

Perbandingan Algoritme *Naïve Bayes* dan *Decision Tree* Pada Analisis Sentimen Data Komentar Siswa Pada Aplikasi *Digital Teacher Assessment*

Ferat Kristanto^{1*}, Wing Wahyu Winarno², Asro Nasiri³

^{1,3}Teknik Informatika, Universitas Amikom Yogyakarta, Yogyakarta, Indonesia

²STIE YKPN, Yogyakarta, Indonesia

*e-mail *Corresponding Author*: feratkristanto@gmail.com

Abstract

Telkom Purwokerto Vocational High School is a school managed by the Telkom Education Foundation. They use Digital Teacher Assessment (DITA) apps. Students provide comments to the teacher using the DITA application. The collected comment data will be grouped into three categories, namely positive, negative, and neutral. Based on the category of comments requires sentiment analysis in grouping these comments. Sentiment analysis uses lexicon-based. After getting sentiment analysis using lexicon-based, then the words are weighted using TF-IDF and then classified and evaluated. This study uses an algorithm naïve Bayes and a decision tree. So the results of the comparative research on the accuracy of the naïve Bayes algorithm and the decision tree with the decision tree algorithm have the highest level of accuracy, namely 99%. So it can be concluded that using the decision tree algorithm is better at classifying student comment sentiment analysis data.

Keywords: *Digital Teacher Assessment; Naïve Bayes; Decision Tree; Lexicon Based; Sentiment Analysis*

Abstrak

Sekolah Menengah Kejuruan (SMK) Telkom Purwokerto adalah sekolah yang dikelola oleh Yayasan Pendidikan Telkom. Mereka menggunakan aplikasi *Digital Teacher Assessment* (DITA). Siswa memberikan komentar ke Guru menggunakan aplikasi DITA. Data komentar yang terkumpul akan dikelompokkan menjadi tiga kategori yaitu komentar positif, negatif, dan netral. Berdasarkan kategori komentar membutuhkan analisis sentimen dalam mengelompokkan komentar tersebut. Analisis sentimen menggunakan *lexicon based*. Setelah mendapatkan analisis sentimen menggunakan *lexicon based*, selanjutnya kata-kata tersebut diberi bobot menggunakan *TF-IDF* lalu di klasifikasi dan di evaluasi. Dalam penelitian ini menggunakan algoritme *Naïve bayes* dan *Decision tree*. Maka hasil penelitian perbandingan akurasi dari algoritme *Naïve bayes* dan *decision tree* dengan algoritme *Decision tree* memiliki tingkat akurasi yang paling tinggi yaitu 99%. Maka dapat disimpulkan bahwa dengan menggunakan algoritme *Decision tree* lebih baik dalam mengklasifikasi data analisis sentimen komentar siswa.

Kata kunci: *Digital Teacher Assessment; Naïve Bayes; Decision Tree; Lexicon Based; Analisis Sentimen*

1. Pendahuluan

Sekolah Menengah Kejuruan (SMK) Telkom Purwokerto adalah sekolah yang dikelola oleh Yayasan Pendidikan Telkom. Mereka menggunakan aplikasi *Digital Teacher Assessment* (DITA) untuk terus meningkatkan kualitas pendidikan yang mereka berikan. Aplikasi DITA digunakan untuk mengumpulkan masukan dan penilaian dari siswa tentang pekerjaan gurunya guna membantu meningkatkan kinerjanya. Penilaian diberikan pada akhir semester atau pada setiap akhir proses pembelajaran.

Proses pengumpulan data difokuskan pada komentar siswa. Data komentar yang terkumpul akan dikelompokkan menjadi tiga kategori yaitu komentar positif, negatif, dan netral. Berdasarkan kategori komentar membutuhkan analisis sentimen dalam mengelompokkan komentar tersebut. Analisis sentimen adalah studi komputasi opini, perasaan, dan emosi yang

diungkapkan dalam teks. Tugas mendasar analisis sentimen adalah mengklasifikasikan polaritas dokumen, kalimat, atau teks opini. Polaritas memiliki arti untuk apa teks itu. Sebuah dokumen, pernyataan, atau pendapat memiliki aspek positif atau negatif. Analisis sentimen biasanya digunakan untuk menilai suka dan tidak suka publik terhadap barang dan jasa [1].

Machine learning adalah cara untuk mempelajari sesuatu dengan cepat dan mudah dengan menggunakan komputer. Ini dilakukan dengan menggunakan algoritme dan metode khusus untuk memprediksi sesuatu, mengenali pola, dan mengklasifikasikan sesuatu. Algoritme-algoritme di *Machine learning* yaitu *Naural Network*, *Decision tree*, *K-nearest neighbor*, *Naïve bayes*, *Random forest* dan lain sebagainya [2]. Dalam penelitian ini menggunakan menggunakan algoritme *naïve bayes* dan *decision tree* dalam *machine learning*. Algoritme *naïve bayes* adalah jenis klasifikasi yang menggunakan probabilitas dan metode statistik. Metode ini lebih cepat dan mudah digunakan dari pada metode lainnya karena hanya membutuhkan sedikit (*training data*) [3]. Algoritme *Decision tree* adalah metode berbasis pohon yang digunakan untuk mengklasifikasikan data. Metode ini dapat ditingkatkan dengan meningkatkan dan mengantongi, karena sering kali terlalu sesuai dengan data [4].

Lexicon based adalah cara untuk menganalisis sentimen tanpa harus melatih data apa pun sebelumnya [5]. Pendekatan ini menggunakan daftar kata yang telah ditentukan sebelumnya, masing-masing dengan skor sentimen yang sesuai [6]. Mudah digunakan dan praktis, menjadikannya pilihan yang baik untuk analisis sentimen ulasan atau komentar [7]. Setelah mendapatkan analisis sentimen menggunakan *lexicon based*, selanjutnya kata-kata tersebut diberi bobot menggunakan TF-IDF. TF-IDF adalah proses transformasi data dari data tekstual menjadi data numerik untuk setiap kata atau fitur yang akan diberi bobot [8].

Penelitian yang akan dilakukan bertujuan untuk analisis sentimen komentar siswa terhadap guru dengan menggunakan perbandingan algoritme *naïve bayes* dan *decision tree*. Perbedaan dari penelitian terdahulu yaitu untuk melakukan analisis sentimen komentar siswa menggunakan proses *preprocessing* lalu menggunakan *lexicon based* untuk analisis sentimen yang dibandingkan dengan menentukan sentimen secara manual untuk mengetahui hasil perbedaan yang diperoleh dengan mengelompokkan sentimen positif, negatif, dan netral kemudian menggunakan pembobotan kata TF-IDF dan hasilnya membandingkan tingkat akurasi di setiap algoritme.

2. Tinjauan Pustaka

Pada penelitian sebelumnya yang dilakukan oleh Franly Salmon Pattihia dan Hendry [9] yang berjudul "Perbandingan Metode *K-NN*, *Naïve Bayes*, *Decision Tree* untuk Analisis Sentimen Tweet Twitter Terkait Opini Terhadap PT PAL Indonesia". Tujuan penelitian untuk analisis sentimen terhadap opini publik pada media sosial Twitter dengan menggunakan data yang telah dikumpulkan menjadi sebuah dataset dan diolah menggunakan tools *Rapidminer*. Penelitian ini menggunakan metode *naïve bayes*, *K-NN* dan *decision tree* untuk melakukan perbandingan dengan melihat tingkat akurasi dari ketiga metode yang digunakan. Hasil dari penelitian menunjukkan bahwa metode *naïve bayes* memiliki tingkat akurasi sebesar 84,08%, metode *K-NN* adalah 83,38% dan *decision tree* adalah 81,09%. Hasil dari penelitian ini dapat menunjukkan metode *naïve bayes* memiliki tingkat akurasi lebih tinggi dari metode lain yang digunakan dengan tingkat akurasi sebesar 84,08%. Selanjutnya penelitian yang dilakukan oleh Toby Wiratama Putra dkk [10] yang berjudul "Analisis Sentimen Pembelajaran Daring menggunakan Metode *naïve bayes*, *KNN*, dan *decision tree*". Tujuan penelitian untuk mendapatkan ketepatan hasil klasifikasi berdasarkan tingkat akurasi dari persepsi mahasiswa yang menggunakan twitter mengenai pemberlakuan kuliah secara online. Dalam penelitian ini ada tiga metode yang akan dipakai untuk proses klasifikasi. Metode yang digunakan yaitu *naïve bayes*, *KNN* dan *decision tree*. Data yang dipakai dalam penelitian ini merupakan data twitter yang diambil dengan cara *crawling data*. Dari hasil pengujian diketahui metode *decision tree* memiliki nilai yang lebih tinggi diantara metode lainnya dengan akurasi sebesar 61.92%, presisi 73.63%, dan *recall* 11.42%.

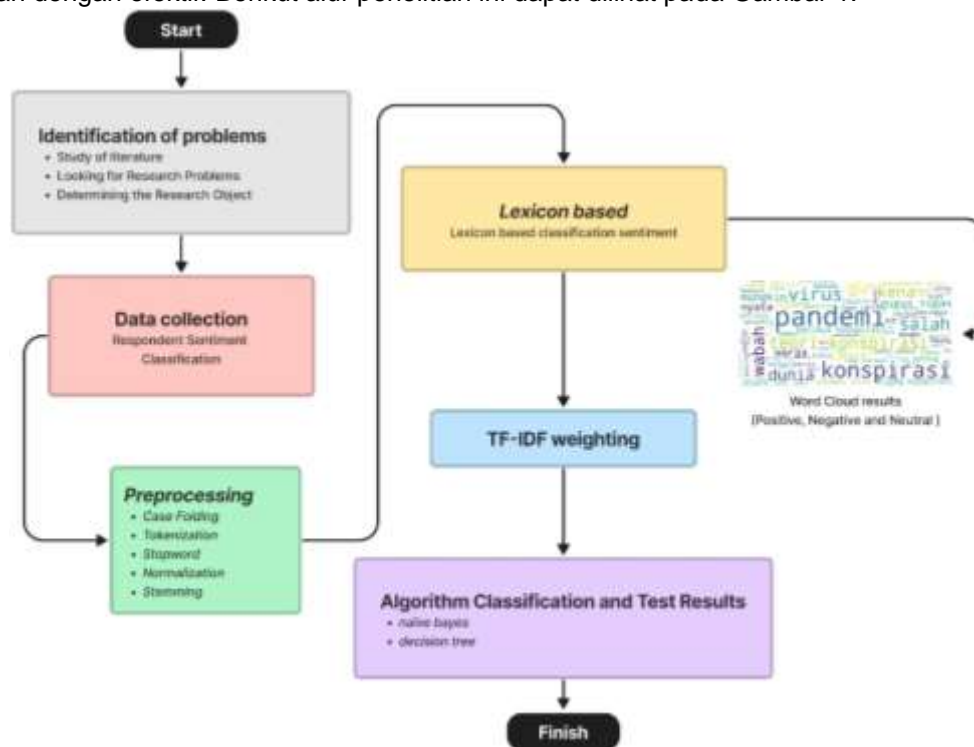
Berdasarkan penelitaian sebelumnya tidak menunjukkan seberapa banyak data training dan data testing yang digunakan, hasil perhitungan manual untuk menghitung akurasi, presisi dan *recall* ada sedikit perbedaan nilai dari perhitungan manual dan menggunakan *Rapid Miner*, dan berdasarkan hasil penelitian untuk nilai tiap akurasi, presisi dan *recall* masih memiliki persentase (%) kurang dari 85%. Penelitian yang akan dilakukan adalah untuk mengelompokkan sentimen komentar siswa terhadap guru yang dilakukan proses

preprocessing data untuk membersihkan dari noise kata lalu di lakukan analisis sentimen menggunakan *Lexicon Based* untuk mengelompokkan sentimen menjadi positif, negatif, dan netral setelah itu proses pembobotan kata menggunakan TF-IDF dengan data sentimen *Lexicon Based* dan selanjutnya melakukan pembagian data training dan data testing sebagai pengujian tingkat akurasi algoritme *Naïve bayes*, dan *Decision tree* secara acak untuk mendapatkan hasil terbaik dari penelitian ini.

3. Metodologi

3.1. Alur Penelitian

Pada penelitian ini dibutuhkan langkah-langkah yang benar agar penelitian ini dapat berjalan dengan efektif. Berikut alur penelitian ini dapat dilihat pada Gambar 1.



Gambar 1. Alur Penelitian

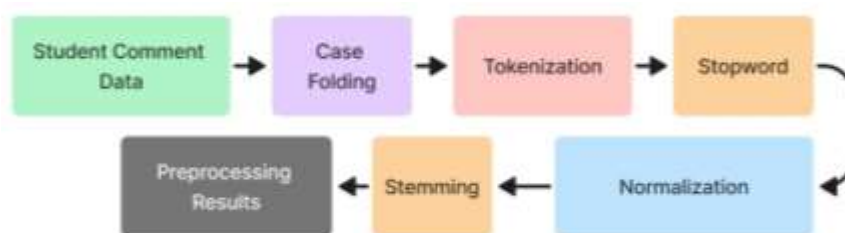
Berdasarkan Gambar 1 alur penelitian terdapat tahapannya, penjelasan tahapan sebagai berikut:

1. Identifikasi Masalah

Pada penelitian ini langkah pertama yaitu identifikasi masalah, didalam identifikasi masalah terdapat studi literatur, mencari permasalahan penelitian, dan menentukan objek. Tahap studi literatur merupakan proses mencari, menggunakan, dan mempelajari berbagai macam literatur berupa jurnal, buku, paper dan lain sebagainya yang berkaitan dengan penelitian analisis sentimen. Permasalahan dan objek yang di gunakan dalam penelitian ini bagaimana melakukan analisis sentimen terhadap komentar siswa di aplikasi *Digital Teacher Assessment (DITA)* dengan menggunakan *lexicon based*, algoritme *naïve bayes* dan *decision tree*. Siswa dapat berkomentar di aplikasi DITA untuk ditujukan ke Guru. Proses pengumpulan data difokuskan pada komentar siswa. Data komentar yang terkumpul akan dikelompokkan menjadi tiga kategori yaitu komentar positif, negatif, dan netral. Berdasarkan kategori komentar membutuhkan analisis sentimen dalam mengelompokkan komentar tersebut.

2. Pengumpulan Data

Pada tahap ini dilakukan pengumpulan data komentar siswa di aplikasi DITA. Data yang diambil lalu diklasifikasi sentimen pada proses labelling sentimen manual.



Gambar 2. Alur Preprocessing

3. *Preprocessing*

Data komentar siswa yang telah terkumpul memiliki banyak *noise* sehingga perlu adanya tahapan proses menghilangkan *noise* agar proses analisis sentimen menjadi lebih akurat [11]. Alur *preprocessing* pada penelitian ini dapat dilihat pada Gambar 2.

4. Lexicon Based

Setelah data di proses preprocessing selanjutnya yaitu klasifikasi data menggunakan *lexicon based*. Dengan mengelompokkan analisis sentimen menjadi 3 kelompok yaitu positif, negatif, dan netral [12]. Hasil *lexicon based* akan di bandingkan dengan data sentimen manual. Selanjutnya hasil *lexicon based* dibuat visualisasi.

5. Pembobotan TF-IDF

Pembobotan kata TF-IDF menghitung nilai bobot setiap kata untuk setiap dokumen. Ini dibagi menjadi dua proses yaitu TF dan IDF. TF (Term Frequency) menghitung jumlah kemunculan setiap kata dalam dokumen, dan dengan kemunculan kata terbanyak, nilai kata tersebut adalah yang terbesar. IDF (*Inverse Document Frequency*) menghitung jumlah dokumen untuk setiap kata yang jarang muncul dalam dokumen yang dianggap memiliki nilai terbesar. Jika kata tersebut memiliki banyak kemunculan kata dalam dokumen, hasilnya akan memiliki nilai yang kecil.

6. Klasifikasi Algoritme dan Hasil Pengujian

Pada klasifikasi algoritme dilakukan pada dataset lexicon based dengan pembobotan TF-IDF lalu di proses dari masing-masing algoritme *Naïve bayes* dan *Decision tree*.

Hasil dari melakukan klasifikasi algoritme menurut sentimen lexicon based adalah perbandingan tingkat akurasi dari setiap algoritme yang menggunakan parameter pengukuran kinerja algoritme, yaitu presisi, *recall*, *f1-measure*, dan akurasi.

3.2. Naïve Bayes

Algoritme *Naïve Bayes* melewati tahapan pelatihan dan klasifikasi dalam proses klasifikasi. Pada tahap pelatihan, proses analisis dilakukan terhadap sampel dokumen, memilih kata-kata yang mungkin muncul dalam kumpulan dokumen sampel sebanyak mungkin untuk mewakili dokumen [13]. Dari sampel dokumen, dicari probabilitas awal untuk setiap kategori. Untuk tahapan klasifikasi satu dokumen, nilai kategori ditentukan berdasarkan term yang muncul pada dokumen yang telah diklasifikasi [14].

3.3. Decision Tree

Algoritme *Decision Tree* adalah metode penambahan data untuk mengklasifikasikan data. Variabel atau fitur adalah root node, internal node, dan terminal node [15]. Label kelas adalah node terminal. Untuk menghasilkan pohon keputusan sering digunakan metode Decision Tree. Data dalam pohon keputusan dinyatakan dalam bentuk tabel dengan atribut dan *record*. Atribut yang ada dievaluasi menggunakan ukuran statistik berupa perolehan informasi, untuk mengukur keefektifan atribut saat mengklasifikasikan sekumpulan sampel data [9].

3.4. Prediksi Klasifikasi

Sistem ini digunakan peneliti untuk mengetahui akurasi data dengan menghitung nilai *precision* (Prec) dan *recall* (Rec) [16]. Matriks Kebingungan kemudian digunakan untuk mendapatkan hasil data aktual, yang kemudian digunakan untuk menghitung prediksi klasifikasi [4]. *Confusion Matrix* dapat dilihat pada Tabel 1.

Table 1. *Confusion Matrix*

		Label atau kelas	
		Positif	Negatif
Positif	True Positive		
	False Negative		
Positif	False Positive		
	True Negative		

Berdasarkan *confusion matrix* pada Tabel 1, dapat diketahui berbagai parameter pengukuran kinerja algoritme, yaitu *presisi*, *recall*, *f1-measure*, dan akurasi [17]. Presisi merupakan parameter untuk mengukur ketepatan dari suatu algoritme. Menghitung presisi dengan rumus yang ditunjukkan pada persamaan (1).

$$precision = \frac{TP}{TP+FP} \quad (1)$$

Recall merupakan parameter untuk mengukur kelengkapan sebuah algoritme. Menghitung *recall* dengan rumus yang ditunjukkan pada persamaan (2).

$$recall = \frac{TP}{TP+FN} \quad (2)$$

F-measure merupakan rata-rata harmonik dari presisi dan *recall*. Nilai tertinggi adalah 1 dan nilai terendah adalah 0. Menghitung *f-measure* dengan rumus yang ditunjukkan pada persamaan (3).

$$F - measure = \frac{2 \times precision \times recall}{precision + recall} \quad (3)$$

Akurasi merupakan perhitungan yang umum digunakan untuk mengevaluasi kinerja dari sebuah algoritme. Akurasi dihitung berdasarkan rasio jumlah data yang diprediksi secara benar oleh algoritme dengan jumlah semua data yang ada pada *dataset*. Menghitung akurasi dengan rumus yang ditunjukkan pada persamaan (4).

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (4)$$

4. Hasil dan Pembahasan

Hasil penelitian analisis sentimen komentar siswa terhadap guru pada aplikasi DITA dengan menggunakan algoritme *Lexicon Based* dan Naive Bayes serta pohon keputusan. Pengumpulan data yang digunakan pada penelitian ini adalah data komentar siswa di aplikasi DITA. Pengolah data menggunakan aplikasi *Jupyter Notebook*.

Data komentar siswa diambil lalu diklasifikasi sentimen dengan proses labelling sentimen secara manual untuk menghasilkan sentimen yang dikelompokkan menjadi 3 yaitu positif, negatif, dan netral. Data excel dengan format CSV yang dilabelling manual ditunjukkan pada Gambar 3.

Mencari tahu Siswa Komentar	Label	Kategori
1000000105 adalah tagornya jgn yg susah-susah ya pak	Positif	Positif
1000000108 sangat baik	Positif	Positif
1000000104 Alungku bisa di Seri materi bertulis	Netral	Netral
1000000109 Es ok	Positif	Positif
1000000112 Sudah cukup baik	Positif	Positif
1000000104 Mohon maaf apabila saat kegiatan pembelajaran saya melakukan kesalahan	Netral	Netral
1000000109 sebelumnya saya sangat berterimakasih atas yaitu dan usahanya dalam pen-	Positif	Positif
1000000104 Tidak ada	Netral	Netral
1000000101 Baik sekali	Positif	Positif
1000000117 terima kasih pak, sudah mengajar kami dengan baik dan sabar. semangat rre!	Positif	Positif
1000000402 Menurut saya sudah baik	Positif	Positif
1000000109 Baik	Positif	Positif
1000000409 Terimakasih pak semoga sehat selalu	Positif	Positif
1000000108 banyak praktik jadi mudah di pahami	Positif	Positif
1000000112 Nice	Positif	Positif
1000000109 semoga kedepannya lebih baik lagi dari sebelumnya	Negatif	Negatif
1000000105 terimakasih pak sudah sabar membimbing kami sampai kami paham	Positif	Positif
1000000082 kurang karangin lagi indahnya pak	Negatif	Negatif
1000000101 mohon maaf kalo selama ini banyak kesalahan yang ditunggu maaf pak	Negatif	Negatif
1000000105 Lebih baik lagi	Negatif	Negatif
1000000108 Tetap kalen pak	Netral	Netral
1000000104 Terimakasih sudah membimbing 2 tahunya	Positif	Positif
1000000402 semoga ketika 11 oktober nya bisa tetap nuka	Netral	Netral
1000000085 The best	Positif	Positif
1000000105 Baik	Positif	Positif
1000000108 Sudah baik dalam mengajar, mungkin saran dari saya materi yang disampaikan	Positif	Positif
1000000105 Sudah baik pak	Positif	Positif
1000000099 semoga bisa lebih baik untuk kedepannya	Netral	Netral

Gambar 3. Data Komentar Labeling Manual

Berdasarkan hasil Gambar 3 maka data komentar yang akan diolah sebanyak 676 komentar siswa, dengan field nomor induk siswa, komentar, dan klasifikasi. Seperti yang di tunjukkan pada Gambar 4.

	Nomor Induk Siswa	Komentar	Klasifikasi
0	3105200165	okeelah. tugasnya jgn yg susah-susah ya pak :v	Positif
1	3105200176	Sangat baik, terba	Positif
2	3105200184	Mungkin bisa di beri mater tertulis	Netral
3	3105200379	Its ok	Positif
4	3105200012	Sudah cukup baik.	Positif
...
671	3105190386	Baikk	Positif
672	3105190548	Materi mudah di pahami	Positif
673	3105200351	Pembelajaran yang disampaikan oleh Pak Agus cu...	Positif
674	3105190500	cukup baik	Positif
675	3105190455	Baik	Positif

676 rows x 3 columns

Gambar 4. Tampilan Awal Data

Tahap *text preprocessing* diperlukan karena data yang dikumpulkan dari komentar siswa banyak mengandung *noise*, dan tidak mengandung informasi yang berguna untuk proses klasifikasi sentimen seperti simbol, tanda baca, angka, dan penggunaan bahasa asing. Tahapan preprocessing yaitu *case folding, tokenization, stopwords, normalization, stemming*.

Tahapan *case folding* dipergunakan untuk mengubah huruf besar menjadi *lower case* atau huruf kecil. Contoh "Mungkin" menjadi "mungkin". Berikut kode untuk tahapan *case folding* seperti yang ditunjukkan pada Gambar 5.

```
#Preprocessing
#case folding
data['Komentar_lower'] = data['Komentar'].str.lower()
data
```

Gambar 5. Proses Case Folding

Tahapan *tokenization* dipergunakan untuk menghapus karakter yang tidak berpengaruh terhadap proses klasifikasi sentimen. Karakter yang di hapus seperti tanda baca koma (,), titik (.), simbol, karakter seperti huruf, angka dan memisahkan kata perkata dari setiap komentar siswa. Berikut kode untuk tahapan *tokenization* seperti yang ditunjukkan pada Gambar 6.

```
def clean_text(text):
    # Remove non-alphabetic characters
    text = re.sub(r'[^a-zA-Z]', '', text)
    # Convert to lowercase
    text = text.lower()
    # Split into words
    words = text.split()
    # Remove empty strings
    words = [word for word in words if word]
    return words

def tokenize(text):
    # Clean text
    words = clean_text(text)
    # Tokenize
    tokens = words
    return tokens

def preprocess(text):
    # Clean text
    words = clean_text(text)
    # Tokenize
    tokens = words
    return tokens
```

Gambar 6. Proses Tokenization

Tahapan *stopword* dipergunakan untuk menghapus kata-kata yang tidak penting berdasarkan pada kamus *stopword*. Berikut kode untuk tahapan *stopword* seperti yang ditunjukkan pada Gambar 7.

```

#Stopword
list_stopwords = set(stopwords.words('indonesian','english'))

def stopwords_removal(words):
    return [word for word in words if word not in list_stopwords]

data['komentar_token_stopword'] = data['komentar_token'].apply(stopwords_removal)

print(data['komentar_token_stopword'].head())
data

```

Gambar 7. Proses *Stopword*

Tahapan *normalization* dipergunakan untuk dilakukan proses untuk menormalisasi kata terhadap setiap kata-kata yang mengandung kata tidak baku maupun noise menjadi kata yang baku dan siap diolah. Berikut kode untuk tahapan *normalization* seperti yang ditunjukkan pada Gambar 8.

```

#Normalization
normalized_word = pd.read_excel("normalisasi.xlsx")

normalized_word_dict = {}

for index, row in normalized_word.iterrows():
    if row[0] not in normalized_word_dict:
        normalized_word_dict[row[0]] = row[1]

def normalized_term(document):
    return [normalized_word_dict[term] if term in normalized_word_dict else term for term in document]

data['komentar_normalized'] = data['komentar_token_stopword'].apply(normalized_term)

print(data['komentar_normalized'].head())
data

```

Gambar 8. Proses *Normalization*

Tahapan *stemming* digunakan untuk mengembalikan kata ke bentuk dasarnya. Contoh "Tugasnya" menjadi "tugas". Berikut kode untuk tahapan *stemming* seperti yang ditunjukkan pada Gambar 9.

```

#Stemming
from nltk.stem import PorterStemmer, SnowballStemmer
import nltk

factory = PorterStemmer()
stemmer = factory.create_stemmer()

def stemmed_mapper(term):
    return stemmer.stem(term)

term_dict = {}

for document in data['komentar_normalized']:
    for term in document:
        if term not in term_dict:
            term_dict[term] = ''

print(len(term_dict))
print("-----")

for term in term_dict:
    term_dict[term] = stemmed_mapper(term)
    print(term, " ", term_dict[term])

print(term_dict)
print("-----")

def get_stemmed_term(document):
    return [term_dict[term] for term in document]

data['komentar_token_stemmer'] = data['komentar_normalized'].apply(get_stemmed_term)
data.head()
data

```

Gambar 9. Proses *Stemming*

Proses *preprocessing* telah selesai. Berikut hasil tahapan *preprocessing* dari *case folding*, *tokenization*, *stopword*, *normalization*, *stemming* dan terakhirnya menjadi komentar baru setelah melalui tahap *preprocessing* yang di tunjukkan pada Gambar 10.

Asi	Komentar_lower	Komentar_token	Komentar_token_kemuncutan	Komentar_token_Stopword	Komentar_normalized	Komentar_token_stemmed	Komentar_baru
utf	okelah tugasnya jgn yg susah-susah ya pak	[okelah, tugasnya, jgn, yg, susah-susah, ya, pak]	{'okelah': 1, 'tugasnya': 1, 'jgn': 1, 'yg': 1, ...}	[okelah, tugasnya, jgn, yg, susah-susah, ya]	[okelah, tugasnya, jgn, yg, susah-susah, ya]	[oke, tugas, jgn, yg, susah-susah, ya]	oke tugas jgn yg susah-susah ya
utf	sangat bakterba	[sangat, bakterba]	{'sangat': 1, 'bakterba': 1}	[bakterba]	[bakterba]	[bakterba]	bakterba
utf	mungkin bisa di beri materi tertulis	[mungkin, bisa, di, beri, materi, tertulis]	{'mungkin': 1, 'bisa': 1, 'di': 1, 'beri': 1, ...}	[materi, tertulis]	[materi, tertulis]	[materi, tulis]	materi tulis
utf	its ok	[its, ok]	{'its': 1, 'ok': 1}	[its, ok]	[its, ok]	[its, ok]	its ok
utf	sudah cukup baik	[sudah, cukup, baik]	{'sudah': 1, 'cukup': 1, 'baik': 1}	[]	[]	[]	
utf	baik	[baik]	{'baik': 1}	[baik]	[baik]	[baik]	baik
utf	materi mudah di pahami	[materi, mudah, di, pahami]	{'materi': 1, 'mudah': 1, 'di': 1, 'pahami': 1}	[materi, mudah, pahami]	[materi, mudah, pahami]	[materi, mudah, pahami]	materi mudah pahami
utf	penbeajaran yang disampaikan oleh pak agus cu...	[penbeajaran, yang, disampaikan, oleh, pak, agus, cu, ...]	{'penbeajaran': 1, 'yang': 1, 'disampaikan': ...}	[penbeajaran, agus, mudah, dipahami]	[penbeajaran, agus, mudah, dipahami]	[Agi, agus, mudah, pahami]	agi agus mudah pahami
utf	cukup baik	[cukup, baik]	{'cukup': 1, 'baik': 1}	[]	[]	[]	
utf	baik	[baik]	{'baik': 1}	[]	[]	[]	

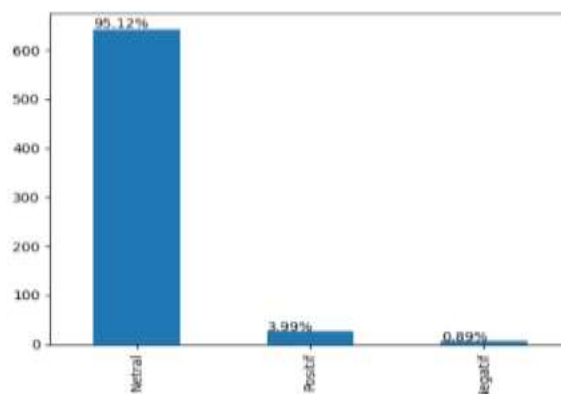
Gambar 10. Hasil proses *preprocessing*

Proses klasifikasi menggunakan *Lexicon based* dengan data yang sudah melalui tahap *preprocessing*. Seperti pada Gambar 10 yaitu menggunakan data pada *field* komentar baru. *Lexicon based* digunakan untuk klasifikasi sentimen dan untuk mengidentifikasi kata-kata dengan sentimen positif, negatif, atau netral dengan menghitung nilai polarity. Jika nilai polarity < 0 maka sentimen negatif, jika nilai polarity = 0 maka sentimen netral dan jika nilai polarity > 0 maka sentimen positif yang di tunjukkan pada Gambar 11.

Komentar_token_stemmed	Komentar_baru	Compound_Score	Sentiments
[oke, tugas, jgn, yg, susah-susah, ya]	oke tugas jgn yg susah-susah ya	0.000	Netral
[bakterba]	bakterba	0.000	Netral
[materi, tulis]	materi tulis	0.000	Netral
[its, ok]	its ok	0.296	Positif
[]		0.000	Netral

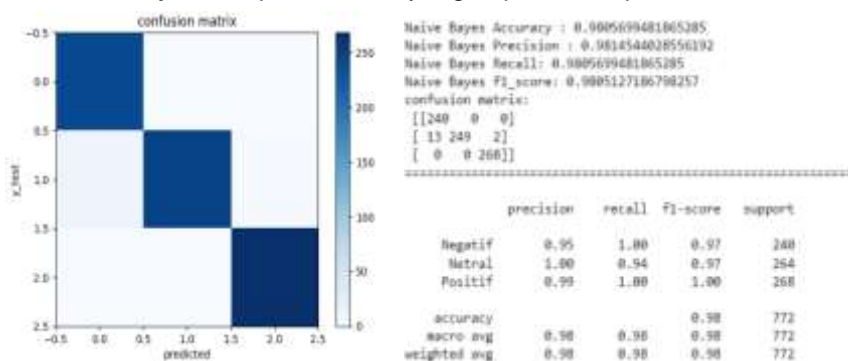
Gambar 11. Proses klasifikasi *lexicon based*

Berdasarkan hasil klasifikasi menggunakan *lexicon based* terdapat 3 kelompok sentimen, yang mana komentar siswa lebih banyak masuk kelompok netral sebanyak 95.12 % dan kelompok negatif paling rendah yaitu 0.89% dengan ditunjukkan oleh grafik pada Gambar 12.



Gambar 12. Grafik Hasil Klasifikasi Sentimen *Lexicon based*

Hasil algoritme *Naïve bayes* didapatkan hasil yang dapat dilihat pada Gambar 16.



Gambar 16. Hasil Algoritme *Naïve Bayes*

Berdasarkan Gambar 16 hasil algoritme *Naïve bayes* dengan nilai akurasi yaitu 98% dalam analisis komentar siswa dan ditunjukkan *confusion matrix*. Hasil algoritme *naïve bayes* dan *decision tree* dengan menggunakan data uji 60 % dan data testing 40% maka dapat di lihat *confusion matrix* yang dihasilkan, dapat diketahui berbagai parameter pengukuran kinerja algoritme, yaitu presisi, *recall*, *f1-measure*, dan akurasi. Perbandingan algoritme *naïve bayes* dan *decision tree* dapat dilihat pada Tabel 2.

Table 2. Hasil Perbandingan Performansi *Naïve Bayes* dan *Decision Tree*

Model	Accuracy	Precision	Recall	F1-score
Decision Tree	99%	0.99	0.99	0.99
Naïve Bayes	98%	0.98	0.98	0.98

Disimpulkan dari Tabel 2 maka hasil perbandingan akurasi dari algoritme *Naïve bayes* dan *Decision tree*. Algoritme *decision tree* memiliki tingkat akurasi yang paling tinggi yaitu 99%. Maka dapat disimpulkan bahwa dengan menggunakan algoritme *Decision tree* lebih baik dalam mengklasifikasi data analisis sentimen komentar siswa.

Berdasarkan hasil dari penelitian yang menjadi perbedaan dengan peneliti sebelumnya yaitu tingkat akurasi yang didapat lebih dari 95% semua yang ditunjukkan pada Table 2. Berdasarkan klasifikasi juga menjadi 3 analisis sentimen yaitu negatif, positif, dan netral dan menggunakan klasifikasi *lexicon based*.

5. Simpulan

Berdasarkan hasil pengujian menggunakan *Naïve Bayes* dan *Decision tree* dengan menggunakan data training 60% dan data testing 40% maka ada beberapa hal yang dihasilkan, sebagai berikut:

1. Hasil ketepatan klasifikasi menggunakan metode *Naïve Bayes* diperoleh *accuracy* sebesar 98%, *precision* sebesar 98%, *recall* sebesar 98%, dan *f1_score* sebesar 98%.
2. Hasil ketepatan klasifikasi menggunakan metode *decision tree* diperoleh *accuracy* sebesar 99%, *precision* sebesar 99%, *recall* sebesar 99%, dan *f1_score* sebesar 99%.
3. Secara keseluruhan perbandingan performansi metode *Naïve Bayes* dan *Decision tree* menunjukkan hasil bahwa *decision tree* lebih baik dalam mengklasifikasi data. Berdasarkan hasil analisis dan kesimpulan, peneliti dapat memberi saran yaitu untuk penelitian selanjutnya, dalam memperoleh data dapat mengambil data pada media sosial populer lainnya seperti *Facebook* dan *Instagram* agar data yang didapat lebih bervariasi dan dapat menggunakan metode klasifikasi yang lain sehingga diperoleh hasil klasifikasi yang lebih spesifik dan baik.

Daftar Referensi

- [1] N. Tri Romadloni, I. Santoso, and S. Budilaksono, "Perbandingan Metode Naive Bayes, Knn Dan Decision Tree Terhadap Analisis Sentimen Transportasi Krl Commuter Line," *J. IKRA-ITH Inform.*, vol. 3, no. 2, pp. 1–9, 2019.

- [2] N. R. Muntiari and K. H. Hanif, "Klasifikasi Penyakit Kanker Payudara Menggunakan Perbandingan Algoritme Machine Learning," *J. Ilmu Komput. dan Teknol.*, vol. 3, no. 1, pp. 1–6, 2022, doi: 10.35960/ikomti.v3i1.766.
- [3] M. K. Anam, B. N. Pikir, and M. B. Firdaus, "Penerapan Naïve Bayes Classifier, K-Nearest Neighbor (KNN) dan Decision Tree untuk Menganalisis Sentimen pada Interaksi Netizen dan Pemerintah," *MATRIK J. Manajemen, Tek. Inform. dan Rekayasa Komput.*, vol. 21, no. 1, pp. 139–150, 2021, doi: 10.30812/matrik.v21i1.1092.
- [4] M. Alfi, R. Reynaldhi, and Y. Sibaroni, "Analisis Sentimen Review Film pada Twitter menggunakan Metode Klasifikasi Hybrid SVM, Naïve Bayes, dan Decision Tree," vol. 8, no. 5, pp. 10127–10137, 2021.
- [5] Y. Wang, G. Huang, J. Li, H. Li, Y. Zhou, and H. Jiang, "Refined Global Word Embeddings Based on Sentiment Concept for Sentiment Analysis," *IEEE Access*, vol. 9, no. 1, pp. 37075–37085, 2021, doi: 10.1109/ACCESS.2021.3062654.
- [6] X. Fu, J. Yang, J. Li, M. Fang, and H. Wang, "Lexicon-Enhanced LSTM with Attention for General Sentiment Analysis," *IEEE Access*, vol. 6, pp. 71884–71891, 2018, doi: 10.1109/ACCESS.2018.2878425.
- [7] A. FATIHIN, "ANALISIS SENTIMEN TERHADAP ULASAN APLIKASI MOBILE MENGGUNAKAN METODE SUPPORT VECTOR MACHINE (SVM) DAN PENDEKATAN LEXICON BASED," 2022.
- [8] J. A. Septian, T. M. Fachrudin, and A. Nugroho, "Analisis Sentimen Pengguna Twitter Terhadap Polemik Persepakbolaan Indonesia Menggunakan Pembobotan TF-IDF dan K-Nearest Neighbor," *J. Intell. Syst. Comput.*, vol. 1, no. 1, pp. 43–49, 2019, doi: 10.52985/insyst.v1i1.36.
- [9] F. S. Pattihha and Hendry, "Perbandingan Metode K-NN, Naïve Bayes, Decision Tree untuk Analisis Sentimen Tweet Twitter Terkait Opini Terhadap PT PAL Indonesia," *J. Ris. Komputer*, vol. 9, no. 2, pp. 2407–389, 2022, doi: 10.30865/jurikom.v9i2.4016.
- [10] T. Wiratama Putra, A. Triayudi, and A. Andrianingsih, "Analisis Sentimen Pembelajaran Daring Menggunakan Metode Naïve Bayes, KNN, dan Decision Tree," *J. JTik (Jurnal Teknol. Inf. dan Komunikasi)*, vol. 6, no. 1, pp. 20–26, 2022, doi: 10.35870/jtik.v6i1.368.
- [11] H. Han, Y. Zhang, J. Zhang, J. Yang, and X. Zou, "Improving the performance of lexicon-based review sentiment analysis method by reducing additional introduced sentiment bias," *PLoS One*, vol. 13, no. 8, pp. 1–11, 2018, doi: 10.1371/journal.pone.0202523.
- [12] A. S. Rusydiana, I. Firmansyah, and L. Marlina, "Sentiment Analysis of Microtakaful Industry: Comparison between Indonesia and Malaysia," *Int. J. Nusant. Islam*, vol. 6, no. 1, pp. 20–34, 2019, doi: 10.15575/ijni.v6i1.3004.
- [13] N. N. Wilim and R. S. Oetama, "Sentiment Analysis About Indonesian Lawyers Club Television Program Using K-Nearest Neighbor, Naïve Bayes Classifier, And Decision Tree," *IJNMT (International J. New Media Technol.)*, vol. 8, no. 1, pp. 50–56, 2021, doi: 10.31937/ijnmt.v8i1.1965.
- [14] I. Hilmy Khairi Idris, Mochammad Ali Fauzi, "Klasifikasi Teks Bahasa Indonesia Pada Dokumen Pengaduan Sambat Online Menggunakan Metode K-Nearest Neighbors Dan Chi-square," *Syst. Inf. Syst. Informatics J.*, vol. 3, no. 1, pp. 25–32, 2017, doi: 10.29080/systemic.v3i1.191.
- [15] S. Panggabean, W. Gata, and T. A. Setiawan, "Analysis of Twitter Sentiment Towards Madrasahs Using Classification Methods," *J. Appl. Eng. Technol. Sci.*, vol. 4, no. 1, pp. 375–389, 2022.
- [16] M. Rezapour, "Sentiment classification of skewed shoppers' reviews using machine learning techniques, examining the textual features," *Eng. Reports*, vol. 3, no. 1, pp. 1–13, 2021, doi: 10.1002/eng2.12280.
- [17] W. Li, P. Liu, Q. Zhang, and W. Liu, "An improved approach for text sentiment classification based on a deep neural network via a sentiment attention mechanism," *Futur. Internet*, vol. 11, no. 4, 2019, doi: 10.3390/FI11040096.
- [18] N. Z. Dina, "Tourist sentiment analysis on TripAdvisor using text mining: A case study using hotels in Ubud, Bali," *African J. Hosp. Tour. Leis.*, vol. 9, no. 2, pp. 1–10, 2020.