ANALISIS SENTIMEN ULASAN APLIKASI THREADS DI GOOGLE PLAYSTORE MENGGUNAKAN ALGORITMA NAÏVE BAYES

Nurzaman, Nana Suarna, Willy Prihartono
Teknik Informatika, STMIK IKMI Cirebon
Jalan Perjuangan No.10B, Karyamulya, Kec. Kesambi, Kota Cirebon, Jawa Barat 45135
nurzaman0240@gmail.com

ABSTRAK

Aplikasi Threads merupakan aplikasi perpesanan yang dikembangkan oleh Meta *Platform*s, Inc. (Meta). Aplikasi ini memungkinkan pengguna untuk berkomunikasi dengan teman dan keluarga melalui pesan teks, gambar, dan video. Permasalahan ini masih ada beberapa pengguna yang memberikan ulasan negatif tentang aplikasi ini, dan Ulasan-ulasan negatif ini dapat menurunkan reputasi aplikasi dan membuat pengguna lain enggan untuk menggunakan aplikasi tersebut. Tujuan utama yakni untuk mengklasifikasikan pendapat positif, netral, dan negatif berdasarkan nilai akurasi, presisi, dan recall yang dihasilkan dari analisis sentimen menggunakan algoritma naive bayes. Oleh karena itu, diperlukan sistem yang dapat mengkategorikan komentar dan menganalisis sentimen pengguna. Metode Penelitian ini menggunakan *Knowledge Discovery in Database* (KDD), web *scraping* Python untuk mengumpulkan data ulasan pengguna aplikasi Threads di Google PlayStore. Data yang dikumpulkan mencakup dari periode Agustus hingga November 2023 dengan jumlah 1500 data teks. Metode Naïve Bayes Classifier, metode klasifikasi ini dipilih karena memiliki kecepatan pemrosesan menghasilkan akurasi yang tinggi dan dapat memproses data jumlah yang besar. Hasil penelitian bahwa Pengujian analisis sentimen dilakukan dengan mengukur nilai recall, precision, dan accuracy. Menunjukkan bahwa nilai accuracy mencapai 73% yang menunjukkan bahwa model analisis sentimen mampu memprediksi sentimen dengan akurasi yang tinggi. Nilai precision juga mencapai 70%, Recall 99%.

Kata kunci : *Analisis Sentimen, Aplikasi Threads, Google Playstore, Algoritma Naïve Bayes Classifier*

1. PENDAHULUAN

Di era digital saat ini, penggunaan aplikasi mobile telah menjadi bagian yang tak terpisahkan dari kehidupan sehari-hari. Aplikasi mobile menawarkan beragam fitur dan fungsi yang memungkinkan pengguna berinteraksi dengan teknologi secara praktis. Seiring dengan kemajuan teknologi, pengguna aplikasi mobile juga semakin cerdas dalam memilih aplikasi yang ingin mereka gunakan. Google Play Store, sebagai salah satu platform utama distribusi aplikasi Android, menyediakan berbagai macam aplikasi, termasuk aplikasi media sosial. Salah satu aplikasi media sosial yang cukup terkenal adalah Threads, yang dikembangkan oleh Meta. Permasalahan utama yang ingin dipecahkan oleh penelitian ini adalah Bagaimana cara mengklasifikasikan pendapat positif, netral, dan negatif berdasarkan nilai akurasi, presisi, dan recall yang dihasilkan dari analisis sentimen menggunakan algoritma naive bayes. Ulasan tersebut dapat memberikan gambaran tentang kelebihan dan kekurangan suatu aplikasi. Namun, ulasan tersebut belum dapat diklasifikasikan secara otomatis berdasarkan sentimennya, yaitu positif, negatif, serta dapat memberikan pandangan yang berharga tentang preferensi dan harapan pengguna terhadap aplikasi sosial media. Penelitian ini akan memberikan wawasan yang berharga bagi pemangku kepentingan, seperti pemasar dan manajemen produk, yang ingin memahami cara mengintegrasikan Threads dengan kebutuhan pengguna yang ada. Faktor-faktor yang mungkin mencakup kinerja aplikasi, fitur-fitur yang ditawarkan, atau pengalaman pengguna dalam berinteraksi dengan teman-teman mereka melalui Threads akan diidentifikasi untuk memberikan wawasan yang lebih dalam tentang mengapa pengguna merasa seperti yang mereka ungkapkan dalam ulasan. Penelitian ini juga akan menambah literatur dalam analisis sentimen dan penggunaan Naïve Bayes Classifier dalam konteks ulasan aplikasi mobile. Penelitian ini dapat menjadi referensi penting bagi penelitian selanjutnya dalam bidang ini dan menyumbangkan pemahaman lebih lanjut tentang sentimen pengguna terhadap aplikasi mobile. Dengan demikian, penelitian ini akan mengisi celah pengetahuan yang ada dalam pemahaman sentimen ulasan aplikasi sosial media melalui penggunaan metode klasifikasi sentimen.

Metode analisis data *mining*, menggunakan tahapan *Knowledge Discovery in Database* (KDD) yang terdiri dari tahapan data selection, data *Cleaning*, data *transformation*, data *mining*, *pattern evaluation*, dan *knowledge presentation* Data yang digunakan dalam penelitian ini adalah ulasan aplikasi Threads di Google Play Store . Ulasan-ulasan tersebut akan dikumpulkan menggunakan web *scraping* Python dari Google Play Store, dan data dibersihkan terlebih dahulu untuk menghilangkan noise dan kesalahan [1]. Data ulasan dibagi menjadi dua set, yaitu set pelatihan dan set pengujian. Set pelatihan akan digunakan untuk melatih model Naïve Bayes Classifier, sedangkan set pengujian akan digunakan untuk berkontribusi pada keputusan akhir secara independen. Metode ini terbukti lebih efisien secara komputasi dibandingkan

dengan pengklasifikasi teks lainnya. Salah satu keunggulan metode ini adalah kemampuannya dalam memprediksi probabilitas dengan tingkat akurasi dan kecepatan yang tinggi bahkan ketika diterapkan pada database yang sangat besar.

2. TINJAUAN PUSTAKA

Pada bagian ini, akan dipaparkan literatur ilmiah yang relevan dengan penelitian yang dilakukan saat ini. Literatur ilmiah tersebut akan digunakan sebagai landasan teori untuk penelitian ini.

2.1. Algoritma Naïve Bayes

Metode klasifikasi Naive Bayes adalah metode klasifikasi yang menggunakan teorema Bayes. Teorema Bayes adalah metode klasifikasi yang dinamai sesuai dengan penemunya, Thomas Bayes, seorang ilmuwan Inggris yang mempelajari klasifikasi. Setelah kematiannya, penelitiannya tentang metode Naive Bayes dilanjutkan oleh teman-temannya. Metode Naive Bayes mengasumsikan bahwa setiap variabel independen satu sama lain, sehingga hasil dari satu variabel tidak mempengaruhi hasil variabel lainnya.[2].

2.2. Analisis Sentimen

Analisis sentimen atau opinion mining adalah studi komputasi mengenai pendapat, perilaku dan emosi seseorang terhadap entitas. Entitas tersebut dapat berupa individu, kejadian atau topik. Analisis sentimen merupakan proses memahami, mengekstrak dan mengolah data tekstual secara otomatis untuk mendapatkan informasi sentimen yang terkandung dalam suatu kalimat. Karena pengaruh dan manfaat dari analisis sentimen yang besar, penelitian dan aplikasi berbasis analisis sentimen berkembang pesatt[3].

2.3. *Term Frequency Inverse Document Frequency* (TF-IDF)

Metode *Term Frequency-Inverse Document Frequency* (TF-IDF) digunakan untuk menghitung seberapa sering kata muncul dalam sebuah dokumen atau artikel, serta seberapa penting kata tersebut dalam dokumen. Metode ini dikenal efisien, mudah, dan akurat. Nilai TF-IDF adalah ukuran statistik yang digunakan untuk menentukan tingkat kepentingan kata dalam sebuah dokumen atau kelompok kata. Frekuensi kemunculan kata dalam sebuah kalimat menunjukkan seberapa penting kata tersebut dalam kalimat tersebut. Frekuensi dokumen yang mengandung kata tersebut menunjukkan seberapa umum kata tersebut. Bobot sebuah kata akan meningkat jika kata tersebut sering muncul dalam sebuah dokumen dan akan menurun jika kata tersebut muncul dalam banyak dokumen. [4].

2.4. Labeling

Untuk melabeli jenis sentimen dari setiap ulasan pengguna, dilakukan pelabelan data. Ulasan yang didapatkan berupa teks dalam bahasa Indonesia dan rating dalam bentuk bintang 1 sampai 5. Data yang diperoleh dari Google Play Store tidak memiliki label kelas sentimen. Oleh karena itu, rating dapat digunakan untuk melakukan pelabelan. Metode berbasis leksikon digunakan untuk mengekstrak kalimat opini secara otomatis dengan menggunakan kamus kata opini yang akan digunakan sebagai acuan dalam klasifikasi. Kamus tersebut menjadi dasar untuk mengidentifikasi kalimat opini dengan nilai presisi yang tinggi. Pendekatan berbasis leksikal ini tidak memerlukan pembelajaran model dan pelatihan dataset karena merupakan pendekatan berbasis kamus. Kamus leksikon yang digunakan dalam penelitian ini adalah Inset Lexicon. Inset Lexicon terdiri dari kamus positif dan kamus negatif. Berdasarkan kamus tersebut, setiap kata dalam ulasan diberi skor berdasarkan skor yang ada di kamus. Hasil dari jumlah *score* kata positif dan kata negatif di sebuah ulasan digunakan untuk menilai sentimen ulasan tersebut [5] .

2.5. *Confusion matrix*

Confusion matrix adalah matriks yang digunakan untuk mengukur akurasi klasifikasi suatu algoritma dalam kelas tertentu. Matriks ini penting untuk mengevaluasi kualitas model statistik atau algoritma data mining. Ada beberapa jenis *Confusion matrix* yang digunakan untuk menguji model data mining, seperti akurasi dan lain-lain. Tabel 1 di bawah ini menunjukkan bentuk *Confusion matrix* [6].

Tabel 1. *Confusion matrix*

Actual Class			
Percobaan 1			
Prediksi Percobaan	Percobaan 1	*True Positive* (TP)	*False Negative* (FN)
		False Positive (FP)	*True Negative* (TN)

Deskripsi:

1) *True Positive* (TP) adalah jumlah data yang sebenarnya diklasifikasikan sebagai kelas positif.
2) *True Negative* (TN) adalah jumlah data yang benar-benar diklasifikasikan sebagai kelas negatif.
3) *False Positive* (FP) adalah jumlah data yang seharusnya tidak Termasuk dalam kelas positif, tetapi salah diklasifikasikan sebagai kelas positif.
4) *False negative* (FN) adalah jumlah data yang harus diklasifikasikan sebagai kelas positif, tetapi salah diklasifikasikan sebagai kelas negatif.

Pengujian menggunakan *Confusion matrix* menghasilkan perhitungan yang menghasilkan tiga keluaran, antara lain:

a. Akurasi adalah ukuran ketepatan model dalam mengklasifikasikan data. Rumus untuk menghitung nilai akurasi secara keseluruhan dapat dilihat pada persamaan (1) di bawah ini.

$$Accuracy = \frac{TP + TN}{(\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN})} \qquad (1)$$

b. Presisi digunakan untuk mengukur ketepatan prediksi positif, yaitu jumlah prediksi positif yang benar terhadap jumlah total prediksi positif. Rumus berikut ini dapat digunakan untuk menghitung nilai presisi. Rumus tersebut dapat dilihat pada persamaan 2 di bawah ini.

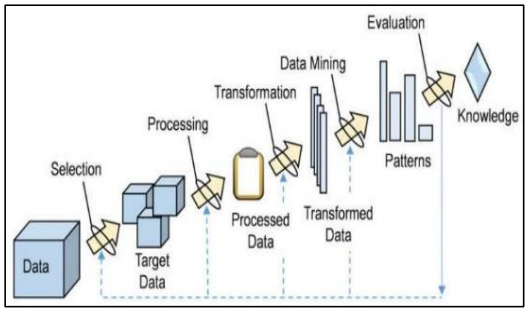
$$Precision = \frac{TP}{\mathrm{TP} + \mathrm{FP}} \qquad (2)$$

c. Recall digunakan untuk mengukur seberapa banyak data positif yang benar-benar terdeteksi sebagai positif, yaitu banyaknya prediksi positif yang benar terhadap total data aktual yang positif. Rumus persamaan pada 3 dapat digunakan untuk menghitung nilai recall.

$$Recall = \frac{TP}{\mathrm{TP} + \mathrm{FN}} \qquad (3)$$

3. METODE PENELITIAN

Penelitian sentimen ulasan aplikasi Threads pada Google Play Store ini menggunakan metode Knowledge Discovery in Data (KDD) yang terdiri dari enam tahapan, yaitu data selection, data cleaning, data transformation, data mining, pattern evaluation, dan knowledge presentation. [7]. Berikut adalah gambaran mengenai perincian metode penelitian yang akan dilaksanakan, sebagaimana ditunjukkan pada Gambar 1.



Gambar 1. Proses *Knowledge Discovery in Database* Sumber Refrensi [8]

a. Data
Penelitian ini menggunakan sumber data sekunder, yaitu data publik yang diambil langsung dari internet. Data tersebut dikumpulkan dengan teknik "scraping" menggunakan bahasa pemrograman Python di Google Colab. Langkah pertama adalah mengunjungi situs resmi Google Play Store dengan kata kunci "Threads". Kemudian, data dikumpulkan dengan fokus pada halaman informasi aplikasi "Threads" yang memiliki paket nama "com.instagram.barcelona." Jumlah data yang dikumpulkan adalah 1500 data sampel ulasan pengguna aplikasi "Threads" di Google Play Store yang diambil sejak bulan Agustus 2023 hingga November 2023.

b. Selection
Pemilihan atribut mendahului praktik pengumpulan data yang sebenarnya melalui scraping data di Google Colab dimulai dengan mencari ulasan pengguna pada halaman informasi aplikasi Threads dengan paket nama "com.instagram.barcelona." Setelah mendapatkan data tersebut dapat diketahui terdapat 11 atribut dan ditahapan ini kita akan menyeleksi atribut yang relevan dengan penelitan ini. Dan didapatkan ada 3 atribut yang relevan dengan penelitian ini yaitu username, content dan score.

c. Preprocessing
Preprocessing adalah proses mengubah data yang tidak terstruktur menjadi data terstruktur yang sesuai untuk Text Mining.
Tahapan Preprocessing yang digunakan meliputi: *Case Folding*, *Tokenizing*, *Filtering*, dan *Stemming*.
1) *Case Folding* mengubah semua huruf dalam dokumen menjadi huruf kecil dan menghilangkan karakter non-huruf.
2) *Tokenizing* adalah proses memecah suatu teks atau dokumen menjadi unit – unit diskrit yang disebut token. Token – token ini bisa berupa kata – kata atau simbol individu yang memiliki makna atau signifikan dalam konteks yang diberikan.
3) *Stopword Removal* adalah proses yang akan menghapus kata umum yang biasanya muncul dalam jumlah besar dan dianggap tidak memiliki makna. Pada tahap ini menggunakan library Sastrawi untuk melakukan Stopword Removal.
4) *Stemming* proses Stemming akan memfilter kata yang memiliki imbuhan akan dikembalikan ke dalam bentuk baku. Seperti kata “memukul” akan menjadi “pukul”.

d. Transformation
Pada tahap tranformation Pada tahap labeling, peneliti melakukan perubahan data yang bersifat tidak terbimbing (*unsupervised learning*) menjadi terbimbing (*supervised learning*). Kelas sentimen ini akan digunakan untuk mengidentifikasi kata positif dan negatif pada suatu kalimat. Pada tahap ini, dilakukan pembagian ke dalam tiga kelas sentimen, yaitu negatif, netral, dan positif. Untuk mengkonversi nilai ke dalam bentuk numerik, digunakan metode TF-IDF. Metode TF-IDF merupakan metode untuk menentukan frekuensi relatif dari setiap kata atau token. Setiap kata akan diberi pembobotan berupa nilai berdasarkan pentingnya suatu kata dalam dokumen. Nilai bobot tersebut ditentukan berdasarkan jumlah kemunculan kata dalam suatu dokumen dan mengukur kata-kata

tersebut terhadap keseluruhan dokumen yang ada.

e. Data Mining

Data Mining adalah suatu metode yang digunakan untuk mengidentifikasi pola-pola dan menggali informasi berharga lainnya dari kumpulan data yang berskala besar. Tujuan utama dari teknik ini adalah untuk menemukan informasi yang memiliki nilai signifikan dalam suatu dataset. Dari jumlah keseluruhan tersebut, 536 ulasan menunjukkan adanya sentimen positif, sementara 301 ulasan mengekspresikan sentimen negatif.

f. Pattern Evaluation

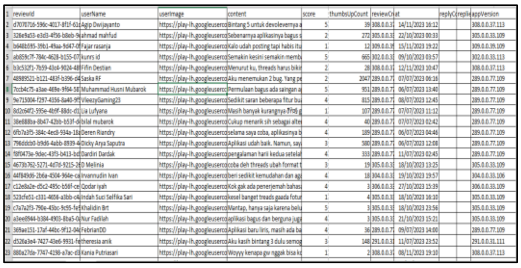
Dalam konteks analisis sentimen aplikasi Threads pada ulasan di Google Playstore menggunakan algoritma Naïve Bayes Classifier, tahapan evaluasi pola (*Pattern Evaluation*) merupakan langkah kunci dalam proses penelitian ini. Pada tahap ini, dilakukan analisis mendalam terhadap pola-pola sentimen yang muncul dari ulasan pengguna, dengan tujuan untuk mengidentifikasi dan mengevaluasi pola-pola sentimen yang signifikan dan informatif yang terkandung dalam dataset ulasan tersebut. Evaluasi pola ini dapat memberikan wawasan yang berharga terkait dengan bagaimana pengguna secara kolektif merespons aplikasi Threads, sehingga dapat membantu dalam pemahaman yang lebih mendalam terhadap persepsi dan umpan balik terhadap aplikasi tersebut di platform Google Playstore.

4. HASIL DAN PEMBAHASAN

Dalam penelitian ini, tahap pembahasan hasil uji menggunakan matriks konfusi yang memanfaatkan library sklearn metrics. Tujuannya adalah untuk mengetahui tingkat akurasi proses klasifikasi yang telah dilakukan sistem. Tahap ini dilakukan sebanyak satu kali percobaan, dengan pembagian data latih dan uji. Dalam implementasinya, perhitungan menggunakan Python untuk menghitung akurasi, presisi, recall.

4.1. Data

Penelitian ini menggunakan sumber data sekunder yang diambil langsung dari internet. Data tersebut dikumpulkan dengan teknik "scraping" menggunakan bahasa pemrograman Python di Google Colab. Langkah pertama adalah mengunjungi situs resmi Google Play Store dengan kata kunci "Threads". Kemudian, data dikumpulkan dengan fokus pada halaman informasi aplikasi "Threads" yang memiliki paket nama "com.instagram.barcelona". Jumlah data yang dikumpulkan adalah 1500 sampel ulasan pengguna aplikasi "Threads" di Google Play Store yang diambil sejak bulan Agustus 2023 hingga November 2023. Di atas ini memperlihatkan hasil data pada tahapan scraping melalui Google Colab dan didapatkan 1500 data pada proses tahapan scraping ini, data pada proses tahapan scraping terdapat pada Gambar 2.



Gambar 2 Dataset *Scraping*

4.2. Data Selection

Data yang digunakan dalam penelitian ini adalah data ulasan aplikasi Threads. Data tersebut terdiri dari 1500 record dan merupakan data publik. Dari 11 atribut yang tersedia, hanya 3 atribut yang digunakan. Dari hasil *scrapping*, didapatkan banyak kolom. Oleh karena itu, peneliti perlu melakukan proses filter untuk mendapatkan kolom *username*, *score*, dan *content*. Hal ini dikarenakan 8 atribut lainnya tidak menjadi fokus penelitian terkait ulasan aplikasi Threads, Terdapat pada Gambar 3.

	userName	score	content
0	Agip Dwijayanto	5	Bintang 5 untuk devolevernya agar semangat nge...
1	ahmad mahfud	2	Sebenarnya aplikasinya bagus simpel gak nego2....
2	Fajar rasanja	1	Kalo udah posting tapi habis itu tiba2 formatn...
3	kunrs id	5	Semakin kesini semakin membaik kualitas Thread...
4	Fifin Destian	2	Menurut ku, threads harus bikin fitur dm soaln...

Gambar 3. Data Selection

4.3. Data Preprocessing

Pada tahap *preprocessing*, data ulasan yang telah terkumpul dan memiliki label diproses dengan enam langkah agar dapat digunakan pada tahapan selanjutnya. Langkah-langkah tersebut adalah *cleaning*, *case folding*, *tokenizing*, *stopword removal, dan stemming*. Proses *preprocessing* ini dilakukan dengan bantuan tools Google Colaboratory menggunakan bahasa pemrograman Python.

a. *Cleaning*

Pada tahap ini, dilakukan pembersihan data dengan menghapus atribut yang tidak berpengaruh terhadap klasifikasi, seperti karakter simbol, angka, dan emotikon. Data mentah yang dihasilkan dari scraping masih mengandung simbol, URL, tanggal, pengguna, dan sebagainya. Unsur-unsur tersebut tidak memiliki arti dalam analisis sentimen pada tahap berikutnya. Oleh karena itu, diperlukan proses pembersihan data. Proses ini bertujuan untuk membersihkan data dari unsur-unsur yang tidak memiliki arti agar data dapat digunakan dalam analisis sentimen.[9], pada Tabel 2 menyajikan bentuk sampel data *Cleaning*.

Tabel 2. Data *Cleaning*

Sebelum	Sesudah
Sebenarnya aplikasinya bagus simpel gak nego2. Dan cenderung sedikit bot karena log in lewat Instagram. Tapi tolong ditambahkan gitu DM karena jadi gak perlu buka ig. Dan ada bug ketika dialihkan ke ig gak bisa kembali harus ditutup atau gak kembali ke home screen	sebenarnya aplikasinya bagus simpel gak nego dan cenderung sedikit bot karena log in lewat instagram tapi tolong ditambahkan gitu dm karena jadi gak perlu buka ig dan ada bug ketika dialihkan ke ig gak bisa kembali harus ditutup atau gak kembali ke home screen

b. *Case Folding*
Mengubah semua huruf dalam dokumen menjadi huruf kecil dan menghilangkan karakter non-huruf[10], Bentuk text *Case Folding* dapat dilihat pada Tabel 3.

Tabel 3. *Case Folding*

Sebelum	Sesudah
Sebenarnya aplikasinya bagus simpel gak nego2. Dan cenderung sedikit bot karena log in lewat Instagram. Tapi tolong ditambahkan gitu DM karena jadi gak perlu buka ig. Dan ada bug ketika dialihkan ke ig gak bisa kembali harus ditutup atau gak kembali ke home screen.	sebenarnya aplikasinya bagus simpel gak nego dan cenderung sedikit bot karena log in lewat instagram tapi tolong ditambahkan gitu dm karena jadi gak perlu buka ig dan ada bug ketika dialihkan ke ig gak bisa kembali harus ditutup atau gak kembali ke home screen.

c. *Tokenizing*
Proses memecah suatu teks atau dokumen menjadi unit – unit diskrit yang disebut token. Token – token ini bisa berupa kata – kata atau simbol individu yang memiliki makna atau signifikan dalam konteks yang diberikan [11], Bentuk *Tokenizing* dapat dilihat pada Tabel 4.

Tabel 4. *Tokenizing*

Sebelum	Sesudah
Sebenarnya aplikasinya bagus simpel gak nego2. Dan cenderung sedikit bot karena log in lewat Instagram. Tapi tolong ditambahkan gitu DM karena jadi gak perlu buka ig. Dan ada bug ketika dialihkan ke ig gak bisa kembali harus ditutup atau gak kembali ke home screen	['sebenarnya', 'aplikasinya', 'bagus', 'simpel', 'gak', 'nego', 'dan', 'cenderung', 'sedikit', 'bot', 'karena', 'log', 'in', 'lewat', 'instagram', 'tapi', 'tolong', 'ditambahkan', 'gitu', 'dm', 'karena', 'jadi', 'gak', 'perlu', 'buka', 'ig', 'dan', 'ada', 'bug', 'ketika', 'dialihkan', 'ke', 'ig', 'gak', 'bisa', 'kembali', 'harus', 'ditutup', 'atau', 'gak', 'kembali', 'ke', 'home', 'screen']

d. *Stopword Removal*
Tahap ini dilakukan proses yang akan menghapus kata umum yang biasanya muncul dalam jumlah besar dan dianggap tidak memiliki makna. Pada tahap ini menggunakan *library* Sastrawi untuk melakukan *Stopword Removal*, Bentuk *Stopword Removal* dapat dilihat pada Tabel 5.

Tabel 5. *Stopword Removal*

Sebelum	Sesudah
Sebenarnya aplikasinya bagus 971ias971e gak nego2. Dan cenderung sedikit bot karena log in lewat Instagram. Tapi tolong ditambahkan gitu DM karena jadi gak perlu buka ig. Dan ada bug ketika dialihkan ke ig gak 971ias kembali harus ditutup atau gak kembali ke home screen.	[‘aplikasinya’, ‘bagus’, ‘simpel’, ‘gak’, ‘nego’, ‘cenderung’, ‘bot’, ‘log’, ‘in’, ‘instagram’, ‘tolong’, ‘gitu’, ‘dm’, ‘gak’, ‘buka’, ‘ig’, ‘bug’, ‘dialihkan’, ‘ig’, ‘gak’, ‘ditutup’, ‘gak’, ‘home’, ‘screen’]

e. *Stemming*
Proses *Stemming* akan memfilter kata yang memiliki imbuhan akan dikembalikan ke dalam bentuk baku. Seperti kata “memukul” akan menjadi “pukul”, Bentuk *Stemming* dapat dilihat pada Tabel 6.

Tabel 6. *Stemming*

Sebelum	Sesudah
Sebenarnya aplikasinya bagus simpel gak nego2. Dan cenderung sedikit bot karena log in lewat Instagram. Tapi tolong ditambahkan gitu DM karena jadi gak perlu buka ig. Dan ada bug ketika dialihkan ke ig gak bisa kembali harus ditutup atau gak kembali ke home screen	aplikasi bagus simpel gak nego cenderung bot log in instagram tolong gitu dm gak buka ig bug alih ig gak tutup gak home screen

4.4. Transformation

Tahap transformasi dilakukan untuk mengubah data menjadi bentuk yang dapat diolah pada tahapan data mining. Pada tahap ini, data akan dibagi menjadi satu skenario terlebih dahulu, yaitu skenario 1 (80% data training dan 20% data testing). Setelah itu, data akan dibobotkan dengan *Term Frequency - Inverse Document Frequency* (TF-IDF), yang berguna untuk mengubah data berupa teks menjadi vektor bobot numerik. Metode TF-IDF digunakan untuk mendapatkan nilai bobot setiap kata pada data yang digunakan. Nilai pembobotan terhadap kemunculan kata dalam tiap ulasan tersebut kemudian digunakan untuk membantu proses klasifikasi pada tahap data mining. Informasi lebih lanjut dapat dilihat pada Tabel 7.

Tabel 7. *Transformation TF-IDF*

Term	**TF**	**TF-IDF**
Bintang	0.03225806451612903	0.10383470402800647
Devolevernya	0.03225806451612903	0.21355074859775341
semangat	0.03225806451612903	0.18399298305342582
ngembangin	0.03225806451612903	0.21355074859775341
apk	0.12903225806451613	0.3371277446835981

Setelah data diproses dengan metode pembobotan kata, langkah selanjutnya adalah membagi data tersebut menjadi data latih dan data uji. Data latih digunakan untuk melatih model, sedangkan data uji digunakan untuk menguji kinerja model. Rasio pembagian data yang penulis gunakan adalah 80:20. Jumlah data setelah dilakukan pembagian dapat dilihat pada Tabel 8.

Tabel 8. Data *training* dan *testing*

Data Training	**Data Testing**
80% (1200 Baris Data)	20% (300 Baris Data)

Setelah data dibagi menjadi data latih dan data uji, langkah selanjutnya adalah membuat model klasifikasi Naïve Bayes. Untuk mengetahui rasio pembagian data yang menghasilkan akurasi, dilakukan pengujian dengan menggunakan rasio pembagian data yang telah ditentukan.

4.5. Data Mining

```
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import accuracy_score, classification_report

# Ekstraksi fitur menggunakan TF-IDF
vectorizer = TfidfVectorizer()
X_train_tfidf = vectorizer.fit_transform(X_train)
X_test_tfidf = vectorizer.transform(X_test)

# Inisialisasi dan latih model Naive Bayes
nb_classifier = MultinomialNB()
nb_classifier.fit(X_train_tfidf, y_train)

# Prediksi label sentimen pada data uji
y_pred = nb_classifier.predict(X_test_tfidf)

# Evaluasi model
accuracy = accuracy_score(y_test, y_pred)
classification_rep = classification_report(y_test, y_pred)

print("MultinomialNB Accuracy:", accuracy_score(y_test,y_pred)*100,'%')
print("MultinomialNB Precision:", precision_score(y_test,y_pred, average="binary", pos_label="Positif")*100,'%')
print("MultinomialNB Recall:", recall_score(y_test,y_pred, average="binary", pos_label="Positif")*100,'%')
print("MultinomialNB f1_score:", f1_score(y_test,y_pred, average="binary", pos_label="Positif")*100,'%')

print(f'confusion_matrix:\n {confusion_matrix(y_test, y_pred)}')
print('==========================================================\n')
print(classification_report(y_test, y_pred, zero_division=0))
```

Gambar 4. Model *klasifikasi Naïve Bayes*

Proses data mining adalah pengolahan data dengan menerapkan model klasifikasi Naïve Bayes. Model klasifikasi Naïve Bayes yang digunakan adalah model Multinominal Naïve Bayes dari *scikit-learn*. Untuk mengetahui model klasifikasi Naïve Bayes yang paling optimal, dilakukan pengujian dengan menggunakan rasio split data yang berbeda, yaitu 80:20. Algoritma Naïve Bayes membagi data menjadi data latih dan data uji. Konsep algoritma Naïve Bayes dalam menentukan kelompok kelas dokumen adalah melalui proses probabilitas. Kelebihan algoritma ini adalah dapat menghasilkan akurasi yang tinggi dan dapat memproses data dalam jumlah yang besar. Dapat lihat Gambar 4 untuk lebih jelasnya.

4.6. Evaluation

Dalam penelitian ini, evaluasi model klasifikasi Naïve Bayes dilakukan dengan menggunakan rasio split data 80:20. Pemisahan data training dan data testing yang paling optimal adalah dengan distribusi 80% data latih dan 20% data uji, sehingga menghasilkan akurasi sebesar 73%. Data dibagi menjadi dua bagian, yaitu data latih dan data uji. Data latih digunakan untuk melatih model, sedangkan data uji digunakan untuk menguji kinerja model. Evaluasi menggunakan matriks konfusi untuk mengetahui performa hasil uji dari algoritma Naive Bayes. Hasil evaluasi berupa nilai akurasi, presisi, dan recall. Informasi lebih lanjut dapat dilihat pada Gambar 5.

```
# Ekstraksi fitur menggunakan TF-IDF
vectorizer = TfidfVectorizer()
X_train_tfidf = vectorizer.fit_transform(X_train)
X_test_tfidf = vectorizer.transform(X_test)

# Inisialisasi dan latih model Naive Bayes
nb_classifier = MultinomialNB()
nb_classifier.fit(X_train_tfidf, y_train)

# Prediksi label sentimen pada data uji
y_pred = nb_classifier.predict(X_test_tfidf)

# Evaluasi model
accuracy = accuracy_score(y_test, y_pred)
classification_rep = classification_report(y_test, y_pred)

print("MultinomialNB Accuracy:", accuracy_score(y_test,y_pred)*100,'%')
print("MultinomialNB Precision:", precision_score(y_test,y_pred, average="binary", pos_label="Positif")*100,'%')
print("MultinomialNB Recall:", recall_score(y_test,y_pred, average="binary", pos_label="Positif")*100,'%')
print("MultinomialNB f1_score:", f1_score(y_test,y_pred, average="binary", pos_label="Positif")*100,'%')

print(f'confusion_matrix:\n {confusion_matrix(y_test, y_pred)}')
print('==========================================================\n')
print(classification_report(y_test, y_pred, zero_division=0))
```

Gambar 5. Hasil *Evaluasi Akurasi Naïve Bayes*

Berdasarkan hasil evaluasi pemodelan naive bayes pada Tabel 9, dapat dilihat bahwa hasil klasifikasi terbaik pada aplikasi Threads diperoleh dengan rasio 80:20, yaitu dengan akurasi sebesar 73%, presisi sebesar 70%, recall sebesar 99%.

Tabel 9. *Evaluasi* pemodelan *naive bayes*

Keterangan	**Hasil Analisis**
Precision	70%
Recall	99%
Accuracy	73%

Tahap ini dilakukan sebanyak satu kali percobaan, dengan pembagian data latih dan uji. Dalam implementasinya, perhitungan menggunakan Python untuk menghitung akurasi, presisi, recall, Tedapat Pada Gambar 6.

```
MultinomialNB Accuracy: 72.61904761904762 %
MultinomialNB Precision: 70.0 %
MultinomialNB Recall: 99.05660377358491 %
MultinomialNB f1_score: 82.03125 %
confusion_matrix:
 [[ 17  45]
 [  1 105]]
====================================================

              precision    recall  f1-score   support

     Negatif       0.94      0.27      0.42        62
     Positif       0.70      0.99      0.82       106

    accuracy                           0.73       168
   macro avg       0.82      0.63      0.62       168
weighted avg       0.79      0.73      0.67       168
```

Gambar 6. Hasil pengujian *Confusion matrix*

Pada tahap hasil uji menggunakan *Confusion matrix* , dilakukan uji 80% data latih dan 20% data uji. Hasil uji tersebut menunjukkan bahwa kemampuan prediksi kelas positif yang tepat adalah 3 ulasan, dan kemampuan prediksi kelas negatif yang tepat adalah 17 ulasan. Sementara itu, tingkat kesalahan yang dapat terjadi dalam melakukan prediksi kelas positif (false negative) adalah 1 ulasan, dan tingkat kesalahan yang dapat terjadi dalam melakukan prediksi kelas negatif (false positive) adalah 45 ulasan. Berdasarkan hasil tersebut, dapat disimpulkan bahwa model memiliki akurasi sebesar 73%, presisi 70%, recall 99%. Hasil analisis evaluasi akurasi, precision, recall, dan accuracy dengan metode Naïve Bayes dapat dilihat pada Tabel 10.

Tabel 10. Hasil Data Uji *Evaluasi Naïve Bayes*

Keterangan	**Hasil Analisis**
Data *Training*	80%
Data *Testing*	20%
True Positive	105
True Negative	17
False Positive	45
False Negative	1
Precision	70%
Recall	99%
F1-Score	82%
Accuracy	73%

5. KESIMPULAN DAN SARAN

Data yang telah dikumpulkan dari ulasan penggunaan aplikasi Threads pada website Google Play Store melalui proses *scraping*, kemudian diklasifikasikan. Proses klasifikasi dimulai dengan *preprocessing* untuk melakukan pemrosesan awal terhadap teks ulasan. Selanjutnya, dilakukan pelabelan menggunakan metode *lexicon-based*. Metode ini menggunakan kamus untuk mengidentifikasi apakah kata tersebut Termasuk sentimen positif atau negatif. Kamus tersebut dibuat dengan cara memberikan bobot pada setiap kata. Hasil dari proses pelabelan menunjukkan bahwa kelas sentimen positif sebanyak 536 ulasan, kelas sentimen negatif sebanyak 301 ulasan dari total 837 ulasan. Berdasarkan proses klasifikasi dengan menggunakan algoritma Naïve Bayes yang dilakukan melalui skenario data latih 80% dan data uji 20%, menunjukkan nilai akurasi sebesar 73% dengan precision sebesar 70%, recall sebesar 99%.

Sedangkan hasil *Confusion matrix* adalah kemampuan prediksi kelas positif yang tepat benar adalah 105 ulasan dan kemampuan prediksi kelas negatif yang tepat benar adalah 17 ulasan. Sedangkan tingkat kesalahan yang dapat terjadi dalam melakukan prediksi kelas positif (*false negative*) sebesar 1 ulasan dan tingkat kesalahan yang dapat terjadi dalam melakukan prediksi kelas negatif (*false positive*) sebesar 45 ulasan. Saran untuk penelitian selanjutnya adalah Metode Naive Bayes adalah metode klasifikasi yang sederhana dan mudah diimplementasikan. Namun, metode ini memiliki beberapa keterbatasan, seperti tidak dapat menangani data yang tidak linier. Oleh karena itu, penelitian berikutnya dapat dilakukan untuk meningkatkan akurasi klasifikasi dengan menggunakan metode klasifikasi lain yang lebih kompleks, seperti *SVM, Random Forest,* atau *Deep Learning*.

DAFTAR PUSTAKA

[1] F. Setya Ananto and F. N. Hasan, "Implementasi Algoritma Naïve Bayes Terhadap Analisis Sentimen Ulasan Aplikasi MyPertamina pada Google Play Store," *J. ICT Inf. Commun. Technol.*, vol. 23, no. 1, pp. 75–80, 2023, [Online]. Available: https://ejournal.ikmi.ac.id/index.php/jict-ikmi

[2] M. F. El Firdaus, N. Nurfaizah, and S. Sarmini, "Analisis Sentimen Tokopedia Pada Ulasan di Google Playstore Menggunakan Algoritma Naïve Bayes Classifier dan K-Nearest Neighbor," *JURIKOM (Jurnal Ris. Komputer)*, vol. 9, no. 5, p. 1329, 2022, doi: 10.30865/jurikom.v9i5.4774.

[3] S. M. Saputra, Suwanda Aditya, Didi Rosiyadi, Windu Gata Husain, "Analisis Sentimen E-Wallet Pada Google Play Menggunakan Algoritma," vol. 1, no. 10, pp. 3–8, 2021.

[4] G. F. Santoso *et al.*, "Respon masyarakat terhadap kebijakan psbb sebagai penekan angka covid-19," *JIKO (Jurnal Inform. dan Komputer)*, vol. 5, no. 2, pp. 38–48, 2021.

[5] S. Roiqoh and B. Zaman, "Analisis Sentimen Berbasis Aspek Ulasan Aplikasi *Mobile* JKN dengan *Lexicon* Based dan Naïve Bayes," vol. 7, pp. 1582–1592, 2023, doi: 10.30865/mib.v7i3.6194.

[6] N. F. Arminda *et al.*, "IMPLEMENTASI ALGORITMA MULTINOMIAL NAIVE BAYES PADA ANALISIS SENTIMEN TERHADAP ULASAN PENGGUNA APLIKASI BRIMO," vol. 7, no. 3, pp. 1817–1822, 2023.

[7] B. Z. Ramadhan, R. I. Adam, and I. Maulana, "Analisis Sentimen Ulasan pada Aplikasi E-Commerce dengan Menggunakan Algoritma Naïve Bayes," *J. Appl. Informatics Comput.*, vol. 6, no. 2, pp. 220–225, 2022, doi:

10.30871/jaic.v6i2.4725.

[8] N. Purwati, "Analisa Pengaruh Iklan tanpa Label Harga pada media sosial Menggunakan Algoritma Naive Bayes," *Infomatek*, vol. 24, no. 1, pp. 45–50, Jun. 2022, doi: 10.23969/infomatek.v24i1.4622.

[9] J. S. Komputer *et al.*, "Analisis Sentimen Menggunakan Metode Naive Bayes Berbasis Particle Swarm Optimization Terhadap Pelaksanaan Program Merdeka Belajar Kampus Merdeka," vol. 6, no. September, pp. 916–930, 2022.

[10] A. P. Calvin Matulessy, "ANALISIS SENTIMEN TERHADAP REVIEW PENGGUNA INDRIVE DI GOOGLE PLAYSTORE MENGGUNAKAN ALGORITMA SUPPORT VECTOR MACHINE," vol. 31, no. 1, pp. 1015–1025, 2023.

[11] A. Santosa, I. Purnamasari, and Mayasari Rini, "Pengaruh *Stopword Removal* dan *Stemming*Terhadap Performa Klasifikasi Teks KomentarKebijakan New Normal Menggunakan AlgoritmaLSTM," *J. Sains Komput. Inform.*, vol. 6, pp. 81–93, 2022.

.