Penerapan Vektor di Ruang Euclidean dalam Analisis Sentimen Kalimat Bahasa Indonesia

Muhamad Nazih Najmudin — 13523144[1]
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
[1]13523144@std.stei.itb.ac.id, [2]nazihnajmudin@gmail.com

Abstrak— Media sosial menghasilkan volume data teks yang besar dengan berbagai opini dan sentimen yang membuat analisis sentimen menjadi semakin penting. Penelitian ini mengeksplorasi penerapan vektor ruang Euclidean dalam analisis sentimen bahasa Indonesia menggunakan algoritma *Support Vector Machine* (SVM). Metode ini mengubah teks menjadi representasi numerik dengan TF-IDF untuk menghasilkan vektor fitur, yang kemudian diklasifikasikan oleh SVM ke dalam kelas sentimen. Hasil eksperimen menunjukkan kombinasi vektor ruang Euclidean dan SVM mampu menghasilkan akurasi yang baik, meskipun terdapat keterbatasan dalam menangani kalimat kompleks dan kualitas dataset. Penelitian ini diharapkan dapat berkontribusi pada pengembangan sistem analisis sentimen bahasa Indonesia untuk berbagai aplikasi, seperti pemantauan opini publik, analisis produk, dan pengembangan *chatbot*.

***Kata Kunci*— analisis sentimen, vektor ruang Euclidean, Support Vector Machine, TF-IDF, bahasa Indonesia**

I. PENDAHULUAN

Saat ini, dunia telah memasuki era digital, yaitu era di mana segala hal di dunia ini memiliki ketergantungan dengan instrumen digital. Perkembangan instrumen-instrumen digital kian memuncak, salah satunya dalam penyebaran informasi dan komunikasi di dunia maya. Media sosial menjadi salah satu instrument digital yang paling banyak digunakan oleh manusia saat ini dalam berbagi informasi, berkomunikasi, dan menyuarakan pendapat mereka. Jutaan pengguna aktif di berbagai *platform* seperti Twitter, Facebook, Instagram, TikTok, dan Youtube setiap harinya menghasilkan data dalam jumlah besar mencakup opini, ulasan, dan tanggapan terhadap berbagai isu maupun produk barang atau jasa. Kondisi ini membuat media sosial menjadi salah satu sumber informasi yang kaya untuk memahami pandangan masyarakat secara luas.

Bagi industri, memahami informasi yang terkandung dalam media sosial menjadi kebutuhan yang tidak dapat diabaikan. Salah satu aspek pentingnya adalah menganalisis sentimen publik, yaitu mengidentifikasi apakah opini yang diungkapkan bernada positif, negatif, atau netral. Analisis sentimen ini sangat relevan dalam berbagai konteks, seperti pengembangan strategi pemasaran, peningkatan layanan pelanggan, serta pemantauan opini masyarakat terhadap isu-isu tertentu. Namun, kompleksitas data yang bersifat tidak terstruktur, seperti teks dalam bahasa alami, menimbulkan tantangan tersendiri dalam proses pengolahan dan analisis.

Untuk mengatasi tantangan ini, teknologi berbasis Natural Language Processing (NLP) telah berkembang pesat. NLP memungkinkan komputer untuk memahami dan memproses bahasa manusia dengan menggunakan pendekatan matematis dan komputasi. Salah satu teknik utama dalam NLP adalah representasi teks dalam bentuk vektor di ruang Euclidean. Representasi ini mengubah kata, frasa, atau kalimat menjadi bentuk numerik yang dapat dianalisis secara matematis, memungkinkan komputer untuk mengenali pola dan hubungan semantik dalam teks.

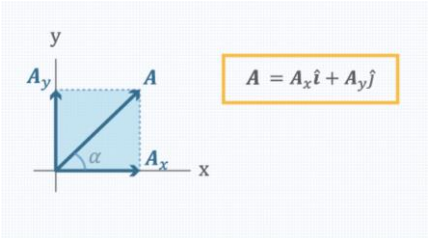
Dalam konteks analisis sentimen, penerapan vektor di ruang Euclidean memberikan kemampuan untuk mengukur kesamaan, relevansi, dan konteks antar kata atau kalimat. Teknik ini memungkinkan model untuk mendeteksi sentimen dalam teks secara lebih akurat dengan memanfaatkan representasi berbasis ruang, seperti word embeddings atau sentence embeddings. Makalah ini akan membahas penerapan vektor di ruang Euclidean dalam analisis sentimen kalimat berbahasa Indonesia, serta peranannya dalam meningkatkan akurasi dan efisiensi proses analisis tersebut.

II. DASAR TEORI

A. Vektor di Ruang Euclidean

Vektor adalah objek matematis yang memiliki panjang (*magnitude*) dan arah, biasanya direpresentasikan dalam bentuk barisan bilangan yang menunjukkan koordinatnya dalam suatu ruang. Secara formal, vektor sering dinyatakan dalam ruang Euklides, yaitu himpunan titik-titik yang diatur sedemikian rupa sehingga memenuhi sifat-sifat dasar geometri Euklides, seperti jarak dan sudut. Dalam ruang ini, vektor direpresentasikan sebagai titik atau panah yang berawal dari titik asal (origin) dan berakhir pada koordinat tertentu.Vektor dilambangkan dengan huruf kecil yang diberi panah diatasnya, atau

dicetak tebal. Vektor dapat dituliskan dalam notasi baris, notasi kolom, serta dalam notasi vector satuan *i-j-k*.



Gambar 1. Penggambaran vektor dan penulisan vektor dalam notasi vektor satuan.
Sumber: https://www.aisyahnestria.com/2019/12/komponen-vektor-dan-vektor-satuan.html (diakses pada 1 Januari 2025)

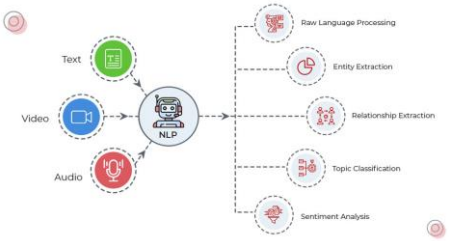
Ruang Euklides, atau biasa disebut juga Ruang Euclidean R^n adalah himpunan semua n-tupel terurut $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ dimana $x_1, x_2, \ldots, x_n \in R$. Panjang suatu vektor disebut sebagai norm, ditulis sebagai $||\mathbf{x}||$ dapat dihitung dengan rumus:

$$||x||_2 = \sqrt{\sum_{i=1}^{n} |x_i|^2}$$

Operasi dasar pada vektor meliputi penjumlahan, pengurangan, dan perkalian skalar. Penjumlahan vektor dilakukan dengan menjumlahkan elemen-elemen yang bersesuaian, sedangkan pengurangan mengikuti prinsip serupa dengan menggunakan selisih elemen. Perkalian skalar melibatkan pengalian setiap elemen vektor dengan suatu konstanta. Selain itu, terdapat operasi lanjutan seperti perkalian dot (dot product) dan perkalian silang (cross product), yang banyak digunakan untuk menghitung proyeksi dan luas area dalam ruang tertentu.

B. Natural Language Processing

Natural Language Processing (NLP) merupakan bidang interdisiplin yang mengaplikasikan teknik komputasi untuk menganalisis, memahami, dan menghasilkan bahasa manusia. NLP mencakup berbagai tingkatan analisis bahasa, mulai dari struktur kalimat (sintaksis) hingga makna kontekstual (semantik dan pragmatik). Tujuan utama NLP adalah untuk memungkinkan komputer berinteraksi dengan manusia secara alami melalui bahasa.



Gambar 2. Ilustrasi NLP
Sumber: https://beyondvoice.ai/natural-language-processing-consulting/ (diakses pada 1 Januari 2025)

Natural Language Processing (NLP) menganalisis bahasa manusia melalui berbagai tingkat bahasa. Tingkatan pemrosesan bahasa dalam NLP meliputi:

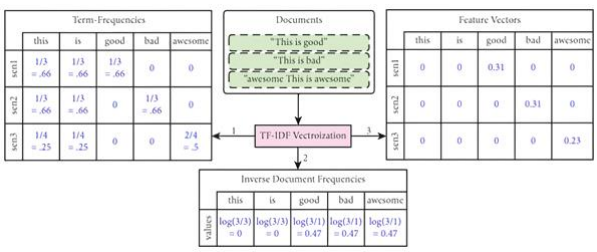
1. Fonologi: Menganalisis suara dalam kata, termasuk tekanan dan intonasi.
2. Morfologi: Menguraikan kata menjadi morfem (unsur terkecil yang bermakna).
3. Leksikal: Menentukan makna kata, termasuk bagian kata dan representasi semantik.
4. Sintaksis: Menganalisis struktur kalimat dan hubungan antar kata.
5. Semantik: Menentukan makna keseluruhan kalimat, termasuk disambiguasi kata.
6. Diskursus: Menganalisis teks lebih panjang, memahami hubungan antar kalimat, dan menyelesaikan referensi.
7. Pragmatik: Menganalisis penggunaan bahasa dalam konteks tertentu, termasuk niat dan tujuan pembicara.

Sistem NLP modern berusaha mengintegrasikan semua tingkat bahasa ini untuk mencapai pemahaman bahasa yang lebih komprehensif. Melalui NLP, komputer dapat memahami bahasa manusia, dan melakukan pemrosesan untuk kebutuhan komputasi lainnya.

C. TF-IDF (Term Frequency-Inverse Document Frequency)

Term Frequency-Inverse Document Frequency (TF-IDF) adalah salah satu teknik vektorisasi teks yang paling umum digunakan dalam *Natural Language Processing* (NLP). Metode ini bertujuan untuk mengukur pentingnya suatu kata dalam sebuah dokumen atau korpus teks. Konsep dasarnya adalah bahwa kata yang sering muncul dalam suatu dokumen, namun jarang muncul di dokumen lain dalam korpus, dianggap cenderung lebih relevan dan informatif. TF-IDF menggabungkan dua metrik utama, yaitu:

1. *Term Frequency* (TF): Metrik ini menunjukkan seberapa sering suatu kata muncul dalam sebuah dokumen. TF yang tinggi mengindikasikan bahwa kata tersebut sangat relevan dengan dokumen tersebut.
2. *Inverse Document Frequency* (IDF): Metrik ini mengukur seberapa umum atau jarang suatu kata muncul dalam seluruh korpus. IDF yang tinggi mengindikasikan bahwa kata tersebut unik dan membedakan satu dokumen dengan dokumen lainnya.



Gambar 3. Cara kerja TF-IDF

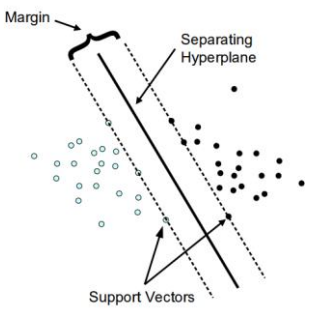
Sumber: https://www.researchgate.net/publication/354354484_Automated_Prediction_of_Good_Dictionary_EXamples_GDEX_A_Comprehensive_Experiment_with_Distant_Supervision_Machine_Learning_and_Word_Embedding-Based_Deep_Learning_Techniques (diakses pada 2 Januari 2025)

Mudahnya, TF-IDF diperoleh dari hasil perkalian TF dan IDF, berdasarkan rumus *TF-IDF(t, d) = TF(t, d) * IDF(t)* dengan TF(t, d) adalah frekuensi kata t dalam dokumen d, dan IDF(t) adalah *inverse document frequency* dari kata t, yang dihitung sebagai logaritma natural dari jumlah total dokumen dibagi dengan jumlah dokumen yang mengandung kata t.. Secara matematis, ditulis sebagai berikut.

$$a_{ij} = tf_{ij} \times \log\left(\frac{N}{df_i}\right)$$

D. SVM (Support Vector Machine)

Support Vector Machine (SVM) merupakan salah satu algoritma pembelajaran mesin yang digunakan untuk klasifikasi dan regresi. Secara umum, SVM mencari hyperplane yang memisahkan data ke dalam dua kelas dengan margin terbesar, yaitu jarak antara hyperplane dan titik data terdekat dari kedua kelas, yang disebut sebagai *support vectors*. Dengan memaksimalkan margin, SVM berusaha meminimalkan kesalahan klasifikasi dan meningkatkan generalisasi model. SVM dapat digunakan baik untuk masalah klasifikasi linier maupun non-linier, dengan menggunakan pemetaan data ke dalam ruang dimensi yang lebih tinggi.



Gambar 4. Ilustrasi Support Vector Machine
Sumber: https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=4850faab0aab5c4b40fcdd5a77d9e7626a163db5 (diakses pada 29 Desember 2024)

Kelebihan SVM mampu menangani data yang besar dan kompleks, serta mampu untuk bekerja dengan baik pada masalah klasifikasi dengan dimensi fitur yang tinggi. SVM juga dapat menangani masalah klasifikasi non-linier dengan menggunakan *kernel trick*, dan memiliki performa yang baik meskipun data tidak terdistribusi secara *uniform*. Namun, SVM juga memiliki beberapa kekurangan, seperti kebutuhan untuk memilih kernel yang tepat dan parameter yang optimal, serta kesulitan dalam menangani dataset yang sangat besar dengan waktu komputasi yang tinggi. Selain itu, SVM tidak memberikan probabilitas sebagai output. Hal ini dapat menjadi masalah dalam beberapa aplikasi yang membutuhkan interpretasi probabilistik.

III. IMPLEMENTASI

A. Metodologi

Pada analisis sentiment kalimat bahasa Indonesia, vektor memiliki peranan penting sebagai titik-titik dalam bidang berdimensi banyak, dengan setiap titik merepresentasikan dokumen/kalimat yang menjadi sumber model, maupun kalimat yang hendak dianalisa. Pada prosesnya meliputi hal-hal berikut.

1. Pemuatan dataset sebagai sumber data untuk melakukan analisis yang akuran. Pada makalah ini, data diperoleh dari *website* https://www.kaggle.com/datasets/ruditdota/sentimen-positif-negatif-indonesia yang diunggah oleh Raditya Pratama.
2. Pra-pemrosesan kalimat meliputi pembersihan, penyesuaian huruf kapital, dan stematisasi, yaitu pengembalian kata ke bentuk asalnya.
3. Pemrosesan kalimat menjadi vektor, yang direpresentasikan dalam bentuk list dengan elemen berupa angka-angka yang mewakili nilai dari dimensi vektor. Proses ini menggunakan algoritma TF-IDF.
4. Pembuatan dan pelatihan model SVM berdasarkan dataset yang telah divektorisasi tersebut.
5. Proses penginputan kalimat yang hendak dianalisis, pra-pemrosesan, dan vektorisasi, kemudian analysis prediksi sentiment kalimat tersebut berdasarkan model yang telah dibentuk.

B. Program Source Code

Program ditulis dalam bahasa Python menggunakan *library* luar sebagai alat utama, yaitu *pandas* untuk mengakses dataframe, *nltk* dan *Sastrawi* untuk mempermudah dalam NLP, serta *Scikit-Learn* untuk mempermudah dalam vektorisasi dan pemodelan. Berikut adalah tangkapan *source code* dalam implementasi program.

Fungsi load_dataset(filepath) digunakan untuk melakukan pemuatan dataset dari file CSV secara lokal.

```
# Step 1: Input Dataset
def load_dataset(filepath):
    """Load dataset from a CSV file."""
    return pd.read_csv(filepath)
```

Gambar 5. Load Dataset
Sumber: Dokumen Penulis

Fungsi preprocess_text(text) digunakan untuk melakukan pra-pemrosesan string kalimat menjadi kalimat yang mudah untuk divektorisasi.

```python
# Step 2: Preprocessing
def preprocess_text(text):
    """Clean and preprocess the text."""
    if not isinstance(text, str):
        text = str(text)  # Konversi nilai non-string ke string
    # Hapus angka
    text = re.sub(r'\d+', '', text)
    # Hapus tanda baca
    text = re.sub(r'[^\w\s]', '', text)
    # Ubah menjadi huruf kecil
    text = text.lower()
    # Hapus spasi ganda
    text = re.sub(r'\s+', ' ', text).strip()

    # Stop words removal
    stop_words = set(stopwords.words('indonesian'))
    wordList = [word for word in text.split(' ') if not word in stop_words]

    # Stemming
    factory = StemmerFactory()
    stemmer = factory.create_stemmer()
    stemWordList = [stemmer.stem(word) for word in wordList]

    # Convert back to text
    return ' '.join(stemWordList)
```

Gambar 6. Preprocessing
Sumber: Dokumen Penulis

Fungsi train_model(dataset, text_column, label_column) digunakan untuk melakukan vektorisasi menggunakan TF-IDF, sekaligus membuat dan melakukan *training* pada model SVE.

```python
# Step 3: Modelling using TF-IDF and Support Vector Machine
def train_model(dataset, text_column, label_column):
    """Train SVM model with TF-IDF for sentiment analysis."""
    # Preprocessing the dataset
    dataset[text_column] = dataset[text_column].apply(preprocess_text)

    # Split dataset into train and test sets
    X_train, X_test, y_train, y_test = train_test_split(
        dataset[text_column], dataset[label_column], test_size=0.2, random_state=42
    )

    # Vectorization using TF-IDF
    vectorizer = TfidfVectorizer(max_features=5000)
    X_train_tfidf = vectorizer.fit_transform(X_train)
    X_test_tfidf = vectorizer.transform(X_test)

    # Train SVM
    svm_model = SVC(kernel='linear', class_weight='balanced', probability=True)
    svm_model.fit(X_train_tfidf, y_train)

    # Evaluate the model
    y_pred = svm_model.predict(X_test_tfidf)
    print("Model Evaluation:")
    print(classification_report(y_test, y_pred))

    # Save vectorizer and model into a pipeline-like dictionary
    pipeline = {"vectorizer": vectorizer, "model": svm_model}
    return pipeline
```

Gambar 7. Modelling and Training
Sumber: Dokumen Penulis

Fungsi analyze_sentiment(pipeline, sentence) digunakan untuk melakukan proses analisis kalimat yang sudah diinput.

```python
# Step 4: Input Sentence for Analysis
def analyze_sentiment(pipeline, sentence):
    """Analyze sentiment of a given sentence."""
    # Preprocessing and Vectorization
    processed_sentence = preprocess_text(sentence)
    vectorized_sentence = pipeline["vectorizer"].transform([processed_sentence])

    # Classification
    prediction = pipeline["model"].predict(vectorized_sentence)[0]

    # Output Sentiment Result
    return prediction
```

Gambar 8. Proses Analisis Kalimat Input
Sumber: Dokumen Penulis

Adapun dataset yang digunakan dalam makalah ini terdiri dari 435 data dengan dua kolom, kolom ‘text’ berisi kalimat, dan kolom ‘label’ berisi sentimen positif atau negatif. Data ini berisi kalimat sehari-hari dalam bahasa Indonesia, yang memiliki kepolaran sentimen positif atau negatif. Tujuan dipilihnya data ini adalah untuk membuat model yang dapat digunakan secara general dalam menganalisis sentimen, serta lebih mudah dalam menganalisisnya ketimbang ulasan produk maupun opini publik.

newdata.csv > data

```
text,label
"Saya merasa sangat sedih dan tidak berdaya.",negatif
"Tidak ada yang peduli padaku, aku merasa sangat kesepian.",nega
"Semua terasa hampa dan tidak bermakna.",negatif
"Aku sering menangis tanpa alasan yang jelas.",negatif
"Rasanya seperti dunia ini tidak adil padaku.",negatif
"Aku sering merasa cemas tanpa alasan yang jelas.",negatif
"Tidurku selalu terganggu oleh mimpi buruk.",negatif
"Aku merasa tidak berharga dan tidak berguna.",negatif
"Kehidupan ini terasa sangat berat bagiku.",negatif
"Aku merasa tidak ada jalan keluar dari masalahku.",negatif
"Rasanya seperti aku tidak bisa menikmati apa pun lagi.",negatif
"Aku selalu merasa tegang dan tidak bisa rileks.",negatif
"Aku kehilangan minat pada hal-hal yang biasanya aku sukai.",neg
"Aku merasa terjebak dalam lingkaran kesedihan.",negatif
"Aku sering merasa sangat marah tanpa alasan.",negatif
"Aku merasa seperti beban bagi orang lain.",negatif
"Tidak ada hal yang membuatku bahagia.",negatif
"Aku sering merasa lelah meskipun tidak melakukan apa-apa.",nega
```

Gambar 9. Dataset yang digunakan
Sumber: Dokumen Penulis

C. Tampilan Program

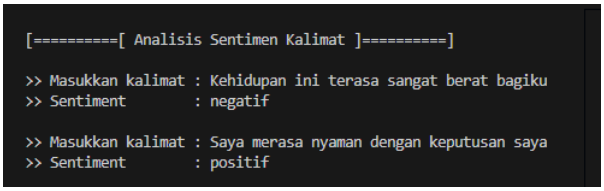
Penggunaan program meliputi proses input model atau dataset, dilanjut dengan masuk ke tampilan utama untuk menginput kalimat baru, lalu memberikan output berupa sentiment kalimat tersebut.

```
[==========[ Analisis Sentimen Kalimat ]==========]

[nltk_data] Downloading package punkt_tab to
[nltk_data]     C:\Users\Nazih\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt_tab is already up-to-date!
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\Nazih\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!

*note: enter if you don't have.
>> Masukkan file Model  :
>> Masukkan dataset     : ../newdata.csv

[program] loading dataset ...
[program] load dataset berhasil

>> Masukkan nama kolom kalimat : text
>> Masukkan nama kolom sentimen: label

[program] membuat model SVM ...
[program] melatih model SVM ...
Model Evaluation:
              precision    recall  f1-score   support

     negatif       0.95      0.89      0.92        46
     positif       0.89      0.95      0.92        41

    accuracy                           0.92        87
   macro avg       0.92      0.92      0.92        87
weighted avg       0.92      0.92      0.92        87

[program] model SVM siap digunakan

>> Ketik enter untuk masuk:
```

Gambar 10. Tampilan Awal Program
Sumber: Dokumen Penulis

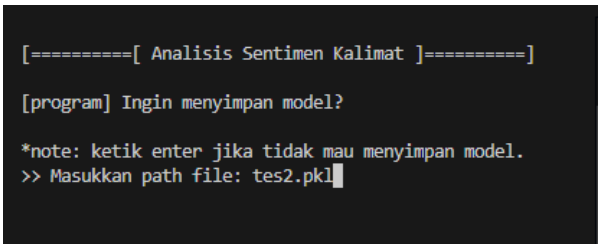
Setelah melakukan input dataset atau model, proses

selanjutnya adalah input kalimat yang hendak dianalisis.



Gambar 11. Tampilan Utama untuk Analisis
Sumber: Dokumen Penulis

Model yang telah dibentuk dari dataset, dapat disimpan dalam file dengan ekstensi *.pkl untuk mempercepat komputasi pada penggunaan selanjutnya.



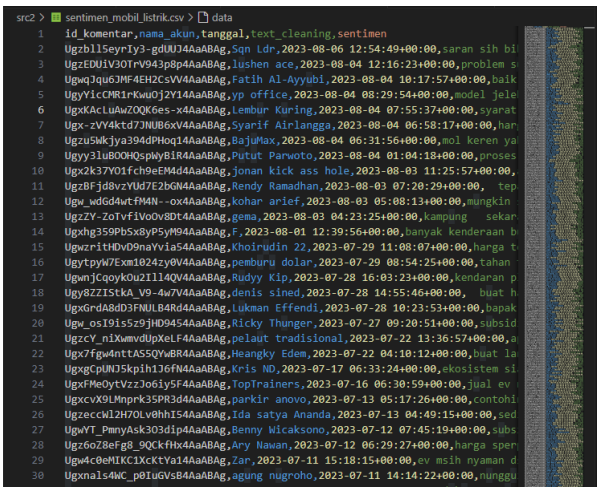
Gambar 12. Penyimpanan Model
Sumber: Dokumen Penulis

D. Pengujian

Pengujian program dilakukan untuk memastikan program dapat melakukan analisis sentimen kalimat dengan akurat. Pada pengujian ini, program diberi input berupa dataset lain yang memiliki kalimat dan label sentimen. Dataset ini diiterasi untuk dilakukan pengecekan sentiment oleh program, kemudian dilakukan validasi terhadap hasil analisis sentiment yang dihasilkan. Validasi berupa pembuatan *confusion matrix*, yaitu matriks yang menunjukkan ketepatan dalam memprediksi terhadap jawaban yang diinginkan.

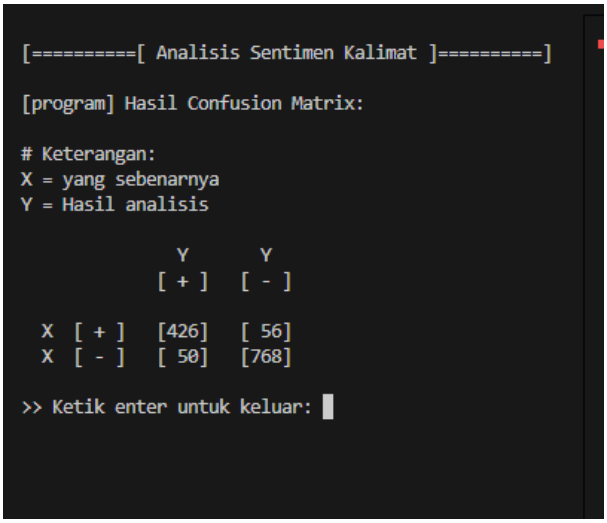
Data yang digunakan sebagai data uji pada makalah ini diperoleh dari *website* https://www.kaggle.com/datasets/billycemerson/analisis-sentimen-terkait-intensif-mobil-listrik yang diunggah oleh Biliarto Sastro Cemerson. Data ini berisi komentar-komentar masyarakat terkait pemeberlakuan intensif mobil listrik di Indonesia. Komentar-komentar yang terdapat dalam dataset ini merupakan komentar yang diambil dari berbagai video yang membahas tentang pemeberlakuan intensif mobil listrik pada media Youtube menggunakan API resmi dari Youtube.

Sebelum digunakan, dilakukan proses filter terhadap dataset uji untuk menghapus data dengan sentiment netral, sehingga menyisakan sentiment positif dan negatif saja, sesuai dataset yang digunakan dalam model.



Gambar 13. Potongan dataset yang digunakan untuk pengujian.
Sumber: Dokumen Penulis

Pengujian dilakukan terhadap 1300 kalimat berbeda dalam dataset uji. Hasil pengujian ditunjukkan oleh gambar berikut.



Gambar 14. Hasil pengujian
Sumber: Dokumen Penulis

Berdasarkan hasil *confusion matrix* pengujian yang telah dilakukan, diperoleh ketepatan sebesar 88,38%, yaitu 426 dari 482 kalimat, dalam memprediksi kalimat bersentimen positif, serta ketepatan sebesar 93,89%, yaitu 768 dari 818 kalimat, dalam memprediksi kalimat bersentimen negatif.

IV. PEMBAHASAN

A. Penerapan Vektor di Ruang Euclidean

Vektor di ruang Euclidean digunakan untuk merepresentasikan teks dalam bentuk numerik sehingga dapat dianalisis secara matematis oleh computer. Proses ini mencakup prapemrosesan kalimat, tokenisasi, dan vektorisasi hingga terbentuk matriks vektor. Representasi vektor mempermudah model untuk mendeteksi pola

semantik seperti hubungan antar kata dan intensitas sentiment. Proses vektorisasi kalimat setelah ditokenisasi dapat dilakukan dengan berbagai cara. Dalam makalah ini, cara yang digunakan adalah *Term Frequency-Inverse Document Frequency* (TF-IDF). Teknik ini memungkinkan pemilihan fitur teks yang lebih relevan dan informatif melalui identifikasi kata-kata kunci yang berkontribusi signifikan terhadap klasifikasi sentimen.

Terdapat serangkaian proses yang dilakukan oleh TF-IDF. Pertama, dimensi vektor ditentukan oleh banyaknya kata berbeda yang terdapat dalam korpus dataset, sebut saja sebanyak N. Setiap kalimat diubah menjadi vektor berdimensi N dengan setiap dimensinya mewakili bobot suatu kata unik dari korpus berdasarkan kata yang terdapat dalam kalimat tersebut. Besarnya bobot ini ditentukan oleh seberapa banyak kata/dimensi tersebut muncul pada kalimat (TF) dan seberapa jarang kata tersebut muncul pada seluruh dataset/dokumen (IDF). Semakin memenuhi TF dan IDF, semakin besar pula bobotnya. Dengan melakukan proses ini, teks kalimat dapat direpresentasikan dalma vektor numerik sehingga lebih mudah untuk dianalisis.

B. Peran SVM dalam Analisis Sentimen

Support Vector Machine (SVM) digunakan sebagai algoritma utama dalam klasifikasi sentiment berdasarkan vektor yang telah dihasilkan pada proses sebelumnya. SVM bekerja dengan memanfaatkan margin maksimum pada *hyperplane* untuk memisahkan data sentiment positif dan negatif. Setelah kalimat yang hendak dianalisis divektorisasi, selanjutnya disebut sebagai vektor input, vektor input dipetakan ke dalam ruang vektor berdimensi N, yaitu tempat di mana vektor-vektor dataset dipetakan. Setelah itu, SVM memprediksi klasifikasi vektor input menggunakan *separating hyperlane* dengan mempertimbangkan margin pada *support vectors* yang telah berlabel. Untuk meningkatkan performa, model SVM dilatih terlebih dahulu setelah vektor-vektor dataset dipetakan, sebelum siap digunakan pada vektor input. Proses latihan ini dilakukan dengan membagi vektor pada dataset menjadi vektor-vektor uji dan vektor-vektor latih untuk memperoleh model yang optimal.

C. Hasil Implementasi dan Pengujian

Setelah dilakukan implementasi program analisis sentimen kalimat bahasa Indonesia dan dilakukan pengujian, diperoleh hasil yang memuaskan. Program dapat memprediksi sentimen dengan akurasi 88,38% untuk kalimat bersentimen positif, dan 93,89% untuk kalimat bersentimen negatif. Hal ini karena program seperti mengabaikan kata-kata yang terlalu sering beririsan dengan kalimat yang berbeda sentimen, dan juga program berprilaku seperti membuat kata kunci dengan pembobotan yang besar untuk kata yang jarang muncul. Hal ini terjadi karena program menggunakan TF-IDF dalam vektorisasinya.

Meskipun memiliki performa yang baik untuk bahasa Indonesia, penerapan TF-IDF memiliki kelemahan pada kalimat dengan struktur bahasa yang tidak mudah dan banyak berdialektika karena TF-IDF hanya memvektorkan jumlah kata saja, tidak memvektorkan makna semantik dan hubungan antar struktur kata. Namun, hal ini tidak terlalu menjadi masalah pada bahasa Indonesia yang sederhana, sehingga dapat memperoleh hasil yang baik.

Selain TF-IDF, pada penerapan ini, SVM juga masih memiliki kelemahan, yaitu apabila hasil pemetaan vektor-vektor dataset memiliki margin yang tidak terlalu berbeda untuk sentimen yang berbeda. Hal ini dapat terjadi apabila kualitas dataset buruk, dan sulit untuk divektorisasi dengan benar. bisa juga disebabkan oleh dimensi vektor terlalu besar akibat tata bahasa yang terlalu kompleks. Selain itu, SVM juga lebih baik digunakan untuk klasifikasi biner, sehingga program memerlukan pendekatan yang berbeda untuk sentimen yang lebih variative seperti netral, atau sentimen yang berskala.

Secara garis besar, penerapan TF-IDF dan SVM dalam analisis sentimen positif atau negatif dari kalimat bahasa Indonesia sudah cukup baik untuk diterapkan.

V. KESIMPULAN

Penerapan vektor di ruang Euclidean pada analisis sentimen kalimat bahasa Indonesia terbukti memiliki efisiensi dan akurasi yang baik. Dengan merepresentasikan teks dalam bentuk vektor menggunakan TF-IDF, algoritma SVM dapat memanfaatkan pola jumlah kata dalam teks untuk membedakan sentimen positif dan negatif. Proses yang sistematis, mulai dari pra-pemrosesan hingga pelatihan model, memungkinkan analisis ini diterapkan secara luas pada berbagai konteks, seperti pemasaran, pemantauan isu sosial, layanan produk, dan pengembangan *chatbot*. Meski memiliki beberapa keterbatasan, seperti kekurangnannya dalam menganalisis stuktur kalimat berkompleksitas tinggi, pendekatan ini menunjukkan potensi signifikan dalam analisis data tidak terstruktur di media sosial.

REFERENSI

[1] R. Carey, Linear Algebra 2024 Notes. bookdown.org, 2024. [Online]. Available: https://bookdown.org/rachaelmcarey/lanotes/introduction-to-vectors.html

[2] D. Sun, "Lecture 9 Textual Data: Vector Space Model and TF-IDF," Stanford University DATASCI 112, Jan. 29, 2024. [Online]. Available: https://web.stanford.edu/class/datasci112/lectures/lecture9.pdf m

[3] E. D. Liddy, "Natural Language Processing," SURFACE at Syracuse University, 2001. [Online]. Available: https://surface.syr.edu/cgi/viewcontent.cgi?article=1043&context=istpub

[4] J. Eisentein, Natural Language Processing. MIT Press, Nov. 13, 2018. [Online]. Available: https://princeton-nlp.github.io/cos484/readings/eisenstein-nlp-notes.pdf

[5] Chopra, A. Prashar, and C. Sain, "Natural Language Processing," International Journal of Technology Enhancements and Emerging Engineering Research, vol. 1, no. 4, 2013. [Online]. Available:

https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=eeace1d14e266a5cd44fe781a874c662928602fd

[6] Trstenjak, S. Mikac, and D. Donko, "KNN with TF-IDF based Framework for Text Categorization," Procedia Engineering, vol. 69, 2013. [Online]. Available: https://doi.org/10.1016/j.proeng.2014.03.129

[7] Boswell, "Introduction to Support Vector Machines," Aug. 6, 2002. [Online]. Available: https://pzs.dstu.dp.ua/DataMining/svm/bibl/IntroToSVM.pdf

[8] V. Jakkula, "Tutorial on Support Vector Machine (SVM)," School of EECS, Washington State University, 2011. [Online]. Available: https://course.khoury.northeastern.edu/cs5100f11/resources/jakkula.pdf

[9] W. Liao, "Chapter 9 – Support Vector Machines," School of Mathematics, Georgia Institute of Technology, 2019. [Online]. Available: https://wliao60.math.gatech.edu/19FallDataScience/LectureSlides/Chapter9.pdf

[10] D. Meyer, "Support Vector Machines: The Interface to libsvm," Jan. 6, 2009. [Online]. Available: https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=4850faab0aab5c4b40fcdd5a77d9e7626a163db5

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Jatinangor, 2 Januari 2025

Muhamad Nazih Najmudin
NIM: 13523144