The 6th Conference on Innovation and Application of Science and Technology (CIASTECH)

Website Ciastech 2023 : https://ciastech.net
Open Confrence Systems : https://ocs.ciastech.net
Proceeding homepage : https://publishing-widyagama.ac.id/ejournal-v2/index.php/ciastech/issue/view/236

P-ISSN : 2622-1276
E-ISSN: 2622-1284

KINERJA *AUTO LABELLING* PADA ANALISIS SENTIMEN TERHADAP PASANGAN CALON PRESIDEN 2024 DI MEDIA SOSIAL X

Syahroni Wahyu Iriananda[1*)], Rangga Pahlevi Putra[2)], Ahmad Farhan[3)]

[1,2,3)] Program Studi S1 Teknik Informatika, Fakultas Teknik, Universitas Widyagama Malang

INFORMASI ARTIKEL

Data Artikel :

Naskah masuk, 30 November 2023
Direvisi, 4 Desember 2023
Diterima, 12 Desember 2023

Email Korespondensi :
syahroni@widyagama.ac.id

ABSTRAK

Di era pasca Covid-19 pemanfaatan *platform* X telah meningkat pesat dan memainkan peran penting sebagai sumber opini publik. Media sosial memungkinkan individu untuk menyatakan pendapat mereka tentang peristiwa terkini. *Respons* positif terhadap demokrasi dan kebebasan berekspresi telah terlihat dalam penggunaan *platform* ini. Banyak individu menggunakan *Twitter* untuk menyampaikan sudut pandang dan komentar, *Term*asuk dalam konteks politik khususnya pada masa menjelang Pemilihan Presiden dan Wakil Presiden Indonesia 2024. Penelitian ini menekankan tantangan dalam menganalisis sentimen dalam *tweet* berbahasa Indonesia dan menyoroti penggunaan algoritma pembelajaran mesin untuk analisis sentimen. Penelitian ini menggunakan teknik *web scraping* untuk mengumpulkan data dari *platform* media sosial dan menerapkan metode *preprocessing* untuk membersihkan dan mengatur data. Ekstraksi fitur dilakukan menggunakan metode TF-IDF dan kombinasi *Unigram*, *Bigram*, dan *Trigram* (*UniBiTri*). Pendekatan *Auto Labelling* digunakan dengan algoritma VADER dan *TextBlob*, diikuti oleh klasifikasi SVM dengan mempertimbangkan rasio data latih dan data uji serta pengujian dengan berbagai *kernel*. Evaluasi dilakukan melalui *confusion matrix* dan laporan klasifikasi untuk menilai kinerja model. Temuan menunjukkan bahwa kinerja *Auto Labelling TextBlob* melampaui VADER, mencapai akurasi maksimum 99,33%, *Precision* 99,35%, dan *recall* 99,33%.

Kata Kunci : *Vader, TextBlob, Auto Labelling, Twitter, Pilpres 2024*

1. PENDAHULUAN

Pemanfaatan internet *pasca* Covid-19 khususnya melalui *platform* media sosial seperti X (*Twitter*), telah meningkat pesat dan memainkan peran kunci sebagai sumber opini publik. Media sosial memungkinkan individu terhubung dengan individu lain dan dapat mengekspresikan pendapat

mereka tentang peristiwa terkini [1]. Studi menunjukkan *respons* positif terhadap demokrasi dan kebebasan berpendapat di *platform* ini [2]. Individu banyak menggunakan X untuk menyampaikan sudut pandang dan komentar, *Term*asuk masalah politik [3]. Platfotm X berfungsi sebagai media sosial dengan basis pengguna yang besar [4], menjadikannya sumber informasi yang berharga untuk menganalisis opini publik selama periode pemilihan [5][6].

Analisis jejaring sosial menyoroti peran media sosial dalam menyebarkan sudut pandang dan memfasilitasi interaksi[7]. Kerangka kerja penambangan opini telah dirancang untuk menilai opini publik di *Twitter*, mencakup evaluasi sentimen dan pemanfaatan kata kunci dalam *tweet* [8].

Penggunaan kerangka kerja penambangan opini di X juga mencerminkan dampak besar internet dan media sosial terhadap pembentukan opini publik. Penambangan teks opini (*Opinion Mining*) yang kemudian lebih umum disebut sebagai Analisis Sentimen, khususnya klasifikasi teks, dapat digunakan untuk mengembangkan mesin yang dapat menganalisis dan mengklasifikasikan sejumlah besar *tweet*, memberikan wawasan tentang sentimen publik terhadap kandidat presiden. Para peneliti menggunakan teknik analisis sentimen untuk menganalisis *tweet* terkait pemilihan dan memahami perilaku penerbitan pengguna, sentimen, keselarasan politik, dan topik diskusi [9].

Opini Masyarakat dalam Bahasa Indonesia yang bersumber data dari media sosial, seperti *platform* X sangat bervariasi dan memiliki tantangan yang besar dalam menanganinya, menimbulkan tantangan ekstra karena teksnya cenderung singkat dan gaya bahasa yang bermacam-macam, masih terdapat bahasa *slang* atau bahasa gaul, campuran dengan Bahasa daerah, dan belum dinormalisasi. Dalam menghadapi kendala-kendala tersebut, pelabelan otomatis dapat mengandalkan analisis polaritas sentimen, ekspresi emosi, dan tingkat toksisitas [10]. Pelabelan otomatis data *Twitter* dapat diterapkan melalui berbagai metode, seperti *TextBlob*, Vader, dan Afinn [11].

Analisis sentimen pada data *platform* X mengacu pada proses menganalisis sentimen atau pendapat yang diungkapkan dalam *tweet*. Ini melibatkan mengklasifikasikan *tweet* sebagai positif, negatif, atau netral [4][9] berdasarkan sentimen yang mereka sampaikan.

Teknik pra-pemrosesan *(preprocessing)* seperti tokenisasi, *case folding*, *data cleansing*, *stemming*, *Filtering* diterapkan pada teks sebelum analisis dilakukan [6], dilanjutkan dengan menghapus *stopwords*. Setelah proses *preprocessing* telah dilakukan, berikutnya adalah proses memberikan label pada setiap *tweet*. Pada bagian ini penerapan label awal pada data media sosial merupakan tantangan terbuka bagi para peneliti ketika melakukan analisis sentimen [11].

Memberi label pada korpus dalam skala besar menjadi suatu tugas yang rumit dan kurang efisien jika dilakukan secara manual. Keterbatasan pelabel Bahasa Indonesia yang memenuhi syarat menjadi hambatan utama.

Penerapan pendekatan pelabelan otomatis ini dapat menjadi alternatif efisien terhadap pelabelan oleh manusia secara manual [10]. Metode-metode ini memanfaatkan analisis sentimen berbasis leksikon untuk mengklasifikasikan sentimen dalam teks. Penelitian sebelumnya mengusulkan penerapan metode pelabelan Rata-rata dan Biner untuk pelabelan otomatis [13]. Metode-metode ini melibatkan pelabelan manual teks berdasarkan acuan tertentu. Penelitian berikutnya membahas kendala pelabelan otomatis dalam bahasa Bahasa Melayu, dan mengusulkan metode yang menggabungkan polaritas sentimen, emosi, dan toksisitas untuk pelabelan otomatis *tweet cyberbully* [10].

Akurasi dari metode-metode pelabelan otomatis ini diukur dengan membandingkan label yang dihasilkan dengan data yang telah dilabeli manual. Teknik *Auto Labelling* ini dapat menjadi sangat berguna dalam mengorganisir dan mengkategorikan sejumlah besar data, seperti posting media

sosial atau ulasan pelanggan. Dengan memanfaatkan model analisis sentimen seperti Vader dan *TextBlob* bersamaan dengan klasifikasi SVM, para peneliti bertujuan untuk menentukan efisiensi pelabelan otomatis dalam mengorganisir dan menganalisis data teks [14]. VADER telah terbukti mengungguli *TextBlob* dalam hal akurasi skor sentimen saat menganalisis teks media sosial, seperti yang ditunjukkan oleh studi seperti yang dilakukan oleh S. Kiritchenko dalam (Mehta & Verma, 2021) [14]. Dalam konteks analisis sentimen pada teks media sosial, VADER cenderung memiliki kinerja lebih baik dibandingkan dengan *TextBlob* dalam hal akurasi dan presisi.

Tugas klasifikasi pada analisis sentimen dilakukan menggunakan algoritma pembelajaran mesin tradisional seperti *Support Vector Machine* (SVM), *Naive Bayes Classifier* (NBC), serta algoritma jaringan saraf dalam seperti *Convolutional Neural Networks* (CNNs), *Recurrent Neural Networks* (RNN), dan *glove-DCNN* [6]. Tujuannya adalah untuk mengklasifikasikan sentimen *tweet* ke dalam kategori positif, negatif dan netral. Analisis tersebut memberikan wawasan tentang perilaku pemilih di media sosial selama kampanye pemilu.

Dalam penelitian terdahulu, beberapa metode telah diterapkan untuk menganalisis polaritas sentimen dan mengklasifikasikan *tweet* menjadi sentimen positif dan negatif, khususnya dalam konteks Pemilu Presiden RI 2019[9]. Namun, penting untuk dicatat bahwa detail spesifik tentang *Twitter*, pemilihan, dan presiden dapat bervariasi tergantung pada konteks masing-masing makalah. Berbagai penelitian telah menggunakan analisis sentimen pada *platform* media sosial *Twitter* untuk mengevaluasi opini publik yang berlaku mengenai para pesaing untuk presiden Indonesia. Lukmana (2019) mencapai akurasi 86% yang mengesankan dalam membedakan *tweet* sebagai positif atau negatif melalui pemanfaatan *Support Vector Machine* [15]. Silitonga (2019) dan Fitriyyah (2019) keduanya menggunakan *Klasifikasi Bayesian Naïve* [5], [16], dengan Silitonga (2019) mencapai tingkat akurasi 77,62% dan Fitriyyah (2019) mencatat tingkat akurasi 88% dalam mengategorikan *tweet* sebagai positif, negatif, atau netral. Secara kolektif, studi-studi ini berfungsi sebagai indikasi potensi analisis sentimen di *Twitter* sebagai sarana untuk memahami sentimen publik terhadap calon presiden.

Metode atau algoritma yang digunakan untuk analisis sentimen dalam makalah ini terutama difokuskan pada pendekatan pembelajaran mesin. Beberapa makalah menyebutkan penggunaan algoritma pembelajaran mesin tradisional seperti *Support Vector Machine* (SVM), *Naive Bayes Classifier* (NBC) [5], [9], [16]–[19]dan Entropi Maksimum [6]. Pada penelitian sebelumnya tingkat akurasi Algoritma *Naïve Bayes* paling maksimum adalah 86,4% [20]. Metode NBC ini bertujuan untuk secara akurat memprediksi sentimen *tweet* yang terkait dengan berbagai topik, *Term*asuk pemilihan presiden di Indonesia pada tahun 2019 [19], [5], [16], [6], [17], [20]. Algoritma pembelajaran mendalam (*Deep Learning*), *Term*asuk *Convolutional Neural Networks* (CNN), *Recurrent Neural Networks* (RNN), Memori jangka pendek panjang (LSTM) [21], CNN-LSTM, *Gated Recurrent Unit* (GRU) -LSTM, dan LSTM Bidirectional, juga telah digunakan untuk analisis *sentiment* [14] [15]. Pembelajaran transfer, yang melibatkan pelatihan model pada *dataset* yang berbeda dan menggunakannya untuk prediksi pada *dataset* baru, telah digunakan untuk mengatasi kurangnya data berlabel [24]. Selain itu, metode berbasis leksikon, seperti *SentiWordNet* dan *WordNet*, telah digunakan untuk analisis sentimen. Secara keseluruhan, makalah ini menyoroti penggunaan pembelajaran mesin tradisional dan algoritma pembelajaran mendalam untuk analisis sentimen dalam berbagai konteks.

Menggunakan Vader dan *TextBlob* untuk analisis sentimen dan klasifikasi SVM untuk pelabelan otomatis, mereka dapat secara akurat menetapkan label atau kategori pada data teks berdasarkan

sentimen, sehingga dapat mengorganisir dan menganalisis data dengan efektif. Penelitian ini mengimplementasikan langkah-langkah berikut: akuisisi data, pra-pemrosesan, pelabelan otomatis (*Auto Labelling*) menggunakan Vader dan *TextBlob*, dan klasifikasi menggunakan SVM [14].
Penelitian ini bertujuan untuk menguji efektivitas penggunaan model seperti Vader dan *TextBlob* sebagai pelabelan otomatis *(Auto Labelling)* dengan klasifikasi SVM. Penelitian ini dilakukan untuk menginvestigasi bagaimana kinerja dan penggunaan pelabelan otomatis (*Auto Labelling*) data teks berdasarkan *sentiment* untuk *dataset* Pasangan Calon Presiden dan Calon Wakil Presiden Indonesia 2024 menggunakan Algoritma *Support Vector Machine* (SVM). Penelitian ini menggunakan berbagai teknik seperti pemrograman *Python*, API *Twitter*, APIFY *Web Scraper*, *Natural Language Tool Kit* (NLTK) *library*, Sastrawi *Stemmer* dan visualisasi untuk menganalisis sentimen dalam data teks [14]

2. METODE PENELITIAN

2.1 Dataset

Langkah pertama dalam analisis sentimen *Twitter* terkait pasangan calon presiden 2024 adalah melakukan pengumpulan data melalui teknik *web scraping*. Dalam konteks ini, *web scraping* dapat dilakukan dengan mengekstrak *tweet* yang mencakup nama atau referensi terhadap kedua calon dari situs web *Twitter*. Proses *web scraping* ini melibatkan penggunaan perangkat lunak atau *script* khusus yang dapat menelusuri halaman *Twitter*, mengidentifikasi *tweet* yang relevan, dan mengekstrak informasi yang dibutuhkan. Penggunaan antarmuka pemrograman aplikasi (API) *Twitter* juga bisa menjadi pilihan, tergantung pada ketersediaan akses dan kebutuhan spesifik analisis. Data *Twitter* dari *platform* X diambil dengan kriteria tanggal data adalah 13 – 14 November 2023. Tanggal 13 November ini merupakan momen KPU menetapkan pasangan Capres dan Cawapres pada Pilpres 2024, sedangkan tanggal 14 adalah waktu pengundian Nomor Urut pasangan calon tersebut. Kemudian untuk membentuk *dataset* ini ditentukan beberapa kriteria pencarian (*searchQuery*), tagar (*hashtag*) dengan hasil jumlah data pada masing-masing *dataset* adalah sebagai berikut:

Tabel 1. *Dataset* Pilpres 2024

Dataset	***searchQuery*/Kata Kunci/*Hashtag***	**Jumlah Data**	**Jumlah Data Unik**
aniesmuhaimin	#aniespresiden, #AMIN, #aniesmuhaimin, #aniescakimin, #anies2024, #aniesbaswedan, #koalisiperubahan, #muhaimin, #anies, #cakimin,	3045	987
prabowogibran	#prabowo2024, #prabowogibran, #koalisiindonesiamaju, #prabowo, #gibran, #KIM, #indonesiamaju	1826	248
ganjarmahfud	#ganjar2024, #mahfudmd, #mahfud, #ganjarpranowo, #GAMA, #ganjar, #ganjarpresiden, #ganjarmahfud	2435	2072
	Total	7306	3307

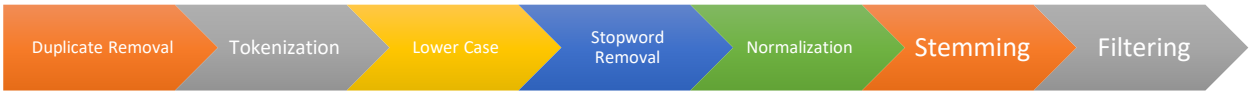
Dari hasil *web scraping* tersebut ditunjukkan pada Tabel 1. Jumlah data yang berhasil didapatkan untuk *Dataset* **"aniesmuhaimin"** adalah sebanyak 3.045 data *Twitter*, dengan jumlah data unik adalah 987 opini atau 32.42% dari jumlah total *dataset*. Sedangkan jumlah data yang berhasil didapatkan untuk *Dataset* **"prabowogibran"** adalah sebanyak 1.826 data *Twitter*, dengan jumlah data unik adalah 248 opini atau 13.59% dari jumlah total *dataset*. Jumlah data yang berhasil didapatkan untuk *Dataset* **"ganjarmahfud"** adalah sebanyak 2.435 data *Twitter*, dengan jumlah data unik adalah 2.072 opini atau 85.07% dari jumlah total *dataset*. Sehingga didapatkan jumlah total dari

seluruh *dataset* adalah 7.306 data *tweet* dengan total data unik tanpa duplikasi adalah 3.307 data *tweet*.
Setelah data berhasil diambil, langkah selanjutnya adalah membersihkan dan merapikan *dataset*. Proses pembersihan ini mencakup penghapusan *tweet* duplikat, karakter khusus, dan elemen yang tidak relevan sehingga memastikan keakuratan dan ketelitian data yang akan digunakan dalam analisis sentimen. Proses ini dilakukan pada tahap *pre-processing* selanjutnya.

2.2 Pre-processing

Setelah berhasil mengumpulkan data, langkah berikutnya adalah melakukan pembersihan dan penataan *dataset*. Langkah *pre-processing* memiliki peran sentral dalam algoritma *data mining* dan *Deep Learning* dengan tujuan meningkatkan mutu data dan kinerja algoritma [13] Metode *pre-processing* mencakup *lemmatization*, penghapusan *stopwords*, dan *case folding*, yang berfungsi untuk menstandarisasi dan membersihkan data teks [13]. Ekstraksi fitur merupakan bagian lain dari proses *pre-processing*, di mana fitur-fitur terpilih disimpan dalam representasi *Bag of Words* [11]. Seleksi fitur memiliki peranan yang sangat penting dalam tugas klasifikasi, karena dapat menghilangkan fitur yang tidak relevan dan meningkatkan akurasi [11].

Pre-processing juga melibatkan pembersihan dan transformasi data, seperti penghapusan kata-kata, digit, dan istilah yang tidak diinginkan. Teknik seperti *word embedding*, contohnya GloVe, digunakan untuk mengubah data teks menjadi representasi kata yang berdimensi rendah dan padat [13]. Secara umum, pra-pemrosesan memiliki peran krusial dalam persiapan data, ekstraksi fitur, serta peningkatan akurasi dan kinerja algoritma *data mining* dan *Deep Learning*.
Proses pembersihan melibatkan penghapusan *tweet* duplikat, karakter khusus, dan elemen yang tidak relevan, memastikan bahwa data yang digunakan untuk analisis sentimen memiliki tingkat keakuratan dan ketelitian yang tinggi. Ini adalah gambaran proses *pre-processing* yang dilakukan seperti pada gambar 1 berikut



Gambar 1. Proses *Pre-processing* Teks *Platform* X (*Twitter*)

Dengan *dataset* yang telah dibersihkan, langkah selanjutnya adalah menerapkan teknik analisis sentimen menggunakan algoritma atau model *machine learning*. Model ini berfungsi untuk mengklasifikasikan setiap *tweet* ke dalam kategori sentimen yang sesuai, seperti positif, negatif, atau netral. Pada proses selanjutnya dilakukan pelabelan terhadap setiap *Twitter* pada *dataset*. Pelabelan ini diperlukan untuk klasifikasi supervised *machine learning* yaitu menggunakan *Support Vector Machine*.

2.3. Automatic Labelling

Auto labeling (pelabelan otomatis) atau proses pemberian label atau tag secara otomatis pada data atau dokumen tanpa campur tangan manusia, sebuah metode yang efisien dalam aspek waktu dan biaya jika dibandingkan dengan pelabelan manual. *Auto Labelling* Berfungsi sebagai teknik yang bertujuan meningkatkan efisiensi dan kualitas dalam berbagai tugas, *Term*asuk analisis sentimen,

klasifikasi teks, dan annotasi data. Dalam konteks khusus dari paper yang diberikan, *Auto labeling* secara eksplisit ditekankan dalam ranah analisis sentimen yang diterapkan pada data *Twitter*.

Algoritma *Auto Labelling* seperti *TextBlob* dan VADER digunakan untuk mengategorikan *tweet* secara mandiri ke dalam sentimen positif, negatif, atau netral. Tujuan utama di balik penerapan *Auto labeling* adalah untuk mengurangi kebutuhan annotasi manual, sehingga membuat proses menjadi lebih *scalable* dan efisien. Berbagai algoritma dan teknik, *Term*asuk pendekatan berbasis leksikon seperti *TextBlob* dan VADER, yang bergantung pada kamus atau aturan yang telah ditentukan sebelumnya untuk memberikan label sentimen pada teks, dapat digunakan untuk mencapai *Auto labeling*. Penting untuk diakui, bagaimanapun, bahwa annotasi otomatis atau *Auto labeling* mungkin tidak selalu seakurat annotasi manual dan dapat menunjukkan keterbatasan dan kelemahan bawaan [11].

2.3.1 VADER

Vader merupakan teknik pelabelan sentimen otomatis yang digunakan dalam analisis sentimen untuk data media sosial. Ini merupakan salah satu dari tiga metode pelabelan otomatis yang dibandingkan dalam penelitian tersebut [11]. Vader menggunakan skor sentimen berdasarkan kombinasi fitur leksikal, aturan gramatikal, dan urutan kata untuk menetapkan label sentimen pada data teks. VADER adalah alat analisis perasaan berbasis leksikon dan aturan yang secara khusus peka terhadap sentimen yang diungkapkan dalam media sosial. VADER menggunakan kombinasi fitur leksikal (misalnya, kata-kata) yang umumnya ditandai dengan arah semantiknya sebagai positif atau negatif. Dengan demikian, VADER tidak hanya memberikan informasi tentang skor polaritas, tetapi juga memberikan informasi sejauh mana suatu opini bersifat positif atau negatif [25].

VADER menggunakan *lexicon* yang terdiri dari kata-kata dan frasa yang telah dikaitkan dengan sentimen tertentu, yaitu positif, negatif, atau netral. VADER menggunakan *lexicon* ini untuk menghitung nilai polaritas dari teks. Nilai polaritas berkisar antara -1 (sangat negatif) hingga 1 (sangat positif). Selain menggunakan *lexicon*, VADER juga menggunakan beberapa aturan untuk menghitung sentimen dari teks. Aturan-aturan ini digunakan untuk menangani beberapa kasus khusus, seperti: 1) Kata-kata yang dapat memiliki sentimen yang berbeda tergantung pada konteksnya. 2) Kata-kata yang dapat memiliki sentimen yang berbeda tergantung pada struktur kalimatnya. VADER telah dilatih dengan kumpulan data teks yang besar. Kumpulan data ini berisi teks dengan berbagai sentimen, yaitu positif, negatif, dan netral.

```
import vader_Sentiment
analyzer = vader_Sentiment.Sentiment_IntensityAnalyzer()
text = "Film ini sangat bagus! Aku sangat menikmatinya."
sentiment = analyzer.Polarity_scores(text)
print(sentiment)
```

Dari contoh *coding* diatas maka akan menghasilkan:

```
{'neg': 0.0000, 'neu': 0.6000, 'pos': 0.4000, 'compound': 0.6000}
```

Dalam contoh tersebut, nilai neg adalah 0, nilai neu adalah 0,6, nilai pos adalah 0,4, dan nilai *compound* adalah 0,6. Dengan demikian, dapat disimpulkan bahwa teks tersebut adalah positif dengan nilai polaritas 0,6. Nilai **NEG** *(Negative)* dan **POS** *(Positive)* menunjukkan seberapa banyak kata-kata dan frasa positif dan negatif yang muncul dalam teks. Nilai neg yang rendah dan nilai pos

yang tinggi menunjukkan bahwa teks tersebut positif. Nilai **NEU** *(Neutral)* menunjukkan seberapa banyak kata-kata dan frasa netral yang muncul dalam teks. Nilai neu yang tinggi menunjukkan bahwa teks tersebut netral. Nilai ***Compound*** adalah nilai yang paling penting untuk menentukan sentimen dari teks. Nilai *compound* yang tinggi menunjukkan bahwa teks tersebut positif. Perlu diingat bahwa hasil Vader hanya merupakan perkiraan. Hasil yang lebih akurat dapat diperoleh dengan menggunakan metode pembelajaran mesin salah satunya dengan menggunakan *Support Vector Machine* (SVM).

2.3.2 TextBlob

TextBlob adalah alat analisis sentimen populer lainnya yang menawarkan pendekatan yang lebih sederhana dan umum. *TextBlob* menggunakan model analisis sentimen yang telah dipelajari sebelumnya dan mengandalkan leksikon polaritas untuk mengklasifikasikan sentimen dari setiap kata dalam teks. Meskipun VADER dan *TextBlob* umumnya digunakan untuk analisis sentimen, VADER dianggap lebih valid dan akurat dalam menganalisis sentimen, terutama dalam konteks situs web media sosial.[14] *TextBlob*, di sisi lain, adalah perpustakaan pemrosesan bahasa alami yang bersifat umum dan menawarkan berbagai fitur, *Term*asuk analisis sentimen. Analisis sentimen *TextBlob* bekerja dengan memberikan skor sentimen untuk setiap kata dalam teks dan kemudian menghitung skor sentimen keseluruhan untuk seluruh teks. [14]

TextBlob akan menggunakan model pembelajaran mesin untuk menentukan sentimen dari teks yang telah di*preprocessing*. *TextBlob* menggunakan model *Naive Bayes* untuk melakukan klasifikasi sentimen menjadi tiga kelas, yaitu: Positif, Netral dan Negatif. Model *Naive Bayes* dalam *TextBlob* bekerja dengan menghitung probabilitas bahwa sebuah kata atau frasa memiliki sentimen tertentu. Probabilitas ini dihitung berdasarkan frekuensi kemunculan kata atau frasa tersebut dalam kumpulan data yang telah dilatih. Berikut adalah contoh penggunaan *TextBlob*.

```
import TextBlob
text = "Film ini sangat bagus! Aku sangat menikmatinya."
sentiment = TextBlob.sentiment(text)
print(sentiment)
```

Dari contoh *coding* diatas maka akan menghasilkan :

```
Sentiment (Subjectivity=0.6, Polarity=0.8)
```

Dari output tersebut, dapat dilihat bahwa sentimen dari teks tersebut adalah positif dengan nilai polaritas 0.8 dan nilai subjektivitas 0.6. ***Subjectivity*** merupakan nilai subjektivitas dari teks, yaitu nilai yang menunjukkan seberapa subjektif teks tersebut. Nilai subjektivitas berkisar antara 0 (tidak subjektif) hingga 1 (sangat subjektif), dan ***Polarity*** merupakan nilai polaritas dari teks, yaitu nilai yang menunjukkan sentimen dari teks, yaitu positif, negatif, atau netral. Nilai polaritas berkisar antara -1 (sangat negatif) hingga 1 (sangat positif) dan Nol adalah netral.

2.4 Feature Extraction (TFIDF) dan N-Gram

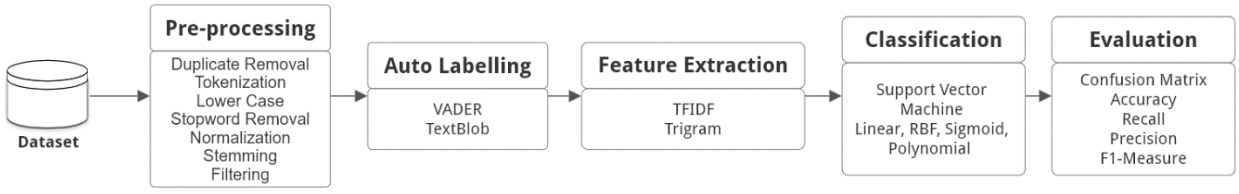
Proses ekstraksi fitur menggunakan metode *Term Frequency-Inverse Document Frequency* (TF-IDF) dalam penelitian ini melibatkan pembobotan kata pada data komentar setelah melalui tahap *preprocessing* dan *Auto Labelling*. Pada langkah ini, setiap kata atau *Term* yang telah melalui proses

preprocessing diberi bobot berdasarkan urutan ID, dengan penambahan *frekuensi Term* (TF) apabila terdapat *Term* yang sama. Proses selanjutnya melibatkan penghitungan jumlah komentar dan N-*Term*, yang diikuti oleh perhitungan TF-IDF untuk menghasilkan pembobotan TF-IDF. Tahapan ini diulang untuk setiap *Term* pada setiap komentar, memastikan bobot TF-IDF mencerminkan tingkat signifikansi relatif suatu *Term* dalam suatu komentar.

Langkah berikutnya melibatkan pemilihan fitur terbaik dengan memanfaatkan teknik *N-Gram*, yang dapat dikonfigurasi untuk mencari rentang *N-Gram (N-Gram range)* yang optimal dan membatasi jumlah fitur maksimal (*max feature*) guna mengontrol kompleksitas model dan menghindari *overfitting*. Dalam penelitian ini, konfigurasi *N-Gram* diatur pada ngram_range=(1,3) untuk *Unigram* + *Bigram* + *Trigram* (*UniBiTri*) [26]. Setelah menentukan jangkauan *N-Gram* yang diinginkan, dilakukan perhitungan nilai TF-IDF pada setiap jangkauan *N-Gram*. Proses selanjutnya mencakup normalisasi nilai TF-IDF, menghasilkan *Feature Vector Matrix* yang digunakan dalam analisis berikutnya. Normalisasi dilakukan untuk membandingkan nilai dengan efektif, dan *Feature Vector Matrix* yang telah dinormalisasi digunakan sebagai dasar analisis yang lebih lanjut dalam penelitian ini.

2.4 MODEL ANALISIS SENTIMEN

Langkah awal dalam implementasi model analisis sentimen ini adalah melakukan pengumpulan data dari *platform* media sosial, khususnya *Twitter* (X). Setelah itu, data disiapkan dan melibatkan serangkaian teknik *preprocessing* yang mencakup tahapan pembersihan data dari karakter khusus, penghilangan tautan atau URL, serta pemisahan kata-kata dalam teks. Konsep umum model Analisis Sentimen *Auto Labelling* digambarkan pada Gambar 2 berikut:



Gambar 2. Model Analisis Sentimen *Auto Labelling* menggunakan SVM

Proses *preprocessing* ini melibatkan sejumlah teknik, seperti *Duplicate Removal* (penghapusan duplikat pada *tweet*), *Tokenization* (pemisahan teks menjadi kata-kata individu), *Lower Case* (pengonversian huruf teks menjadi huruf kecil), Stopword Removal (penghilangan *stopwords* seperti "*a*", "*the*", "dan" yang tidak membawa makna signifikan), *Normalization* (penormalkan format), Stemming (pengubahan kata ke bentuk dasarnya), dan *Filtering* untuk menghilangkan karakter khusus, angka, URL, *hashtag*, dan sebagainya. Penerapan teknik-teknik ini bertujuan untuk membersihkan data secara menyeluruh, meningkatkan kualitas data, dan memastikan bahwa data yang akan digunakan untuk analisis sentimen telah terstandarisasi dengan baik. Dengan demikian, langkah-langkah *preprocessing* ini menjadi kunci dalam mempersiapkan data agar siap digunakan dalam pembuatan model analisis sentimen yang akurat dan dapat diandalkan.

Tahap berikutnya adalah proses pelabelan *Twitter* menggunakan pendekatan *Auto Labelling* (pelabelan otomatis) yaitu dengan menggunakan VADER dan *TextBlob*. Vader menggunakan *lexicon* untuk menghitung sentimen dari teks, sedangkan *TextBlob* menggunakan metode *lexicon based* dan *machine learning*. Setelah dilakukan *Auto Labelling* dengan VADER dan *TextBlob*, *dataset* kemudian dibagi menjadi dua bagian utama: data *training* dan data *testing*. Pada penelitian ini menggunakan

skenario 75:25 yaitu data *training* sebesar 75% dan data *testing* adalah 25% dari total data *sample* yang digunakan.

Proses berlanjut dengan penggunaan model pada proses seleksi yang memanfaatkan data *training*. Pada tahap ini, berbagai model dapat diuji dan dievaluasi untuk menemukan model yang paling sesuai dengan karakteristik *dataset* dan tujuan analisis sentimen yang diinginkan. Keseluruhan proses ini dapat disimak melalui ilustrasi yang dijelaskan lebih lanjut pada gambar yang tidak disertakan dalam deskripsi ini. Dengan pendekatan ini, model analisis sentimen diharapkan mampu memberikan hasil yang akurat dan dapat diandalkan dalam menganalisis opini dari data *Twitter*.

Dalam langkah ini, dapat diterapkan pencarian optimal untuk jangkauan *N-Gram* (*N-Gram* range) dan batasan maksimal fitur (*max feature*) pada setiap prosesnya. Penerapan ini bertujuan untuk memastikan bahwa konfigurasi parameter tersebut optimal dan memberikan kontribusi terbaik dalam pembentukan model analisis sentimen. Proses *Sentiment Prediction* menggunakan data *Testing* yang telah ditentukan dan dibagi dari langkah-langkah sebelumnya. Prediksi sentimen ini dilakukan dengan memanfaatkan model *Support Vector Machine* (SVM) yang optimal dari proses sebelumnya. Dengan demikian, diperoleh nilai akurasi, *Precision*, *recall* yang optimal pada tahap evaluasi.

Dengan memastikan konfigurasi parameter yang optimal dan menggunakan model SVM terbaik, hasil evaluasi pada proses *Sentiment Prediction* diharapkan memberikan performa analisis sentimen yang akurat dan dapat diandalkan. Proses ini menjadi langkah kritis dalam menguji dan mengonfirmasi keefektifan model analisis sentimen yang telah dikembangkan sebelumnya.

2.5 Evaluasi Confusion matrix

Confusion matrix merupakan tabel yang berperan penting dalam mengevaluasi efektivitas suatu algoritma klasifikasi. Fungsinya melibatkan visualisasi dan summarization untuk menyajikan sejauh mana kinerja algoritma dalam melakukan klasifikasi data [27] **.** *Confusion matrix* ditunjukkan pada Tabel 2 berikut.

Tabel 2. ***Confusion matrix***

	Predicted value	
Actual Value	**Positif**	**Negatif**
Positif	TP	FN
Negatif	FP	TN

Dalam *machine learning* dan *data science*, salah satu metrik evaluasi paling umum untuk model klasifikasi adalah ***Confusion matrix*****,** yang terdiri dari empat karakteristik dasar atau angka. Metrik ini digunakan untuk menilai kinerja algoritma klasifikasi, dengan komponen sebagai berikut: **True Positive (TP):** Menunjukkan jumlah label yang diklasifikasikan dengan benar sebagai sentimen Positif, yang berarti label tersebut sesuai dengan klasifikasinya. **True Negative (TN):** Mewakili jumlah label Negatif yang diklasifikasikan dengan benar. **False Positive (FP):** Menyatakan jumlah label Positif yang salah diklasifikasi$Accuracy = \frac{(TP+TN)}{(TP+FP+FN+TN)}$al seharusnya Positif. **False Negative (FN):** Menunjukkan jumlah label Nn sebagai Positif, namun seharusnya Negatif. Beberapa metrik kinerja algoritma melibatkan TP, TN, FP, dan FN di atas, seperti akurasi, presisi, *recall*, dan skor F1. Pengukuran ini menjadi dasar untuk mengevaluasi keefektifan algoritma klasifikasi [27]. Rumus metrik kinerja tersebut disajikan pada formula Akurasi (1), *Precision* (2), *Recall* (3), dan skor pengukuran F1 (4) berikut:

(1)

$$Precision = \frac{TP}{TP+FP} \quad (2)$$

$$\text{Re}\,call = \frac{TP}{TP+FN} \quad (3)$$

$$F1\ \text{Score} = 2 \times \frac{\text{Precision} \times \text{Sensitivity}}{\text{Precision} + \text{Sensitivity}} \quad (4)$$

Akurasi (*accuracy*) memberikan proporsi jumlah prediksi yang benar dari total prediksi. ***Precision*** atau nilai prediksi positif, adalah fraksi nilai positif dari total prediksi positif, yang menunjukkan seberapa akurat model dalam mengidentifikasi positif. Dengan kata lain, presisi adalah proporsi nilai positif yang diidentifikasi dengan benar. ***Recall***, juga disebut *Sensitivity* atau *True Positive Rate* (TPR), adalah bagian dari nilai positif dari total kasus positif aktual, yaitu proporsi kasus positif aktual yang diidentifikasi dengan benar. **Skor F1, Skor F, atau *F-measure*** adalah rata-rata harmonik dari presisi dan *recall*. Skor ini memberikan penilaian yang seimbang antara kedua faktor, memperhitungkan baik keakuratan model dalam mengidentifikasi positif maupun kemampuannya mengidentifikasi sebanyak mungkin kasus positif yang sebenarnya [28]

3. HASIL DAN PEMBAHASAN

3.1 Rasio Data Latih dan Data Uji

Pada pengujian tahap ini menggunakan 3000 *sample* dari total 3307 baris data dari seluruh *dataset* pasangan calon presiden dengan rasio perbandingan data latih dan data uji disajikan pada Tabel 1 berikut:

Tabel 1. Jumlah Rasio Perbandingan Label Positif dan Negatif

Label	Jumlah	Rasio 75:25
Positif	1000	750:250
Netral	1000	750:250
Negatif	1000	750:250
Total	**3000**	**2250:750**

Label Positif dan Negatif memiliki jumlah yang sama yaitu total 3000 baris data pada setiap rasio perbandingan data latih dan data uji. Seperti yang disajikan pada tabel tersebut diatas, pada rasio 75:25 memiliki masing-masing 750 data latih dan 250 data uji pada setiap kelasnya.

3.2 Kinerja Klasifikasi SVM

Pada bagian ini eksperimen dalam rangka komparasi *Auto Labelling* VADER dan *TextBlob* dilakukan. Jumlah data sampel yang digunakan dalam proses pelatihan *(training)* dan pengujian *(testing)* algoritma SVM adalah 750, 1500, 2250 dan 3000 baris data dengan komposisi masing-masing kelas *sentiment* yang seimbang *(balanced dataset)*. Sebagai contoh jumlah data 750 baris data terdiri dari komposisi label Positif (POS), Netral (NEU) dan Negatif (NEG) masing-masing 250 baris data. Demikian juga dengan jumlah *dataset* 1500, terdiri dari label POS, NEU dan NEG masing-masing 500 baris data seimbang. Untuk jumlah sampel data 2250 terdiri dari 750 data seimbang pada

masing-masing label *sentiment*, dan jumlah sampel 3000 yang berarti terdapat 1000 data yang digunakan untuk setiap label sentimen.

Ekstraksi fitur menggunakan TFIDF dan Teknik kombinasi *Unigram*, *Bigram*, dan *Trigram* (*UniBiTri*) atau *ngram_range=(1,3)* [26]. Klasifikasi ini didasarkan pada rasio data latih *(training)* sebesar 75% dan data uji *(testing)* sebesar 25% atau rasio 75:25. Dalam penelitian ini menggunakan *library* Python scikit-learn SVC *Classifier* dengan setelan *hyperparameter* C=1 dan Gamma (γ) = 1. *Kernel* yang digunakan dalam penelitian ini adalah *Linear*, *Sigmoid*, *Polynomial* dan *Radiant Basis Function* (RBF). Hasil klasifikasi disajikan pada Tabel 2. Berikut:

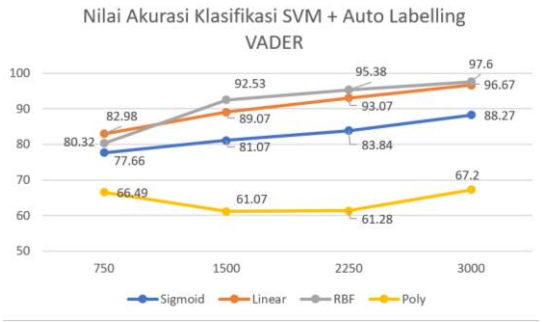
Tabel 2. Nilai Akurasi *Auto Labelling* VADER dan *TextBlob* Algoritma SVM dalam (%)

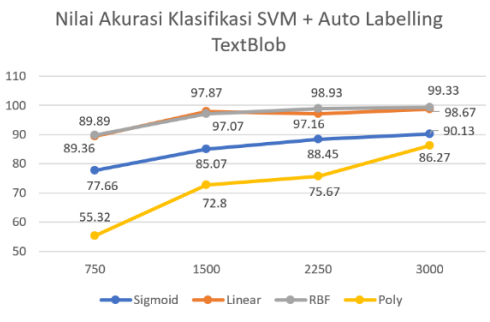
	Akurasi VADER (%)				**Akurasi *TEXTBLOB* (%)**			
Jumlah *Dataset*	***Sigmoid***	***Linear***	**RBF**	***Poly***	***Sigmoid***	***Linear***	**RBF**	***Poly***
750	77.66	**82.98**	80.32	66.49	77.66	89.36	**89.89**	55.32
1500	81.07	89.07	**92.53**	61.07	85.07	**97.87**	97.07	72.8
2250	83.84	93.07	**95.38**	61.28	88.45	97.16	**98.93**	75.67
3000	88.27	96.67	**97.6**	67.2	90.13	98.67	**99.33**	86.27

Berdasarkan hasil klasifikasi pada Tabel 2 tersebut, pada pengujian dengan jumlah data 750 baris, nilai akurasi maksimum untuk klasifikasi SVM dengan *Auto Labelling* VADER adalah 82.98% yaitu pada *kernel Linear* sedangkan akurasi paling minimum adalah 66.49% dengan *kernel Polynomial*. Sedangkan untuk klasifikasi SVM dengan *Auto Labelling TextBlob*, menghasilkan nilai akurasi maksimum menggunakan *kernel* RBF yaitu 89.89% dan nilai minimum dicapai oleh *kernel Polynomial*. Dengan demikian untuk perbandingan *Auto Labelling* dengan jumlah data 750 baris data, dalam pengujian ini *TextBlob* memiliki nilai akurasi lebih baik daripada VADER.

Berikutnya pengujian dengan jumlah baris data 1500, VADER menghasilkan nilai akurasi maksimum 92.53% menggunakan *kernel* RBF dan akurasi minimum didapatkan pada *kernel Polynomial*. Sedangkan pada *TextBlob* menghasilkan nilai akurasi maksimum menggunakan *kernel Linear* adalah 97.87% dan nilai akurasi minimum 72.80% pada *kernel Polynomial*. Dengan demikian, nilai akurasi *Auto Labelling TextBlob* lebih baik daripada VADER.

Ekperimen dengan jumlah data 2250 baris, nilai akurasi maksimum *Auto Labelling* VADER didapatkan oleh *kernel* RBF yaitu 95.38%, dan minimum 61.28% pada *kernel Polynomial*. Sedangkan pada *TextBlob* menghasilkan nilai akurasi maksimum adalah 98.93%, dan minimum 75.67% pada *kernel*. Pada pengujian dengan jumlah 3000 baris data, nilai akurasi maksimum VADER diperoleh dengan *kernel* RBF yaitu 97.60% dan minimum akurasi diperoleh dengan *kernel Polynomial* adalah 67.20%. Sementara *TextBlob* menghasilkan nilai akurasi paling maksimum diantara tahap pengujian lainnya yaitu 99.33% sementara itu nilai akurasi minimum yang dicapai oleh *kernel Polynomial* adalah 86.27%.





(a) **Gambar 3(a).** Nilai akurasi SVM VADER

(b) **Gambar 3(b).** Nilai akurasi SVM *TextBlob*

Berdasarkan Tabel 2 dan eksperimen yang telah dilakukan, digambarkan visualisasi seperti pada gambar 3a untuk nilai akurasi *Auto Labelling* VADER sedangkan gambar 3b visualisasi *TextBlob*. Secara umum nilai akurasi pada gambar 3a dan 3b meningkat pada setiap peningkatan jumlah *dataset*. Terdapat dua *kernel* yang memiliki nilai akurasi tinggi pada VADER dan *TextBlob* yaitu *kernel Linear* dan RBF. Namun terdapat visual yang berbeda pada *kernel Polynomial* pada gambar 3a, dimana nilai akurasi *dataset* 750 adalah 66.49%, kemudian pada *dataset* 1500 nilai akurasi turun menjadi 61.07%, sedangkan untuk *dataset* 2250 nilai akurasi tidak terpaut jauh dari *dataset* sebelumnya.

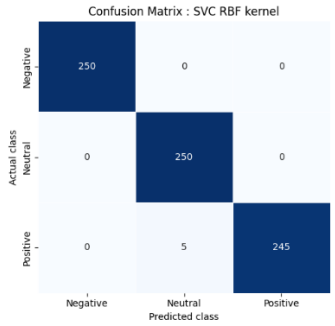
Dalam klasifikasi SVM terdapat laporan yang memberikan gambaran lebih detail tentang akurasi, presisi, *recall*, dan *F1-score*, ini merupakan metrik yang digunakan untuk mengevaluasi kinerja model klasifikasi untuk setiap kelas. Pada penelitian ini nilai akurasi tertinggi dicapai pada *Auto Labelling* Texblob pada *dataset* 3000 dengan *kernel* RBF. Laporan klasifikasi tersebut dapat dipresentasikan pada Gambar 4 berikut:

Menggunakan Metode :SVC RBF kernel

	precision	recall	f1-score	support
Negative	1.00	1.00	1.00	250
Neutral	0.98	1.00	0.99	250
Positive	1.00	0.98	0.99	250
accuracy			0.99	750
macro avg	0.99	0.99	0.99	750
weighted avg	0.99	0.99	0.99	750

Accuracy score: 99.33%
Precision score: 99.35%
Recall score: 99.33%

(a) **Gambar 4(a).** *Classification Report*



(b) **Gambar 4(b).** *Confusion matrix* RBF *TextBlob* 1000

Berdasarkan Gambar 4a. yang disajikan bahwa *Auto Labelling TextBlob* menghasilkan nilai akurasi paling maksimum diantara tahap pengujian lainnya yaitu 99.33% dengan nilai *Precision* 99.35% dan *Recall* 99.33%. Nilai *Precision* untuk kelas *Positive* (POS) dan *Negative* (NEG) masing-masing bernilai 1.0. Ketika nilai *Precision*, *recall*, dan *F1-score* sebesar 1 untuk Kelas Positif dan Kelas Negatif menunjukkan kinerja klasifikasi yang sempurna. Ini berarti semua instansi yang diprediksi sebagai positif atau negatif adalah benar *(True Positive)* atau *(True Negative)*, dan tidak ada positif palsu *(False Positive)* atau negatif palsu *(False Negative)* untuk kelas-kelas ini. Semua prediksi kelas positif (POS) dan kelas negative (NEG) dinilai akurat.

Sedangkan pada kelas netral (NEU) nilai presisi sebesar 0,98 berarti 98% dari instansi yang diprediksi sebagai netral adalah benar-benar netral, dan terdapat 2% positif palsu. *Recall* sebesar 1,0 berarti model mengidentifikasi semua instansi netral yang sebenarnya, dan tidak ada negatif palsu. *F1-score* sebesar 0,99 merupakan nilai rata-rata harmonik dari presisi dan *recall*. Ini menunjukkan bahwa model Anda berkinerja sangat baik, terutama dalam mengklasifikasikan Kelas Positif (POS) dan Negatif (NEG) dengan sempurna, dan mencapai kinerja tinggi pada Kelas Netral (NEU) juga.

Beberapa alasan kemungkinan untuk presisi yang lebih rendah pada kelas netral adalah a) Ambiguitas data: Instansi kelas netral mungkin ambigu atau sulit untuk dibedakan dari contoh-contoh negatif atau positif. Ambiguitas ini dapat menyebabkan pengklasifikasi salah mengklasifikasikan beberapa instansi netral sebagai positif atau negatif. b) Bias model: Pengklasifikasi mungkin cenderung mementingkan identifikasi contoh negatif dan positif,

menyebabkan kesalahan klasifikasi pada beberapa instansi netral. c) Keterbatasan fitur: Fitur yang digunakan untuk klasifikasi mungkin tidak cukup untuk menangkap semua aspek kelas netral, menyebabkan kesalahan klasifikasi. Pada umumnya pencapaian kinerja ini merupakan hasil yang baik, terutama ketika berurusan dengan *dataset* yang tidak seimbang di mana contoh positif jarang terjadi. Namun, penting untuk mempertimbangkan metrik lain seperti *recall* dan skor F1 untuk mendapatkan gambaran yang lebih komprehensif tentang kinerja model.

4. KESIMPULAN

Dalam ekperimen yang telah dilakukan pada penelitian ini, terdapat pola yang dapat diamati pada Tabel 2 tersebut yaitu semakin banyak jumlah *dataset* yang digunakan untuk pengujian, maka semakin besar pula potensi untuk mendapatkan nilai akurasi yang maksimum. Kelas Positif (POS) dan Negatif (NEG) hasil klasifikasi yang sempurna dengan presisi, *recall*, dan *F1-score* semuanya sama dengan 1. Sedangkan pada kelas Netral (NEU): Presisi, *recall*, dan *F1-score* yang sangat tinggi, meskipun tidak sempurna. Terdapat beberapa instansi yang diprediksi sebagai netral yang sebenarnya bukan, tetapi secara keseluruhan, kinerjanya sangat baik.

Dalam penelitian ini ditemukan pola bahwa *kernel Polynomial* memiliki nilai paling minimum dibandingkan dengan *kernel* yang lainnya. Dengan ini dapat disimpulkan berdasarkan pengujian yang dilakukan dan hasil pada Tabel 2 bahwa pada eksperimen ini kinerja *Auto Labelling TextBlob* lebih baik daripada VADER. Hasil nilai akurasi yang maksimum mencapai 99.33%, *Precision* mencapai 99.35% dan *recall* 99.33% membuat peneliti tidak merasa puas dan terus tertarik untuk melakukan investigasi lebih mendalam pada penelitian selanjutnya

Tentunya hasil yang telah dicapai dalam penelitian ini masih banyak kekurangan. Eksperimen yang dilakukan saat ini masih perlu di uji ulang, salah satunya dengan menerapkan *Cross Validation* (CV). Maka dari itu disarankan pada penelitian selanjutnya algoritma klasifikasi sudah melibatkan cross-validation. Penyetelan *Hyperparameter* masih dilakukan secara statis dan belum dilakukan optimasi. Belum melakukan eksperimen dengan menggunakan algoritma yang lain seperti NBC, *K-Nearest Neighbor* (KNN) Random Forest, yang pada penelitian ini belum digunakan.

Sebagai saran untuk penelitian berikutnya adalah diharapkan dapat mengatasi inkonsistensi kelas Netral yang memiliki skore lebih rendah dapat dilakukan beberapa hal sebagai berikut: a) Menganalisis instansi yang salah diklasifikasikan dengan meneliti instansi netral yang salah diklasifikasikan untuk memahami penyebab kesalahan tersebut. b) Menyesuaikan parameter model melalui tuning *hyperparameter* SVM yang berpotensi dapat meningkatkan kinerja klasifikasi pada kelas netral. c) Menambahkan variasi data pelatihan, dengan meningkatkan jumlah contoh netral dalam data pelatihan untuk memberikan informasi yang lebih baik kepada model untuk klasifikasi yang akurat. d) Menjelajahi fitur alternatif dengan berbagai eksperimen dengan fitur atau teknik rekayasa fitur yang berbeda untuk menangkap karakteristik kelas netral dengan lebih efektif. Dengan menyelidiki kelas netral lebih lanjut dan mengatasi potensi penyebab penurunan sedikit presisi, Anda dapat berupaya meningkatkan kinerja keseluruhan pengklasifikasi SVM Anda dan mencapai akurasi yang lebih konsisten di semua kelas. e) Menggunakan pendekatan *Lexicon* dan *Machine learning* secara bersamaan menggabungkan kelebihan-kelebihan antara *Lexicon* dan *Machine learning* untuk dapat mengujicoba metode *Hybrid* agar mendapatkan suatu metode yang lebih efektif dan efisien.

5. UCAPAN TERIMA KASIH

Terima kasih kepada Lembaga Penelitian dan Pengabdian Masyarakat (LPPM) Universitas Widyagama Malang yang telah mendukung penelitian ini melalui program Hibah Penelitian Internal (PERINTIS).

6. REFERENSI

[1] Y. Zhang, F. Chen, and K. Rohe, "Social Media Public Opinion as Flocks in a Murmuration: Conceptualizing and Measuring Opinion Expression on Social Media," *Journal of Computer-Mediated Communication*, vol. 27, no. 1, p. zmab021, Jan. 2022, doi: 10.1093/jcmc/zmab021.

[2] H. Lawelai, A. Sadat, and A. Suherman, "DEMOCRACY AND FREEDOM OF OPINION IN SOCIAL MEDIA: *SENTIMENT* ANALYSIS ON *TWITTER*," *PRAJA: Jurnal Ilmiah Pemerintahan*, vol. 10, no. 1, Art. no. 1, Feb. 2022, doi: 10.55678/prj.v10i1.585.

[3] M. A. Beg and D. Jadon, "*Opinion Mining* from Social Media," *IJRASET*, vol. 10, no. 5, pp. 3643–3647, May 2022, doi: 10.22214/ijraset.2022.43233.

[4] M. Marpaung and F. Humida, "ANALISIS SENTIMEN OPINI MASYARAKAT INDONESIA MENGENAI CALON GUBERNUR DKI JAKARTA 2017 MENGGUNAKAN *NAÏVE BAYES CLASSIFIER*," s1, UAJY, 2017. Accessed: Nov. 19, 2023. [Online]. Available: http://e-journal.uajy.ac.id/11785/

[5] S. N. J. Fitriyyah, N. Safriadi, and E. E. Pratama, "Analisis Sentimen Calon Presiden Indonesia 2019 dari Media Sosial *Twitter* Menggunakan Metode *Naive Bayes*," *JEPIN (Jurnal Edukasi dan Penelitian Informatika)*, vol. 5, no. 3, pp. 279–285, Dec. 2019, doi: 10.26418/jp.v5i3.34368.

[6] A. F. Sabily, P. P. Adikara, and M. A. Fauzi, "Analisis Sentimen Pemilihan Presiden 2019 pada *Twitter* menggunakan Metode Maximum Entropy," 2019.

[7] R. Rinaldo, A. P. Sari, and E. Fardiana, "DIGITAL OPINION #PUANADALAHHARAPAN DI MEDIA SOSIAL *TWITTER* MENGGUNAKAN SOCIAL NETWORK ANALYSIS," *Jurnal Ilmiah Multidisiplin*, vol. 2, no. 01, Art. no. 01, Jan. 2023, doi: 10.56127/jukim.v2i01.407.

[8] S. S. Arumugam, "Development of argument *based Opinion Mining* model with *sentiment*al data analysis from *Twitter* content," *Concurrency and Computation: Practice and Experience*, vol. 34, no. 15, p. e6956, 2022, doi: 10.1002/cpe.6956.

[9] A. F. Hidayatullah, S. Cahyaningtyas, and A. M. Hakim, "*Sentiment* Analysis on *Twitter* using Neural Network: Indonesian Presidential Election 2019 *Dataset*," *IOP Conf. Ser.: Mater. Sci. Eng.*, vol. 1077, no. 1, p. 012001, Feb. 2021, doi: 10.1088/1757-899X/1077/1/012001.

[10] R. Maskat, M. Faizzuddin Zainal, N. Ismail, N. Ardi, A. Ahmad, and N. Daud, "Automatic Labelling of Malay *Cyberbully*ing *Twitter* Corpus using Combinations of *Sentiment*, Emotion and Toxicity Polarities," in *2020 3rd International Conference on Algorithms, Computing and Artificial Intelligence*, Sanya China: ACM, Dec. 2020, pp. 1–6. doi: 10.1145/3446132.3446412.

[11] S. Biswas, K. Young, and J. Griffith, "A Comparison of Automatic Labelling Approaches for *Sentiment* Analysis:," in *Proceedings of the 11th International Conference on Data science, Technology and Applications*, Lisbon, Portugal: SCITEPRESS - Science and Technology Publications, 2022, pp. 312–319. doi: 10.5220/0011265900003269.

[12] G. Buntoro, R. Arifin, G. Syaifuddiin, A. Selamat, O. Krejcar, and F. Hamido, "THE IMPLEMENTATION OF THE *MACHINE LEARNING* ALGORITHM FOR THE *SENTIMENT* ANALYSIS OF INDONESIA'S 2019 PRESIDENTIAL ELECTION," *IIUMEJ*, vol. 22, no. 1, pp. 78–92, Jan. 2021, doi: 10.31436/iiumej.v22i1.1532.

[13] V. O. Tama, Y. Sibaroni, and Adiwijaya, "Labeling Analysis in the Classification of Product Review *Sentiment*s by using Multinomial *Naive Bayes* Algorithm," *J. Phys.: Conf. Ser.*, vol. 1192, p. 012036, Mar. 2019, doi: 10.1088/1742-6596/1192/1/012036.

[14] U. Mehta and D. D. Verma, "*Sentiment* Analysis of Facebook Using *TextBlob* and Vader," vol. 4, p. 5, 2021.

[15] D. T. Lukmana, S. Subanti, and Y. Susanti, "ANALISIS SENTIMEN TERHADAP CALON PRESIDEN 2019 DENGAN *SUPPORT VECTOR MACHINE* DI *TWITTER*," *Seminar & Conference Proceedings of UMT*, no. 0, Art. no. 0, Jun. 2019, doi: 10.31000/cpu.v0i0.1693.

[16] W. H. Silitonga and J. I. Sihotang, "Analisis Sentimen Pemilihan Presiden Indonesia Tahun 2019 Di *Twitter* Berdasarkan Geolocation Menggunakan Metode *Naïve Bayes*ian Classification," *TeIKa*, vol. 9, no. 02, pp. 115–127, Oct. 2019, doi: 10.36342/teika.v9i02.2199.

[17] A. R. Abdillah and F. N. Hasan, "Analisis Sentimen Terhadap Kandidat Calon Presiden Berdasarkan *Tweet*s Di Sosial Media Menggunakan *Naive Bayes Classifier*," *SMATIKA*, vol. 13, no. 01, pp. 117–130, Jul. 2023, doi: 10.32664/smatika.v13i01.750.

[18] A. Khare, A. Gangwar, S. Singh, and S. Prakash, "*Sentiment* Analysis and Sarcasm Detection of Indian General Election *Tweets*," *ArXiv*, Jan. 2022, Accessed: Nov. 06, 2023. [Online]. Available: https://www.semanticscholar.org/paper/*Sentiment*-Analysis-and-Sarcasm-Detection-of-Indian-Khare-Gangwar/583f7a31ce1df0d3c974b9a19a6a249248463070

[19] L. A. Andika, P. A. N. Azizah, and R. Respatiwulan, "Analisis Sentimen Masyarakat terhadap Hasil Quick Count Pemilihan Presiden Indonesia 2019 pada Media Sosial *Twitter* Menggunakan Metode *Naive Bayes Classifier*," *IJAS*, vol. 2, no. 1, p. 34, Jul. 2019, doi: 10.13057/ijas.v2i1.29998.

[20] E. B. Santoso and A. Nugroho, "Analisis Sentimen Calon Presiden Indonesia 2019 Berdasarkan Komentar Publik Di Facebook," *Jurnal Eksplora Informatika*, vol. 9, no. 1, pp. 60–69, Sep. 2019, doi: 10.30864/eksplora.v9i1.254.

[21] C. Humam and A. D. Laksito, "Implementasi Aplikasi Sentimen Pada Data *Twitter* Jelang Pemilu 2024," *JPIT*, vol. 8, no. 2, pp. 105–112, May 2023, doi: 10.30591/jpit.v8i2.5051.

[22] H. N. Chaudhry *et al.*, "*Sentiment* Analysis of before and after Elections: *Twitter* Data of U.S. Election 2020," *Electronics*, vol. 10, no. 17, p. 2082, Aug. 2021, doi: 10.3390/electronics10172082.

[23] N. Hayatin, G. I. Marthasari, and L. Nuraini, "Optimization of *Sentiment* Analysis for Indonesian Presidential Election using *Naïve Bayes* and Particle Swarm Optimization," *Jurnal Online Informatika*, vol. 5, no. 1.

[24] H. Atsqalani, N. Hayatin, and C. S. K. Aditya, "*Sentiment* Analysis from Indonesian *Twitter* Data Using *Support Vector Machine* And Query Expansion Ranking," *join*, vol. 7, no. 1, p. 116, Jun. 2022, doi: 10.15575/join.v7i1.669.

[25] V. Nurcahyawati and Z. Mustaffa, "Vader *Lexicon* and *Support Vector Machine* Algorithm to Detect Customer *Sentiment* Orientation," *J. Inf. Syst. Eng. Bus. Intell.*, vol. 9, no. 1, pp. 108–118, Apr. 2023, doi: 10.20473/jisebi.9.1.108-118.

[26] S. W. Irianandra, R. W. Budiawan, A. Y. Rahman, and I. Istiadi, "KINERJA SELEKSI FITUR *N-GRAM* PADA ANALISIS SENTIMEN ULASAN MOBILE GAME DI GOOGLE PLAYSTORE," *The 5th Conference on Innovation and Application of Science and Technology (CIASTECH)*, 2022.

[27] P. Singh, N. Singh, K. K. Singh, and A. Singh, "Chapter 5 - Diagnosing of disease using *machine learning*," in *Machine learning and the Internet of Medical Things in Healthcare*, K. K. Singh, M.

Elhoseny, A. Singh, and A. A. Elngar, Eds., Academic Press, 2021, pp. 89–111. doi: 10.1016/B978-0-12-821229-5.00003-3.

[28] D. K. Sharma, M. Chatterjee, G. Kaur, and S. Vavilala, "3 - *Deep Learning* applications for disease diagnosis," in *Deep Learning for Medical Applications with Unique Data*, D. Gupta, U. Kose, A. Khanna, and V. E. Balas, Eds., Academic Press, 2022, pp. 31–51. doi: 10.1016/B978-0-12-824145-5.00005-8.