

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Landasan Teori**

##### **2.1.1 Analisis Sentimen**

Analisis sentimen digunakan untuk menentukan apakah suatu opini memiliki sentimen positif atau negatif terhadap suatu entitas atau objek. Analisis sentimen telah banyak digunakan untuk membantu menggali pendapat pelanggan tentang produk yang diminati (Fransiska, Rianto and Irham Gufroni, 2020). Tujuan dari analisis sentimen adalah untuk menemukan opini positif atau negatif seseorang, yang kemudian dapat digunakan untuk mengambil keputusan (Adhi Putra, 2021). Penelitian dengan topik analisis sentimen masih belum banyak dilakukan. Hal ini disebabkan karena tidak ada internet di zaman dahulu dan pengumpulan data hanya dapat dilakukan melalui pendapat teman dan keluarga. Setelah kemunculan internet, orang dapat dengan mudah memposting ulasan produk melalui situs *web* dan mengungkapkan pendapat dalam suatu forum internet, grup diskusi, dan postingan media sosial (Khatami, Irawan and Setianingsih, 2020).

Proses analisis sentimen diawali dengan pembuatan data latih, yang kemudian diolah dengan algoritma klasifikasi tertentu untuk menghasilkan model klasifikasi yang digunakan dalam proses klasifikasi. Proses pembuatan data latih dimulai dari data yang awalnya berisi beberapa komentar yang telah dilakukan *labeling* pada tahap *preprocessing*. Tahapan *preprocessing* diantaranya *case folding*, *tokenizing*, *filtering* dengan proses eliminasi

*stopword*, dan *lemmatization* untuk menghilangkan infleksi kata ke bentuk dasarnya. Tabel *confusion matrix* digunakan untuk menampilkan hasil klasifikasi. Informasi dalam *confusion matrix* diperlukan untuk menentukan kinerja model klasifikasi (Ilmawan and Mude, 2020).

### **2.1.2 Web Scraping**

*Web Scraping* adalah proses yang melibatkan ekstraksi informasi dari berbagai situs *web*. Secara umum, situs *web* berisi data tidak terstruktur. *Web Scraping* dapat digunakan untuk mengambil data, mengekstrak informasi, mengubahnya serta menyimpan menjadi format yang dapat dipahami seperti *Comma-Separated Values* (CSV). (Thota and Ramez, 2021).

### **2.1.3 Data Preprocessing**

*Data Preprocessing* adalah proses pembersihan dan penyiapan teks untuk proses klasifikasi (Pramono, Didi Rosiyadi and Windu Gata, 2019). *Data Preprocessing* meliputi beberapa proses seperti *Case Folding*, *Tokenizing*, *Filtering*, dan *Lemmatization*.

#### **a. Case Folding**

*Case Folding* merupakan proses perubahan teks yang tidak beraturan dalam penggunaan huruf pada suatu teks, mengakibatkan teks yang ditulis yang tidak konsisten. *Case Folding* berfungsi untuk mengubah huruf dalam teks dari huruf besar ke huruf kecil (Kurniawan et al., 2017).

#### **b. Tokenizing**

*Tokenizing* adalah proses pemisahan setiap kata dalam suatu dokumen. Secara umum, karakter spasi mengidentifikasi atau

memisahkan setiap kata dari kata lain, sehingga proses *Tokenizing* bergantung pada karakter spasi dalam dokumen untuk melakukan pemisahan kata (Fitriyah, Warsito and Maruddani, 2020).

c. *Filtering*

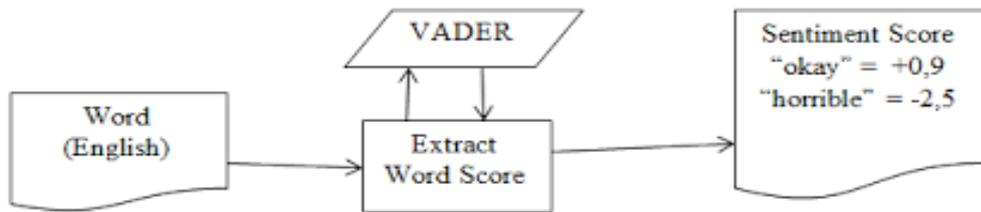
*Filtering* adalah proses menghilangkan kata-kata yang sering muncul tetapi dianggap tidak memiliki makna (*stopwords*) (Yulita et al., 2021). Proses *Filtering* sering disebut juga sebagai proses *removal stopwords*. *Stopwords* biasanya berisi kata-kata umum yang muncul dalam sejumlah besar kalimat dan konjungsi atau kata sambung (Hendra and Fitriyani, 2021).

d. *Lemmatization*

*Lemmatization* adalah proses menemukan bentuk dasar dari suatu kata (Syah and Witanti, 2022). Tujuan dari proses *Lemmatization* adalah mengubah kata-kata yang memiliki arti yang sama menjadi satu bentuk sehingga sistem dapat membacanya sebagai data yang berbeda dan mempersulit proses analisis (Lesmana and Andarsyah, 2022).

#### **2.1.4 Valence Aware Dictionary and sEntiment Reasoner (VADER)**

VADER biasanya digunakan sebagai model analisis sentimen untuk menentukan keragaman data berdasarkan kekuatan sentimen dalam kamus data leksikon (Pamungkas and Darmawan, 2022). Kalimat, sentimen, dan frasa dinilai menggunakan kamus leksikon untuk menghitung nilai polaritas dari suatu kata. Alur penentuan nilai polaritas suatu kalimat pada metode VADER dapat dilihat pada Gambar 2.1.



Gambar 2.1 Alur penentuan nilai polaritas

Sumber: (Abimanyu et al., 2022)

Berdasarkan Gambar 2.1 nilai polaritas yang didapatkan digunakan untuk menentukan skor negatif, positif, netral dan *compound score*. *Compound score* adalah metrik yang digunakan menghitung jumlah semua nilai kosakata antara -1 (sangat negatif) dan +1 (sangat positif) (Effendi and Noviana, 2021). Perhitungan nilai *compound score* dilakukan menggunakan rumus pada persamaan 2.1.

$$CS = \frac{x}{\sqrt{x^2 + \alpha}} \quad (2.1)$$

Keterangan:

CS : *Compound Score*

$x$  : jumlah nilai valensi kata penyusun

$\alpha$  : konstanta normalisasi (nilai *default* = 15)

Proses penentuan kelas sentimen bergantung pada hasil perhitungan *compound score* yang telah dilakukan. Ketentuan nilai *compound score* terhadap pelabelan sentimen dengan VADER dapat dilihat pada Tabel 2.1.

Tabel 2.1 Ketentuan pelabelan VADER

No	Sentimen	Compound Score (CS)	Keterangan
1	Positif	$CS \geq 0.05$	Bobot kata lebih banyak mengandung positif
2	Netral	$-0.05 < CS < 0.05$	Bobot kata lebih banyak mengandung netral atau tidak mengandung positif dan negatif
3	Negatif	$CS \leq -0.05$	Bobot kata lebih banyak mengandung negatif

Berdasarkan Tabel 2.1 penentuan kelas sentimen pada data ulasan menggunakan VADER didasarkan pada pembobotan *compound score*. Jika nilai *compound score* lebih besar sama dengan 0,05 maka ulasan tersebut masuk pada kelas sentimen positif. Jika nilai *compound score* berada pada rentang kurang dari 0,05 dan lebih besar dari -0,05 maka ulasan tersebut masuk pada kelas sentimen netral. Jika nilai *compound score* kurang dari -0,05 maka ulasan tersebut masuk pada kelas sentimen negatif.

### 2.1.5 Term Frequency – Inverse Document Frequency (TF-IDF)

TF-IDF adalah algoritma yang menunjukkan seberapa sering kata-kata muncul dalam dokumen. *Term Frequency* (TF) adalah seberapa sering kata-kata muncul dalam dokumen, sedangkan *Inverse Document Frequency* (IDF) adalah jumlah kemunculan suatu kata pada koleksi dokumen (Rizkia, Budi Setiawan and Puspendari, 2019). Tahapan pembobotan kata dengan TF-IDF adalah sebagai berikut.

- a. Menghitung nilai *term frequency* dengan persamaan 2.2.

$$TF(t_i, d) = \frac{\text{frekuensi } t \text{ pada } d}{\text{jumlah } t \text{ pada } d} \quad (2.2)$$

Keterangan:

$t_i$  : kata ke-  $i$   
 $d$  : Dokumen

- b. Menghitung nilai *document frequency* untuk mengetahui frekuensi kata pada seluruh dokumen.
- c. Menghitung nilai *inverse document frequency* dengan persamaan 2.3.

$$(IDF(t_i) = \ln \frac{|D|+1}{DF(t_i)+1} + 1 \quad (2.3)$$

Keterangan:

$t_i$  : kata ke-  $i$   
 $\ln$  : logaritma alami  
 $|D|$  : jumlah seluruh dokumen  
 $DF(t_i)$  : Jumlah dokumen yang mengandung  $t_i$

- d. Menghitung bobot akhir TF-IDF dengan persamaan 2.4.

$$TFIDF = TF(t_i, d) * IDF(t_i) \quad (2.4)$$

- e. Dalam *library scikit-learn*, hasil vektor TF-IDF kemudian dilakukan normalisasi menggunakan *Euclidean* yang dihitung dengan persamaan 2.5.

$$V_{norm} = \frac{v}{\sqrt{v_1^2 + v_2^2 + \dots + v_n^2}} \quad (2.5)$$

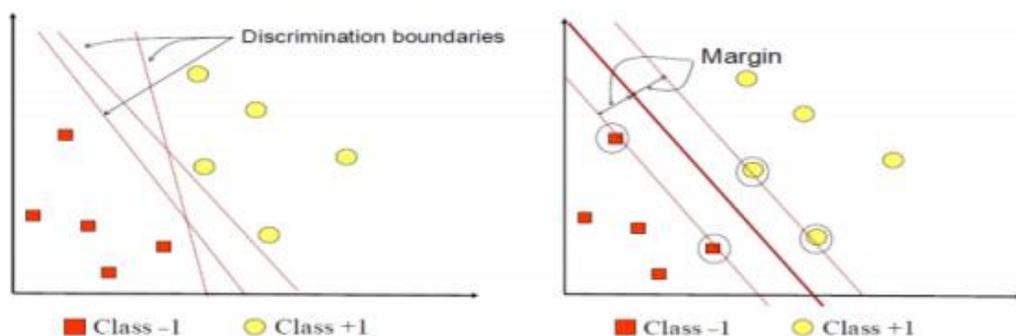
Keterangan:

$v$  : nilai vektor TF-IDF

### 2.1.6 Support Vector Machine (SVM)

SVM merupakan algoritma *supervised learning* dengan tingkat presisi/ketajaman dan kualitas yang tinggi, sehingga menjadikannya sebagai algoritma klasifikasi yang paling banyak digunakan dibandingkan dengan algoritma lainnya (Nurhafida and Sembiring, 2022). Berdasarkan prosesnya, kelas positif dan negatif dibagi menggunakan garis pemisah (*hyperplane*) dalam proses klasifikasi. Menurut aturannya, semakin lebar margin maka semakin kecil kesalahan generalisasi dari pemilah. Margin digunakan sebagai pemisahan antara kelas titik vektor dan *hyperplane*. Oleh karena itu, pembatas yang baik adalah yang terjauh dari titik data latih terdekat dari setiap kelas (Erfina et al., 2020). Inti dari algoritma ini adalah menemukan *hyperplane* terbaik dari setiap kemungkinan (Nasution and Hayaty, 2019).

Ilustrasi penemuan *hyperplane* terbaik dapat dilihat pada Gambar 2.2.



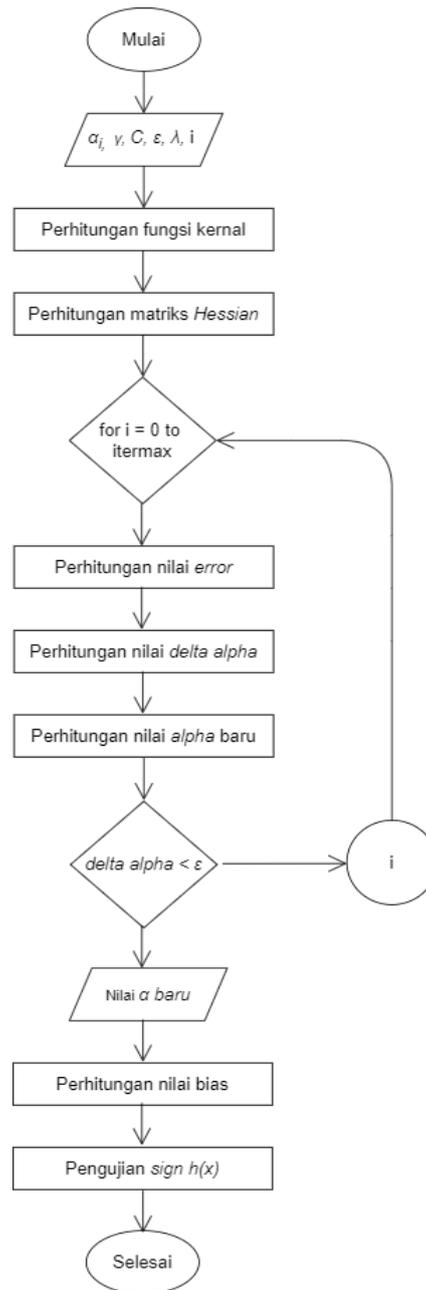
Gambar 2.2 Ilustrasi *hyperplane* SVM

Sumber: (Drajana, 2017)

Berdasarkan Gambar 2.2 menggambarkan antara dua kelas data yang terpisah. Garis merah tebal adalah *hyperplane* yang memisahkan dua titik,

dengan titik data di satu sisi diberi label kelas negatif yaitu -1, dan kelas positif diberi label +1. Berdasarkan garis merah kecil yang berpotongan dan memiliki pengaruh paling besar, titik data (vektor) yang paling dekat dengan *hyperplane* disebut sebagai *support vector*. Margin adalah jarak antara *hyperplane* dan *support vector*. Permasalahan *non-linear* pada klasifikasi SVM dapat diselesaikan dengan memodifikasi *trick kernel* ke dalam SVM yang bertindak sebagai pemisah kelas atau *hyperplane* dalam ruang vektor. Fungsi *kernel* yang digunakan dalam penelitian ini adalah *kernel linear*, karena *kernel* tersebut paling sederhana dan umumnya digunakan pada kasus klasifikasi teks (Rahman Isnain et al., 2021).

Penelitian ini menggunakan *Sequential Training* yang merupakan metode yang digunakan untuk memproses data latih dari algoritma SVM yang digunakan dengan waktu yang cukup cepat dibandingkan algoritma lain (Pravina, Cholissodin and Adikara, 2019). Alur dari proses *Sequential Training* dapat dilihat pada Gambar 2.3.



Gambar 2.3 Alur *Sequential Training*

Sumber: (Ramadan, Syauqy and Tibyani, 2019)

Berdasarkan Gambar 2.3 tahapan dari *Sequential Training* pada proses klasifikasi SVM dijelaskan sebagai berikut.

- a. Melakukan inisiasi terhadap parameter yang digunakan, yaitu:

$\alpha_i = \text{alpha}$ , digunakan untuk menentukan nilai *support vector*

$\gamma = \text{konstanta gamma}$ , digunakan untuk mengontrol kecepatan

$C = \text{variabel slack/Complexity}$

$\varepsilon = \text{epsilon}$ , digunakan untuk mencari nilai *error*

$\lambda = \text{turunan batas teoritis}$

$i = \text{iterasi maksimum}$

- b. Menghitung fungsi *kernel linear* dengan persamaan 2.6.

$$K(x_i, x_j) = (x_i * x_j) \quad (2.6)$$

- c. Menghitung matriks *Hessian* dengan persamaan 2.7.

$$D_{ij} = y_i y_j (K(x_i, x_j) + \lambda^2) \quad (2.7)$$

Keterangan:

$D_{ij}$  : elemen matriks ke – ij

$y_i$  : kelas atau label data ke – i

$y_j$  : kelas atau label data ke – j

$\lambda$  : turunan batas teoritis

- d. Menghitung nilai  $E_i$ ,  $\delta\alpha_i$ , dan  $\alpha_i$  sesuai iterasi maksimum yang ditentukan dengan persamaan 2.8, persamaan 2.9 dan persamaan 2.10.

$$E_i = \sum_{j=1}^i \alpha_i D_{ij} \quad (2.8)$$

$$\delta\alpha_i = \min\{\max[\gamma(1 - E_i), \alpha_i], C - \alpha_i\} \quad (2.9)$$

$$\alpha_i = \alpha_i + \delta\alpha_i \quad (2.10)$$

Keterangan:

- $E_i$  : nilai *error* data ke - i
- $\delta\alpha_i$  : parameter *delta alpha* untuk ke- i
- $\gamma$  : *gamma*
- $C$  : variabel *slack*
- $\alpha_i$  : parameter *alpha* untuk ke - i

e. Mencari nilai bias dengan persamaan 2.11.

$$b = -\frac{1}{2} * \left( \sum_{i=0}^N \alpha_i y_i (K(x_i, x^+)) + \sum_{i=0}^N \alpha_i y_i (K(x_i, x^-)) + \sum_{i=0}^N \alpha_i y_i (K(x_i, x^0)) \right) \quad (2.11)$$

Keterangan:

- $\alpha_i y_i (K(x_i, x^+))$  : Nilai *kernel* data x dengan data x kelas positif yang memiliki nilai  $\alpha_i$  tertinggi
- $\alpha_i y_i (K(x_i, x^-))$  : Nilai *kernel* data x dengan data x kelas negatif yang memiliki nilai  $\alpha_i$  tertinggi
- $\alpha_i y_i (K(x_i, x^0))$  : Nilai *kernel* data x dengan data x kelas netral yang memiliki nilai  $\alpha_i$  tertinggi

f. Setelah proses *Sequential Training* dilakukan selanjutnya pengujian pada dokumen data uji dengan menentukan nilai fungsi *kernel* data uji terhadap dokumen data latih terlebih dahulu. Nilai fungsi *kernel* data uji digunakan untuk perhitungan *sign h(x)* yang menentukan kelas sentimen data uji. Jika nilai hasil perhitungan keputusan  $> 0$  maka nilai *sign h(x)* adalah +1, maka termasuk pada kelas positif. Jika nilai hasil perhitungan keputusan sama dengan 0 maka nilai *sign h(x)* adalah 0, maka termasuk pada kelas netral. Jika nilai hasil perhitungan keputusan  $< 0$  maka nilai *sign h(x)* adalah -1, maka termasuk pada kelas negatif. Perhitungan *sign h(x)* ditentukan

dengan menggunakan persamaan 2.12 (Afrillia, Rosnita and Siska, 2022).

$$h(x) = \sum_{i=1}^m \alpha_i y_i K(x, x_i) + b \quad (2.12)$$

### 2.1.7 Confusion Matrix

*Confusion matrix* merupakan matriks yang berisi informasi tentang hasil yang diprediksi dan aktual (Afdhal et al., 2022). Tabel *confusion matrix* dapat dilihat pada Tabel 2.2.

Tabel 2.2 *Confusion matrix*

<i>Confusion Matrix</i>		Kelas Aktual		
		Positif	Netral	Negatif
Kelas Prediksi	Positif	TF	FP	FP
	Netral	FNeu	TNeu	FNeu
	Negatif	FN	FN	TN

Berdasarkan Tabel 2.2 *confusion matrix* memiliki 4 nilai keluaran yang dapat dihitung untuk mengukur performa algoritma klasifikasi, yaitu:

a. Akurasi

Akurasi merupakan nilai yang dihasilkan dari perbandingan semua hasil prediksi dengan seluruh data. Rumus menghitung nilai akurasi dapat dilihat pada persamaan (2.13).

$$Akurasi = \frac{TP + TNeu + TN}{TP + TNeu + TN + FP + FNeu + FN} \quad (2.13)$$

Keterangan:

- True Positive (TP)* : Banyaknya nilai data positif yang tergolong positif
- False Positive (FP)* : Banyaknya nilai data positif yang tergolong negatif atau netral
- False Negative (FN)* : Banyaknya nilai data negatif yang tergolong positif atau netral
- True Negative (TN)* : Banyaknya nilai data negatif yang tergolong negatif
- True Neutral (TNeu)* : Banyaknya nilai data netral yang tergolong netral
- False Neutral (FNeu)* : Banyaknya nilai data netral yang tergolong negatif atau positif

b. Presisi

Presisi merupakan nilai yang dihasilkan dari perbandingan nilai prediksi benar dan hasil prediksi keseluruhan. Rumus menghitung nilai presisi dapat dilihat pada persamaan (2.14).

$$Presisi = \frac{TP}{TP + FP} \quad (2.14)$$

c. Recall

Recall merupakan nilai yang dihasilkan dari perbandingan nilai prediksi benar dan hasil keseluruhan hasil yang benar. Rumus menghitung nilai *recall* dapat dilihat pada persamaan (2.15)

$$Recall = \frac{TP}{TP + FN + FNeu} \quad (2.15)$$

## 2.2 Penelitian Terkait dan Kebaruan Penelitian

### 2.2.1 *State of The Art* Bidang Analisis Sentimen

Perbandingan penelitian yang berhubungan dengan fokus permasalahan penelitian dan hasil penelitian yang dihasilkan dalam proses analisis sentimen dapat dilihat pada Tabel 2.3.

Tabel 2.3 *State of The Art*

No	Nama Peneliti	Tahun	Judul	Hasil Penelitian
1	Sigit Pamungkas, J.B. Budi Darmawan	2022	Klasifikasi Sentiment Tweet Pelanggan IndiHome Selama Pandemi Covid-19 Menggunakan Algoritma Multinomial Naive Bayes	Berdasarkan hasil pengujian, model dengan akurasi tertinggi sebesar 76.06% dari 11 iterasi pengujian.
2	Ibnu Afdhal, Rahmad Kurniawan, Iwan Iskandar, Roni Salambue, Elvia Budianita, Fadhilah Syafria	2022	Penerapan Algoritma Random Forest Untuk Analisis Sentimen Komentar Di YouTube Tentang Islamofobia	Akurasi yang dihasilkan mencapai 79% dengan F1-Score sebesar 86,26% yang menandakan bahwa algoritma random forest cukup baik dalam melakukan klasifikasi sentimen pada komentar youtube terhadap topik islamofobia tersebut.

Tabel 2.3 *State of The Art* (Lanjutan 1)

No	Nama Peneliti	Tahun	Judul	Hasil Penelitian
3	Asep Hendra, Fitriyani	2021	Analisis Sentimen Review Halodoc Menggunakan Naïve Bayes Classifier	Algoritma yang digunakan adalah Naïve Bayes Classifier dengan hasil akurasi 81.68% dan nilai AUC sebesar 0.756, sehingga termasuk ke dalam <i>fair classification</i> .
4	Aluisius Dwiki Adhi Putra, Safitri Juanita	2021	Analisis Sentimen Pada Ulasan Pengguna Aplikasi Bibit dan Bareksa Dengan Algoritma KNN	Berdasarkan hasil yang diperoleh dari tahapan modelling dengan menggunakan algoritma k-nearest neighbors dan dengan pemisahan 60:40 antara data pelatihan dan pengujian, nilai presisi dan akurasi recall yang dihasilkan oleh setiap aplikasi masing-masing adalah 85,14%, 91,91%, dan 76,44% untuk bibit sedangkan untuk bareksa yaitu 81,70%, 87,15%, 75,73%

Tabel 2.3 *State of The Art* (Lanjutan 2)

No	Nama Peneliti	Tahun	Judul	Hasil Penelitian
5	Christian Cahyaningtyas, Yessica Nataliani, Indrastanti Ratna Widiarsari	2021	Analisis Sentimen Pada Rating Aplikasi Shopee Menggunakan Metode Decision Tree Berbasis SMOTE	Hasil penelitian dengan menggunakan algoritma Decision Tree dengan SMOTE ( <i>Synthetic Minority Oversampling Technique</i> ) nilai akurasinya menghasilkan 99,91%, AUC (Area Under Curve) 0,999 recall 99,88% dan nilai precision 99,98%. Hasil menggunakan algoritma Decision Tree tanpa SMOTE nilai accuracynya menghasilkan 99,89%, AUC (Area Under Curve) 0,950, recall 99,88% dan nilai precision 99,98%.
6	Siti Maryam Gaib, Muis Nanja, Hastuti Dalai	2021	Analisis Efektivitas Pembelajaran Daring Google Classroom Menggunakan Metode Naïve Bayes	Nilai akurasi yang diperoleh antara lain kedua data responden yaitu data guru dan data siswa masing masing memiliki tingkat akurasi yang sangat baik yaitu antara lain 95% untuk data guru dan 96,97% untuk data siswa.

Tabel 2.3 *State of The Art* (Lanjutan 3)

No	Nama Peneliti	Tahun	Judul	Hasil Penelitian
7	Eendi dan Rina Noviana	2021	Perancangan Web Aplikasi Analisis Sentimen Media Sosial Twitter Dengan Metode Valence Aware Dictionary and Sentiment Reasoner (VADER) Menggunakan PHP dan MySQL pada Pemerintah Kota Bekasi	Dengan banyaknya jumlah pengguna twitter di Indonesia khususnya Kota Bekasi maka mempermudah Pemerintah Kota Bekasi dalam merespon keluhan dan masukan dari masyarakat secara cepat.
8	Muhammad Ghudafa Taufik Akbar, Dwi Budi Srisulistiowati	2021	Analisa Sentimen Efektifitas Vaksin terhadap Varian COVID 19 Omicron Berbasis Leksikon	Algoritma SVM mengungguli pelabelan komentar dengan metode TextBlob dalam hal akurasi ketika diterapkan pada kumpulan data yang diberi label dengan metode VADER.

Tabel 2.3 *State of The Art* (Lanjutan 4)

No	Nama Peneliti	Tahun	Judul	Hasil Penelitian
9	Miranda Sari, Ibnu Rasyid Munthe, Irmayani	2021	Metode Clustering K-Medoids untuk Aplikasi Pembelajaran Masa Pandemi COVID-19	Pada penelitian ini ditemukan bahwa clustering rapid miner menghasilkan pembelajaran yang paling disukai selama pandemi COVID 19 saat ini menggunakan Moodle dan WhatsApp, sedangkan penggunaan zoom tidak disukai dalam proses pembelajaran.
10	Fajar Pramono, Didi Rosiyadi, Windu Gata	2021	Integrasi N-gram, Information Gain, Particle Swarm Optimization di Naïve Bayes untuk Optimasi Sentimen Google Classroom	Penggunaan N-gram dan PSO sangat membantu dalam mengoptimalkan hasil Akurasi dan AUC, namun penambahan Information Gain dalam penelitian di sini tidak memberikan peningkatan yang berbeda.

Tabel 2.3 *State of The Art* (Lanjutan 5)

No	Nama Peneliti	Tahun	Judul	Hasil Penelitian
11	Lutfi Budi Ilmawana, Muhammad Aliyazid Mudeb	2020	Perbandingan Metode Klasifikasi Support Vector Machine dan Naïve Bayes untuk Analisis Sentimen pada Ulasan Tekstual di Google Play Store	Klasifikasi dengan SVM memiliki nilai yang lebih tinggi dibandingkan dengan klasifikasi dengan Nave Bayes Classifier, dengan classifier SVM memiliki akurasi 81,46% dan classifier Nave Bayes memiliki akurasi 75,41%.
12	Nuraeni Herlinawati, Yuri Yuliani, Siti Faizah, Windu Gata, Samudi.	2020	Analisis Sentimen Zoom Cloud Meetings di Play Store Menggunakan Naïve Bayes dan Support Vector Machine	Evaluasi model menggunakan nilai 10 k-fold cross-validation diperoleh nilai akurasi dan nilai AUC dari masing- masing algoritma yaitu untuk NB nilai akurasi = 74,37% dan nilai AUC = 0,659. Algoritma SVM memiliki nilai akurasi sebesar 81,22% dan nilai AUC sebesar 0,886.

Tabel 2.3 *State of The Art* (Lanjutan 6)

No	Nama Peneliti	Tahun	Judul	Hasil Penelitian
13	Susanti Fransiska, Rianto, Acep Irham Gufroni	2020	Sentiment Analysis Provider by.U on Google Play Store Reviews with TF-IDF and Support Vector Machine (SVM) Method	Akurasi rata-rata yang dihasilkan 84,7%, presisi 84,9%, recall 84,7%, dan f-ukuran 84,8%. Hasil akurasi tertinggi pada fold 2, 86,1%.
14	Faiz Adil Khatami, Budhi Irawan, S.Si., M.T., Casi Setianingsih, S.T., M.T.	2020	Analisis Sentimen Terhadap Review Aplikasi Layanan E-Commerce Menggunakan Metode Convolutional Neural Network	Sistem dapat menganalisis sentimen menjadi 3 kategori yaitu positif, negatif, dan netral dengan tingkat akurasi paling tinggi sebesar 86,6%.
15	Arini, Luh Kesuma Wardhani, Dimas Octaviano	2020	Perbandingan Seleksi Fitur Term Frequency & Tri-Gram Character Menggunakan Algoritma Naïve Bayes Classifier (Nbc) Pada Tweet Hashtag #2019gantipresiden	Hasil akurasi Algoritma Naïve Bayes dengan seleksi fitur Tri-Gram Character yang lebih tinggi dibanding Term-Frequency diperkuat dalam suatu penelitian, nilai akurasi Tri-Gram sebesar 86% dan Term-Frequency sebesar 68,5%.

Tabel 2.3 *State of The Art* (Lanjutan 7)

No	Nama Peneliti	Tahun	Judul	Hasil Penelitian
16	Venkateswarlu Bonta, Nandhini Kumaresh, N. Janardhan	2019	A Comprehensive Study on Lexicon Based Approaches for Sentiment Analysis	Menghasilkan performa klasifikasi metode VADER yang memiliki akurasi sebesar 77%, sedangkan <i>TextBlob</i> memiliki akurasi 74% dan <i>SentiWordNet</i> memiliki akurasi 62%.
17	Shakina Rizkia, Erwin Budi Setiawan S.Si., M.T, Diyas Puspanhari S.S., M.Pd.	2019	Analisis Sentimen Kepuasan Pelanggan Terhadap Internet Provider Indihome di Twitter Menggunakan Metode Decision Tree dan Pembobotan TF-IDF	Dengan akurasi 80,1%, hasil tersebut tidak memuaskan karena kurangnya data dan beberapa fitur yang tidak cukup untuk mengklasifikasikan sentimen.
18	Ananda Fitri Niasita, Putra Pandu Adikara, Sigit Adinugroho	2019	Analisis Sentimen Pembangunan Infrastruktur di Indonesia dengan Automated Lexicon Word2Vec dan Naive-Bayes	Metode Naive-Bayes memiliki presisi 0,36, recall 0,818, f-measure 0,5, dan akurasi 64% saat digunakan untuk mengklasifikasikan kelas sentimen.

Tabel 2.3 *State of The Art* (Lanjutan 8)

No	Nama Peneliti	Tahun	Judul	Hasil Penelitian
19	James Effendi, M Jorgi Ramadhan	2018	Analisis Cluster Aplikasi pada Google play Store dengan Menggunakan Metode K-Mean	Nilai rating, jumlah review, dan jumlah download yang tinggi merupakan karakteristik aplikasi yang cukup baik di cluster ketiga. tapi karena jarak harga dan ukuran pada klaster ketiga jauh lebih besar dibandingkan dengan klaster lainnya.
20	Ferly Gunawan, Mochammad Ali Fauzi, Putra Pandu Adikara	2017	Analisis Sentimen Pada Ulasan Aplikasi Mobile Menggunakan Naive Bayes dan Normalisasi Kata Berbasis Levenshtein Distance (Studi Kasus Aplikasi BCA Mobile)	Penelitian menghasilkan bahwa metode Naive Bayes Classifier dengan Levenshtein Distance memiliki tingkat akurasi 96,9% dibandingkan dengan metode Naive Bayes Classifier tanpa menerapkan Levenshtein Distance dengan tingkat akurasi sebesar 94,4%.

Berdasarkan Tabel 2.3 *State of The Art* (SOTA) penelitian terkait, terdapat persamaan dan perbedaan latar belakang permasalahan yang digunakan dan hasil penelitian yang telah dilakukan. Perbedaan pada latar belakang permasalahan, algoritma atau metode yang digunakan, objek penelitian, ruang lingkup, serta faktor-faktor lain dapat mempengaruhi hasil penelitian.

Berdasarkan hal tersebut, pada penelitian yang dilakukan juga memiliki faktor-faktor pembeda yang dapat mempengaruhi hasil penelitian, seperti perbedaan dalam ruang lingkup, jumlah data, objek penelitian hingga beberapa indikator dari variabel yang dipakai dalam penelitian.

### **2.2.2 Matriks Penelitian**

Matriks penelitian digunakan untuk memetakan keseluruhan isi penelitian terkait serta penelitian yang dibuat. Matriks penelitian mencakup informasi seputar penulis, judul, dan ruang lingkup yang berisi metode atau algoritma yang digunakan serta tujuan yang berisi keluaran dari penelitian yang dilakukan. Matriks penelitian dapat dilihat pada Tabel 2.4.















Tabel 2.4 Matriks Penelitian (Lanjutan 7)

No	Penulis	Judul	Ruang Lingkup												
			Metode									Tujuan			
			SVM	NBC	KNN	DT	CNN	RF	MNB	KMEANS	KMEDOIDS	VADER	Aplikasi	Klasifikasi	Klasterisasi
21	Yuda Maulana (2022)	Optimalisasi <i>Support Vector Machine</i> menggunakan Pelabelan VADER pada Analisis Sentimen Ulasan Google Classroom	√	-	-	-	-	-	-	-	-	√	-	√	-

Berdasarkan Tabel 2.4 Matriks Penelitian, terdapat beberapa penelitian sebelumnya yang menggunakan algoritma yang sama dengan penelitian yang dilakukan yaitu algoritma *Support Vector Machine*. Algoritma klasifikasi *Support Vector Machine* dipilih karena akurat dan bekerja dengan baik di ruang berdimensi tinggi. SVM juga memiliki kelebihan seperti penentuan jarak menggunakan *support vector* sehingga mempercepat proses komputasi (Octaviani, Yuciana Wilandari and Ispriyanti, 2014). Selain itu, berdasarkan penelitian sebelumnya, algoritma SVM memperoleh tingkat akurasi cukup baik yaitu 81,22% dibanding algoritma *Naïve Bayes* yang hanya memperoleh tingkat akurasi 74,37% (Herlinawati et al., 2020).

### 2.2.3 Relevansi Penelitian

Penelitian ini memiliki relevansi atau keterkaitan dengan penelitian terkait sebelumnya (Fransiska, Rianto and Irham Gufroni, 2020). Perbandingan terkait relevansi penelitian dapat dilihat pada Tabel 2.5.

Tabel 2.5 Relevansi Penelitian

Peneliti	(Fransiska, Rianto and Irham Gufroni, 2020)	(Yuda Maulana, 2022)
Judul	Sentiment Analysis Provider by.U on Google Play Store Reviews with TF-IDF and Support Vector Machine (SVM) Method	Optimalisasi <i>Support Vector Machine</i> menggunakan Pelabelan VADER pada Analisis Sentimen Ulasan Google Classroom
Masalah Penelitian	Aplikasi by.U memiliki review yang cukup banyak dan tidak terstruktur sehingga diperlukan suatu teknik untuk menyajikan review tersebut agar lebih informatif bagi pengguna maupun pengembang aplikasi atau produk.	Banyak sekali aplikasi pembelajaran <i>online</i> lain yang memiliki <i>rating</i> hampir sama dengan Google Classroom, sehingga predikat terbaik bagi aplikasi tersebut menjadi tidak relevan. Selain itu, banyak ulasan yang diberikan tidak sesuai dengan <i>rating</i> yang diberikan sehingga tidak dapat dijadikan acuan.
Objek Penelitian	Ulasan by.U pada Google Play Store	Ulasan Google Classroom pada Google Play Store
Algoritma/Metode	<i>Support Vector Machine</i> , TF-IDF	<i>Support Vector Machine</i> , VADER, TF-IDF

Tabel 2.5 Relevansi Penelitian (Lanjutan 1)

Peneliti	<b>(Fransiska, Rianto and Irham Gufroni, 2020)</b>	<b>(Yuda Maulana, 2022)</b>
Dataset	Data ulasan by.U berasal dari situs Google Play Store dengan jumlah 10.000 data berbahasa Indonesia. Data yang digunakan dalam penelitian ini adalah data terbaru dari tanggal 17 Maret 2020.	Data ulasan dengan jumlah 10.000 data berbahasa inggris. Data yang digunakan dalam penelitian ini adalah data ulasan pada kurun waktu 1 Februari 2022 hingga 6 September 2022.

Berdasarkan Tabel 2.5 Relevansi Penelitian, terdapat satu penelitian yang dapat dijadikan tolak ukur dalam melakukan penelitian ini. Penelitian yang dilakukan memiliki latar belakang permasalahan, *dataset* serta algoritma dan metode yang digunakan cukup relevan, namun dari penelitian sebelumnya yang dilakukan perlu adanya optimalisasi menggunakan metode pelabelan VADER sehingga meningkatkan performa atau tingkat akurasi dari algoritma *Support Vector Machine*.