi *Zero-Shot*: Skenario ini dimulai dengan mendefinisikan dua deskripsi label yang untuk setiap kelas. *Embedding* untuk kedua deskripsi ini dihasilkan menggunakan model Embedding-3. Kemudian, untuk setiap ulasan, *cosine similarity*-nya dihitung. Ulasan selanjutnya diklasifikasikan ke dalam kelas yang memiliki nilai kemiripan tertinggi.

4.1. Analisis Hasil Pengujian

4.1.1. Skenario Klasifikasi *Supervised Learning*

Tabel 1. Hasil Klasifikasi *Supervised* dengan Dataset *Cleaned*

Model		Akurasi	Presisi	*Recall*	*F1-score*
W2V-LR	*Default*	77.68%	77.71%	77.68%	77.68%
	Tuned	77.92%	77.94%	77.92%	77.91%
W2V-SVM	*Default*	77.40%	77.44%	77.40%	77.39%
	Tuned	77.73%	77.76%	77.73%	77.72%
Embedding-3-LR	*Default*	92.12%	92.14%	92.12%	92.12%
	Tuned	92.45%	92.49%	92.45%	92.45%
Embedding-3-SVM	***Default***	**92.55%**	**92.57%**	**92.55%**	**92.54%**
	Tuned	92.45%	92.49%	92.45%	92.45%

Hasil percobaan menunjukkan superioritas dari model *embedding* kontekstual OpenAI dibandingkan dengan *embedding* statis. Pada dataset tanpa pembersihan, model Word2Vec terbaik (W2V-LR *Default*) hanya mencapai *F1-score* sebesar 78.01%. Sebaliknya, model terbaik yang menggunakan Embedding-3 (Embedding-3-SVM *Default*) mencapai *F1-score* 93.20%. Hal ini mengindikasikan bahwa kemampuan model *embedding* kontekstual dalam menangkap nuansa semantik, sintaksis, dan hubungan antar kata jauh lebih unggul dibandingkan *embedding* statis.

Tabel 2. Hasil Klasifikasi *Supervised* dengan Dataset *Uncleaned*

Model		Akurasi	Presisi	*Recall*	*F1-score*
W2V-LR	*Default*	78.01%	78.04%	78.01%	78.01%
	Tuned	77.31%	77.32%	77.31%	77.31%
W2V-SVM	*Default*	76.98%	77.00%	76.98%	76.98%
	Tuned	77.78%	77.81%	77.78%	77.77%
Embedding-3-LR	*Default*	92.59%	92.61%	92.59%	92.59%
	Tuned	92.92%	92.94%	92.92%	92.92%
Embedding-3-SVM	***Default***	**93.20%**	**93.23%**	**93.20%**	**93.20%**
	Tuned	92.97%	92.99%	92.97%	92.97%

Untuk model yang menggunakan W2V, *F1-score* tertinggi terletak pada dataset *uncleaned*. Ini menunjukkan W2V tidak banyak mengambil manfaat dari tanda baca. Kinerja model Embedding-3 secara konsisten juga lebih tinggi pada dataset *uncleaned*. Model terbaik secara keseluruhan adalah Embedding-3-SVM *Default* pada dataset *uncleaned* dengan *F1-score* 93.20%. Hal ini mengimplikasikan bahwa model *embedding* OpenAI mampu memanfaatkan tanda baca sebagai fitur sentimen.

Ketika menggunakan *embedding* W2V, performa antara LR dan SVM relatif sebanding, dengan LR menunjukkan keunggulan pada parameter *default*. Sedangkan ketika menggunakan Embedding-3, model SVM secara konsisten memberikan hasil yang terbaik pada parameter *default*. Ini menunjukkan bahwa kemampuan SVM memberikan sedikit keuntungan dalam memisahkan kelas di ruang fitur berdimensi tinggi.

Pada model W2V, proses *hyperparameter tuning* memberikan peningkatan kinerja terkecuali untuk model W2V-LR pada dataset *uncleaned*. Pada model berbasis Embedding-3, *hyperparameter tuning* meningkatkan performa dalam beberapa kasus. Hal ini mengindikasikan bahwa parameter *default* dari model klasifikasi seringkali sudah cukup optimal.

Kombinasi Embedding-3 dengan model SVM pada data yang tidak dibersihkan (*uncleaned*) dengan *hyperparameter default* terbukti menjadi konfigurasi paling unggul dalam penelitian ini.

4.1.2. Skenario Klasifikasi *Zero-Shot*

Tabel 3. Hasil Klasifikasi *zero-shot*

Klasifikasi		Akurasi	Presisi	*Recall*	*F1-score*
Zero-Shot	*Cleaned*	84.77%	84.78%	84.77%	84.77%
	Uncleaned	**86.29%**	**86.30%**	**86.29%**	**86.29%**

Serupa dengan hasil pada skenario klasifikasi *supervised* yang menggunakan model Embedding-3, pendekatan *zero-shot* juga menunjukkan kinerja yang lebih baik pada dataset *uncleaned*. *F1-score* meningkat dari 84.77% menjadi 86.29%, sebuah peningkatan sebesar 1.52%. Temuan ini menunjukkan bahwa Embedding-3 mampu mengekstrak informasi sentimen tambahan dari karakter non-alfanumerik

seperti tanda baca.

Performa *zero-shot* yang cukup baik pada dataset MR ini mungkin tidak sepenuhnya mencerminkan kemampuan generalisasi model pada domain yang baru. Dataset MR adalah salah satu *benchmark* paling populer dan telah ada sejak lama dalam penelitian NLP [41]. Sangat mungkin bahwa data dari dataset ini telah menjadi bagian dari data pelatihan yang digunakan oleh OpenAI.

5. Kesimpulan

Penelitian ini berhasil menunjukkan bahwa model *embedding* teks kontekstual Embedding-3 dari OpenAI memiliki performa yang superior untuk klasifikasi sentimen pada dataset ulasan film dibandingkan dengan *embedding* statis.

Selain itu, penelitian ini membuktikan Embedding-3 sangat efektif untuk klasifikasi *zero-shot*. Kinerja ini melampaui kinerja *embedding* statis dalam klasifikasi *supervised learning* secara telak. Hasil ini sudah bagus, tetapi kinerja tertinggi masih didapatkan melalui *supervised learning* dengan model Embedding-3.

Terdapat beberapa arah pengembangan yang dapat dieksplorasi. Untuk mendapatkan gambaran performa yang lebih luas, disarankan untuk membandingkan Embedding-3 dengan model *embedding* kontekstual lainnya.

Mengenai model yang diuji, penelitian ini terbatas pada model ML tradisional (SVM dan LR). Akan sangat menarik untuk menginvestigasi bagaimana performa *embedding* OpenAI jika dipasangkan dengan arsitektur DL.

Metodologi ini juga dapat diuji pada dataset dari domain yang berbeda dan dalam bahasa lain, khususnya Bahasa Indonesia.

Model Embedding-3 telah mendukung pemotongan dimensi vektor. Penelitian selanjutnya dapat menganalisis dampak dari pengurangan dimensi ini terhadap akurasi, kecepatan komputasi, dan kebutuhan sumber daya pada tugas klasifikasi sentimen.

Daftar Pustaka

[1] B. Liu, “Sentiment Analysis and Opinion Mining”.

[2] B. Csanády, L. Muzsai, P. Vedres, Z. Nádasdy, dan A. Lukács, “LlamBERT: Large-scale low-cost data annotation in NLP,” 23 Maret 2024, *arXiv*: arXiv:2403.15938. doi: 10.48550/arXiv.2403.15938.

[3] P. S. Ghatora, S. E. Hosseini, S. Pervez, M. J. Iqbal, dan N. Shaukat, “Sentiment Analysis of Product Reviews Using Machine Learning and Pre-Trained LLM,” *BDCC*, vol. 8, no. 12, hlm. 199, Des 2024, doi: 10.3390/bdcc8120199.

[4] S. A. Salloum, R. Alfaisal, A. Basiouni, K. Shaalan, dan A. Salloum, “Effectiveness of Logistic Regression for Sentiment Analysis of Tweets About the Metaverse,” dalam *Breaking Barriers with Generative Intelligence. Using GI to Improve Human Education and Well-Being*, vol. 2162, A. Basiouni dan C. Frasson, Ed., dalam Communications in Computer and Information Science, vol. 2162. , Cham: Springer Nature Switzerland, 2024, hlm. 32–41. doi: 10.1007/978-3-031-65996-6_3.

[5] R. Patil, S. Boit, V. Gudivada, dan J. Nandigam, “A Survey of Text Representation and Embedding Techniques in NLP,” *IEEE Access*, vol. 11, hlm. 36120–36146, 2023, doi: 10.1109/ACCESS.2023.3266377.

[6] “New *embedding* models and API updates | OpenAI.” Diakses: 31 Maret 2025. [Daring]. Tersedia pada: https://openai.com/index/new-*embedding*-models-and-api-updates/

[7] K. Kheiri dan H. Karimi, “SentimentGPT: Exploiting GPT for Advanced Sentiment Analysis and its Departure from Current Machine Learning,” 23 Juli 2023, *arXiv*: arXiv:2307.10234. doi: 10.48550/arXiv.2307.10234.

[8] “Vector *embedding*s - OpenAI API.” Diakses: 21 Maret 2025. [Daring]. Tersedia pada: https://platform.openai.com

[9] Nikhil Sanjay Suryawanshi, “Sentiment analysis with machine learning and deep learning: A survey of techniques and applications,” *Int. J. Sci. Res. Arch.*, vol. 12, no. 2, hlm. 005–015, Jul 2024, doi: 10.30574/ijsra.2024.12.2.1205.

[10] B. Pang dan L. Lee, “Seeing stars: exploiting class relationships for sentiment categorization with respect to rating scales,” dalam *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics - ACL ’05*, Ann Arbor, Michigan: Association for Computational Linguistics, 2005, hlm. 115–124. doi: 10.3115/1219840.1219855.

[11] R. Huang, Q. Chen, J. Tang, dan J. Song, “The Influence of Word Embeddings on the Performance of Sentiment Classification,” *IJCIT*, vol. 4, no. 1, hlm. 1, Okt 2023, doi: 10.56028/ijcit.1.4.1.2023.

[12] Y. Jin dan A. Zhao, “Bert-based graph unlinked *embedding* for sentiment analysis,” *Complex Intell. Syst.*, vol. 10, no. 2, hlm. 2627–2638, Apr 2024, doi: 10.1007/s40747-023-01289-9.

[13] A. Deniz, M. Angin, dan P. Angin, “Sentiment and Context-refined Word Embeddings for Sentiment Analysis,” dalam *2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, Melbourne, Australia: IEEE, Okt 2021, hlm. 927–932. doi: 10.1109/SMC52423.2021.9659189.

[14] M. Zulqarnain, R. Ghazali, M. Aamir, dan Y. M. M. Hassim, “An efficient two-state GRU based on feature attention mechanism for sentiment analysis,” *Multimed Tools Appl*, vol. 83, no. 1, hlm. 3085–3110, Jan

2024, doi: 10.1007/s11042-022-13339-4.
[15] I. N. Khasanah, "Sentiment Classification Using fastText Embedding and Deep Learning Model," *Procedia Computer Science*, vol. 189, hlm. 343–350, 2021, doi: 10.1016/j.procs.2021.05.103.
[16] F. Giglietto, "Evaluating Embedding Models for Clustering Italian Political News: A Comparative Study of Text-Embedding-3-Large and UmBERTo," 20 Agustus 2024. doi: 10.31219/osf.io/2j9ed.
[17] Z. Huang, Y. Long, K. Peng, dan S. Tong, "An Embedding-Based Semantic Analysis Approach: A Preliminary Study on Redundancy Detection in Psychological Concepts Operationalized by Scales," *J. Intell.*, vol. 13, no. 1, hlm. 11, Jan 2025, doi: 10.3390/jintelligence13010011.
[18] I. Keraghel, S. Morbieu, dan M. Nadif, "Beyond Words: A Comparative Analysis of LLM Embeddings for Effective Clustering," dalam *Advances in Intelligent Data Analysis XXII*, vol. 14641, I. Miliou, N. Piatkowski, dan P. Papapetrou, Ed., dalam Lecture Notes in Computer Science, vol. 14641. , Cham: Springer Nature Switzerland, 2024, hlm. 205–216. doi: 10.1007/978-3-031-58547-0_17.
[19] N. B. Korade, M. B. Salunke, A. A. Bhosle, P. B. Kumbharkar, G. G. Asalkar, dan R. G. Khedkar, "Strengthening Sentence Similarity Identification Through OpenAI Embeddings and Deep Learning," *IJACSA*, vol. 15, no. 4, 2024, doi: 10.14569/IJACSA.2024.0150485.
[20] K. Ajroudi, M. I. Khedher, O. Jemai, dan M. A. EI-Yacoubi, "Exploring the Efficacy of Text Embeddings in Early Dementia Diagnosis from Speech," dalam *2024 16th International Conference on Human System Interaction (HSI)*, Paris, France: IEEE, Jul 2024, hlm. 1–6. doi: 10.1109/HSI61632.2024.10613581.
[21] S. K. Lho *dkk.*, "Large Language Models and Text Embeddings for Detecting Depression and Suicide in Patient Narratives," *JAMA Netw Open*, vol. 8, no. 5, hlm. e2511922, Mei 2025, doi: 10.1001/jamanetworkopen.2025.11922.
[22] D. Venkatesh dan S. Raman, "BITS Pilani at SemEval-2024 Task 1: Using text-*embedding*-3-large and LaBSE *embedding*s for Semantic Textual Relatedness," dalam *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, Mexico City, Mexico: Association for Computational Linguistics, 2024, hlm. 865–868. doi: 10.18653/v1/2024.semeval-1.124.
[23] R. Chandra, J. Sonawane, dan J. Lande, "An Analysis of Vaccine-Related Sentiments on Twitter (X) from Development to Deployment of COVID-19 Vaccines," *BDCC*, vol. 8, no. 12, hlm. 186, Des 2024, doi: 10.3390/bdcc8120186.
[24] A. L. Jiménez-Preciado, J. Álvarez-García, S. Cruz-Aké, dan F. Venegas-Martínez, "The Power of Words from the 2024 United States Presidential Debates: A Natural Language Processing Approach," *Information*, vol. 16, no. 1, hlm. 2, Des 2024, doi: 10.3390/info16010002.
[25] M. Rodríguez-Ibánez, A. Casánez-Ventura, F. Castejón-Mateos, dan P.-M. Cuenca-Jiménez, "A review on sentiment analysis from social media platforms," *Expert Systems with Applications*, vol. 223, hlm. 119862, Agu 2023, doi: 10.1016/j.eswa.2023.119862.
[26] P. Monika, C. Kulkarni, N. Harish Kumar, S. Shruthi, dan V. Vani, "Machine learning approaches for sentiment analysis: A survey," *ijhs*, hlm. 1286–1300, Apr 2022, doi: 10.53730/ijhs.v6nS4.6119.
[27] I. Nawawi, K. F. Ilmawan, M. R. Maarif, dan M. Syafrudin, "Exploring Tourist Experience through Online Reviews Using Aspect-Based Sentiment Analysis with Zero-Shot Learning for Hospitality Service Enhancement," *Information*, vol. 15, no. 8, hlm. 499, Agu 2024, doi: 10.3390/info15080499.
[28] "Zero-shot classification with *embedding*s | OpenAI Cookbook." Diakses: 30 Maret 2025. [Daring]. Tersedia pada: https://cookbook.openai.com/examples/zero-shot_classification_with_*embedding*s
[29] C. Liu, Y. Sheng, Z. Wei, dan Y.-Q. Yang, "Research of Text Classification Based on Improved TF-IDF Algorithm," dalam *2018 IEEE International Conference of Intelligent Robotic and Control Engineering (IRCE)*, Lanzhou: IEEE, Agu 2018, hlm. 218–222. doi: 10.1109/IRCE.2018.8492945.
[30] E. Rudkowsky, M. Haselmayer, M. Wastian, M. Jenny, Š. Emrich, dan M. Sedlmair, "More than Bags of Words: Sentiment Analysis with Word Embeddings," *Communication Methods and Measures*, vol. 12, no. 2–3, hlm. 140–157, Apr 2018, doi: 10.1080/19312458.2018.1455817.
[31] M. T. Pilehvar dan J. Camacho-Collados, "Embeddings in Natural Language Processing".
[32] T. Mikolov, K. Chen, G. Corrado, dan J. Dean, "Efficient Estimation of Word Representations in Vector Space," 7 September 2013, *arXiv*: arXiv:1301.3781. doi: 10.48550/arXiv.1301.3781.
[33] J. Pennington, R. Socher, dan C. Manning, "Glove: Global Vectors for Word Representation," dalam *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar: Association for Computational Linguistics, 2014, hlm. 1532–1543. doi: 10.3115/v1/D14-1162.
[34] A. Vaswani *dkk.*, "Attention Is All You Need," 2 Agustus 2023, *arXiv*: arXiv:1706.03762. doi: 10.48550/arXiv.1706.03762.
[35] J. Devlin, M.-W. Chang, K. Lee, dan K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding".
[36] M. Peters *dkk.*, "Deep Contextualized Word Representations," dalam *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, New Orleans, Louisiana: Association for Computational Linguistics, 2018, hlm. 2227–2237. doi: 10.18653/v1/N18-1202.
[37] O. Galal, A. H. Abdel-Gawad, dan M. Farouk, "Rethinking of BERT sentence *embedding* for text

classification," *Neural Comput & Applic*, vol. 36, no. 32, hlm. 20245–20258, Nov 2024, doi: 10.1007/s00521-024-10212-3.
[38] A. Radford, K. Narasimhan, T. Salimans, dan I. Sutskever, "Improving Language Understanding by Generative Pre-Training".
[39] T. B. Brown *dkk.*, "Language Models are Few-Shot Learners".
[40] A. Neelakantan *dkk.*, "Text and Code Embeddings by Contrastive Pre-Training," 24 Januari 2022, *arXiv*: arXiv:2201.10005. doi: 10.48550/arXiv.2201.10005.
[41] Q. Li *dkk.*, "A Survey on Text Classification: From Traditional to Deep Learning," *ACM Trans. Intell. Syst. Technol.*, vol. 13, no. 2, hlm. 1–41, Apr 2022, doi: 10.1145/3495162.
[42] H.-T. Duong dan T.-A. Nguyen-Thi, "A review: preprocessing techniques and data augmentation for sentiment analysis," *Comput Soc Netw*, vol. 8, no. 1, hlm. 1, Des 2021, doi: 10.1186/s40649-020-00080-x.
[43] P. R. Amalia dan E. Winarko, "Aspect-Based Sentiment Analysis on Indonesian Restaurant Review Using a Combination of Convolutional Neural Network and Contextualized Word Embedding," *Indonesian J. Comput. Cybern. Syst.*, vol. 15, no. 3, hlm. 285, Jul 2021, doi: 10.22146/ijccs.67306.
[44] C. Kuo, *The handbook of NLP with Gensim: leverage topic modeling to uncover hidden patterns, themes, and valuable insights within textual data*, 1st edition. Birmingham, UK: Packt Publishing Ltd., 2023.
[45] O. Ozyurt dan A. Ayaz, "Twenty-five years of education and information technologies: Insights from a topic modeling based bibliometric analysis," *Educ Inf Technol*, vol. 27, no. 8, hlm. 11025–11054, Sep 2022, doi: 10.1007/s10639-022-11071-y.
[46] E. Hokijuliandy, H. Napitupulu, dan Firdaniza, "Application of SVM and Chi-Square Feature Selection for Sentiment Analysis of Indonesia's National Health Insurance Mobile Application," *Mathematics*, vol. 11, no. 17, hlm. 3765, Sep 2023, doi: 10.3390/math11173765.
[47] "What is the default threshold in Sklearn logistic regression? - GeeksforGeeks." Diakses: 16 Juni 2025. [Daring]. Tersedia pada: https://www.geeksforgeeks.org/data-science/what-is-the-default-threshold-in-sklearn-logistic-regression/

Lampiran