

## ANALISIS SENTIMEN TERHADAP ULASAN PADA WHIZ PRIME HOTEL SUDIRMAN PEKANBARU MENGGUNAKAN SUPPORT VECTOR MACHINE DAN NAIVE BAYES

Alfariz Priofani<sup>1</sup>, \*Mariza Devega<sup>2</sup>, Yuhelmi<sup>3</sup>, Walhidayat<sup>4</sup>

(<sup>1,2</sup> Program Studi Teknik Informatika, <sup>3</sup>Program Studi Sistem Informasi, <sup>4</sup>Program Studi Bisnis Digital Fakultas Ilmu Komputer Universitas Lancang Kuning)  
Jl. Yos Sudarso KM. 8 Rumbai, Pekanbaru, Riau, telp. 0811 753 2015)  
e-mail: <sup>1</sup>alfariz.unilak2020@gmail.com, <sup>2</sup>yuhelmi@unilak.ac.id, <sup>3</sup>marizadevega@unilak.ac.id,  
<sup>4</sup>walhidayat@unilak.ac.id

### Abstrak

Kemajuan teknologi informasi telah mengubah cara bisnis dijalankan, termasuk dalam industri perhotelan melalui platform perdagangan elektronik. Dalam penelitian ini, telah dilakukan analisis sentimen terhadap ulasan pelanggan Whiz Prime Hotel Sudirman Pekanbaru yang diperoleh dari situs Traveloka. Sebanyak 4336 ulasan dikumpulkan, kemudian diproses menggunakan metode frekuensi istilah-frekuensi invers dokumen. Setelah tahap praproses, sebanyak 2165 ulasan dianalisis. Hasilnya menunjukkan distribusi sentimen positif sebesar 72,79 persen, negatif 15,38 persen, dan netral 11,82 persen. Dua algoritma klasifikasi, yaitu mesin vektor pendukung dan Naive Bayes, telah digunakan dalam analisis ini. Mesin vektor pendukung menghasilkan akurasi tertinggi sebesar 77 persen, sedangkan Naive Bayes mencapai akurasi sebesar 73 persen. Temuan ini memberikan wawasan bagi pihak manajemen hotel untuk memahami kualitas layanan yang ditawarkan dan meningkatkan kepuasan pelanggan.

**Kata kunci:** Analisis Sentimen, Support Vector Machine, Naive Bayes, Hotel, Machine Learning

### Abstract

Advances in information technology have changed the way businesses are run, including in the hospitality industry through electronic commerce platforms. In this research, sentiment analysis of Whiz Prime Hotel Sudirman Pekanbaru customer reviews obtained from Traveloka website has been conducted. A total of 4336 reviews were collected, then processed using the document inverse term-frequency method. After the preprocessing stage, 2165 reviews were analyzed. The results showed a positive sentiment distribution of 72.79 percent, negative 15.38 percent, and neutral 11.82 percent. Two classification algorithms, namely support vector machine and Naive Bayes, have been used in this analysis. The support vector machine produced the highest accuracy of 77 percent, while Naive Bayes achieved an accuracy of 73 percent. The findings provide insights for hotel management to understand the quality of services offered and improve customer satisfaction.

**Keywords:** Sentiment Analysis, Support Vector Machine, Naive Bayes, Hotel, Machine Learning

### 1. PENDAHULUAN

Kemajuan teknologi yang pesat telah memudahkan manusia dalam berbagai aspek kehidupan, termasuk bidang bisnis. Berdasarkan survei APJII tahun 2016, jumlah pengguna internet di Indonesia mencapai 132,7 juta, sekitar 51,8% dari total penduduk, yang memberikan dampak positif bagi dunia usaha[1]. Salah satu sektor yang berkembang pesat adalah *Electronic-Commerce (E-Commerce)*, yang mempermudah konsumen dalam mendapatkan barang atau jasa[2]. Traveloka, sebagai salah satu platform *E-Commerce* terkemuka di Indonesia, menyediakan layanan pemesanan tiket pesawat dan hotel yang mendominasi pasar. Berdasarkan

survei *DailySocial*, 50,67% masyarakat Indonesia memilih Traveloka untuk reservasi hotel, sementara riset Avara *Research* menunjukkan 79% generasi milenial menggunakan Traveloka untuk pemesanan tiket dan hotel.

Whiz Prime Hotel Sudirman Pekanbaru, hotel bintang 3 yang berlokasi strategis di Pekanbaru, memanfaatkan platform seperti Traveloka untuk meningkatkan pemesanan dan mendapatkan feedback dari pelanggan. Sebelum menggunakan Traveloka, hotel ini mengandalkan metode kuesioner, yang kurang efektif karena tingkat partisipasi yang rendah. Melalui Traveloka, pelanggan dapat memberikan ulasan yang membantu hotel meningkatkan layanan dan menarik tamu baru. Namun, banyaknya komentar yang masuk sulit dianalisis secara manual, sehingga diperlukan teknik pengolahan data, seperti analisis sentimen.

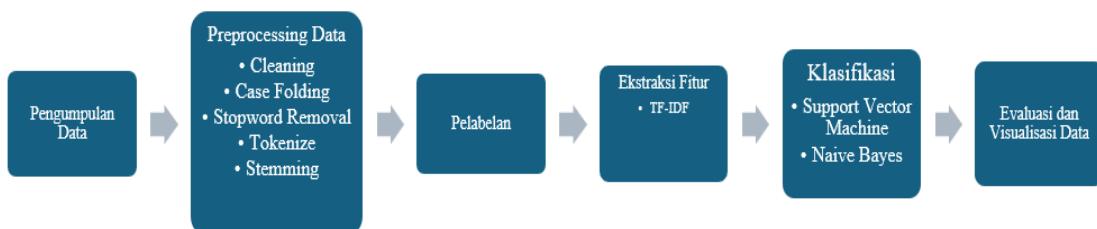
Analisis sentimen, atau opinion mining, digunakan untuk mengelola komentar pelanggan dengan mengidentifikasi tingkat kepuasan dan sikap pengguna terhadap layanan hotel[3]. Dalam penelitian ini, digunakan metode klasifikasi seperti *Support Vector Machine* (SVM) dan *Naive Bayes*. Studi sebelumnya menunjukkan bahwa SVM dengan kernel linier sering kali lebih unggul dibandingkan *Naive Bayes* dalam analisis sentimen. Penelitian yang dilakukan sebelumnya mencatat bahwa SVM mencapai akurasi 93% dengan presisi 100%, sementara *Naive Bayes* menghasilkan akurasi 86% dengan presisi 87%[4]. Studi lain juga menunjukkan keunggulan SVM dengan kernel RBF dalam analisis sentimen di Twitter, dengan akurasi 86,04% dibandingkan *Naive Bayes* yang mencapai 75,47%[5].

Dengan menggunakan metode ini, Whiz Prime Hotel Sudirman Pekanbaru dapat mengelola komentar pelanggan secara efektif, meningkatkan layanan, dan mempertahankan daya saing di industri perhotelan yang semakin kompetitif.

## 2. METODE PENELITIAN

Pada penelitian ini proses yang dijalankan yaitu melakukan klasifikasi sebuah sentimen dari ulasan yang telah didapat dan diimplementasikan dengan *Algoritma Naïve Bayes* sehingga dapat dihasilkan hasil akurasi dari sebuah sentimen. Data diproses menggunakan *website Google Colab* untuk mengambil data ulasan dan menjalankan proses klasifikasi data.

Berikut akan ditampilkan alur dari penelitian ini:



**Gambar 1** Metode Penelitian

### 2.1. Pengumpulan Data

Dalam melakukan pengambilan data, penelitian ini akan menggunakan bantuan *extension browser* yaitu *Web Scraper*. Dengan mengatur extension tersebut terlebih dahulu agar mendapatkan data ulasan yang dibutuhkan. *Dataset* yang telah diubah kedalam format csv sebelum melakukan *preprocessing* dilakukan data *preparation* yaitu menghapus kolom yang

tidak diperlukan seperti: rating dan tanggal, menghapus data duplikat dan *missing value*. Pada penelitian ini *dataset* semula berjumlah 4336 data, setelah dilakukannya *data preparation* menjadi 2166. Setelah itu baru dilakukan tahap *text preprocessing*. *Dataset* akan dilakukan pengolahan menggunakan bahasa pemrograman *Python* dengan *Google Colab*.

### **2.2. Preprocessing Data**

Pra-pemrosesan adalah tahap penting yang dilakukan setelah pengumpulan *dataset* dan sebelum proses klasifikasi sentimen, bertujuan untuk mengolah data agar lebih bersih, konsisten, dan siap digunakan dalam proses selanjutnya [6]. Proses ini mencakup beberapa langkah utama, seperti *cleaning*, yang membersihkan data dari karakter non-abjad seperti titik (.), koma (,), tanda tanya (?), tanda seru (!), dan karakter lainnya. *Case folding* mengubah teks menjadi huruf kecil sehingga hanya huruf "a" hingga "z" yang diterima, sedangkan *stopword removal* menghilangkan kata-kata yang tidak memiliki makna signifikan, seperti "a", "and", "the", dan "but", yang tidak memengaruhi proses klasifikasi sentimen. Selanjutnya, tokenizing memecah kalimat menjadi kata-kata individu, atau token, contohnya "kami mahasiswa jawa timur" menjadi "kami, mahasiswa, jawa, timur" [7]. Proses terakhir adalah *stemming*, yang mengubah kata menjadi bentuk dasarnya untuk menyeragamkan kata, seperti "membantu" menjadi "bantu" dengan menghilangkan imbuhan seperti prefiks, sufiks, dan konfiks [8]. Langkah-langkah ini memastikan data yang lebih bersih dan terstruktur, sehingga meningkatkan efisiensi dan akurasi dalam proses *machine learning*.

### **2.3. Pelabelan**

Pelabelan data dalam penelitian ini dilakukan secara otomatis menggunakan *library Python TextBlob* dengan konsultasi sebelumnya kepada ahli. Data diberi tiga label sentimen: positif, negatif, dan netral. Label positif mencerminkan emosi seperti senang dan puas, serta kata-kata yang bersifat mendukung atau memuji. Label negatif mencerminkan emosi seperti marah, kesal, dan kecewa, serta kata-kata yang bersifat keluhan atau sindiran. Sementara itu, label netral diberikan pada teks yang bersifat objektif, tanpa menunjukkan emosi positif atau negatif yang kuat.

### **2.4. TF-IDF**

*Term Frequency-Inverse Document Frequency* (TF-IDF) adalah algoritma yang dapat digunakan untuk menganalisis sebuah kalimat dengan sekumpulan dokumen dan metode untuk menghitung bobot setiap kata. Rumus TF-IDF adalah kombinasi dari perhitungan *Term Frequency* (TF) dan *Inverse Document Frequency* (IDF) dengan mengalikan nilai TF dan IDF. Tujuan penggunaan TF-IDF adalah untuk mengekstrak kata menggunakan *Machine Learning* ke dalam bentuk numerik agar data dapat diolah ke tahap berikutnya [9]. Pada *Term Frequency*, nilai frekuensi ini sering muncul dalam dokumen. Semakin sering suatu kata muncul dalam dokumen, semakin besar bobot atau nilai yang diberikan.

### **2.5. Metode Support Vector Machine**

*Support Vector Machine* (SVM) adalah metode pembelajaran linier yang bertujuan menemukan *hyperplane* optimal untuk memisahkan dua kelas, positif dan negatif, dengan memaksimalkan jarak antar kelas [10]. Dalam 2-D, *hyperplane* digambarkan sebagai garis, dalam 3-D sebagai bidang, dan pada ruang berdimensi lebih tinggi tetap disebut *hyperplane*. SVM bekerja dengan memperhitungkan data terluar yang paling dekat dengan *hyperplane*, disebut *support vector*, yang berperan penting karena letaknya yang hampir tumpang tindih dengan kelas lain, sehingga menjadi objek paling sulit untuk diklasifikasikan. Efektivitas SVM sangat bergantung pada jenis fungsi kernel yang digunakan, seperti *linear*, *polynomial*, dan *radial*. Kernel *linear* dianggap lebih cepat dibandingkan kernel *nonlinier* dan memberikan akurasi yang baik untuk masalah berdimensi tinggi, seperti klasifikasi dokumen[11]. Untuk klasifikasi data multikelas, SVM dapat menggunakan pendekatan satu formula untuk semua data atau teknik satu-ke-satu dengan SVM

biner. Banyak penelitian melaporkan SVM sebagai salah satu metode paling akurat untuk klasifikasi teks, berkat kemampuannya menemukan *hyperplane* yang maksimal antara dua kelas.

### 2.6. Metode *Naive Bayes*

Metode *Naive Bayes* adalah algoritma klasifikasi probabilistik sederhana yang bekerja dengan menghitung probabilitas berdasarkan frekuensi kombinasi nilai dalam dataset, menggunakan pendekatan statistik berbasis conditional probabilities atau probabilitas bersyarat [12]. Metode ini didasarkan pada *Teorema Bayes*, yang pertama kali dikemukakan oleh *Thomas Bayes*, dengan asumsi bahwa variabel-variabel bersifat independen atau tidak saling bergantung [13]. Salah satu keunggulan utama *Naive Bayes* adalah kemampuannya menggunakan jumlah data pelatihan yang relatif kecil untuk menentukan parameter yang diperlukan dalam proses klasifikasi. Karena asumsi independensi antar variabel, algoritma ini hanya memerlukan informasi variabel dari sebuah kelas, bukan keseluruhan matriks kovarians, sehingga menjadikannya efisien dalam proses komputasi [14].

### 2.7. Evaluasi

Pada tahap evaluasi model dilakukan untuk mengukur akurasi (*accuracy*), presisi (*precision*), *recall*, dan *F1-Score* menggunakan *confusion matrix*. Proses evaluasi ini diterapkan pada ulasan yang telah diklasifikasikan menggunakan *algoritma Naive Bayes* dan *Support Vector Machine*. Tujuan utama evaluasi adalah untuk menilai sejauh mana model mampu secara efektif mengidentifikasi dan mengklasifikasikan ulasan pelanggan menjadi sentimen positif, negatif, atau netral. Hasil evaluasi memberikan wawasan mendalam tentang performa dan efisiensi kedua algoritma, sehingga dapat menentukan metode yang paling optimal untuk menganalisis sentimen ulasan pelanggan.

## 3. HASIL DAN PEMBAHASAN

### 3.1 Hasil

**Tabel 1. Hasil**

<b>Metode</b>	<b>Test ing</b>	<b>Trai ning</b>	<b>Akurasi</b>	<b>Precision</b>			<b>Recall</b>			<b>F1 Score</b>		
				<b>+</b>	<b>O</b>	<b>—</b>	<b>+</b>	<b>O</b>	<b>—</b>	<b>+</b>	<b>O</b>	<b>—</b>
<b>SVM</b>	10	90	77%	88%	29%	79%	84%	26%	77%	86%	27%	78%
	20	80	75%	88%	18%	74%	80%	23%	80%	84%	20%	77%
<b>NB</b>	10	90	73%	87%	20%	77%	82%	35%	70%	84%	26%	73%
	20	80	71%	88%	17%	75%	78%	36%	67%	83%	23%	71%

Dalam penelitian ini, dilakukan analisis sentimen terhadap ulasan pelanggan Whiz Prime Hotel Sudirman Pekanbaru menggunakan dua metode, yaitu *Support Vector Machine* (SVM) dan *Naive Bayes* (NB). Pengujian dilakukan dengan membagi data menjadi dua jenis, yakni data Testing dan data Training. Pada metode SVM, ketika data Testing sebesar 10% dan Training 90%, diperoleh akurasi sebesar 77% dengan nilai *Precision* kelas Positif 88%, Netral 29%, dan Negatif 79%. Nilai *Recall* untuk kelas Positif adalah 84%, Netral 26%, dan Negatif 77%, sedangkan *F1 Score* untuk kelas Positif 86%, Netral 27%, dan Negatif 78%. Ketika data Testing dinaikkan menjadi 20% dan Training 80%, akurasi metode SVM menurun menjadi 75%, dengan nilai *Precision* kelas Positif 88%, Netral 18%, dan Negatif 74%. Nilai *Recall* untuk kelas Positif 80%, Netral 23%, dan Negatif 80%, sedangkan nilai *F1 Score* untuk kelas Positif 84%, Netral 20%, dan Negatif 77%.

Sementara itu, pada metode *Naive Bayes* (NB), ketika data Testing 10% dan Training 90%, diperoleh akurasi sebesar 73% dengan nilai *Precision* kelas Positif 87%, Netral 20%, dan

Negatif 77%. Nilai *Recall* pada kelas Positif adalah 82%, Netral 35%, dan Negatif 70%, sedangkan *F1 Score* untuk kelas Positif 84%, Netral 26%, dan Negatif 73%. Ketika data Testing 20% dan *Training* 80%, akurasi metode *Naive Bayes* menurun menjadi 71%, dengan nilai *Precision* kelas Positif 88%, Netral 17%, dan Negatif 75%. Nilai *Recall* yang diperoleh adalah 78% untuk kelas Positif, 36% untuk Netral, dan 67% untuk Negatif, sedangkan *F1 Score* untuk kelas Positif 83%, Netral 23%, dan Negatif 71%.

Hasil penelitian menunjukkan bahwa metode *Support Vector Machine* (SVM) memiliki performa yang lebih baik dibandingkan dengan *Naive Bayes* (NB) dalam analisis sentimen ulasan pada Whiz Prime Hotel Sudirman Pekanbaru.

### 3.2 Pembahasan

#### 3.2.1 Pengambilan Data

Penelitian yang telah dilakukan dari proses pengumpulan data menggunakan Teknik *web scrapping* pada ulasan Whiz Prime Hotel Sudirman Pekanbaru didapatkan data ulasan sebanyak 4336 ulasan dan telah dilakukan *preprocessing* menjadi 2165 dengan data sentimen positif (2) sebanyak 1260 data, kemudian untuk sentimen netral (0) sebanyak 201 data dan data sentimen negatif (1) sebanyak 704 data.

#### 3.2.2 Text Processing

Pada tahap *preprocessing* dilakukan beberapa tahap untuk mendapatkan data acuan yang siap untuk diproses. Tahap yang dilakukan sebagai berikut

##### 1. Cleaning

Pada tahapan ini, data komentar mengalami pembersihan dengan menghilangkan tab, *newline*, *backslash*, karakter non-ASCII, *mention*, *link*, *hashtag*, URL, angka, tanda baca, serta karakter tunggal yang berdiri sendiri untuk mengurangi *noise* dan memastikan hanya kata-kata relevan yang tersisa, kemudian hasil dari proses *cleaning* ini ditampilkan dalam beberapa baris untuk direview dan disimpan dalam *DataFrame* baru.

**Tabel 2.** Hasil Cleansing

Ulasan	Ulasan_Cleaning
Alhamdulillah memuaskan, tempat strategis, kamar bersih, pelayanan juga mantap, terima kasih. 😊	Alhamdulillah memuaskan tempat strategis kamar bersih pelayanan juga mantap terima kasih
Nyaman dan bersih. Walaupun kamarnya smoking room, tapi tidak bau rokok, good. ☺	Nyaman dan bersih Walaupun kamarnya smoking room tapi tidak bau rokok good

##### 2. Case Folding

Pada tahapan ini, case folding berfungsi untuk mengubah semua huruf menjadi *lowercase* (huruf kecil), kemudian hasil dari proses ini ditampilkan beberapa baris untuk direview dan disimpan dalam *DataFrame* baru.

**Tabel 3.** Case Folding

Ulasan_Cleaning	Ulasan_Case Folding
Alhamdulillah memuaskan tempat strategis kamar bersih pelayanan juga mantap terima kasih	alhamdulillah memuaskan tempat strategis kamar bersih pelayanan juga mantap terima kasih
Nyaman dan bersih Walaupun kamarnya smoking room tapi tidak bau rokok good	nyaman dan bersih walaupun kamarnya smoking room tapi tidak bau rokok good

### 3. Tokenizing

Pada proses *tokenizing*, teks dipisahkan menjadi token berupa kata, frasa, atau simbol menggunakan *library nltk* yang di-install dan di-import, kemudian hasilnya ditampilkan dan dibandingkan dengan *DataFrame* sebelumnya.

**Tabel 4.** Hasil Tokenizing

Ulasan_Case Folding	Ulasan_Tokenizing
alhamdulillah memuaskan tempat strategis kamar bersih pelayanan juga mantap terima kasih	['alhamdulillah', 'memuaskan', 'tempat', 'strategis', 'kamar', 'bersih', 'pelayanan', 'juga', 'mantap', 'terima', 'kasih']
nyaman dan bersih walaupun kamarnya smoking room tapi tidak bau rokok good	['nyaman', 'dan', 'bersih', 'walaupun', 'kamarnya', 'smoking', 'room', 'tapi', 'tidak', 'bau', 'rokok', 'good']

### 4. Stopwords Removal

Pada tahap ini, peneliti menambahkan beberapa kata dalam proses stopwords removal menggunakan fungsi *extend ()*.

**Tabel 5.** Hasil Stopwords Removal

Ulasan_Tokenizing	Ulasan_Stopwords Removal
['alhamdulillah', 'memuaskan', 'tempat', 'strategis', 'kamar', 'bersih', 'pelayanan', 'juga', 'mantap', 'terima', 'kasih']	['memuaskan', 'strategis', 'kamar', 'bersih', 'pelayanan', 'mantap', 'terima', 'kasih']
['nyaman', 'dan', 'bersih', 'walaupun', 'kamarnya', 'smoking', 'room', 'tapi', 'tidak', 'bau', 'rokok', 'good']	['nyaman', 'bersih', 'kamarnya', 'smoking', 'room', 'bau', 'rokok', 'good']

### 5. Stemming

Pada tahap stemming, kata-kata berimbuhan dalam dokumen diubah menjadi kata dasarnya

**Tabel 6.** Hasil Stemming

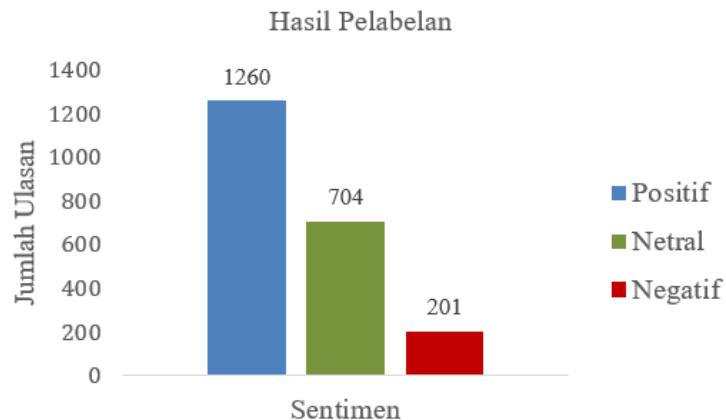
Ulasan_Tokenizing	Ulasan_Stemming
['memuaskan', 'strategis', 'kamar', 'bersih', 'pelayanan', 'mantap', 'terima', 'kasih']	['muas', 'strategis', 'kamar', 'bersih', 'ber', 'pelayan', 'mantap', 'terima', 'kasih']
['nyaman', 'bersih', 'kamarnya', 'smoking', 'room', 'bau', 'rokok', 'good']	['nyaman', 'bersih', 'kamar', 'smoking', 'room', 'bau', 'rokok', 'good']

### 3.2.3 Pelabelan

Tahapan pelabelan data pada penelitian ini akan menggunakan pelabelan otomatis menggunakan *Library python textblob* dan sebelumnya sudah berkonsultasi dengan ahli pakar. Pelabelan data terdiri dari 3 sentimen yaitu: Label Positif, Label Negatif, dan Label Netral.

**Tabel 7.** Hasil dari Pelabelan yang telah dilakukan

Sentimen	Jumlah
Positif	1260
Netral	704
Negatif	201



Gambar 1. Diagram Bar Hasil dari Pelabelan

Dari berbagai ulasan pelanggan Whiz Prime Hotel Sudirman Pekanbaru, kata-kata yang paling dominan atau sering muncul ditampilkan pada gambar berikut.



Gambar 2. Wordcloud Sentimen Pelanggan terhadap Whiz Prime Hotel Sudirman Pekanbaru

Gambar ini menampilkan visualisasi kata-kata yang paling sering muncul dalam ulasan pelanggan berdasarkan kategori sentimen.

1. Gambar sebelah kiri menunjukkan wordcloud untuk ulasan positif, dengan kata-kata dominan seperti kamar, hotel, baik, bagus, dan bersih.
2. Gambar tengah menunjukkan wordcloud untuk ulasan netral, dengan kata-kata seperti kamar mandi, sarapan, layanan, dan lokasi.
3. Gambar sebelah kanan menampilkan wordcloud untuk ulasan negatif, dengan kata-kata dominan seperti sangat, ac, dingin, parkir, dan mandi.

Visualisasi ini membantu mengidentifikasi aspek-aspek layanan hotel yang paling banyak dibahas pelanggan dalam masing-masing kategori sentimen.

### 3.2.4 Fitur Ekstraksi

TF-IDF (*Term Frequency-Inverse Document Frequency*) adalah metode pembobotan yang mengintegrasikan *term frequency* (frekuensi kemunculan kata) dengan *inverse document frequency* (frekuensi terbalik dokumen). Dalam penelitian ini, metode TF-IDF digunakan untuk memilih fitur berupa bobot kata yang berfungsi sebagai hasil ringkasan. Proses penerapan TF-IDF dilakukan menggunakan *TfidfVectorizer* dari *scikit-learn*, yang mengubah teks menjadi representasi numerik berdasarkan perhitungan bobot kata menggunakan formula TF-IDF[15]. Data dalam penelitian ini dibagi dengan perbandingan 90:10, di mana 90% digunakan sebagai data training dan 10% sebagai data testing.

```
# TF-IDF Vectorization
tfidf = TfidfVectorizer(max_features=5000)
tfidf_df = tfidf.fit_transform(data['cleanhotel']).toarray()

# Menyimpan rasio yang ingin diuji
test_sizes = [0.1, 0.2]
# Tahap TF-IDF
X = data['cleanhotel'].values.astype('U')
sentimen = data['sentimen_check']
tfidf_vectorizer = TfidfVectorizer()
tfidf_vector = tfidf_vectorizer.fit_transform(X)
print(tfidf_vector.shape)
print(tfidf_vector)

# Visualisasi Dataframe Dari Hasil Pembobotan Kata Menggunakan TF IDF
df = pd.DataFrame(tfidf_vector.todense().T,
                   index=tfidf_vectorizer.get_feature_names_out(),
                   columns=[f'D{i+1}' for i in range(len(X))])

Bentuk Data Hasil dari Tahap TF-IDF :
(2165, 2288)
Hasil dari Tahap TF-IDF :
(0, 797)      0.10313273418845492
(0, 1827)     0.0973999732206307
(0, 280)       0.17765996765432368
(0, 1088)     0.07287492032157114
(0, 5)         0.049790692630908316
(0, 1115)     0.0973999732206307
(0, 2271)     0.06829698283612266
```

Gambar 3. Source Code TF IDF

### 3.2.5 Klasifikasi

Setelah dilakukan pembobotan kata, tahapan selanjutnya adalah perhitungan dan pemisahan terhadap model klasifikasi, yaitu *Support Vector Machine* dan *Naive Bayes*. Sebelum tahapan ini dilakukan, data ulasan dalam bentuk teks perlu dikonversikan menjadi bentuk numerik atau angka. Untuk itu, digunakan metode TF-IDF tahap selanjutnya merupakan proses ketika melakukan pemodelan dan pengujian pada dataset dengan proporsi pembagian data training dan data testing yaitu 90:10, dan 80:20 dengan menggunakan 2 perbandingan data tersebut akan mendapatkan proporsi pembagian data terbaik.

```
# Rasio yang ingin diuji
test_sizes = [0.10, 0.20]

# Urutan label sentimen yang konsisten
sentiment_labels = ['negatif', 'neutra', 'positif']

for test_size in test_sizes:
    # Splitting the data
    X_train, X_test, y_train, y_test = train_test_split(tfidf_vector, data['sentimen_check'],
                                                       test_size=test_size, random_state=42)

    # Mengatasi ketidakseimbangan kelas dengan SMOTE
    smote = SMOTE(random_state=42)
    X_train_res, y_train_res = smote.fit_resample(X_train, y_train)

    # Initialize and train SVM classifier
    svm_classifier = SVC(kernel='linear')
    svm_classifier.fit(X_train_res, y_train_res)

    # Calculate SVM scores
    svm_train_score = svm_classifier.score(X_train_res, y_train_res)
    svm_test_score = svm_classifier.score(X_test, y_test)

    # Predict using the trained model
    svm_test_pred = svm_classifier.predict(X_test)
```

Gambar 4. Source Code Support Vector Machine

```
test_sizes = [0.10, 0.20]

# Urutan label sentimen yang konsisten
sentiment_labels = ['negatif', 'netral', 'positif']

# Iterasi melalui rasio yang ingin diuji
for test_size in test_sizes:
    # Splitting the data
    X_train, X_test, y_train, y_test = train_test_split(tfidf_vector, data['sentimen_check'],
                                                        test_size=test_size, random_state=42)

    # Mengatasi ketidakseimbangan kelas dengan SMOTE
    smote = SMOTE(random_state=42)
    X_train_res, y_train_res = smote.fit_resample(X_train, y_train)

    # Initialize and train Naive Bayes classifier
    nb_classifier = MultinomialNB()
    nb_classifier.fit(X_train_res, y_train_res)

    # Calculate Naive Bayes scores
    nb_train_score = nb_classifier.score(X_train_res, y_train_res)
    nb_test_score = nb_classifier.score(X_test, y_test)

    # Predict using the trained model
    nb_train_pred = nb_classifier.predict(X_train_res)
    nb_test_pred = nb_classifier.predict(X_test)

    # Evaluate the Naive Bayes model
    nb_train_score = accuracy_score(y_train_res, nb_train_pred)
    nb_test_score = accuracy_score(y_test, nb_test_pred)
```

Gambar 5. Source Code Naïve Bayes

*Confusion matrix* digunakan untuk mengevaluasi akurasi *algoritma Support Vector Machine* (SVM) dan *Naive Bayes* pada *dataset*. *Matrix* ini mencakup nilai *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN). Berdasarkan hasil ini, metrik seperti akurasi, *precision*, *recall*, dan *F1-score* dihitung untuk memberikan gambaran kinerja kedua model secara lebih mendalam. *Confusion matrix* membantu mengidentifikasi sejauh mana model mampu melakukan prediksi dengan benar dan mengklasifikasikan data secara akurat, sehingga memberikan pemahaman yang komprehensif tentang efektivitas algoritma SVM dan *Naive Bayes*.

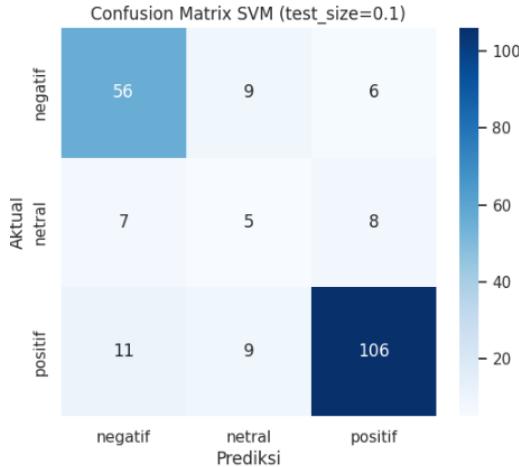
### 3.2.6 Evaluasi dan visualisasi

Pengujian dilakukan menggunakan metode confusion matrix. Selama proses evaluasi, *algoritma Support Vector Machine* (SVM) dan *Naive Bayes* digunakan untuk menghitung metrik performa, yaitu akurasi (*accuracy*), presisi (*precision*), *recall*, dan *F1-score*.

#### 3.2.6.1 Support Vector Machine

Dalam pengujian menggunakan *Machine Learning*, metode *Support Vector Machine* (SVM) mencapai akurasi sebesar 77%. Gambar di bawah menunjukkan *confusion matrix* beserta nilai akurasi, *precision*, *recall*, dan *F1-score* yang terkait dengan pengujian menggunakan *dataset*.

```
== Model SVM (test_size=0.1) ==
Akurasi pada Data Testing: 0.7695852534562212
Laporan Klasifikasi pada Data Testing:
precision    recall   f1-score   support
negatif      0.76     0.79     0.77      71
netral       0.22     0.25     0.23      20
positif      0.88     0.84     0.86     126
accuracy        -       -       -       217
macro avg      0.62     0.63     0.62     217
weighted avg   0.78     0.77     0.77     217
```



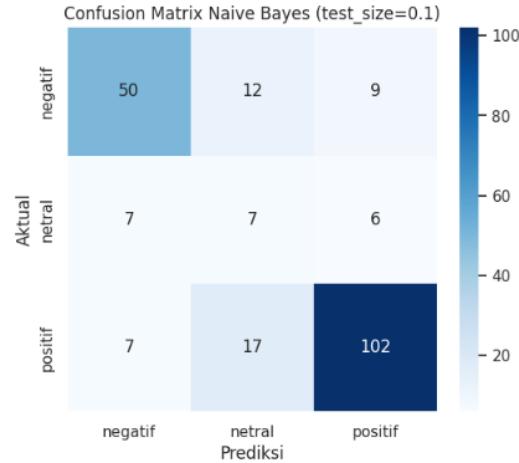
**Gambar 6.** Akurasi Menggunakan Support Vector Machine dan Evaluasi Confusion Matrix

### 3.2.6.1 Naïve Bayes

Dalam pengujian menggunakan *Machine Learning*, metode *Naive Bayes* mencapai akurasi sebesar 73%. Gambar di bawah menunjukkan confusion matrix beserta nilai akurasi, *precision*, *recall*, dan *F1-score* yang terkait dengan pengujian menggunakan *dataset*.

```
Hasil Dari Naive Bayes Score Data Testing : 0.7327188940092166
Laporan Klasifikasi pada Data Testing (Naive Bayes):
precision    recall   f1-score   support
```

```
negatif      0.78     0.70     0.74      71
netral       0.19     0.35     0.25      20
positif      0.87     0.81     0.84     126
accuracy        -       -       -       217
macro avg      0.62     0.62     0.61     217
weighted avg   0.78     0.73     0.75     217
```



**Gambar 7.** Akurasi Menggunakan Naïve Bayes dan Evaluasi Confusion Matrix

**Tabel 8.** Rata-rata Confusion Matrix Support Vector Machine

No	Label	Akurasi	Precision	Recall	F1 Score
1	Positif		88%	84%	86%
2	Netral	77%	29%	26%	27%
3	Negatif		79%	77%	78%

**Tabel 9.** Rata-rata Confusion Matrix Naive Bayes

No	Label	Akurasi	Precision	Recall	F1 Score
1	Positif		87%	82%	84%
2	Netral	73%	20%	35%	26%
3	Negatif		77%	70%	73%

### 3.2.7 Pembahasan

Penelitian ini menganalisis sentimen ulasan Whiz Prime Hotel Sudirman Pekanbaru menggunakan data sebanyak 4336 ulasan yang, setelah *preprocessing*, menjadi 2165 ulasan dengan distribusi sentimen positif (1260), netral (201), dan negatif (704). Penelitian dilakukan menggunakan *Google Collaboratory* dengan *Python* melalui tahapan *text preprocessing* (*cleaning*, *case folding*, *stopwords removal*, *tokenizing*, *stemming*), pelabelan data dengan bantuan ahli dan *TextBlob*, serta pemodelan klasifikasi menggunakan *Support Vector Machine* (SVM) dan *Naive Bayes*.

Evaluasi hasil klasifikasi menggunakan *confusion matrix* menunjukkan bahwa SVM unggul dengan akurasi 77% dibandingkan *Naive Bayes* dengan 73%. Visualisasi dalam bentuk *bar chart* dan *wordcloud* mengungkapkan kata-kata positif seperti "bersih," "ramah," dan "lokasi" yang mencerminkan keunggulan hotel, sedangkan kata negatif seperti "parkir" menunjukkan area perbaikan yang dapat meningkatkan kepuasan pelanggan.

## 4. KESIMPULAN

Penelitian ini menganalisis sentimen ulasan Whiz Prime Hotel Sudirman Pekanbaru menggunakan *algoritma Support Vector Machine* (SVM) dan *Naive Bayes*. Data diperoleh melalui teknik *scraping*, kemudian diproses dengan tahap *preprocessing*, pelabelan otomatis menggunakan *TextBlob*, dan diklasifikasikan dengan kedua algoritma tersebut. Hasil analisis menunjukkan bahwa 72,79% ulasan bersentimen positif, 15,38% negatif, dan 11,82% netral. Visualisasi data disajikan dalam bentuk bar chart dan *wordcloud*. Algoritma SVM menunjukkan akurasi terbaik sebesar 77%, dengan performa tinggi untuk sentimen positif, meskipun masih kurang optimal untuk sentimen netral. *Naive Bayes* memiliki akurasi tertinggi sebesar 73% dengan hasil yang serupa. Untuk penelitian selanjutnya, disarankan menggunakan *data-set* yang lebih besar serta metode pembobotan lanjutan seperti Word2Vec atau BERT, dan menerapkan teknik ensembel atau parameter tuning untuk meningkatkan akurasi, khususnya pada sentimen netral.

## Daftar Pustaka

- [1] L. Linda and Y. S. M. A. Dharasta, "Pengaruh Perilaku Konsumen Aplikasi Traveloka terhadap Kepuasan Konsumen," *Jurnal Manajemen, Bisnis dan Kewirausahaan*, vol. 2, no. 2, pp. 42–48, 2022.
- [2] W. Sitaresma and S. Ayuni, "Meningkatkan Loyalty Konsumen Traveloka Berbasis E-Service Quality Dan Satisfaction 'Studi Kasus Pada Aplikasi Traveloka,'" *Prosiding Konstelasi Ilmiah Mahasiswa Unissula (KIMU) Klaster Ekonomi*, vol. 1, no. 1, 2021.
- [3] T. Ernayanti, M. Mustafid, A. Rusgiyono, and A. R. Hakim, "Penggunaan Seleksi Fitur Chi-Square Dan Algoritma Multinomial Naïve Bayes Untuk Analisis Sentimen

- Pelanggan Tokopedia,” *Jurnal Gaussian*, vol. 11, no. 4, pp. 562–571, Feb. 2023, doi: 10.14710/j.gauss.11.4.562-571.
- [4] I. P. Rahayu, A. Fauzi, and J. Indra, “Analisis Sentimen Terhadap Program Kampus Merdeka Menggunakan Naive Bayes Dan Support Vector Machine,” *Jurnal Sistem Komputer dan Informatika (JSON)*, vol. 4, no. 2, p. 296, Dec. 2022, doi: 10.30865/json.v4i2.5381.
- [5] R. El Husna, R. Wasono, and M. Al Haris, “Analisis Klasifikasi Sentimen Pada Twitter Mengenai Netflix Yang Diblokir Oleh Telkom Menggunakan Naïve Bayes Classifier Dan Support Vector Machine Jurnal Ilmiah,” 2020. [Online]. Available: <http://repository.unimus.ac.id>
- [6] Z. Alhaq, A. Mustopa, S. Mulyatun, and J. D. Santoso, “Penerapan Metode Support Vector Machine Untuk Analisis Sentimen Pengguna Twitter,” *Journal of Information System Management (JOISM)*, vol. 3, no. 1, pp. 16–21, 2021.
- [7] Syahril Dwi Prasetyo, Shofa Shofiah Hilabi, and Fitri Nurapriani, “Analisis Sentimen Relokasi Ibukota Nusantara Menggunakan Algoritma Naïve Bayes dan KNN,” *Jurnal KomtekInfo*, pp. 1–7, Jan. 2023, doi: 10.35134/komtekinfo.v10i1.330.
- [8] E. Yuniar, D. S. Utsalinah, and D. Wahyuningsih, “Implementasi Scrapping Data Untuk Sentiment Analysis Pengguna Dompet Digital dengan Menggunakan Algoritma Machine Learning,” *Jurnal Janitra Informatika dan Sistem Informasi*, vol. 2, no. 1, pp. 35–42, Apr. 2022, doi: 10.25008/janitra.v2i1.145.
- [9] F. Novianti and K. R. N. Wardani, “Analisis Sentimen Masyarakat Terhadap Data Tweet Traveloka Selama Rapid Test Antigen Menggunakan Algoritma Naïve Bayes,” *JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)*, vol. 8, no. 3, pp. 922–933, 2023.
- [10] M. A. Saddam, E. K. Dewantara, and A. Solichin, “Sentiment Analysis of Flood Disaster Management in Jakarta on Twitter Using Support Vector Machines,” *Sinkron*, vol. 8, no. 1, pp. 470–479, Jan. 2023, doi: 10.33395/sinkron.v8i1.12063.
- [11] N. Herlinawati *et al.*, “Analisis Sentimen Zoom Cloud Meetings Di Play Store Menggunakan Naïve Bayes Dan Support Vector Machine,” 2020.
- [12] A. Triawan, “Penerapan Metode Naïve Bayes Untuk Rekomendasi Topik Tugas Akhir Berdasarkan Daftar Hasil Studi Mahasiswa di Perguruan Tinggi,” *Teknois: Jurnal Ilmiah Teknologi Informasi dan Sains*, vol. 10, no. 2, pp. 58–70, 2020.
- [13] R. Meifitrah, I. Darmawan, and O. N. Pratiwi, “Sentiment analysis of tokopedia application review to service product recommender system using neural collaborative filtering for marketplace in Indonesia,” in *IOP Conference Series: Materials Science and Engineering*, IOP Publishing, 2020, p. 012071.
- [14] A. F. Watratan and D. Moeis, “Implementasi Algoritma Naive Bayes Untuk Memprediksi Tingkat Penyebaran Covid-19 Di Indonesia,” *Journal of Applied Computer Science and Technology*, vol. 1, no. 1, pp. 7–14, 2020.
- [15] J. E. B. Sinulingga and H. C. K. Sitorus, “Analisis Sentimen Opini Masyarakat terhadap Film Horor Indonesia Menggunakan Metode SVM dan TF-IDF,” *Jurnal Manajemen Informatika (JAMIKA)*, vol. 14, no. 1, pp. 42–53, 2024.



ZONAsi: Jurnal Sistem Informasi

Is licensed under a [Creative Commons Attribution International \(CC BY-SA 4.0\)](#)