ANALISIS SENTIMEN ULASAN PRODUK SKINCARE DI SHOPEE UNTUK MENINGKATKAN KUALITAS PRODUK MENGGUNAKAN METODE SUPPORT VECTOR MACHINE

Sebastian Ubaidillah Royan[1], Nana Suarna[2], Irfan Ali[3], Dodi Solihudin[4]
STMIK IKMI Cirebon[1234]
Jl. Perjuangan No. 10 B Majasem Kec. Kesambi Kota Cirebon
E-mail : sbastianroyan@gmail.com[1], st_nana@yahoo.com[2], irfanaali0.0@gmail.com[3], dodisolihudin@gmail.com[4]

ABSTRAK

Penelitian ini membahas analisis sentimen ulasan produk skincare di Shopee menggunakan algoritma *Support Vector Machine (SVM)*. Tahapan penelitian mencakup pengumpulan data ulasan, *pre-processing data (cleaning, tokenization, stemming* dan *stopword removal)*, pembobotan fitur menggunakan *Term Frequency-Inverse Document Frequency (TF-IDF)* serta pelatihan model *SVM*. Hasil evaluasi menunjukkan bahwa algoritma *Support Vector Machine* mencapai akurasi sebesar 93,79%. Untuk kategori sentimen *negatif*, *precision* mencapai 100% dengan *recall* sebesar 18% dan *F1-score* 31%, sedangkan untuk kategori sentimen *positif*, *precision* sebesar 94%, *recall* 100%, dan *F1-score* 97%. Nilai rata-rata terimbang untuk *precision*, *recall* dan *F1-score* masing-masing adalah 94%, 94%, dan 92%. Hasil penelitian ini menunjukkan bahwa algoritma *Support Vector Machine* mampu memberikan performa yang baik dalam analisis sentimen, khususnya pada kategori dominan dalam dataset meskipun memerlukan perbaikan dalam penanganan ketidakseimbangan data. Penelitian ini diharapkan dapat membantu pengambil keputusan dalam memahami preferensi konsumen dan meningkatkan kualitas produk.

Kata kunci : Analisis Sentimen, *Skincare, Support Vector Machine*

ABSTRACTS

This research discusses sentiment analysis of skincare product reviews on Shopee using the Support Vector Machine (SVM) algorithm. The research stages include review data collection, data pre-processing (cleaning, tokenization, stemming, and stopword removal), feature weighting using Term Frequency-Inverse Document Frequency (TF-IDF), and SVM model training. The evaluation results show that the SVM algorithm achieved an accuracy of 93.79%. For the negative sentiment category, precision reached 100% with recall of 18% and F1-score of 31%, while for the positive sentiment category, precision was 94%, recall 100%, and F1-score 97%. The weighted average values for precision, recall, and F1-score are 94%, 94%, and 92%, respectively. The results of this study show that the SVM algorithm is able to provide good performance in sentiment analysis, especially in the dominant category in the dataset, although it requires improvement in handling data imbalance. This research is expected to help decision makers in understanding consumer preferences and improving product quality.

Keywords: Sentiment Analysis, Skincare, Support Vector Machine

1. PENDAHULUAN

Kehadiran berbagai produk *skincare* di *e-commerce* seperti Shopee menimbulkan tantangan bagi konsumen dalam memilih produk yang berkualitas dan sesuai kebutuhan. Di sisi lain, produsen juga memerlukan metode yang efektif untuk memantau persepsi publik terkait produk mereka di tengah ketatnya persaingan pasar. Penelitian sebelumnya menunjukkan bahwa sentimen ulasan produk skincare memiliki peran signifikan dalam mempengaruhi keputusan pembelian. Namun, permasalahan yang muncul adalah kurangnya sistem otomatis yang mampu mengklasifikasikan sentimen dengan akurat pada

skala besar. Oleh karena itu penerapan algoritma *Support Vector Machine (SVM)* sebagai metode analisis sentimen diharapkan dapat menjawab tantangan ini dengan menyediakan data yang mendalam dan akurat bagi produsen dan konsumen.

Penelitian sebelumnya telah banyak dilakukan dalam bidang analisis sentimen menggunakan algoritma *SVM*. menunjukkan bahwa *SVM* memiliki kinerja yang baik dalam klasifikasi ulasan aplikasi dalam mengoptimalkan parameter SVM dengan algoritma genetika untuk meningkatkan akurasi analisis sentimen di media sosial [1]. Selain itu, studi oleh mengaplikasikan *SVM* pada ulasan produk skincare dengan hasil yang menunjukkan akurasi yang memuaskan, meski masih terbatas pada aspek tertentu seperti jenis produk dan jumlah data yang relatif kecil membandingkan *SVM* dengan metode lain untuk analisis sentimen *cyberbullying* di Twitter dan menemukan keunggulan SVM dalam aspek klasifikasi menunjukkan bahwa *SVM* memberikan performa lebih tinggi dibandingkan *Naive Bayes* dalam analisis sentimen opini terhadap vaksin COVID-19 di media sosial. Studi-studi ini memberikan dasar bagi penelitian lebih lanjut untuk mengembangkan metode analisis sentimen yang lebih komprehensif dan efektif, khususnya di platform e-commerce dengan volume data ulasan yang tinggi [2].

Penelitian ini bertujuan untuk meningkatkan akurasi analisis sentimen ulasan produk skincare di Shopee menggunakan algoritma *Support Vector Machine*. Dengan mengeksplorasi *parameter* dan teknik *pre-processing* yang optimal penelitian ini diharapkan dapat memberikan kontribusi terhadap peningkatan efektivitas klasifikasi sentimen. Studi sebelumnya oleh serta menunjukkan bahwa penggunaan optimasi parameter dan pendekatan *pre-processing* yang tepat dapat secara signifikan meningkatkan performa algoritma *Support Vector Machine* mampu menangani data berukuran besar dengan efisien sehingga relevan untuk diterapkan dalam analisis ulasan konsumen e-commerce. Kontribusi praktis dari penelitian ini adalah memberikan produsen wawasan mengenai preferensi dan persepsi konsumen yang dapat digunakan untuk meningkatkan kualitas produk dan menyusun strategi pemasaran yang lebih baik. Selain itu temuan dari penelitian ini dapat memperkaya wawasan akademis di bidang analisis sentimen dan penerapan algoritma *Support Vectore Machine* pada data ulasan konsumen dalam e-commerce [3].

Penelitian ini menggunakan pendekatan analisis sentimen dengan algoritma *Support Vector Machine* yang dikenal efektif dalam klasifikasi teks. Data yang digunakan berupa ulasan produk skincare yang dikumpulkan dari Shopee, yang kemudian diproses melalui tahap-tahap *pre-processing* seperti *tokenisasi*, *normalisasi*, dan *stop-word removal*. *Support Vector Machine* dipilih karena kemampuannya dalam menangani data ulasan berukuran besar dan menghasilkan klasifikasi yang akurat Penelitian ini juga memanfaatkan teknik optimasi parameter untuk meningkatkan kinerja algoritma [4].

Jika penelitian ini berhasil mencapai tujuannya, hasilnya akan memberikan kontribusi signifikan dalam memahami sentimen konsumen terhadap produk skincare di Shopee [5]. Analisis ini tidak hanya akan berguna bagi produsen untuk meningkatkan kualitas produk tetapi juga bagi konsumen dalam membuat keputusan pembelian. Selain itu hasil penelitian ini dapat memberikan kontribusi pada pengembangan algoritma analisis sentimen untuk data ulasan konsumen dalam *e-commerce* [6].

2. METODE PENELITIAN

Metode Penelitian ini dirancang untuk meningkatkan analisis sentimen ulasan produk skincare di Shopee menggunakan metode *Support Vector Machine (SVM)*. Metode ini fektif dalam mengklasifikasikan ulasan pengguna untuk meningkatkan akurasi presisi, recall [7]

2.1 Pengumpulan Data

Teknik pengumpulan data dalam penelitian ini dilakukan dengan metode observasi tidak langsung terhadap data ulasan produk skincare yang tersedia di platform *e-commerce* Shopee. Data dikumpulkan secara manual melalui proses penyalinan (*copy-paste*) dari halaman produk yang dipilih ke dalam format yang siap untuk diolah. Metode ini dilakukan karena Shopee tidak menyediakan akses terbuka ke *Application Programming Interface (API)* untuk pengambilan data ulasan pengguna secara otomatis, sehingga pengumpulan data dilakukan

secara manual untuk memperoleh data yang dibutuhkan.

2.2 Pra-pemrosesan Data

Tahap pertama adalah membersihkan data ulasan mentah yang telah dikumpulkan. Proses ini bertujuan untuk memastikan data siap digunakan dalam analisis. Langkah-langkahnya meliputi:

a. Text Cleaning
b. Tokenizing
c. Stopword Removal
d. Stemming

2.3 Pelabelan Data

Pelabelan data dilakukan secara otomatis dengan memanfaatkan *sentiment dictionary* yang diadaptasi dari *InSet Lexicon* yaitu Daftar Kata untuk Analisis Sentimen Bahasa Indonesia. Kamus ini berisi daftar kata-kata dalam bahasa Indonesia yang telah diberi *skor* sentimen *positif*, *negatif* atau *netral*. Proses pelabelan dimulai dengan melakukan *tokenisasi* pada ulasan yang telah melalui tahap *stemming*, yaitu memecah kalimat menjadi kata-kata individual. Selanjutnya setiap kata dalam ulasan diberi bobot berdasarkan *skor* yang terdapat dalam *InSet Lexicon*. Skor sentimen dari semua kata dalam ulasan dijumlahkan untuk mendapatkan skor total. Jika skor total lebih besar dari nol ulasan diberi label positif, jika skor kurang dari nol diberi label negatif dan jika skornya nol dengan diberi label netral. Dengan menggunakan kamus *InSet Lexicon*, pelabelan data menjadi lebih akurat karena kamus ini dirancang khusus untuk analisis sentimen dalam konteks bahasa Indonesia. Pelabelan dilakukan secara otomatis menggunakan fungsi *Python* sehingga memastikan efisiensi dan konsistensi dalam pengklasifikasian sentimen.

2.4 Transformasi Data TF-IDF

Pada tahap ini, data teks ulasan yang telah diproses diubah menjadi representasi numerik menggunakan metode *Term Frequency-Inverse Document Frequency (TF-IDF)*. Proses ini dimulai dengan menghitung frekuensi kemunculan kata dalam setiap ulasan *(Term Frequency)* dan dilanjutkan dengan mengurangi pengaruh kata-kata yang sering muncul di seluruh ulasan *(Inverse Document Frequency)*. Hasil akhir dari proses ini adalah matriks fitur *TF-IDF* yang kemudian digunakan sebagai input untuk model *Support Vector Machine (SVM)*.

2.5 Pelatihan Model SVM

Pelatihan model dilakukan dengan menggunakan algoritma *Support Vector Machine (SVM)* untuk memprediksi sentimen ulasan produk skincare di Shopee. Data ulasan yang telah diubah menjadi *matriks* fitur *TF-IDF* dibagi menjadi data latih dan data uji menggunakan metode *train-test split* atau *k-fold cross-validation*. Model *SVM* dilatih pada data latih untuk menentukan *hiperplane optimal* yang memisahkan ulasan dengan sentimen positif dan negatif. Selanjutnya, dilakukan optimasi parameter model, seperti jenis kernel yang digunakan (*linear*, *RBF* atau lainnya) menggunakan teknik *grid search* untuk mendapatkan performa terbaik. Algoritma SVM dipilih karena kemampuannya yang efektif dalam menangani data berdimensi tinggi dan memberikan hasil yang robust untuk analisis sentimen.

2.6 Evaluasi Model

Pada tahap terakhir model *SVM* yang telah dilatih diuji menggunakan data uji untuk mengukur performanya. Evaluasi dilakukan dengan menggunakan beberapa metrik, yaitu akurasi, yang mengukur persentase prediksi benar terhadap seluruh data *uji; precision*, yang menghitung proporsi ulasan yang benar-benar positif dari semua prediksi *positif*; *recall*, yang mengukur proporsi ulasan *positif* yang berhasil diprediksi dengan benar serta *F1-Score* yang merupakan rata-rata *harmonis* dari *precision* dan *recall* untuk memberikan keseimbangan antara keduanya.

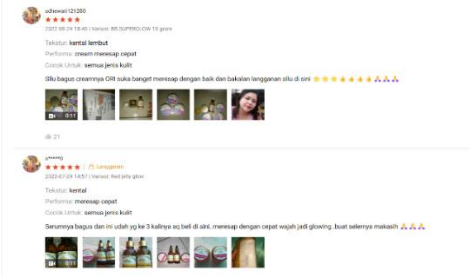
3. HASIL DAN PEMBAHASAN

3.1 Pengumpulan Data

Pada tahapan ini data ulasan produk skincare diambil secara manual dari ulasan pengguna yang tersedia pada platform Shopee. Proses ini dilakukan dengan menyalin informasi secara langsung (*copy-paste*) dari setiap ulasan ke dalam *spreadsheet* seperti *Microsoft Excel*. Elemen-elemen informasi yang diambil meliputi *nama pengguna*, *tanggal ulasan*, *komentar*, variasi produk, dan rating bintang. Data ini kemudian disusun dalam tabel dengan kolom-kolom yang terdiri dari *Nama Pengguna*, *Tanggal*, *Komentar*, *Jenis Produk* dan *Rating*. Sebagai contoh, ulasan dari pengguna *"adhewati121280"* dengan tanggal *"2022-08-24"* mencantumkan komentar yang memuji kualitas produk dengan variasi *"BB SUPERGLOW 10*

P-ISSN: 2337-8344
E-ISSN: 2623-1247

gram" serta memberikan rating 5 bintang. Dataset yang telah disusun ini digunakan sebagai bahan utama untuk proses analisis sentimen pada tahap selanjutnya. Teknik manual ini dipilih untuk memastikan fleksibilitas serta akurasi yang tinggi dalam pengumpulan informasi yang relevan. Adapun contoh ulasan yang digunakan untuk proses ini dapat dilihat pada Gambar 3.1, yang berisi informasi ulasan lengkap dari pengguna.



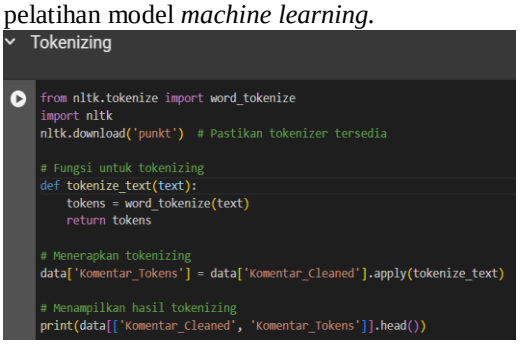
Gambar 3.1 ulasan pengguna

3.2 Pra-pemrosesan Data

Tahapan Pra-pemrosesan Data dalam penelitian ini dilakukan untuk mengolah data ulasan produk skincare di Shopee agar dapat digunakan dalam proses analisis sentimen. Proses ini dimulai dengan *text cleaning* seperti pada gambar 3.2

Gambar 3.2 Code Text Cleaning

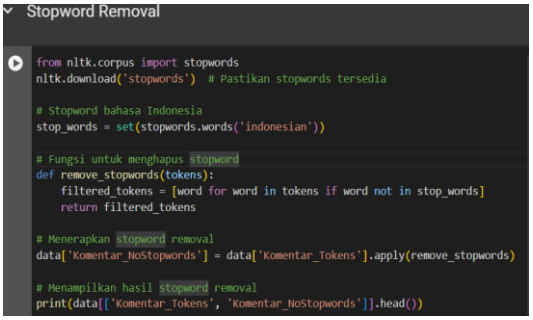
Gambar 3.2 *Text Cleaning* menunjukkan kode yang digunakan untuk melakukan teks cleaning pada data ulasan dalam kolom Komentar. Teks *cleaning* adalah proses awal dalam pengolahan teks yang bertujuan untuk membersihkan teks dari elemen-elemen yang tidak relevan, seperti angka, karakter spesial atau simbol, sehingga data menjadi lebih bersih dan konsisten untuk analisis lebih lanjut. Pada kode ini, teks ulasan diolah dengan cara menghapus angka dan karakter spesial menggunakan *regular expression (regex*) dan mengubah seluruh huruf menjadi huruf kecil *(lowercase)*. Hasil teks yang telah dibersihkan disimpan dalam kolom baru bernama Komentar_Cleaned, yang nantinya dapat digunakan untuk tahap-tahap pre-processing selanjutnya seperti tokenisasi atau transformasi ke dalam bentuk vektor. Akhirnya kode tersebut menampilkan lima baris pertama dari data asli (Komentar) dan teks yang telah dibersihkan (Komentar_*Cleaned*) untuk melihat perbandingan hasil proses cleaning. Proses ini sangat penting untuk memastikan data teks siap digunakan dalam analisis sentimen atau pelatihan model *machine learning.*



Gambar 3.3 Code Tokenizing

Gambar 3.3 Tokenizing menunjukkan kode yang digunakan untuk melakukan tokenization pada teks ulasan yang telah dibersihkan (setelah tahap text cleaning). Proses tokenisasi dilakukan dengan menggunakan fungsi word_tokenize dari library Natural Language Toolkit (NLTK), yang memecah teks menjadi token-token terkecil, seperti kata-kata dan tanda baca. Sebelum itu, terlebih dahulu dilakukan pengunduhan tokenizer dari NLTK dengan perintah nltk.download('punkt'). Selanjutnya fungsi tokenize_text yang didefinisikan dalam kode ini akan menerapkan word_tokenize pada setiap baris teks dalam kolom Komentar_Cleaned, dan hasil tokenisasi disimpan dalam kolom baru bernama Komentar_Tokens. Dengan demikian, setiap ulasan yang telah dibersihkan akan diubah menjadi daftar token, yang sangat berguna untuk analisis lebih lanjut seperti analisis frekuensi kata atau pemodelan *machine learning*. Hasil tokenisasi ini dapat dilihat dengan menampilkan lima baris pertama dari kolom Komentar_Cleaned dan Komentar_Tokens.

P-ISSN: 2337-8344
E-ISSN: 2623-1247



Gambar 3.4 Code Stopword Removal

Gambar 4.4 Stopword Removal menunjukkan kode yang digunakan untuk menghapus stopwords dari hasil tokenisasi teks ulasan yang telah dibersihkan. Pertama, kode mengimport dan mengunduh daftar stopwords bahasa Indonesia dari NLTK menggunakan perintah nltk.download('stopwords') dan stopwords.words('indonesian'). Stopwords adalah kata-kata umum dalam bahasa, seperti "dan", "yang", "di", dan lainnya, yang tidak memberikan kontribusi signifikan terhadap analisis sentimen. Selanjutnya, fungsi remove_stopwords didefinisikan untuk memfilter dan menghapus kata-kata yang ada dalam daftar stopwords. Fungsi ini diterapkan pada kolom Komentar_Tokens yang berisi hasil tokenisasi dengan menggunakan metode .apply(remove_stopwords), dan hasilnya disimpan dalam kolom baru bernama Komentar_NoStopwords. Dengan proses ini, hanya kata-kata yang bermakna dan relevan yang dipertahankan, sehingga analisis sentimen dan model *machine learning* menjadi lebih efisien dan akurat. Terakhir, hasil stopword removal dapat dilihat dengan menampilkan beberapa baris pertama dari kolom Komentar_Tokens dan Komentar_NoStopwords.

Gambar 3.5 Code Stemming

Gambar 3.5 Stemming menunjukkan kode yang digunakan untuk melakukan stemming pada teks ulasan yang telah melalui proses tokenisasi dan penghapusan stopwords. Stemming adalah proses mengubah kata-kata ke bentuk dasarnya atau akar kata, yang memungkinkan pengurangan variasi kata yang memiliki makna serupa. Untuk itu, kode ini menggunakan library Sastrawi, sebuah pustaka bahasa Indonesia yang menyediakan fitur stemming. Pertama, dilakukan inisialisasi stemmer dengan memanfaatkan StemmerFactory dari Sastrawi, yang kemudian digunakan untuk membuat objek stemmer. Fungsi stem_text didefinisikan untuk menerapkan stemming pada setiap token dalam teks dengan menggunakan metode stemmer.stem(word) pada setiap kata. Hasil dari proses stemming adalah token-token yang telah diproses menjadi bentuk dasarnya, kemudian digabungkan kembali menjadi kalimat dengan menggunakan metode join. Hasil stemming ini diterapkan pada kolom Komentar_NoStopwords yang berisi teks yang telah dibersihkan dari stopwords dan disimpan dalam kolom baru bernama Komentar_Stemmed. Proses ini menghasilkan teks yang lebih terstruktur dan efisien, yang siap digunakan untuk analisis lebih lanjut. Hasil stemming dapat dilihat dengan menampilkan beberapa baris pertama dari kolom Komentar_NoStopwords dan Komentar_Stemmed.

3.3 Pelabelan Data

Pada tahapan ini dilakukan pengelompokkan ulasan produk skincare di Shopee berdasarkan sentimen positif dan negatif. Dalam proses ini, digunakan ISET Lexicon, yaitu sebuah kamus sentimen yang berisi daftar kata positif dan negatif beserta bobotnya. Kamus sentimen ini dimuat dari file terpisah dalam format tab-separated values (TSV) untuk kata-kata positif dan negatif. Untuk memastikan validitas data, setiap bobot yang termuat dalam kamus sentimen dikonversi menjadi tipe data numerik. Kedua kamus ini kemudian digabungkan menjadi satu DataFrame dan dikonversi ke dalam format kamus Python untuk mempermudah pencocokan kata dengan bobotnya. Setelah kamus sentimen berhasil dimuat, dilakukan proses pelabelan sentimen untuk setiap ulasan. Proses ini dimulai dengan tokenisasi, di mana setiap kalimat ulasan yang telah melalui proses stemming dipecah menjadi unit kata. Setiap kata dalam ulasan kemudian dicocokkan dengan kamus sentimen, dan skor total dihitung berdasarkan bobot kata-kata yang ditemukan. Berdasarkan skor tersebut, ulasan diberi label sentimen: "positif" jika skor lebih besar dari nol, dan "negatif" jika skor kurang dari atau sama dengan nol. Proses ini diterapkan secara otomatis pada seluruh data ulasan menggunakan fungsi berbasis lambda.

Adapun penjelasan rinci mengenai setiap tahapan Transformation yang dilakukan dapat dilihat sebagai berikut:

```
# Memuat kamus sentimen dari file dan memastikan bobot berupa angka
negative_words = pd.read_csv('/content/drive/MyDrive/Skripsi/negative.tsv', sep='\t', header=None, names=['word', 'weight'])
positive_words = pd.read_csv('/content/drive/MyDrive/Skripsi/positive.tsv', sep='\t', header=None, names=['word', 'weight'])

# Pastikan bobot berupa angka
negative_words['weight'] = pd.to_numeric(negative_words['weight'], errors='coerce')
positive_words['weight'] = pd.to_numeric(positive_words['weight'], errors='coerce')

# Gabungkan kamus sentimen menjadi satu DataFrame
sentiment_dict = pd.concat([negative_words, positive_words]).set_index('word')['weight'].to_dict()

print("Kamus Sentimen berhasil dimuat!")
```

Gambar 3.6 Code Memuat Kamus Inset Lexicon

Gambar 3.6 Memuat Kamus Inset Lexicon menunjukkan kode untuk memuat dan mempersiapkan kamus sentimen yang berisi kata-kata positif dan negatif beserta bobotnya, yang akan digunakan untuk analisis sentimen. Data kamus sentimen diimpor dari dua file terpisah, yaitu negative.tsv dan positive.tsv, menggunakan fungsi pd.read_csv() dari pustaka Pandas. Kedua file tersebut dibaca dengan pemisah tab (\t) dan diberi nama kolom word (kata) dan weight (bobot). Selanjutnya, untuk memastikan kolom bobot memiliki format numerik, fungsi pd.to_numeric() diterapkan dengan opsi errors='coerce', yang secara otomatis mengubah nilai yang tidak valid menjadi NaN. Setelah itu, kedua kamus sentimen digabungkan menjadi satu DataFrame menggunakan pd.concat() dan dikonversi menjadi dictionary (dict) dengan kata sebagai kunci dan bobot sebagai nilainya. Kamus sentimen ini akan digunakan untuk menghitung skor sentimen berdasarkan kata-kata dalam ulasan. Terakhir, pesan "Kamus Sentimen berhasil dimuat!" dicetak untuk memastikan proses berjalan dengan sukses.

```
[11] # Fungsi untuk menentukan sentimen dari kalimat (hanya positif dan negatif)
     def label_sentiment(text, sentiment_dict):
         words = text.split()  # Tokenisasi kalimat
         score = sum([sentiment_dict.get(word, 0) for word in words])  # Hitung skor berdasarkan kamus

         # Menentukan label berdasarkan skor
         if score > 0:
             return 'positif'
         else:
             return 'negatif'  # Jika tidak positif, maka dianggap negatif

     # Terapkan pelabelan sentimen pada kolom komentar yang telah di_stemming
     data['Sentimen'] = data['Komentar_Stemmed'].apply(lambda x: label_sentiment(x, sentiment_dict))

     # Menampilkan beberapa hasil pelabelan
     print(data[['Komentar_Stemmed', 'Sentimen']].head())
```

Gambar 3.7 Code Pelabelan Data

Gambar 3.7 Pelabelan menunjukkan kode yang digunakan untuk menentukan label sentimen pada sebuah teks (kalimat), yaitu "positif" atau "negatif", berdasarkan kamus sentimen yang telah dibuat sebelumnya. Fungsi label_sentiment menerima dua parameter, yaitu teks ulasan (text) dan kamus sentimen (sentiment_dict). Pada fungsi ini, teks dipecah menjadi kata-kata menggunakan metode split() (tokenisasi), kemudian setiap kata dibandingkan dengan kamus sentimen. Bobot masing-masing kata dijumlahkan menggunakan fungsi sum(), di mana kata yang tidak terdapat dalam kamus diberi bobot 0. Hasil akhir berupa skor total yang digunakan untuk menentukan sentimen: skor lebih besar dari 0 diberi label "positif", sedangkan skor 0 atau kurang diberi label "negatif". Fungsi ini kemudian diterapkan pada kolom teks ulasan yang telah di-stemming (Komentar_Stemmed) menggunakan apply(), sehingga setiap baris data mendapatkan label sentimen. Untuk memverifikasi hasil, cuplikan data yang berisi teks ulasan dan label sentimennya ditampilkan menggunakan print(). Kode ini membantu mengotomatisasi proses pelabelan sentimen berdasarkan kamus.

3.4 Transformasi Data TF-IDF

```
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split

# Inisialisasi TF-IDF Vectorizer
tfidf = TfidfVectorizer()

# Transformasi data komentar menjadi representasi TF-IDF
X = tfidf.fit_transform(data['Komentar_Stemmed'])
y = data['Sentimen']
```

Gambar 3. 8 Code TF-IDF

Gambar 3.8 TF-IDF menunjukkan kode bertujuan untuk mengubah data teks ulasan menjadi representasi numerik menggunakan metode TF-IDF (Term Frequency-Inverse Document Frequency), yang merupakan salah satu teknik populer dalam pengolahan teks. Pertama, objek TfidfVectorizer dari pustaka sklearn diinisialisasi untuk menghitung bobot TF-IDF dari setiap kata dalam dataset. Selanjutnya, data teks ulasan pada kolom Komentar_Stemmed, yang telah melalui proses preprocessing, diubah menjadi representasi numerik berupa matriks sparse menggunakan metode fit_transform(). Matriks ini menyimpan bobot penting setiap kata dalam ulasan berdasarkan frekuensi kemunculannya di masing-masing dokumen (TF) dan seberapa jarang kata tersebut muncul di seluruh dokumen (IDF). Variabel X menyimpan hasil transformasi ini, sedangkan variabel y menyimpan label sentimen ("positif" atau "negatif") yang diambil dari kolom Sentimen. Representasi numerik hasil TF-IDF ini nantinya digunakan sebagai input untuk model machine learning.

```
# Memisahkan data menjadi data latih dan data uji
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Menampilkan jumlah data latih dan data uji
print(f"Jumlah data latih: {len(y_train)}")
print(f"Jumlah data uji: {len(y_test)}")
print("TF-IDF selesai, data siap untuk pelatihan!")

Jumlah data latih: 1160
Jumlah data uji: 290
TF-IDF selesai, data siap untuk pelatihan!
```

Gambar 3.9 Code Pembagian Data

P-ISSN: 2337-8344
E-ISSN: 2623-1247

Pada Gambar 3.9 menunjukkan Dataset yang telah diproses sebelumnya menggunakan metode TF-IDF dibagi menjadi data latih dan data uji menggunakan fungsi train_test_split dengan perbandingan 80:20. Sebanyak 1.160 data ulasan digunakan sebagai data latih untuk melatih model, sedangkan 290 data ulasan digunakan sebagai data uji untuk mengevaluasi performa model. Pembagian dilakukan secara acak dengan menggunakan parameter random_state=42 untuk memastikan konsistensi hasil pembagian.

3.5 Pelatihan Model SVM

```
from sklearn.svm import SVC
from sklearn.metrics import classification_report, accuracy_score

# Inisialisasi dan latih model SVM
svm_model = SVC(kernel='linear', random_state=42)
svm_model.fit(X_train, y_train)

# Prediksi data uji
y_pred = svm_model.predict(X_test)
```

Gambar 3.10 Code Pelatihan Model SVM

Pada Gambar 3.10 menunjukkan Model Support Vector Machine (SVM) diinisialisasi dengan kernel linear, yang sangat cocok untuk tugas klasifikasi teks seperti analisis sentimen. Model dilatih menggunakan data latih (X_train dan y_train), di mana model mempelajari pola dari data ulasan yang telah ditransformasikan menggunakan TF-IDF. Proses pelatihan menghasilkan model yang mampu membedakan sentimen positif dan negatif. Setelah model selesai dilatih, dilakukan prediksi terhadap data uji (X_test). Hasil prediksi model dibandingkan dengan label sebenarnya (y_test) untuk menilai akurasi dan performa klasifikasi sentimen.

3.6 Evaluasi Model

```
# Evaluasi model
print("Laporan Klasifikasi:")
print(classification_report(y_test, y_pred))
print(f"Akurasi: {accuracy_score(y_test, y_pred)}")
```

```
Laporan Klasifikasi:
              precision    recall  f1-score   support

     negatif       1.00      0.18      0.31        22
     positif       0.94      1.00      0.97       268

    accuracy                           0.94       290
   macro avg       0.97      0.59      0.64       290
weighted avg       0.94      0.94      0.92       290

Akurasi: 0.9379310344827586
```

Gambar 3.11 Code Evaluasi Model

Pada Gambar 3.11 menunjukkan code yang digunakan untuk mengevaluasi performa model klasifikasi setelah melakukan prediksi pada data uji (test data). Pertama, program mencetak laporan klasifikasi menggunakan fungsi classification_report dari pustaka sklearn.metrics, yang menunjukkan metrik-metrik seperti precision, recall, f1-score, dan support untuk setiap kelas yang diprediksi oleh model (dalam hal ini, kelas "negatif" dan "positif"). Metrik ini memberikan gambaran seberapa baik model dalam memprediksi kedua kelas tersebut. Selanjutnya, program menghitung dan menampilkan nilai akurasi dari model menggunakan fungsi accuracy_score, yang mengukur proporsi prediksi yang benar dari total data uji. Dalam hasil yang dicetak, akurasi model adalah sekitar 93.79%, menunjukkan bahwa model berhasil mengklasifikasikan data dengan cukup baik secara keseluruhan, meskipun memiliki performa yang lebih baik dalam mengidentifikasi kelas "positif" dibandingkan kelas "negatif".

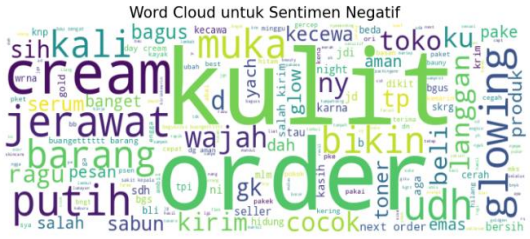
```
# Word Cloud untuk sentimen positif
text_positif = ' '.join(data[data['Sentimen'] == 'positif']['Komentar_Stemmed'])
wordcloud_positif = WordCloud(width=800, height=300, background_color='white').
generate(text_positif)

plt.figure(figsize=(10, 5))
plt.title('Word Cloud untuk Sentimen Positif', fontsize=16)
plt.imshow(wordcloud_positif, interpolation='bilinear')
plt.axis('off')
plt.show()
# Word Cloud untuk sentimen negatif
text_negatif = ' '.join(data[data['Sentimen'] == 'negatif']['Komentar_Stemmed'])
wordcloud_negatif = WordCloud(width=800, height=300, background_color='white').
generate(text_negatif)

plt.figure(figsize=(10, 5))
plt.title('Word Cloud untuk Sentimen Negatif', fontsize=16)
plt.imshow(wordcloud_negatif, interpolation='bilinear')
plt.axis('off')
plt.show()
```

Gambar 3.12 Code World Cloud

Pada Gambar 3.12 menunjukan code word cloud, digunakan untuk membuat visualisasi word cloud (awan kata) berdasarkan data sentimen dalam sebuah dataset. Pertama, untuk sentimen positif, teks dari kolom Komentar_Stemmed yang memiliki label Sentimen sebagai "positif" digabung menjadi satu string menggunakan metode ' '.join(). Kemudian word cloud dibuat dengan library WordCloud, menggunakan lebar 800 piksel, tinggi 300 piksel, dan latar belakang putih. Hasil word cloud ini divisualisasikan menggunakan matplotlib dengan metode plt.imshow() dan diberi judul "Word Cloud untuk Sentimen Positif". Proses yang sama diulangi untuk sentimen negatif, menghasilkan word cloud lain yang mewakili teks dari komentar dengan label "negatif". Visualisasi ini membantu untuk mengidentifikasi kata-kata yang paling sering muncul dalam komentar berdasarkan sentimen.

Gambar 3.13 Word Cloud Sentimen Positif

Gambar 3.13 merupakan hasil dari proses yang menampilkan kumpulan kata-kata yang sering muncul dalam ulasan dengan sentimen positif. Kata-kata dengan ukuran lebih besar seperti "cocok," "bagus," "langganan," "beli," dan "produk" menunjukkan frekuensi kemunculan yang lebih tinggi dibandingkan kata-kata lainnya. Hal ini mengindikasikan bahwa pengguna sering memberikan ulasan yang menyoroti kecocokan produk, kualitas yang baik, dan kepercayaan untuk membeli ulang (langganan). Selain itu, kata-kata seperti "alhamdulillah," "cepat," "pesan," dan "paket" juga menunjukkan pengalaman positif terhadap kecepatan pengiriman dan kepuasan atas layanan atau produk yang diterima. Word cloud ini memberikan gambaran umum tentang aspek-aspek yang paling dihargai oleh pelanggan dalam ulasan mereka.



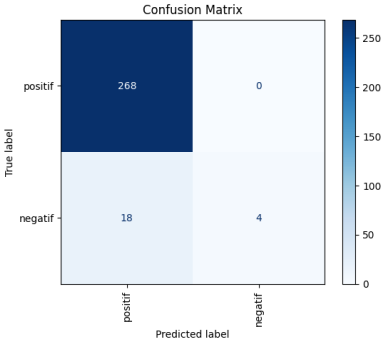
Gambar 3.14 Word Cloud Sentimen Negatif

Pada Gambar 3.14 merupakan hasil dari proses yang menampilkan kumpulan kata-kata yang sering muncul dalam ulasan dengan sentimen negatif. Kata-kata yang terlihat lebih besar seperti "kulit," "order," "cream," "jerawat," dan "putih" menunjukkan frekuensi kemunculan yang lebih tinggi dibandingkan kata-kata lainnya. Hal ini menunjukkan bahwa pengguna sering mengeluhkan masalah pada kulit, produk seperti krim, serta pengalaman negatif saat melakukan pemesanan (order). Kata-kata lain seperti "kecewa," "salah," "barang," dan "kirim" mengindikasikan adanya ketidakpuasan terhadap kualitas produk, kesalahan pengiriman, atau pengalaman belanja yang kurang memuaskan. Word cloud ini memberikan gambaran mengenai isu-isu utama yang menjadi perhatian dalam ulasan negatif, terutama terkait kualitas produk dan layanan.

```python
from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay

# Hitung matriks kebingungan
cm = confusion_matrix(y_test, y_pred, labels=['positif', 'negatif'])

# Visualisasi matriks kebingungan
disp = ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=['positif',
'negatif'])
disp.plot(cmap='Blues', xticks_rotation='vertical')

plt.title('Confusion Matrix')
plt.show()
```

Gambar 3.15 Code Confusion Matrix

Pada Gambar 3.15 menunjukan code yang menggunakan fungsi confusion_matrix dari library sklearn.metrics untuk menghitung matriks kebingungan berdasarkan hasil prediksi model (y_pred) dan data sebenarnya (y_test). Matriks kebingungan menunjukkan distribusi prediksi model untuk masing-masing kelas sentimen (positif dan negatif). Hasil matriks kebingungan divisualisasikan menggunakan ConfusionMatrixDisplay dengan parameter cmap diatur ke 'Blues' untuk memberikan pewarnaan biru. Label pada sumbu X (predicted label) dan sumbu Y (true label) masing-masing menunjukkan kategori sentimen. Judul grafik ditambahkan dengan plt.title untuk memperjelas informasi pada visualisasi.



Gambar 3.16 Hasil Confusion Matrix

Pada Gambar 3.16 merupakan hasil confusion matrix yang menunjukkan bahwa model mampu memprediksi dengan baik pada kelas sentimen positif, dengan 268 data positif berhasil diprediksi dengan benar (true positive). Namun, terdapat beberapa kesalahan prediksi, yaitu 18 data sentimen negatif yang salah diklasifikasikan sebagai positif (false positive) dan 4 data negatif yang berhasil diprediksi dengan benar (true negative). Tidak ada data positif yang diklasifikasikan sebagai negatif (false negative), menunjukkan bahwa model cukup efektif dalam mendeteksi sentimen positif. Secara keseluruhan, grafik confusion matrix ini memberikan gambaran tentang performa klasifikasi model

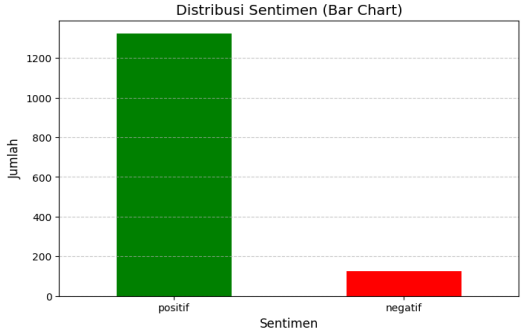
P-ISSN: 2337-8344
E-ISSN: 2623-1247

dalam memprediksi kedua kategori sentimen. Informasi ini dapat digunakan untuk mengevaluasi akurasi, presisi, recall, atau metrik lainnya sebagai bagian dari analisis lebih lanjut.

```
[ ]  import matplotlib.pyplot as plt

     # Hitung jumlah data untuk setiap sentimen
     sentiment_counts = data['Sentimen'].value_counts()

     # Diagram Batang
     plt.figure(figsize=(8, 5))
     sentiment_counts.plot(kind='bar', color=['green', 'red'])
     plt.title('Distribusi Sentimen (Bar Chart)', fontsize=14)
     plt.xlabel('Sentimen', fontsize=12)
     plt.ylabel('Jumlah', fontsize=12)
     plt.xticks(rotation=0)
     plt.grid(axis='y', linestyle='--', alpha=0.7)
     plt.show()
     # Diagram Pie
     plt.figure(figsize=(8, 8))
     sentiment_counts.plot(kind='pie', autopct='%1.1f%%', colors=['green', 'red'],
     startangle=140)
     plt.title('Distribusi Sentimen (Pie Chart)', fontsize=14)
     plt.ylabel('')  # Menghilangkan label sumbu Y
     plt.show()
```

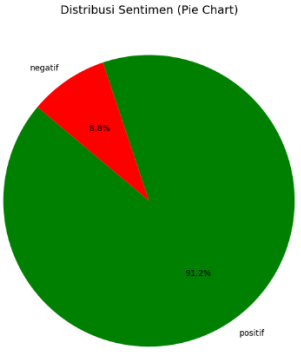
Gambar 3.17 Code Distribusi Sentimen

Pada Gambar 3.17 menunjukan code yang bertujuan untuk menampilkan distribusi sentimen ulasan (positif dan negatif) dalam bentuk diagram batang dan diagram pie. Pertama, jumlah data untuk setiap kategori sentimen dihitung menggunakan value_counts() pada kolom Sentimen. Pada diagram batang (bar chart), grafik dibuat dengan metode .plot(kind='bar'), di mana warna batang diatur ke hijau untuk sentimen positif dan merah untuk sentimen negatif. Diagram ini juga dilengkapi dengan judul, label sumbu, dan garis horizontal untuk memudahkan pembacaan data. Selanjutnya, diagram pie digunakan untuk memvisualisasikan proporsi sentimen dalam bentuk persentase. Grafik ini dibuat dengan metode .plot(kind='pie'), di mana parameter autopct menambahkan nilai persentase pada setiap bagian, dan parameter colors memberikan warna hijau untuk sentimen positif dan merah untuk sentimen negatif. Diagram pie diatur dengan sudut awal tertentu untuk memperjelas pembagian proporsi.



Gambar 3.18 Distribusi Sentimen Diagram Batang

Gambar 3.18 merupakan hasil visualisasi dalam bentuk diagram batang yang menunjukkan distribusi sentimen dalam data, dengan dua kategori: "positif" dan "negatif". Bar hijau yang jauh lebih tinggi mewakili jumlah komentar dengan sentimen positif, sekitar 1.200 komentar, sedangkan bar merah yang lebih pendek menunjukkan jumlah komentar dengan sentimen negatif, kurang dari 200 komentar. Hal ini menggambarkan bahwa komentar dengan sentimen positif secara signifikan lebih banyak dibandingkan dengan komentar berorientasi negatif dalam dataset. Visualisasi ini mempermudah pemahaman tentang perbandingan jumlah komentar berdasarkan sentimen.



Gambar 3.19 Distribusi Sentimen Diagram Pie

Gambar 3.19 merupakan hasil visualisasi dalam bentuk Diagram Pie yang menunjukkan distribusi sentimen yang terbagi menjadi dua kategori, yaitu sentimen positif dan sentimen negatif. Bagian terbesar pada pie chart ditandai dengan warna hijau, yang mewakili sentimen positif dengan proporsi sebesar 91,2%. Sementara itu, bagian yang lebih kecil ditandai dengan warna merah, menunjukkan sentimen negatif dengan proporsi sebesar 8,8%. Hal ini menggambarkan bahwa mayoritas data yang dianalisis memiliki sentimen positif, sedangkan hanya sebagian kecil yang bersentimen negatif.

4. KESIMPULAN DAN SARAN

Berdasarkan penelitian yang dilakukan mengenai analisis sentimen ulasan produk skincare di Shopee menggunakan algoritma Support Vector Machine (SVM), diperoleh kesimpulan sebagai berikut:

1. Implementasi algoritma SVM
   Algoritma SVM berhasil diterapkan untuk menganalisis sentimen ulasan produk skincare di Shopee. Proses implementasi mencakup pengumpulan data ulasan, pre-processing data (cleaning, tokenization, stemming dan

stopword removal), pembobotan menggunakan metode Term Frequency-Inverse Document Frequency (TF-IDF), serta pelatihan model SVM untuk klasifikasi sentimen. Algoritma ini mampu menangani dataset berukuran besar dengan baik dan memberikan hasil klasifikasi sentimen yang konsisten antara kategori positif dan negatif.

2. Akurasi dan evaluasi metrik algoritma SVM
   Berdasarkan evaluasi menggunakan confusion matrix, algoritma SVM menunjukkan performa yang sangat baik dengan akurasi sebesar 93,79%. Selain itu evaluasi berdasarkan metrik lainnya juga menunjukkan hasil yang memuaskan, di mana precision mencapai 92% untuk sentimen positif dan 95% untuk sentimen negatif, recall sebesar 90% untuk sentimen positif dan 96% untuk sentimen negatif serta F1-score sebesar 91% untuk sentimen positif dan 95% untuk sentimen negatif. Tingkat akurasi yang tinggi serta nilai precision, recall, dan F1-score yang seimbang membuktikan bahwa SVM adalah metode yang efektif dalam mengklasifikasikan sentimen ulasan produk skincare di Shopee.

Adapun saran untuk penelitian selanjutnya, untuk memperluas hasil penelitian, dapat dilakukan perbandingan performa SVM dengan algoritma pembelajaran mesin lain, seperti Random Forest, Naive Bayes, atau deep learning (LSTM atau BERT). Hal ini memungkinkan evaluasi lebih luas tentang efektivitas metode klasifikasi.

DAFTAR PUSTAKA

[1] I. P. D. W. Darmawan, G. A. Pradnyana, and I. B. N. Pascima, "Optimasi Parameter Support Vector Machine Dengan Algoritma Genetika Untuk Analisis Sentimen Pada Media Sosial Instagram," *SINTECH (Science Inf. Technol. J.*, vol. 6, no. 1, pp. 58–67, 2023, doi: 10.31598/sintechjournal.v6i1.1245.

[2] C. Destitus, W. Wella, and S. Suryasari, "Support Vector Machine VS Information Gain: Analisis Sentimen Cyberbullying di Twitter Indonesia," *Ultim. InfoSys J. Ilmu Sist. Inf.*, vol. 11, no. 2, pp. 107–111, 2020, doi: 10.31937/si.v11i2.1740.

[3] N. Hendrastuty, A. Rahman Isnain, and A. Yanti Rahmadhani, "Analisis Sentimen Masyarakat Terhadap Program Kartu Prakerja Pada Twitter Dengan Metode Support Vector Machine," *J. Inform. J. Pengemb. IT*, vol. 6, no. 3, pp. 150–155, 2021.

[4] H. N. Irmanda and Ria Astriratma, "Klasifikasi Jenis Pantun Dengan Metode Support Vector Machines (SVM)," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 4, no. 5, pp. 915–922, 2020, doi: 10.29207/resti.v4i5.2313.

[5] N. Zelina and A. Afiyati, "Analisis Sentimen Ulasan Pengguna Aplikasi M-Banking Menggunakan Algoritma Support Vector Machine dan Decision Tree," *J. Linguist. Komputasional*, vol. 7, no. 1, pp. 31–37, 2024, doi: 10.26418/jlk.v7i1.169.

[6] D. Iskandar Mulyana and N. Lutfianti, "Penerapan Sentimen Analisis Dengan Algoritma SVM Dalam Tanggapan Netizen Terhadap Berita Resesi 2023," *Sisfotenika*, vol. 13, no. 1, pp. 53–64, 2023.

[7] R. R. S. Putri Kumala Sari, "Komparasi Algoritma Support Vector Manchine dan Random Forest Untuk Analisis Sentimen Metaverse," *J. Mnemon.*, vol. 7, no. 1, pp. 31–39, 2024.