

Detecting Fake Reviews in E-Commerce: A Case Study on Shopee Using Support Vector Machine and Random Forest

Khoirotulmuadiba Purifyregalia ^{1*}, Khothibul Umam ^{2*}, Nur Cahyo Hendro Wibowo ^{3*}, Maya Rini Handayani ^{4*}

^{*} Teknologi Informasi, Fakultas Sains dan Teknologi, UIN Walisongo Semarang

2208096075@student.walisongo.ac.id ¹, khothibul_umam@walisongo.ac.id ², nur_cahyo@walisongo.ac.id ³, maya@walisongo.ac.id ⁴

Article Info

Article history:

Received 2025-04-26

Revised 2025-06-03

Accepted 2025-06-17

Keyword:

Fake Review Detection,
NLP,
Random Forest,
Shopee,
SVM.

ABSTRACT

The increasing popularity of online shopping, particularly on platforms such as Shopee, has made product reviews a significant factor influencing consumer purchasing decisions. However, the presence of fake reviews generated by non-human agents undermines consumer trust and affects platform credibility. This study aims to detect fake reviews on Shopee by applying a text classification approach using Random Forest and Support Vector Machine (SVM) algorithms. A dataset consisting of 3,686 Shopee product reviews was collected and underwent preprocessing steps including data cleaning, normalization, tokenization, and TF-IDF weighting. Review labeling was performed automatically through the Latent Dirichlet Allocation (LDA) method, categorizing reviews into Original (OR) and Computer-Generated (CG). Model performance was evaluated using accuracy, precision, recall, and F1-score metrics. Experimental results show that the SVM algorithm achieved the highest accuracy at 88.84%, outperforming Random Forest which obtained 80.39%. These findings highlight the effectiveness of SVM in handling high-dimensional text data for fake review detection. The study contributes to the application of automated topic modeling (LDA) for labeling e-commerce reviews in the Indonesian context and opens opportunities for further enhancement using larger datasets and deep learning-based models to improve classification accuracy and scalability.



This is an open access article under the [CC-BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.

I. PENDAHULUAN

Perkembangan teknologi digital di Indonesia terus mengalami pertumbuhan pesat, terutama dalam sektor *e-commerce*. Shopee merupakan salah satu *e-commerce* terpopuler di Indonesia yang banyak dipilih masyarakat untuk melakukan transaksi belanja secara *online*. Dengan meningkatnya jumlah transaksi daring, muncul tantangan baru dalam memastikan keaslian ulasan produk. Ulasan yang diberikan oleh konsumen sangat memengaruhi keputusan pembelian, tetapi tidak semua ulasan dapat dipercaya. Beberapa ulasan palsu atau manipulatif dibuat untuk meningkatkan reputasi produk atau menurunkan reputasi pesaing.

Deteksi ulasan palsu (*fake review detection*) menjadi topik yang krusial dalam menjaga integritas ekosistem *e-commerce*.

Ulasan palsu dapat merugikan konsumen dan mengurangi kepercayaan terhadap *platform e-commerce*. Oleh karena itu, diperlukan metode otomatis yang dapat mengklasifikasikan ulasan ke dalam kategori asli (*organic review/OR*) atau buatan (*computer-generated/CG*). Salah satu pendekatan yang umum digunakan dalam analisis teks adalah pemrosesan bahasa alami (*Natural Language Processing/NLP*), yang memungkinkan pemodelan teks menggunakan teknik seperti *Term Frequency-Inverse Document Frequency* (TF-IDF) dan algoritma *machine learning*.

Beberapa penelitian sebelumnya telah mengeksplorasi pendekatan yang beragam dalam mendeteksi ulasan palsu. Penelitian oleh [1] menggunakan kombinasi model *BERT* dan *Random Forest* untuk mengidentifikasi *fake review* pada ulasan hotel di Lombok dan berhasil mencapai akurasi tertinggi sebesar 84%. Model ini menunjukkan performa yang

cukup baik dalam menangkap konteks kata secara mendalam melalui representasi *BERT*, namun tetap mengandalkan *Random Forest* sebagai *classifier* akhir. Selanjutnya, studi oleh [2] menerapkan algoritma *Random Forest* untuk mendeteksi ulasan palsu pada platform *Amazon* dengan menggunakan 90.000 data uji, menghasilkan akurasi sebesar 79,47%.

Penelitian lain oleh [3] memanfaatkan kombinasi jaringan saraf konvolusi (CNN) dan *Long Short-Term Memory* (LSTM) untuk mendeteksi ulasan palsu dan mencapai akurasi sebesar 88,78%. Model ini mampu menangkap pola sekuensial dan spasial dalam data teks, yang membuatnya efektif untuk memahami konteks dalam ulasan yang kompleks. Sementara itu, studi oleh [4] mengembangkan model berbasis *BERT* dan LSTM yang dipadukan dengan teknik *Monte Carlo Dropout* (MCD), dan diterapkan pada dataset *yelp* berlabel yang terdiri dari 10.000 ulasan hotel di Amerika Utara. Hasilnya, model *BERT* yang tertanam dalam MCD mencatatkan akurasi tertinggi sebesar 91,75% dan terbukti mengungguli performa model LSTM murni dalam hal generalisasi pada data yang belum terlihat sebelumnya.

Selain itu, penelitian oleh [5] melakukan perbandingan kinerja metode SVM dan *Naïve Bayes* untuk deteksi ulasan palsu di Tokopedia. Dataset yang digunakan dibagi menjadi dua tahap: pelatihan model dan prediksi data baru. Pada saat diuji terhadap 29 ulasan tanpa label, model SVM memprediksi 8 ulasan tergolong ke dalam kategori asli dan 21 ulasan tergolong buatan. Sementara itu, *Naïve Bayes* memprediksi 7 data sebagai OR dan 22 sebagai CG. Hasil tersebut mengindikasikan bahwa kedua model memiliki performa yang relatif serupa, namun belum sepenuhnya optimal untuk kasus dengan data terbatas.

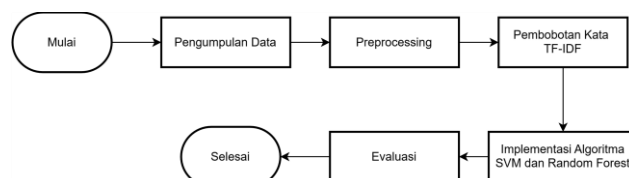
Meskipun berbagai metode telah berhasil diaplikasikan pada domain ulasan produk secara umum, penelitian yang secara spesifik fokus pada deteksi ulasan palsu di platform *e-commerce* lokal seperti *Shopee* masih sangat terbatas. Selain itu, mayoritas penelitian terdahulu menggunakan metode pelabelan data secara manual, yang rentan terhadap subjektivitas dan memerlukan waktu yang cukup lama. Hal ini menunjukkan adanya kebutuhan untuk mengembangkan pendekatan baru yang lebih efisien dan sesuai dengan karakteristik platform lokal.

Penelitian ini menawarkan pendekatan baru dengan memanfaatkan Latent Dirichlet Allocation (LDA) untuk melakukan pelabelan data secara otomatis sebelum proses klasifikasi. Penggunaan LDA diharapkan dapat meningkatkan efisiensi, mengurangi bias subjektif, dan mempercepat proses anotasi data dibandingkan dengan metode pelabelan manual. Dengan penggunaan algoritma *Random Forest* dan *Support Vector Machine* (SVM) dikombinasikan dengan fitur TF-IDF, diharapkan dapat dihasilkan model yang mampu mengklasifikasikan ulasan dengan tingkat akurasi tinggi. Hasil penelitian ini diharapkan dapat memberikan kontribusi dalam pengembangan sistem deteksi ulasan palsu secara otomatis di platform *e-commerce*

dan meningkatkan kepercayaan konsumen terhadap ulasan produk.

II. METODE

Penelitian ini dilaksanakan melalui serangkaian tahapan, mulai dari proses pengumpulan data, dilanjutkan dengan *preprocessing*, pembobotan kata menggunakan TF-IDF, penerapan algoritma *Random Forest* dan SVM, dan diakhiri dengan evaluasi model. Alur dari seluruh proses tersebut disajikan pada Gambar 1.



Gambar 1. Alur Penelitian

A. Pengumpulan Data

Pengumpulan data merupakan tahap fundamental dalam penelitian ilmiah yang bertujuan untuk memperoleh informasi yang relevan dan valid sesuai dengan tujuan studi. Menurut [6], pengumpulan data dilakukan untuk memperoleh data yang sesuai dengan permasalahan penelitian secara sistematis dan objektif, baik melalui sumber primer maupun sekunder. Pengumpulan data dalam penelitian ini dilakukan dengan menggunakan dataset sekunder yang tersedia secara terbuka di repositori GitHub. Dataset tersebut berisi sebanyak 9260 review berbahasa Indonesia yang berasal dari platform *Shopee Indonesia*, pada kategori layanan digital yaitu produk akun premium *Netflix*. Data yang digunakan dalam penelitian ini mencakup kolom *review* dan *rating*, dengan tetap memperhatikan prinsip minimisasi data dan menjaga kerahasiaan informasi pengguna.

B. Preprocessing

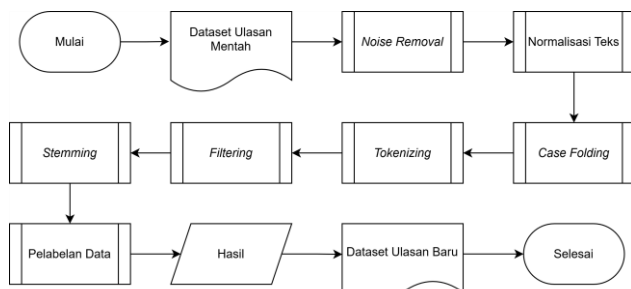
Data ulasan yang diperoleh dari platform *e-commerce* merupakan *unstructured data* yang tidak secara langsung dapat digunakan dalam proses pemodelan. Oleh karena itu, diperlukan tahapan *preprocessing* untuk mengolah data mentah menjadi format yang lebih terstruktur dan sesuai untuk dianalisis secara komputasional. *Preprocessing* mencakup beberapa proses penting seperti *case folding*, *tokenizing*, *filtering*, dan *stemming*. Selain itu, proses *cleaning* seperti penghapusan URL, emoji, dan karakter non-alfabet juga termasuk bagian penting dalam mengurangi *noise* pada data. Proses *text preprocessing* ini bertujuan untuk mengurangi kompleksitas data serta menyusun struktur teks yang dapat direpresentasikan secara numerik melalui metode seperti TF-IDF. Menurut [7], *preprocessing* berperan penting dalam meminimalkan *noise* dan memaksimalkan kualitas fitur, sehingga dapat meningkatkan akurasi model klasifikasi teks.

Selain tahap-tahap dasar seperti *case folding*, *tokenizing*, *filtering*, dan *stemming*, tahapan *preprocessing* dalam

penelitian ini dilanjutkan dengan penerapan metode *Latent Dirichlet Allocation* (LDA) untuk membantu proses pelabelan data. LDA memandang dokumen sebagai campuran topik, dimana setiap topik direpresentasikan oleh distribusi kata-kata yang saling berkaitan. Pada penelitian ini, LDA diterapkan untuk membentuk dua topik utama dari korpus ulasan Shopee. Setiap ulasan kemudian dipetakan ke salah satu topik yang memiliki probabilitas tertinggi.

Untuk menentukan kategori label *Computer Generated* (CG) dan *Original* (OR), dilakukan validasi secara manual oleh peneliti dengan menganalisis kata-kata dominan pada setiap topik yang dihasilkan oleh LDA. Berdasarkan observasi, topik pertama (Topik 0) didominasi oleh kata-kata yang bersifat generik, repetitif, dan umum seperti "mantap", "cepat", "fast", yang cenderung menggambarkan ulasan artifisial atau tidak personal. Oleh karena itu, topik ini dipetakan sebagai *Computer Generated* (CG). Sedangkan topik kedua (Topik 1) memuat kata-kata yang lebih variatif dan natural, mengindikasikan pengalaman pribadi seperti "barang sesuai", "harga murah", "pelayanan baik", sehingga dipetakan sebagai *Original* (OR). Dengan demikian, konversi dari hasil topik LDA menjadi label CG dan OR dilakukan secara objektif berdasarkan analisis pola linguistik kata-kata dominan dan validasi manual oleh peneliti.

Proses keseluruhan tahapan *preprocessing* dan pelabelan topik menggunakan LDA dapat dilihat pada Gambar 2.



Gambar 2. *Preprocessing*

C. Pembobotan Kata TF-IDF

Pentingnya sebuah frasa (*term*) dalam suatu dokumen dibandingkan dengan keseluruhan kumpulan dokumen (*corpus*) dinilai menggunakan pendekatan *Term Frequency-Inverse Document Frequency* (TF-IDF). TF-IDF bekerja dengan menggabungkan dua aspek utama, yaitu jumlah kemunculan suatu istilah dalam satu dokumen (*term frequency*) dan tingkat kekhususan term tersebut terhadap seluruh dokumen dalam korpus (*inverse document frequency*). Nilai bobot TF-IDF suatu kata meningkat seiring dengan frekuensi kemunculannya dalam suatu dokumen dan menurun seiring dengan frekuensi kemunculannya kembali dalam dokumen lain [8]. Rumus perhitungan TF-IDF dapat dilihat pada persamaan (1) [9].

$$W_{ij} = tf_{ij} \times \log\left(\frac{D}{df_j}\right)$$

Dimana W_{ij} merupakan bobot *term j* dalam dokumen *i*. tf_{ij} adalah berapa kali istilah *j* muncul dalam dokumen *i*.

Simbol D merepresentasikan jumlah keseluruhan dokumen dalam korpus. Dan df_j mengacu pada banyaknya dokumen yang mengandung *term ke-j* tersebut.

D. Implementasi

1. Random Forest

Random Forest merupakan salah satu metode *ensemble learning* yang cukup populer dalam klasifikasi maupun regresi. Algoritma ini dikembangkan sebagai penggabungan dari berbagai pohon keputusan (*decision tree*) yang dilatih pada subset data yang berbeda, dengan tujuan untuk meningkatkan akurasi prediksi dan mengurangi risiko *overfitting* yang sering terjadi pada model pohon tunggal [10].

Prinsip kerja *Random Forest* yaitu membangun banyak pohon keputusan selama proses pelatihan, kemudian menggabungkan hasil klasifikasi dari masing-masing pohon menggunakan voting mayoritas (untuk klasifikasi) atau rata-rata (untuk regresi) untuk menghasilkan keputusan akhir [11]. Metode ini efektif dalam menangani data dengan jumlah fitur yang besar dan mampu memberikan estimasi pentingnya variabel dalam klasifikasi.

Algoritma *Random Forest* memiliki keunggulan dalam menangani data dengan jumlah fitur yang besar serta mampu mengatasi masalah data yang tidak seimbang antar kelas. Hal ini disebabkan oleh penerapan metode *bootstrap aggregating* (*bagging*) dalam *Random Forest*, di mana setiap pohon keputusan dilatih pada subset data acak dan subset fitur acak yang berbeda, sehingga membuat model lebih robust. Selain itu, *Random Forest* juga mampu memberikan estimasi pentingnya setiap fitur (*feature importance*), yang berguna dalam proses seleksi atribut pada data berukuran besar. Proses pelatihan *Random Forest* dimulai dengan pembentukan sejumlah pohon keputusan dari subset acak data latih, dan hasil akhir klasifikasi diperoleh berdasarkan mayoritas prediksi dari semua pohon dalam hutan [12].

2. Support Vector Machine

Support Vector Machine (SVM) merupakan algoritma klasifikasi yang dirancang untuk menghasilkan batas pemisah (*hyperplane*) paling tepat antara dua kelas secara optimal. Garis pemisah ini ditentukan dengan mempertimbangkan margin, yaitu jarak antara *hyperplane* terhadap data dari masing-masing kelas yang paling dekat, yang disebut sebagai *support vector*. Semakin lebar margin antara dua kelas, semakin tinggi pula potensi model dalam mengenali pola dari data yang belum pernah dilatih [13].

SVM bekerja dengan membentuk sebuah *decision boundary* dalam bentuk garis (untuk dua dimensi) atau bidang (untuk dimensi yang lebih tinggi) yang berusaha memaksimalkan margin antar kelas. Jika data tidak dapat dipisahkan secara linear, SVM dapat menggunakan teknik *kernel trick* yang memungkinkan transformasi data ke

dalam ruang berdimensi lebih tinggi agar data dapat dipisahkan secara linear dalam ruang tersebut.

Keunggulan utama dari SVM adalah kemampuannya dalam menangani data berdimensi tinggi dan tetap efektif pada jumlah data yang relatif sedikit. Selain itu, SVM juga memiliki ketahanan terhadap overfitting terutama jika parameter yang digunakan telah disetel dengan baik. Namun, kelemahannya terletak pada proses pelatihan yang bisa menjadi lambat ketika dihadapkan dengan dataset yang sangat besar, serta sensitivitas terhadap pemilihan parameter dan jenis kernel yang digunakan.

E. Evaluasi

Tahap evaluasi bertujuan untuk mengukur seberapa baik kinerja model yang telah dibangun dalam mengklasifikasikan data. Data dibagi menggunakan metode *train-test split* dengan proporsi 80% data untuk pelatihan dan 20% data untuk pengujian. Performa model dievaluasi menggunakan *confusion matrix* untuk mengukur *accuracy*, *precision*, *recall*, dan *F1-score* [14].

III. HASIL DAN PEMBAHASAN

Bab ini menyajikan hasil dan analisis dari setiap tahapan yang telah dijelaskan pada bab metodologi penelitian.

A. Hasil Pengumpulan Data

Dataset yang digunakan dalam penelitian ini merupakan data sekunder yang diperoleh dari repositori GitHub, dengan total sebanyak 9260 ulasan (*review*) berbahasa Indonesia. Dataset tersebut berisi ulasan pembeli terhadap produk akun premium Netflix yang dijual di *platform* Shopee Indonesia, pada kategori layanan digital. Dataset dikumpulkan oleh kontributor independen menggunakan metode *web scraping*, dan telah didokumentasikan melalui *platform* Medium serta dibagikan secara terbuka melalui GitHub.

Untuk menjaga prinsip etika pengumpulan data, penelitian ini menerapkan prinsip minimisasi data, yaitu hanya menggunakan atribut yang relevan dengan tujuan analisis, tanpa mengeksploitasi informasi pribadi pengguna. Validitas dataset dipastikan dengan melakukan pemeriksaan manual terhadap sampel data, yang menunjukkan bahwa ulasan sesuai dengan konteks produk layanan digital (akun Netflix) di Shopee Indonesia. Penelitian ini juga menghormati hak kepemilikan data sebagaimana tercantum dalam dokumentasi repositori.

Pada tahap pengumpulan data, dilakukan seleksi awal terhadap kelengkapan data. Ditemukan bahwa sebagian entri memiliki nilai kosong pada atribut penting, seperti kolom *review* dan *rating*. Oleh karena itu, data yang tidak memenuhi kelengkapan atribut tersebut dihapus dari dataset. Setelah proses penyaringan, jumlah data yang digunakan dalam penelitian ini tersisa sebanyak 3686 data ulasan. Namun, pada saat implementasi algoritma klasifikasi, data tersebut dibagi menjadi tiga bagian berdasarkan fungsi *train_test_split* dan skema pembagian data, jumlah data meningkat menjadi 3894.

Perbedaan jumlah ini muncul sebagai akibat dari proses rekonstruksi data selama *pipeline* pemodelan, seperti penyesuaian ulang indeks dan penanganan data duplikat atau *null* saat tahap persiapan akhir, yang menyebabkan jumlah data sedikit berbeda dibanding saat awal pembersihan. Dataset telah disusun dalam format CSV dan mencakup beberapa atribut, seperti nama pembeli, *rating*, nama produk, dan isi ulasan. Contoh isi dari dataset ditampilkan melalui Tabel 1 berikut.

TABEL 1
SAMPEL HASIL PENGUMPULAN DATA

No	Nama	Rating	Review	Nama Produk
1	partianadewi	5.0	work tp proses lama	[Legal] Netflix Premium 1 Tahun
2	arvie2003	5.0	Respon cepat, work, dan proses agak lama	[Legal] Netflix Premium 1 Tahun
3	rama171200	5.0	Mantull Work	[Legal] Netflix Premium 1 Tahun
4	gilang_pras65	5.0	Sangat memuaskan belanja di sini, akun original indonesia, harga murah, penjual respon cepet	Netflix premium lifetime
...
3686	c*****d	5.0	Recommended seller 🍎🍎🍎 🍎 proses cepat gausah tunggu lama langsung jadi	Nord VPN 3 tahun

B. Preprocessing

Setelah tahap pengumpulan data, langkah selanjutnya adalah melakukan tahapan *preprocessing*. Proses ini penting dilakukan karena dataset yang digunakan tergolong sebagai data tidak terstruktur. Adapun beberapa tahapan *preprocessing* yang diterapkan dalam penelitian ini dijelaskan sebagai berikut.

1. Noise Removal

Tahapan ini dilakukan untuk membersihkan elemen-elemen yang tidak relevan dengan teks ulasan, seperti tautan (URL), simbol, serta emotikon yang sering muncul dalam

bentuk emoji dengan cara diubah menjadi representasi teks (*demojize*). Tujuan dari tahap ini adalah untuk menghilangkan gangguan (*noise*) dalam data teks agar hasil pemrosesan menjadi lebih bersih dan akurat. Berikut merupakan contoh hasil dari *noise removal* pada Tabel 2.

TABEL II
NOISE REMOVAL

No	Sebelum Removal	Sesudah Removal
1	work tp proses lama	work tp proses lama
2	Respon cepat, work, dan proses agak lama	Respon cepat, work, dan proses agak lama
3	Mantull Work	Mantull work
4	Sangat memuaskan belanja di sini, akun original indonesia, harga murah, penjual respon cepat	Sangat memuaskan belanja di sini, akun original indonesia, harga murah, penjual respon cepat
...
3686	Recommended seller 🍏🍏🍏🍏🍏proses cepat gausah tunggu lama langsung jadi	Recommended seller red apple red apple red apple red apple red apple proses cepat gausah tunggu lama langsung jadi

2. Normalisasi Teks

Normalisasi teks adalah proses mengubah kata-kata tidak baku, singkatan, atau variasi penulisan yang sering muncul dalam teks ulasan menjadi bentuk kata baku atau bentuk dasar yang lebih standar. Tujuannya adalah untuk menyamakan kata-kata yang memiliki makna sama agar dapat diproses lebih konsisten di tahap selanjutnya seperti tokenisasi dan stemming. Proses ini penting karena dataset yang digunakan merupakan ulasan dari pengguna yang cenderung menggunakan bahasa informal dan singkatan. Contoh *output* dari proses ini disajikan dalam Tabel 3.

TABEL III
NORMALISASI TEKS

No	Review	Normalisasi Teks
1	work tp proses lama	berhasil tapi proses lama
2	Respon cepat, work, dan proses agak lama	Respon cepat, berhasil, dan proses agak lama
3	Mantull Work	Mantap berhasil
4	Sangat memuaskan belanja di sini, akun original indonesia, harga murah, penjual respon cepat	Sangat memuaskan belanja di sini, akun original indonesia, harga murah, penjual respon cepat
...
3686	Recommended seller 🍏🍏🍏🍏🍏proses cepat gausah tunggu lama langsung jadi	Recommended seller red apple red apple red apple red apple red apple proses cepat tidak tunggu lama langsung jadi

3. Case Folding

Case folding merujuk pada langkah normalisasi di mana setiap karakter dalam teks, baik huruf besar maupun huruf

kecil, diubah menjadi huruf kecil semua. Tahapan ini bertujuan untuk menyeragamkan penulisan agar analisis teks menjadi lebih konsisten. Tabel 4 menampilkan contoh hasil yang diperoleh dari tahap ini.

TABEL IV
CASE FOLDING

No	Review	Case Folding
1	work tp proses lama	berhasil tapi proses lama
2	Respon cepat, work, dan proses agak lama	respon cepat, berhasil, dan proses agak lama
3	Mantull Work	mantap berhasil
4	Sangat memuaskan belanja di sini, akun original indonesia, harga murah, penjual respon cepat	sangat memuaskan belanja di sini, akun original indonesia, harga murah, penjual respon cepat
...
3686	Recommended seller 🍏🍏🍏🍏🍏proses cepat gausah tunggu lama langsung jadi	recommended seller red apple red apple red apple red apple proses cepat tidak tunggu lama langsung jadi

4. Tokenizing

Tokenizing adalah proses memecah teks menjadi potongan-potongan kecil yang disebut *token*, biasanya dalam bentuk kata per kata. Setelah dilakukan tokenizing, setiap kalimat dalam dataset akan diubah menjadi kumpulan kata yang berdiri sendiri. Hasil proses *tokenizing* pada Tabel 5.

TABEL V
TOKENIZING

No	Review	Tokenizing
1	work tp proses lama	['berhasil', 'tapi', 'proses', 'lama']
2	Respon cepat, work, dan proses agak lama	['respon', 'cepat', 'berhasil', 'dan', 'proses', 'agak', 'lama']
3	Mantull Work	['mantap', 'berhasil']
4	Sangat memuaskan belanja di sini, akun original indonesia, harga murah, penjual respon cepat	['sangat', 'memuaskan', 'belanja', 'di', 'sini', 'akun', 'original', 'indonesia', 'harga', 'murah', 'penjual', 'respon', 'cepat']
...
3686	Recommended seller 🍏🍏🍏🍏🍏proses cepat gausah tunggu lama langsung jadi	['recommended', 'seller', 'red', 'apple', 'red', 'apple', 'red', 'apple', 'red', 'apple', 'proses', 'cepat', 'tidak', 'tunggu', 'lama', 'langsung', 'jadi']

5. Filtering

Filtering merupakan proses penyaringan kata-kata yang tidak memiliki makna penting, seperti kata hubung, kata depan, atau kata yang sering muncul namun tidak

memberikan informasi signifikan. Pada tahap ini, kata-kata umum tersebut atau yang biasa disebut *stopword* akan dihapus dari dataset. Proses *filtering* menghasilkan *output* yang disajikan dalam Tabel 6.

TABEL VI
FILTERING

No	Sebelum Filtering	Sesudah Filtering
1	work tp proses lama	['berhasil', 'proses', 'lama']
2	Respon cepat, work, dan proses agak lama	['respon', 'cepat', 'berhasil', 'proses', 'agak', 'lama']
3	Mantull Work	['mantap', 'berhasil']
4	Sangat memuaskan belanja di sini, akun original indonesia, harga murah, penjual respon cepat	['sangat', 'memuaskan', 'belanja', 'akun', 'original', 'indonesia', 'harga', 'murah', 'penjual', 'respon', 'cepat']
...
3686	Recommended seller 🍏🍏🍏🍏🍏 proses cepat gausah tunggu lama langsung jadi	['recommended', 'seller', 'red', 'apple', 'red', 'apple', 'red', 'apple', 'red', 'apple', 'red', 'proses', 'cepat', 'tidak', 'tunggu', 'lama', 'langsung', 'jadi']

6. Stemming

Stemming adalah tahapan yang bertujuan untuk mengubah kata berimbuhan menjadi bentuk kata dasar. Tahapan ini dilakukan untuk menyederhanakan variasi kata agar memiliki representasi yang seragam. *Library* Sastrawi yang dirancang khusus untuk Bahasa Indonesia digunakan untuk melakukan prosedur *stemming* dalam penelitian ini. Contoh hasil *stemming* dapat dilihat pada Tabel 7.

TABEL VII
STEMMING

No	Sebelum Stemming	Sesudah Stemming
1	work tp proses lama	['hasil', 'proses', 'lama']
2	Respon cepat, work, dan proses agak lama	['respon', 'cepat', 'hasil', 'proses', 'agak', 'lama']
3	Mantull Work	['mantap', 'hasil']
4	Sangat memuaskan belanja di sini, akun original indonesia, harga murah, penjual respon cepat	['sangat', 'puas', 'belanja', 'akun', 'original', 'indonesia', 'harga', 'murah', 'jual', 'respon', 'cepat']
...
3686	Recommended seller 🍏🍏🍏🍏🍏 proses cepat gausah tunggu lama langsung jadi	['recommended', 'seller', 'red', 'apple', 'red', 'apple', 'red', 'apple', 'red', 'apple', 'red', 'proses', 'cepat', 'tidak', 'tunggu', 'lama', 'langsung', 'jadi']

7. Pelabelan Data

Pelabelan data memiliki peran penting dalam membangun model klasifikasi karena ketepatan label sangat berpengaruh terhadap hasil kinerja model. Pada penelitian ini, proses pemberian label dilakukan dengan memanfaatkan pendekatan *Latent Dirichlet Allocation* (LDA). LDA merupakan teknik pemodelan topik yang bertujuan mengungkap pola tersirat dalam kumpulan dokumen, dengan cara menganalisis sebaran kata yang muncul di dalamnya. Metode ini beroperasi berdasarkan anggapan bahwa sebuah dokumen terdiri atas campuran berbagai topik, di mana masing-masing topik dibentuk oleh kumpulan kata dengan probabilitas tertentu. Oleh karena itu, LDA mampu memetakan dokumen ke dalam topik-topik yang paling dominan secara otomatis tanpa memerlukan label manual.

Berdasarkan hasil penerapan LDA pada dataset, diperoleh dua kategori utama, yaitu *Computer Generated* (CG) sebanyak 1990 ulasan (53,98%) dan *Original* (OR) sebanyak 1696 ulasan (46,02%), dari total 3686 data ulasan. Setiap ulasan dipetakan ke topik dengan probabilitas tertinggi, kemudian dilakukan validasi manual berdasarkan analisis kata-kata dominan dari masing-masing topik untuk memastikan koherensi antara isi topik dan label yang diberikan.

Distribusi label menunjukkan bahwa proporsi antara kedua kategori relatif seimbang, dengan selisih sekitar 7,96%. Meskipun tingkat ketidakseimbangan tidak terlalu signifikan, untuk memastikan model tidak bias terhadap kelas mayoritas, dilakukan penanganan ketidakseimbangan data menggunakan teknik oversampling SMOTE (*Synthetic Minority Over-sampling Technique*) pada data latih.

Untuk memberikan gambaran lebih lanjut mengenai distribusi teks, dilakukan analisis deskriptif terhadap jumlah karakter pada masing-masing kategori label. Gambar 3 menunjukkan ringkasan statistik deskriptif tersebut, yang mencakup jumlah data, nilai rata-rata, simpangan baku, nilai minimum, kuartil, hingga nilai maksimum dari masing-masing kelompok.

	count	mean	std	min	25%	50%	75%	max
label								
CG	1990.0	61.708543	55.009460	0.0	29.0	51.0	75.0	823.0
OR	1696.0	71.085495	68.177976	2.0	37.0	57.0	85.0	1499.0

Gambar 3. Distribusi Data Berlabel Menggunakan Metode LDA
Contoh data yang telah melalui proses pelabelan disajikan dalam Tabel 8.

TABEL VIII
PELABELAN DATA

No	Nama	Rating	Review	Label
1	partianadewi	5.0	work tp proses lama	CG
2	arvie2003	5.0	Respon cepat, work, dan proses agak lama	OR
3	rama171200	5.0	Mantull Work	OR
4	gilang_pras65	5.0	Sangat memuaskan	OR

			belanja di sini, akun original indonesia, harga murah, penjual respon cepet	
...
3686	c*****d	5.0	Recommended seller 🍏🍏🍏🍏🍏 roses cepat gausah tunggu lama langsung jadi	CG

C. Pembobotan Kata TF-IDF

Pembobotan kata dilakukan untuk menentukan seberapa besar kontribusi sebuah kata terhadap keseluruhan isi dokumen. Metode yang digunakan adalah TF-IDF (*Term Frequency-Inverse Document Frequency*), yang memberikan bobot berdasarkan frekuensi kemunculan kata di suatu dokumen dibandingkan dengan seluruh dokumen lain.

Term Frequency (TF) menggambarkan frekuensi kemunculan suatu kata muncul dalam sebuah dokumen, sementara *Inverse Document Frequency* (IDF) mengukur tingkat keunikan kata tersebut di seluruh kumpulan dokumen. Kata yang sering muncul namun jarang ditemukan di dokumen lain akan memiliki nilai TF-IDF yang tinggi, sehingga dianggap lebih penting. Bobot TF-IDF ini kemudian digunakan sebagai representasi numerik dari teks yang akan diproses lebih lanjut pada tahap klasifikasi. Pada tahap ini, TF-IDF diterapkan dengan batas maksimum 5000 fitur, dan diperoleh sebanyak 4265 fitur unik setelah proses vektorisasi. Di bawah ini adalah contoh hasil perhitungan data ulasan dengan menggunakan metode TF-IDF yang disajikan dalam Tabel 9.

TABEL IX
SAMPEL DATA YANG DI VEKTORISASI

Review	TF-IDF					
	man tap	nont on	netflix	harg a	mur ah	makas ih
mantap nonton netflix harga murah makasih	0.33 56	0.45 27	0.3902	0.39 06	0.43 33	0.435 7

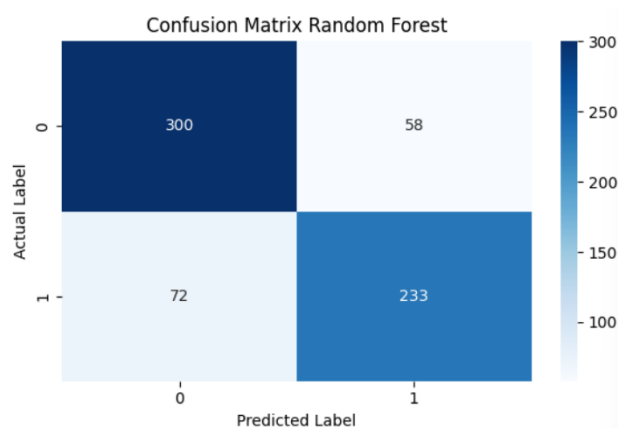
Tabel 9 memperlihatkan bagaimana setiap kata dalam ulasan diberikan bobot berdasarkan tingkat kepentingannya. Sementara itu, Tabel 10 menampilkan sepuluh kata dengan nilai TF-IDF tertinggi dalam dataset, yang merepresentasikan kata-kata paling dominan dan relevan dalam kumpulan ulasan. Sebagian besar kata tersebut mencerminkan kepuasan pelanggan serta konteks transaksi, seperti kecepatan layanan dan kualitas produk.

TABEL X
10 KATA DENGAN NILAI TF-IDF TERTINGGI

Kata	TF-IDF
seller	0.103701
cepat	0.095221
banget	0.091227
respon	0.086636
mantap	0.083630
moga	0.081723
akun	0.067664
beli	0.065343
fast	0.062911
ramah	0.055309

D. Hasil Implementasi Random Forest

Pada tahap ini, algoritma Random Forest digunakan untuk melakukan klasifikasi terhadap data ulasan Shopee yang telah diberi label. Fungsi *train_test_split* dari modul Scikit-learn digunakan untuk membagi dataset menjadi data latih dan uji dalam rasio 80:20, dengan parameter *stratify* untuk memastikan distribusi label yang seimbang pada kedua subset data. Representasi fitur dibuat menggunakan *TfidfVectorizer* dari *library* Sklearn dengan jumlah maksimum fitur sebanyak 5.000. Untuk mengatasi ketidakseimbangan kelas, digunakan metode *oversampling* SMOTE dari *library* Imblearn. Selanjutnya, model Random Forest dikonstruksi dengan menggunakan parameter *n_estimators* = 200, *max_depth* = 10, dan *min_samples_leaf* = 4. Model dilatih menggunakan data latih, kemudian digunakan untuk memprediksi label ada data uji. Evaluasi model dilakukan dengan memanfaatkan fungsi *accuracy_score* dan *classification_report* dari modul Sklearn, dan hasil evaluasi ditampilkan pada layar.



Gambar 4. Confusion Matrix RF

Berdasarkan hasil pengujian, didapatkan akurasi sebesar 80,39% pada *data testing*. Visualisasi hasil prediksi model juga disajikan dalam bentuk *confusion matrix* pada Gambar 4. *Confusion matrix* ini menggambarkan jumlah prediksi yang benar maupun salah dari model terhadap masing-masing kelas. Visualisasi tersebut membantu memahami kesalahan klasifikasi dan distribusi prediksi model secara lebih intuitif. Hasil *confusion matrix* menunjukkan bahwa model

menghasilkan *true positive* (prediksi yang benar terhadap kelas positif) sebanyak 233, *false positive* (prediksi positif terhadap data yang seharusnya negatif) sebanyak 58, *true negative* (prediksi yang benar terhadap kelas negatif) sebanyak 300, dan *false negative* (prediksi negatif terhadap data yang seharusnya positif) sebanyak 72.

Setelah model dilatih menggunakan algoritma Random Forest, dilakukan proses prediksi terhadap 369 data baru yang tidak memiliki label. Pada Gambar 5, ditampilkan hasil deskriptif statistik terhadap *rating* yang dikelompokkan berdasarkan label hasil prediksi. Hasil prediksi pada 369 data uji menunjukkan 214 data yang dikategorikan sebagai CG dan 155 data yang dikategorikan sebagai OR.

	count	mean	std	min	25%	50%	75%	max
label								
CG	214.0	4.948598	0.402075	1.0	5.0	5.0	5.0	5.0
OR	155.0	4.593548	1.166061	1.0	5.0	5.0	5.0	5.0

Gambar 5. Hasil Prediksi Menggunakan RF

Contoh hasil prediksi disajikan dalam Tabel 11.

TABEL XI
HASIL PREDIKSI DATASET BARU

No	Nama	Rating	Review	Label
1	edgardkiddrock	5.0	Cepat respon dan slalu tepat waktu... Mantap jaya....	CG
2	a*****c	5.0	Respon cepat, kualitas sangat baikkk, harga produk baikk	CG
3	maharanitya shapsari	5.0	Makasih banyak raja netflix. Selalu repeat order disini. Pelayanan cepat dan harha terjangkau. Makasih banyak ya	CG
4	purple.merch	5.0	Good 🙌 prosesnya cepat. Thank you	CG
...
369	imrizkia19	5.0	Dalam beberapa hari dipake gak ada kendala. Proses cepet gak pake lama. Mantap 🙌	OR

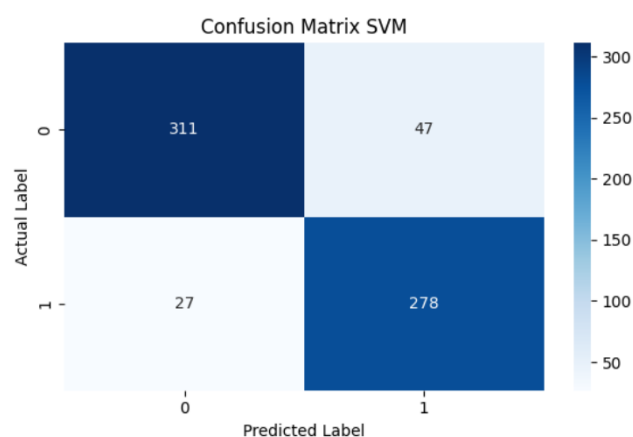
E. Hasil Implementasi SVM

Fungsi *train_test_split* dari *library* Sklearn digunakan dalam implementasi ini untuk membagi dataset menjadi data latih dan data uji dalam rasio 80:20. Ekstraksi fitur dilakukan dengan menggunakan *TfidfVectorizer* untuk mengonversi data teks menjadi representasi numerik yang dapat diproses oleh model. Untuk menangani ketidakseimbangan kelas dalam data latih, digunakan metode *oversampling* SMOTE

(*Synthetic Minority Over-sampling Technique*). Model kemudian dilatih menggunakan algoritma *Support Vector Machine* (SVM) dengan *kernel linear* pada data *training*.

Setelah model dilatih, data uji digunakan untuk menilai kinerja model. Perhitungan akurasi, *recall*, *precision*, *f1-score*, dan *confusion matrix* dilakukan melalui fungsi evaluasi yang disediakan oleh *library* Sklearn. Hasil evaluasi tersebut kemudian ditampilkan ke layar untuk dianalisis lebih lanjut.

Berdasarkan hasil pengujian, model SVM memperoleh akurasi sebesar 88,84% pada data uji. *Confusion matrix* ditampilkan untuk membantu memahami distribusi klasifikasi yang dilakukan oleh model, baik yang tepat maupun yang keliru. Dari Gambar 6 ditunjukkan bahwa model berhasil memprediksi kelas negative secara benar sebanyak 311 (*True Negative*), memprediksi kelas positif secara benar sebanyak 278 (*True Positive*), melakukan kesalahan prediksi positif padahal seharusnya negative sebanyak 47 (*False Positive*), dan kesalahan prediksi negative padahal seharusnya positif sebanyak 27 (*False Negative*).



Gambar 6. Confusion Matrix SVM

Setelah model SVM selesai dilatih, tahap selanjutnya adalah menguji kemampuannya dalam mengklasifikasikan 369 data baru yang sebelumnya tidak memiliki label. Gambar 7 menampilkan hasil distribusi prediksi model terhadap data baru tersebut. Dari total 369 data yang diuji, sebanyak 194 data diprediksi sebagai CG (*Computer Generated*), sementara sisanya yaitu 175 data diprediksi sebagai OR (*Original*). Selain itu, dilakukan juga analisis statistik terhadap *rating* berdasarkan hasil prediksi label, yang memperlihatkan bahwa rata-rata *rating* untuk label CG sedikit lebih tinggi dibandingkan dengan label OR.

	count	mean	std	min	25%	50%	75%	max
label								
CG	194.0	4.963918	0.311668	1.0	5.0	5.0	5.0	5.0
OR	175.0	4.617143	1.138028	1.0	5.0	5.0	5.0	5.0

Gambar 7. Hasil Prediksi Menggunakan SVM

Contoh data hasil klasifikasi oleh model disajikan dalam Tabel 12.

TABEL XII
HASIL PREDIKSI DATASET BARU

No	Nama	Rating	Review	Label
1	edgardkiddrock	5.0	Cepat respon dan slalu tepat waktu... Mantap jaya....	CG
2	a*****c	5.0	Respon cepat, kualitas sangat baikkk, harga produk baikk	CG
3	maharanitya shapsari	5.0	Makasih banyak raja netflix. Selalu repeat order disini. Pelayanan cepat dan harha terjangkau. Makasih banyak ya	CG
4	purple.merch	5.0	Good 🙌 prosesnya cepat. Thank you	CG
...
369	imrizkia19	5.0	Dalam beberapa hari dipake gak ada kendala. Proses cepet gak pake lama. Mantap 🙌	OR

F. Evaluasi

Performa model yang dibangun menggunakan data latih diukur pada tahap evaluasi ini. Evaluasi dilakukan dengan menghitung metrik seperti *precision*, *recall*, *f1-score*, serta *accuracy* menggunakan *confusion matrix* sebagai dasar perhitungan. Perbandingan juga dilakukan terhadap hasil dari dua dataset berbeda, yaitu data latih dan data uji, untuk melihat konsistensi performa model. Model Random Forest yang digunakan telah dilatih dengan total 3525 data, yang terdiri atas 2682 data untuk pelatihan dan 663 data untuk pengujian. Nilai-nilai evaluasi tersebut ditampilkan pada Gambar 8. Berikut ini rumus untuk menghitung nilai akurasi dituliskan dalam persamaan (2).

$$Accuracy = \frac{233 + 300}{233 + 300 + 58 + 72} \times 100\% = 80,39\%$$

Berikut persamaan (3) untuk mencari nilai *precision*.

$$Precision = \frac{(0,81 \times 358) + (0,80 \times 305)}{663} \times 100\% = 80,54\%$$

Berikut persamaan (4) untuk mencari nilai *recall*.

$$Recall = \frac{(0,84 \times 358) + (0,76 \times 305)}{663} \times 100\% = 80,32\%$$

Berikut persamaan (5) untuk mencari nilai *f1-score*.

$$f1 - score = \frac{(0,82 \times 358) + (0,78 \times 305)}{663} \times 100\% = 80,16\%$$

Test Report:

	precision	recall	f1-score	support
0	0.81	0.84	0.82	358
1	0.80	0.76	0.78	305
accuracy			0.80	663
macro avg	0.80	0.80	0.80	663
weighted avg	0.80	0.80	0.80	663

Gambar 8. Hasil *Precision*, *Recall*, dan *F1-score* Random Forest

Sementara itu, implementasi algoritma SVM dilakukan dengan menggunakan sebanyak 3525 data, yang terdiri dari 2862 data latih dan 663 data uji. Evaluasi performa model dilakukan dengan menghitung nilai akurasi, *precision*, *recall*, serta *f1-score*, sebagaimana ditampilkan pada Gambar 9. Adapun perhitungan nilai akurasi ditunjukkan melalui persamaan (6).

$$Accuracy = \frac{278 + 311}{278 + 311 + 47 + 27} \times 100\% = 88,84\%$$

Berikut persamaan (7) untuk mencari nilai *precision*.

$$Precision = \frac{(0,92 \times 358) + (0,86 \times 305)}{663} \times 100\% = 89,24\%$$

Berikut persamaan (8) untuk mencari nilai *recall*.

$$Recall = \frac{(0,87 \times 358) + (0,91 \times 305)}{663} \times 100\% = 88,84\%$$

Berikut persamaan (9) untuk mencari nilai *f1-score*.

$$f1 - score = \frac{(0,89 \times 358) + (0,88 \times 305)}{663} \times 100\% = 88,54\%$$

Test Report:

	precision	recall	f1-score	support
0	0.92	0.87	0.89	358
1	0.86	0.91	0.88	305
accuracy			0.89	663
macro avg	0.89	0.89	0.89	663
weighted avg	0.89	0.89	0.89	663

Gambar 9. Hasil *Precision*, *Recall*, dan *F1-score* SVM

Ketiga metrik tersebut dihitung oleh *library classification_report* menggunakan pendekatan *weighted average*, yaitu mempertimbangkan jumlah data (*support*) dari masing-masing label sehingga hasilnya merepresentasikan performa keseluruhan model secara proporsional.

Sebagai catatan, meskipun dilakukan perhitungan ulang secara manual menggunakan rumus *weighted average* yang sama, hasil akhir bisa tampak sedikit berbeda dibandingkan yang ditampilkan pada gambar. Hal ini dapat terjadi karena *library* menggunakan angka desimal dengan banyak digit di belakang koma dalam proses perhitungan, sementara perhitungan manual biasanya menggunakan angka yang sudah dibulatkan. Perbedaan pembulatan ini menyebabkan sedikit selisih pada hasil akhir.

Secara umum, hasil evaluasi menunjukkan bahwa algoritma SVM memiliki performa yang lebih baik dibandingkan dengan Random Forest. Nilai akurasi SVM sebesar 88,84% lebih tinggi dibandingkan akurasi Random Forest sebesar 80,39%. Salah satu faktor yang menyebabkan SVM mengungguli Random Forest adalah kemampuannya dalam menangani data berdimensi tinggi. Pada kasus ini, penggunaan fitur TF-IDF menghasilkan representasi data dengan dimensi yang sangat besar, karena setiap kata unik pada dataset menjadi satu fitur tersendiri. SVM dikenal efektif untuk klasifikasi pada data berdimensi tinggi karena memaksimalkan margin antar kelas, sehingga lebih mampu menangani *sparsity* dan *noise* pada data teks. Sebaliknya, Random Forest lebih sensitif terhadap *noise* pada fitur-fitur yang tidak terlalu informatif, yang menyebabkan overfitting pada beberapa data latih dan menurunkan performa pada data uji.

Berdasarkan hasil analisis bobot fitur dari model SVM, 10 kata kunci TF-IDF yang paling berpengaruh dalam klasifikasi disajikan pada Tabel 13.

TABEL XIII
10 FITUR TF-IDF BERPENGARUH DI SVM

Kata	TF-IDF
jiwa	2.4752
hand	2.0969
tunggu	2.0673
oke	2.0066
responsif	1.9697
email	1.8606
murah	1.8549
mantab	1.8144
light	1.7952
ganti	1.7684

Kata-kata seperti “jiwa”, “responsif”, “oke”, dan “mantab” mengindikasikan kecenderungan pada ulasan yang positif dan asli (OR), sedangkan istilah seperti “murah” dan “email” dapat berasosiasi dengan pola ulasan buatan (CG) yang sering menekankan transaksi atau proses teknis. Fitur-fitur ini berkontribusi besar dalam membedakan antara dua kelas tersebut.

Selain itu, karakteristik ulasan pada Shopee Indonesia menunjukkan penggunaan gaya bahasa yang cenderung santai dan kasual, dengan banyaknya penggunaan kata-kata informal seperti “mantab”, “oke”, dan “responsif”. Struktur ulasan umumnya singkat, langsung pada inti pengalaman pengguna, serta sering kali disertai emotikon untuk mengekspresikan emosi. Selain gaya santai, ulasan di Shopee Indonesia juga sering kali menilai positif berdasarkan kecepatan pengiriman, tanpa banyak membahas kualitas produk. Adanya pola penggunaan pengulangan kata seperti “bagus bagus bagus” atau “cepat cepat cepat”, serta ulasan yang hanya berisi emotikon tanpa teks, turut membentuk karakteristik khusus pada data ulasan Shopee.

Untuk memberikan insight lebih dalam terhadap performa model, dilakukan analisis terhadap kasus-kasus salah

klasifikasi. Berikut adalah beberapa contoh review yang diklasifikasikan secara salah, disajikan dalam Tabel 14.

TABEL XIV
CONTOH SALAH KLASIFIKASI PADA MODEL SVM

Review	Label Asli	Prediksi Model
makasih kak mantep banget deh dikasihtau sabar...	OR	CG
paket pertama beli cepat tanggap akurat bagus ...	CG	OR
thankyou barang sampe selamat moga awet ya	OR	CG

Dari contoh tersebut, terlihat bahwa model kadang keliru mengklasifikasikan ulasan positif pendek sebagai ulasan buatan, atau sebaliknya. Hal ini menunjukkan bahwa pola bahasa sederhana dan ekspresi umum masih menjadi tantangan bagi model untuk membedakan ulasan asli dan palsu secara akurat. Oleh karena itu, penggunaan fitur berbasis sintaksis atau struktur kalimat tambahan mungkin diperlukan untuk meningkatkan akurasi model dalam kasus serupa.

IV. KESIMPULAN

Penelitian ini bertujuan untuk mendeteksi ulasan palsu pada platform Shopee menggunakan dua algoritma *machine learning*, yaitu Random Forest dan *Support Vector Machine* (SVM). Dataset awal berjumlah 3894 ulasan, lalu dilakukan *preprocessing* hingga menghasilkan data bersih yang dibagi menjadi tiga bagian: 80% data latih (2862 data), 20% data uji (663 data), dan 10% data baru tanpa label (369 data) yang digunakan untuk prediksi akhir.

Dalam tahap pengujian terhadap data baru menggunakan model Random Forest yang telah dilatih, diperoleh prediksi sebanyak 165 data sebagai ulasan asli (*Original*) dan 204 data lainnya sebagai buatan (*Computer Generated*). Sementara itu, pada pengujian menggunakan model SVM, hasil prediksi menunjukkan 170 data diklasifikasikan sebagai OR dan 199 sebagai CG.

Pada proses evaluasi performa model, Random Forest menghasilkan akurasi sebesar 81,48% dengan *precision* 81,24%, *recall* 81,46%, dan *f1-score* sebesar 81,62%. Di sisi lain, model SVM menunjukkan performa yang lebih unggul, dengan akurasi mencapai 89,01%, *precision* 88,70%, *recall* 88,92%, dan *f1-score* sebesar 89,08%. Berdasarkan hasil yang diperoleh, dapat dikatakan bahwa model SVM mengungguli Random Forest dalam mengidentifikasi ulasan palsu pada dataset ini.

Adapun nilai *precision*, *recall*, dan *f1-score* dihitung menggunakan metode *weighted average*, yang mempertimbangkan proporsi masing-masing label. Perbedaan hasil manual dengan hasil dari *library* bisa terjadi karena pembulatan dan tingkat presisi angka).

Penelitian ini memberikan kontribusi dalam penerapan metode pelabelan otomatis berbasis LDA untuk deteksi ulasan palsu di *e-commerce* lokal, serta menunjukkan

efektivitas model SVM dalam menangani teks berdimensi tinggi. Penelitian selanjutnya dapat diarahkan pada pengembangan metode dengan dataset yang lebih luas dan pendekatan *deep learning* untuk meningkatkan akurasi klasifikasi.

DAFTAR PUSTAKA

- [1] Z. Hadi, M. Zulpahmi, . Z., and A. Asrory, "Detecting Fake Reviews Using BERT and Sublinear_TF Methods on Hotel Reviews in the Lombok Tourism Area," *J. Appl. Informatics Comput.*, vol. 8, no. 2, pp. 550–556, Nov. 2024, doi: 10.30871/jaic.v8i2.8721.
- [2] K. Mane, S. Dongre, and M. Madankar, "Fake Review Detection using Random Forest Classifier," in *2025 IEEE International Students' Conference on Electrical, Electronics and Computer Science (SCEECs)*, IEEE, Jan. 2025, pp. 1–6. doi: 10.1109/SCEECs64059.2025.10940605.
- [3] O. Singh, S. Singh, S. Rawat, and S. Nirati, "Fake Reviews Identification Using Deep Learning Techniques," vol. 8, no. 2, pp. 820–827, 2025.
- [4] S. Zabeen, A. Hasan, M. F. Islam, M. S. Hossain, and A. A. Rasel, "Robust Fake Review Detection Using Uncertainty-Aware LSTM and BERT," in *2023 IEEE 15th International Conference on Computational Intelligence and Communication Networks (CICN)*, IEEE, Dec. 2023, pp. 786–791. doi: 10.1109/CICN59264.2023.10402342.
- [5] H. Alamsyah, Y. Cahyana, and A. R. Pratama, "Deteksi Fake Review Menggunakan Metode Support Vector Machine dan Naïve Bayes Di Tokopedia," *Jutisi J. Ilm. Tek. Inform. dan Sist. Inf.*, vol. 12, no. 2, p. 585, Aug. 2023, doi: 10.35889/jutisi.v12i2.1222.
- [6] Sugiyono, *Metode Penelitian Kuantitatif, Kualitatif Dan R&D*. 1967. [Online]. Available: https://www.academia.edu/118903676/Metode_Penelitian_Kuantitatif_Kualitatif_dan_R_and_D_Prof_Sugiono
- [7] K. Kowsari, K. Jafari Meimandi, M. Heidarysafa, S. Mendu, L. Barnes, and D. Brown, "Text Classification Algorithms: A Survey," *Information*, vol. 10, no. 4, p. 150, Apr. 2019, doi: 10.3390/info10040150.
- [8] M. Nurjannah and I. Fitri Astuti, "PENERAPAN ALGORITMA TERM FREQUENCY-INVERSE DOCUMENT FREQUENCY (TF-IDF) UNTUK TEXT MINING Mahasiswa S1 Program Studi Ilmu Komputer FMIPA Universitas Mulawarman Dosen Program Studi Ilmu Komputer FMIPA Universitas Mulawarman," *J. Inform. Mulawarman*, vol. 8, no. 3, pp. 110–113, 2013.
- [9] A. Supriatman, "Pembobotan TF-IDF pada Judul Penelitian Dosen Sebagai Dasar Klasifikasi Menggunakan Algoritma K-NN (Studi Kasus: Universitas Siliwangi)," *J. Serambi Eng.*, vol. 6, no. 1, pp. 1573–1579, 2021, doi: 10.32672/jse.v6i1.2645.
- [10] T. Posangi, L. Yahya, and D. Wungguli, "Implementasi Algoritma Random Forest dengan Forward Selection untuk Klasifikasi Indeks Pembangunan Manusia," *Jambura J. Probab. Stat.*, vol. 4, no. 2, pp. 85–91, 2023, doi: 10.37905/jjps.v4i2.18460.
- [11] I. Afdhal, R. Kurniawan, I. Iskandar, R. Salambue, E. Budianita, and F. Syafria, "Penerapan Algoritma Random Forest Untuk Analisis Sentimen Komentar Di YouTube Tentang Islamofobia," *J. Nas. Komputasi dan Teknol. Inf.*, vol. 5, no. 1, pp. 122–130, 2022, [Online]. Available: <http://ojs.serambimekkah.ac.id/jnkti/article/view/4004/pdf>
- [12] A. Ramadhan, B. Susetyo, and Indahwati, "Penerapan Metode Klasifikasi Random Forest Dalam Mengidentifikasi Faktor Penting Penilaian Mutu Pendidikan," *J. Pendidik. dan Kebud.*, vol. 4, no. 2, pp. 169–182, 2019, doi: 10.24832/jpnk.v4i2.1327.
- [13] K. Putri *et al.*, "Implementasi Algoritma Support Vector Machine dalam Klasifikasi Deteksi Depresi dari Postingan pada Media Sosial," *J. Nas. Teknol. Inf. dan Apl.*, vol. 2, no. 1, pp. 193–202, 2023.
- [14] W. A. Naseer, S. Sarwido, and B. B. Wahono, "Gradient Boosting Optimization with Pruning Technique for Prediction of Bmt Al-hikmah Permata Customer Data," *Jinteks*, vol. 6, no. 3, pp. 719–727, 2024.
- [15] K. Adib, M. R. Handayani, W. D. Yuniarti, and K. Umam, "Opini Publik Pasca-Pemilihan Presiden: Eksplorasi Analisis Sentimen Media Sosial X Menggunakan SVM," *SINTECH (Science Inf. Technol. J.)*, vol. 7, no. 2, pp. 80–91, 2024, doi: 10.31598/sintechjournal.v7i2.1581.
- [16] M. H. Aufan, M. R. Handayani, A. B. Nurjanna, and N. C. Hendro, "THE PERCEPTIONS OF SEMARANG FIVE STAR HOTEL TOURISTS WITH SUPPORT VECTOR MACHINE ON GOOGLE REVIEWS PERSEPSI WISATAWAN HOTEL BINTANG LIMA SEMARANG DENGAN," vol. 5, no. 5, pp. 1241–1247, 2025.
- [17] M. Apriliyani, M. I. Musyaffaq, S. Nur'Aini, M. R. Handayani, and K. Umam, "Implementasi analisis sentimen pada ulasan aplikasi Duolingo di Google Playstore menggunakan algoritma Naïve Bayes," *AITI*, vol. 21, no. 2, pp. 298–311, Sep. 2024, doi: 10.24246/aiti.v21i2.298-311.