

PERBANDINGAN METODE K – NEAREST NEIGHBOR (KNN) DAN NAIVE BAYES TERHADAP ANALISIS SENTIMEN PADA PENGGUNA E-WALLET APLIKASI DANA MENGGUNAKAN FITUR EKSTRAKSI TF-IDF

Muhammad Rayhan Elfansyah^{a,1}, Rudiman^{b,2,*}, Fendy Yulianto^{c,3}

^{abc} Universitas Muhammadiyah Kalimantan Timur Jl. Ir. H. Juanda No.15, Sidodadi, Kec. Samarinda Ulu, Kota Samarinda, Kalimantan Timur 75124

¹2011102441075@umkt.ac.id; ²rudiman@umkt.ac.id*; ³fyi415@umkt.ac.id

* corresponding author

ARTICLE INFO

ABSTRACT

Keywords

Sentiment Analysis, DANA Application, K-Nearest Neighbor, Naive Bayes

This research compares the accuracy of the K-Nearest Neighbor (KNN) and Naive Bayes methods in classifying user sentiment towards the DANA e-wallet application using Term Frequency-Inverse Document Frequency (TF-IDF) feature extraction. User review data was collected through web scraping techniques and labeled by linguists and lexicon models. After undergoing pre-processing steps such as case folding, cleaning, tokenizing, stopword removal, and stemming, the data was classified using the KNN and Naive Bayes methods. The research results indicate that data labeling by linguists significantly improves the accuracy of both classification methods. Additionally, using TF-IDF as a word weighting method proves effective in enhancing the performance of sentiment classification models. Sentiment analysis of user reviews of the DANA application reveals various complaints and issues faced by users, providing information that can be used to improve the features and services offered, thereby increasing user satisfaction. This research also provides a comparison between the KNN and Naive Bayes methods, which can serve as a reference for other researchers in selecting appropriate methods for sentiment analysis on similar datasets.

1. Pendahuluan

Saat ini, teknologi informasi dan komunikasi sedang mengalami perkembangan pesat di era industri 4.0. Terutama aktivitas bisnis dalam sektor niaga menjadi semakin kompleks. Kemudahan bertransaksi online telah mendorong banyak masyarakat beralih ke dalam penggunaan uang elektronik (*E-Money*) [1] Pembayaran non-tunai kini terus berkembang di Indonesia, terutama dengan munculnya berbagai produk *Financial Technology (Fintech)* dompet digital (*E-Wallet*) seperti Aplikasi GoPay, OVO, DANA, LinkAja, Paytren, dan lainnya [2] Setelah GoPay dan OVO, DANA menjadi salah satu aplikasi dompet digital yang populer.

Aplikasi DANA adalah layanan keuangan digital yang berbasis di Jakarta, yang berperan sebagai pembayaran digital. DANA berdiri pada tahun 2018 dan terdaftar di Bank Indonesia, Aplikasi DANA memungkinkan pengguna untuk melakukan transaksi keuangan tanpa uang tunai mulai dari pembayaran kebutuhan sehari-hari hingga penambahan saldo untuk investasi [3] Meskipun demikian keamanan penggunaan uang elektronik masih menjadi titik lemah dan perlu ditingkatkan, Berbagai keluhan pengguna aplikasi DANA tentang layanan dan fitur yang diberikan memiliki opini negatif seperti sulitnya

untuk mengupgrade aplikasi ke premium, seringkali terjadi kehilangan saldo, dan seringkali terjadi error saat ingin mau melakukan transaksi [4]

Tentunya dengan sejumlah fitur layanan yang disediakan oleh DANA, hal ini menciptakan beragam tanggapan dari pengguna yang terdapat dalam ulasan di Google Play Store. Sehingga sering menemukan ulasan yang bersifat positif, negatif, dan netral baik itu dari segi fitur aplikasi, dari segi keamanan, dan lain-lain yang mungkin tidak sepenuhnya relevan dengan *rating* yang diberikan di Google Play Store [5] Pada tahun 2023, Aplikasi DANA sudah mencapai 170 juta pengguna atau meningkat 23 persen dari tahun sebelumnya. Data ulasan atau komentar pengguna aplikasi DANA akan diperoleh menggunakan metode teknik web *scraping* menggunakan bahasa pemrograman python yang bersumber dari kolom komentar pada Google Play Store [6]

Maka dalam penelitian ini Google Play Store menjadi sumber data untuk menganalisis sentimen ulasan pengguna terhadap layanan dompet digital DANA. Analisis sentimen adalah ilmu yang melakukan evaluasi terhadap sentimen dan emosi yang terkait dengan suatu objek, produk, layanan, atau topik tertentu [7] Analisis sentimen juga merupakan proses klasifikasi teks dalam kalimat atau dokumen dengan tujuan untuk menentukan apakah pendapat yang terungkap dalam teks tersebut bersifat positif, negatif atau netral [8]

Terdapat beberapa penelitian sebelumnya yang pertama dilakukan oleh Enggarbela Ogi Intan Pratiwi dan Wiyli Yustanti yang berjudul “Analisis Sentimen Kualitas Layanan Teknologi Pembayaran Elektronik pada Twitter (Studi Kasus OVO dan DANA)” peneliti menggunakan tiga model klasifikasi Naive Bayes Classifier, Support Vector Machine (SVM), dan K-Nearest Neighbor dengan fitur TF-IDF klasifikasi SVM mendapatkan akurasi paling tinggi yaitu 81,33% , Naive bayes 80,63% , dan KNN 75,83% [9] Penelitian kedua oleh M.Kabirur Rifa, Moch.Hafid Totohendarto, dan M.Rafi Muttaqin yang berjudul “Analisis Sentimen Pengguna E-Wallet DANA dan GoPay Pada Twitter Menggunakan Metode Support Vector Machine (SVM)” Peneliti menggunakan metode SVM dengan menggunakan fitur TF-IDF berhasil menganalisis sentimen dengan tingkat akurasi mencapai 92% [10]

Data yang digunakan dalam penelitian ini terdiri dari 700 ulasan dalam bahasa Indonesia, Dalam pemilihan algoritma yang tepat akan sangat bergantung pada tujuan yang diinginkan, dengan begitu, peneliti menggunakan dua metode dalam analisis sentimen untuk memperbandingkan tingkat akurasi dari kedua metode tersebut diantaranya adalah metode KNN Dan Naïve Bayes [11] Algoritma KNN adalah salah satu algoritma yang sangat populer, KNN termasuk dalam kelompok pembelajaran berbasis contoh. Metode KNN merupakan teknik pembelajaran yang lamban (*lazy learning*) [12]

Metode K-Nearest Neighbor digunakan untuk proses klasifikasi dalam penelitian karena kesederhanaannya. Prosesnya didasarkan pada pendekatan pembobotan yang sederhana dan mudah diimplementasikan, diadaptasi, serta proses pembelajarannya. Selain itu, metode ini juga memiliki tingkat akurasi yang tinggi [13] Selain menggunakan metode KNN, peneliti juga menerapkan metode Naïve Bayes. Naïve Bayes adalah sebuah metode *machine learning* yang berfokus pada probabilitas. Dengan kata lain, Naïve Bayes adalah metode untuk klasifikasi teks yang memiliki kecepatan pemrosesan yang tinggi, terutama saat digunakan dalam data yang besar [14]

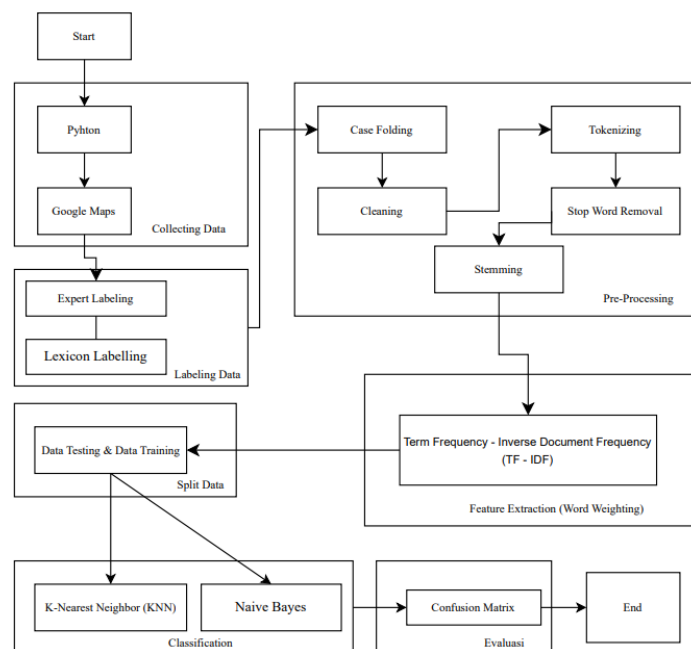
Berdasarkan uraian diatas, peneliti akan meneliti bagaimana melakukan analisis sentimen terhadap ulasan pengguna aplikasi DANA dengan menggunakan dua Metode yaitu KNN dan Naive Bayes, serta fitur TF-IDF untuk meningkatkan efisiensi proses klasifikasi. Dengan membagi sentimen menjadi kelas positif, negatif, dan netral. Data untuk penelitian ini diambil dari ulasan pengguna aplikasi DANA di Google PlayStore dengan menggunakan teknik *scraping* menggunakan bahasa pemrograman python.

Tujuan dari penelitian ini adalah untuk memahami hasil analisis sentimen dalam kelas positif, negatif dan netral serta memberikan evaluasi yang dapat digunakan untuk meningkatkan kualitas layanan pada aplikasi DANA.

2. Metodologi Penelitian

2.1. Alur Penelitian

Dalam penelitian ini, dilakukan teknik web *scraping* pada Google Play Store dengan menggunakan bahasa pemrograman Python kemudian mengambil link aplikasi DANA untuk proses analisis lebih lanjut. Metode yang dipakai dalam penelitian ini bertujuan untuk melakukan analisis sentimen terhadap data pengguna aplikasi DANA di Google Play Store dengan membandingkan dua metode, yaitu K-Nearest Neighbor dan Naive Bayes. Diagram alur penelitian pada Gambar 1.



Gambar 1. Alur Penelitian

Pada Metode Penelitian terdiri dari beberapa tahapan utama, mulai dari Pengumpulan data melalui scrapping menggunakan Python, Labeling Data oleh ahli bahasa (*expert*) dan menggunakan model *lexicon based*, *Pre-processing*, Pembobotan kata (TF-IDF), Pembagian data (*Split data*), Klasifikasi menggunakan metode KNN dan Naive Bayes, serta Evaluasi kinerja model menggunakan *confusion matriks*. Setiap tahapan akan dijelaskan secara rinci dengan memaparkan teori, konsep, dan algoritma yang digunakan.

2.2. Pengumpulan Data

Dalam penelitian ini pengumpulan data dengan menggunakan metode teknik web *scraping* yang bersumber dari kolom komentar pada Google Play Store DANA. dengan menggunakan metode web *scraping* yaitu suatu kemampuan untuk pengambilan data dalam jumlah besar dari sebuah *website*. Proses *scraping* dilakukan dengan mengambil label nama, waktu, komentar, dan jumlah data yang

diambil sebanyak 700 data. Setelah proses *scrapping* selesai, Data yang berhasil diekstrak dapat disimpan dalam format file CSV atau Excel [15] Hasil *crawling* data pada Gambar 2.

```
[ ] my_df.head()
```

	userName	score	at	content
16	Ferry Ahmad	5	2024-03-25 13:19:32	Bagus sih... Cuma sering terjadi kehilangan ua...
148	Rudy Aripin	1	2024-03-25 12:44:50	Saya lupa pin saya, kenapa sih susah banget ga...
183	RAP Project	1	2024-03-25 12:34:52	Dana makin kesini makin ga beres keamanan sald...
106	Deden Hasanudin	1	2024-03-25 12:21:41	Entah kapan sistemnya jadi lebih baik lagi... ..
21	lpank NF	1	2024-03-25 12:14:12	berhati hatilah memakai aplikasi ini, saldo se...

```
[ ] new_df.to_csv("scrapped_data.csv", index = False) #kemudian save menjadi file csv
```

Gambar 2. Hasil Pengumpulan Data

2.3. Labelling Data

Dalam proses klasifikasi teks pada data komentar untuk skripsi tugas akhir, peneliti memerlukan ahli bahasa yang berpengalaman dalam pelabelan data dan memiliki pengetahuan mendalam tentang bahasa Indonesia. Untuk itu, peneliti mengajukan permintaan di *website project.co.id* untuk mencari tenaga ahli yang memenuhi kriteria tersebut. Dalam pengajuan ini, peneliti menjelaskan bahwa diperlukan lulusan jurusan Bahasa Indonesia yang saat ini bekerja di bidang terkait, seperti guru/dosen bahasa Indonesia, penulis, atau ahli bahasa. Calon tenaga ahli diminta untuk mengajukan penawaran dengan menyertakan pekerjaan saat ini, pengalaman relevan dengan bahasa Indonesia, serta gelar akademik yang dimiliki.

Setelah mendapatkan hasil pelabelan data menggunakan *expert*, peneliti juga menggunakan model *lexicon based* untuk memberikan label secara otomatis. *Lexicon* adalah metode yang menggunakan pendekatan kamus untuk menganalisis sentimen. Kamus *lexicon* memproses bobot kata berdasarkan kamus yang memudahkan klasifikasi *automatis* kalimat yang akan dianalisis. Oleh karena itu, *lexicon* dapat digunakan untuk melabeli data sebelum dilakukan pelatihan model [16]

2.4. Pre-Processing

Sebelum melakukan *text mining* dokumen teks perlu disiapkan terlebih dahulu agar dapat digunakan dalam proses utama persiapan ini dikenal sebagai tahap *text preprocessing*, *Text preprocessing* berperan dalam mengubah data mentah yang tidak terstruktur menjadi terstruktur [17] Langkah-langkah yang dilakukan dalam tahap *pre-processing* [18]

- Case Folding* adalah langkah dalam pemrosesan teks yang bertujuan untuk mengubah semua karakter menjadi huruf kecil.
- Cleaning* adalah proses membersihkan dokumen dan menyeleksi kata yang tidak diperlukan seperti menghapus HTML, emoticon, hashtag, mention, dan URL dari teks.
- Tokenizing* adalah proses memecah teks menjadi token, yang bisa berupa kata-kata atau bahkan karakter tergantung pada kebutuhan. Selama proses tokenisasi, karakter pembatas seperti spasi dapat dihilangkan, angka dapat dihapus, dan tanda baca atau karakter non-huruf lainnya juga dapat dihapus, sesuai dengan kebutuhan pemrosesan teks yang diinginkan.
- Stopwords* adalah proses menghapus kata-kata yang tidak memiliki makna atau relevansi penting.
- Steaming* adalah proses untuk menyaring kata-kata seperti kata sambung, kata ganti, dan kata depan, serta mengubahnya menjadi bentuk kata dasar dengan menghilangkan awalan atau akhiran.

2.5. Pembobotan Kata (TF-IDF)

Pada tahap ini, dilakukan pembobotan kata dalam teks menggunakan metode *Term Frequency-Inverse Document Frequency* (TF-IDF). Langkah ini bertujuan untuk menentukan bobot kata dalam sebuah dokumen atau seberapa sering kata tersebut muncul dalam dokumen tersebut, TF-IDF adalah metode yang umum digunakan dalam analisis teks untuk mengevaluasi pentingnya sebuah kata dalam sebuah koleksi dokumen berdasarkan frekuensinya dalam dokumen dan dalam seluruh koleksi [19]

Penggunaan TF dapat menggunakan rumus pada Persamaan (2.1).

$$tf_{ij} = \frac{f_d(i)}{\max f_d(j)} \quad (2.1)$$

TF menunjukkan dokumen (d) seberapa banyak kata (t) yang muncul. Dan terkait untuk rumus IDF dapat dilihat pada Persamaan (2.2)

$$idf_t = \log \frac{N}{df_t} \quad (2.2)$$

N melambangkan jumlah kata dalam teks, df adalah jumlah teks yang memiliki kata t. Dengan menggabungkan TF dan IDF dalam pengerjaan dapat membantu meningkatkan performa. Terkait rumus pembobotan TF-IDF pada Persamaan (2.3).

$$W_{t,d} = tf_{d,t} \times idf_t \quad (2.4)$$

Keterangan :

t	= Kata kunci, <i>term</i>
d	= Dokumen
$W_{d,t}$	= Bobot d terhadap t
Tf	= Banyaknya t (kata) yang dicari dalam dokumen
Idf	= Banyak t kebalikan dari kata yang dicari

2.6. Split Data

Pada tahap ini, *split data* adalah proses membagi dataset yang digunakan dalam penelitian menjadi dua bagian, dataset tersebut umumnya dibagi menjadi data latih (*training*) dan data uji (*testing*). Data latih digunakan untuk melatih algoritma, sedangkan data uji digunakan untuk mengevaluasi kinerja algoritma tersebut [20] Dalam penelitian ini, dataset dibagi menjadi dua tahap dengan menggunakan rasio 90:10, 80:20, dan 70:30 untuk mengevaluasi pengaruh rasio tersebut terhadap kinerja model yang digunakan, yaitu Metode KNN dan Naïve Bayes dengan pembobotan fitur menggunakan TF-IDF. Hasil dari pembagian data ini menunjukkan bagaimana perbedaan rasio dapat mempengaruhi tingkat akurasi model dalam mengklasifikasikan sentimen.

2.7. Klasifikasi

Klasifikasi merupakan salah satu tahap penting dalam *text mining* yang bertujuan untuk mengelompokkan data atau objek baru ke dalam kelas atau label berdasarkan atribut-atribut tertentu. Proses ini melibatkan penggunaan teknik yang melihat variabel dari kelompok data yang sudah ada untuk menentukan aturan pengelompokan. Dengan mempelajari pola dari data yang sudah diberi label, klasifikasi memungkinkan kita untuk memprediksi kelas dari suatu objek yang belum diketahui sebelumnya [21]

2.8. K-Nearest Neighbor (KNN)

Dalam proses klasifikasi peneliti menggunakan dua metode yaitu KNN (*K-Nearest Neighbors*) dan Naive Bayes, KNN merupakan metode klasifikasi yang sederhana dan non-parametrik yang digunakan untuk mengklasifikasikan data. Meskipun algoritma ini sederhana, kinerjanya sangat baik dan menjadi parameter yang penting klasifikasi KNN membutuhkan metrik dan integer positif (K). Aturan KNN memegang posisi sampel pelatihan beserta kelas mereka ketika menghadapi data masuk baru, tujuan dari algoritma ini adalah untuk mengklasifikasikan objek baru berdasarkan nilai atribut dan data latih yang ada [22] Berikut adalah langkah-langkah klasifikasi algoritma KNN :

- a. Tentukan parameter nilai k = banyaknya jumlah tetangga terdekat.
- b. Hitung jarak antara data training dan data testing, rumus dapat dilihat pada Persamaan (2.4).

$$euc = \sqrt{(\sum_{i=1}^n (p_i - q_i)^2)} \quad (2.4)$$

Keterangan :

p_i = sample data / data training

q_i = data uji / data testing

I = variabel data

n = dimensi data

- c. Urutkan jarak-jarak tersebut dan tetapkan tetangga terdekat berdasarkan berdasarkan jarak minimum hingga ke- k .
- d. Periksa kelas dari tetangga terdekat.
- e. Gunakan mayoritas sederhana dari kelas tetangga sebagai nilai prediksi untuk data baru.

2.9. Naive Bayes

Dan algoritma kedua yang digunakan untuk klasifikasi adalah Naive Bayes. Algoritma ini menggunakan perhitungan probabilitas dan statistik dalam pengoperasiannya. Metode klasifikasi Naive Bayes digunakan untuk mengambil keputusan dengan melakukan prediksi suatu kasus berdasarkan hasil dari klasifikasi yang telah diperoleh. Pada penelitian ini, metode Naive Bayes digunakan untuk menentukan sentimen dari data Google Playstore [23]. Berikut ini adalah langkah – langkah rumus Naive Bayes pada Persamaan (2.5).

$$P(A|B) = \frac{P(B|A) \times P(A)}{P(B)} \quad (2.5)$$

Keterangan :

A = Hipotesis data merupakan suatu class spesifik.

B = Data dengan kelas yang masih belum diketahui.

$P(A/B)$ = Probabilitas hipotesis berdasar kondisi.

$P(A)$ = Probabilitas hipotesis berdasar kondisi.

$P(B/A)$ = Probabilitas berdasarkan kondisi pada hipotesis.

$P(B)$ = Probabilitas B.

2.10. Evaluasi

Evaluasi digunakan untuk melihat kinerja dari model klasifikasi yang digunakan. Metode evaluasi yang digunakan adalah *confusion matrix*. Berdasarkan *confusion matrix* pengujian model dilakukan dengan menghitung nilai *Accuracy*, *Presisi*, *Recall*. Untuk mengukur seberapa sering model klasifikasi membuat prediksi yang benar [24].

1. Akurasi

Akurasi dapat dijelaskan sebagai seberapa dekat nilai prediksi dengan nilai sebenarnya. Semakin tinggi akurasi, semakin baik proses klasifikasi tersebut. Rumus untuk menghitung akurasi dalam Persamaan (2.6).

$$Akurasi = \frac{Jumlah\ Prediksi\ Benar}{Total\ Prediksi} \quad (2.6)$$

Pada rumus di atas, cara menentukan akurasi dari sebuah data dapat dilihat dengan menghitung jumlah prediksi yang benar dan membaginya dengan jumlah total prediksi yang dilakukan.

2. Presisi

Presisi merupakan perbandingan antara prediksi positif yang benar dengan keseluruhan hasil yang diprediksi positif terhadap jumlah kelas yang telah diklasifikasikan. Rumus untuk Presisi dalam Persamaan (2.7).

$$Presisi = \frac{TP}{TP+FP} \quad (2.7)$$

Pada rumus diatas dapat dilihat bahwa cara untuk menentukan presisi dari sebuah data. Dimana :

TP = True Positif

FP = False Positif

3. Recall

Recall merupakan rasio antara jumlah prediksi yang tepat dari suatu kelas terhadap jumlah total kasus yang sebenarnya ada dalam kelas tersebut. Rumus untuk recall ditampilkan dalam dalam Persamaan (2.8).

$$Recall = \frac{TP}{TP+FN} \quad (2.8)$$

Dimana :

TP = True Positif

FN= False Negatif

3. Hasil dan Pembahasan

3.1. Hasil Pengumpulan Data

Dalam penelitian ini, data ulasan tentang hasil Aplikasi DANA yang diambil dari media Google Play Store. Sebanyak 700 ulasan yang berhasil dikumpulkan dan disimpan dalam format CSV. Proses pengumpulan data dilakukan melalui metode *crawling* pada tanggal 27 Maret 2024. Hasil pengumpulan data pada Tabel 1.

Tabel 1. Hasil Pengumpulan Data

No.	User Nama	Waktu	Komentar
1.	Nitong99	2024-03-25 08:47:39	Buat APLIKASI DANA kebanyakan update, tapi masih lola lambat saat login maupun bertransaksi super lola, tolong lah update itu kan pembaruan, harusnya tingkatkan untuk kecepatannya,, hapus fitur

fitur yang memberatkan di Aplikasi Dana yang tidak Penting... Segera ditingkatkan!!!

Total	700
-------	-----

3.2. Hasil Pengumpulan Data

Pelabelan data dilakukan oleh ahli bahasa (*expert*) dan model *lexicon based* untuk 700 data ulasan di Google Play Store. Dalam penelitian ini, ulasan dikategorikan menjadi tiga kelompok utama yaitu positif, negatif, dan netral. Dari data yang telah dilabeli *expert*, sebagian besar ulasan menunjukkan sentimen Negatif dan data yang diberi label *lexicon* sebagian besar ulasan menunjukkan Netral. Hasil pelabelan pada Tabel 2 dan 3.

Tabel 2. Hasil Pelabelan Expert

No.	Komentar	Sentimen
1.	Buat APLIKASI DANA kebanyakan update, tapi masih lola lambat saat login maupun bertransaksi super lola, tolong lah update itu kan pembaruan, harusnya tingkatkan untuk kecepatan nya,, hapus fitur fitur yang memberatkan di Aplikasi Dana yang tidak Penting... Segera ditingkatkan!!	Negatif

Tabel 3. Hasil Pelabelan Lexicon

No.	Komentar	Sentimen
1.	Buat APLIKASI DANA kebanyakan update, tapi masih lola lambat saat login maupun bertransaksi super lola, tolong lah update itu kan pembaruan, harusnya tingkatkan untuk kecepatan nya,, hapus fitur fitur yang memberatkan di Aplikasi Dana yang tidak Penting... Segera ditingkatkan!!	Positif
Total		700

3.3. Hasil Pre-Processing

Sebelum data ulasan Aplikasi DANA digunakan, tahap *pre-processing* dilakukan untuk mendapatkan data bersih. Tahapan yang dilakukan antara lain *Cleaning*, *Tokenizing*, *Case Folding*, *Stopword Removal*, dan *Stemming*. Berikut ini adalah sebelum dan setelah dilakukan *Pre-processing*.

1. Case Folding

Tahap pertama dalam *preprocessing* adalah *case folding*. *Case folding* adalah proses mengubah teks menjadi huruf kecil (*lowercase*) untuk menghilangkan variasi antara huruf besar dan huruf kecil dalam analisis teks. Hasil dari data *Case Folding* pada Tabel 4.

Tabel 4. Hasil Case Folding

No.	Case Folding
1.	buat aplikasi dana kebanyakan update, tapi masih lola lambat saat login maupun bertransaksi super lola, tolong lah update itu kan pembaruan, harusnya tingkatkan untuk kecepatan nya,, hapus fitur fitur yang memberatkan di aplikasi dana yang tidak penting... segera ditingkatkan!!!
Total	700



2. *Cleaning*

Tahap berikutnya adalah menghapus karakter-karakter yang tidak diperlukan dari teks. Langkah ini bertujuan untuk membersihkan teks dari elemen-elemen yang tidak relevan dan bisa mengganggu proses analisis. Ulasan dalam bahasa Indonesia seringkali mengandung simbol, kalimat atau kata yang tidak baku, angka, hashtag, dan tautan URL. Hasil dari data *Cleaning* ini pada Tabel 5.

Tabel 5. Hasil Cleaning

No.	Cleaning
1.	buat aplikasi dana kebanyakan update tapi masih lola lambat saat login maupun bertransaksi super lola tolong lah update itu kan pembaruan harusnya tingkatkan untuk kecepatan nya hapus fitur fitur yang memberatkan di aplikasi dana yang tidak penting segera ditingkatkan
Total	700

3. *Tokenizing*

Tahap berikutnya dalam *preprocessing* adalah *Tokenizing*. Proses ini melibatkan pemecahan kalimat menjadi kata-kata individual atau token. Tujuannya adalah untuk mempersiapkan teks agar bisa dianalisis pada tingkat kata. Hasil dari *tokenizing* pada Tabel 6.

Tabel 6. Hasil Tokenizing

No.	Tokenizing
1.	['buat', 'aplikasi', 'dana', 'kebanyakan', 'update', 'tapi', 'masih', 'lola', 'lambat', 'saat', 'login', 'maupun', 'bertransaksi', 'super', 'lola', 'tolong', 'lah', 'update', 'itu', 'kan', 'pembaruan', 'harusnya', 'tingkatkan', 'untuk', 'kecepatan', 'nya', 'hapus', 'fitur', 'fitur', 'yang', 'memberatkan', 'di', 'aplikasi', 'dana', 'yang', 'tidak', 'penting', 'segera', 'ditingkatkan']
Total	700

4. *Stopword Removal*

Tahap *Stopword Removal* digunakan untuk menghapus kata-kata yang tidak memiliki makna penting dalam teks, sehingga hanya kata-kata yang relevan dan signifikan yang tersisa untuk dianalisis. Hasil dari data *Stopword Removal* pada Tabel 7.

Tabel 6. Hasil Stopword Removal

No.	Stopword Removal
1.	['aplikasi', 'dana', 'kebanyakan', 'update', 'lola', 'lambat', 'login', 'bertransaksi', 'super', 'lola', 'tolong', 'update', 'pembaruan', 'tingkatkan', 'kecepatan', 'hapus', 'fitur', 'fitur', 'memberatkan', 'aplikasi', 'dana', 'ditingkatkan']
Total	700

5. *Stemming*

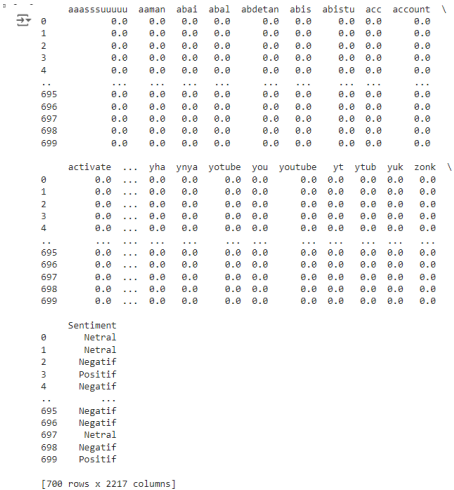
Tahap *stemming* adalah proses mengubah kata berimbuhan menjadi akar katanya. Pada tahap ini, akar kata dari setiap kata dicari menggunakan modul Sastrawi pada Python. Hasil dari *stemming* pada Tabel 8.

Tabel 6. Hasil Stemming

No.	Stemming
1.	aplikasi dana banyak update lola lambat login transaksi super lola tolong update baru tingkat cepat hapus fitur fitur berat aplikasi dana tingkat
Total	700

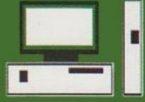
3.4. Hasil Pembobotan Kata (TF-IDF)

Pembobotan kata dilakukan menggunakan metode *Term Frequency-Inverse Document Frequency* (TF-IDF) untuk mengidentifikasi kata-kata penting dalam ulasan aplikasi DANA di Google Play Store. Dataset yang digunakan terdiri dari 700 baris dengan 2214 kolom, di mana setiap kolom mewakili kata unik dalam korpus data. Hasil yang ditampilkan dalam Gambar 3 TF-IDF menunjukkan bahwa sebagian besar kata memiliki bobot rendah dengan nilai 0, yang mengindikasikan bahwa kata-kata tersebut jarang muncul dalam dokumen.



Gambar 3. Pembobotan Kata (TF-IDF)

Berdasarkan hasil skoring keseluruhan kata dengan metode *Term Frequency-Inverse Document Frequency* (TF-IDF), dapat ditemukan kata-kata yang paling sering muncul dan memiliki bobot signifikan dalam korpus data. Gambar pada 4 menunjukkan bahwa kata "dana" memiliki frekuensi tertinggi dengan skor 46.326304, diikuti oleh kata "aplikasi" dengan skor 40.368285, dan "saldo" dengan skor 33.963624. Kata-kata lain yang juga memiliki bobot tinggi termasuk "transaksi" 30.646935, "premium" 29.086346, "upgrade" 29.019483, "tolong" 21.933675, "uang" 20.685666, "akun" 18.191209, dan "baik" 15.989883.



➜ Most frequent terms:

	term	frequency
424	dana	46.326304
102	aplikasi	40.368285
1704	saldo	33.963624
2041	transaksi	30.646935
1586	premium	29.086346
2121	upgrade	29.019483
2027	tolong	21.933675
2087	uang	20.685666
42	akun	18.191209
162	baik	15.989883

Gambar 4. Skoring TF

3.5. Hasil Visualisasi

Kata-kata yang sering muncul dalam teks ulasan divisualisasikan menggunakan *wordcloud*. Dalam pembuatan *wordcloud*, peneliti menggunakan *library wordcloud* dan hasil ekstraksi fitur dengan TF-IDF. Visualisasi kata yang sering muncul dalam ulasan positif pada Gambar 5, visualisasi kata yang sering muncul dalam ulasan negatif pada Gambar 6, dan visualisasi kata yang sering muncul dalam ulasan netral pada Gambar 7.



Gambar 5. Wordcloud Positif



Gambar 6. Wordcloud Negatif



Gambar 6. Wordcloud Netral

3.6. Hasil Evaluasi

Dalam tahap evaluasi peneliti menggunakan dua metode sebagai perbandingan. Evaluasi berfungsi untuk mengetahui akurasi dari model algoritma yang dibuat. Pada penelitian ini, evaluasi yang dilakukan adalah dengan menghitung akurasi. Data yang digunakan sebanyak 700 data. Sebelumnya peneliti melakukan labelling data menggunakan model *Lexicon Based*. Dari proses pelabelan tersebut didapat data kelas positif sebanyak 104, Data kelas negatif sebanyak 76 data, dan Data kelas netral sebanyak 520 data.

Kemudian peneliti juga melakukan *labelling* melalui *expert*, Data yang diberi label oleh ahli bahasa (*expert*) pada kolom sentimen terbagi menjadi 3 kelas dengan kelas positif, negatif, dan netral. Dari proses pelabelan tersebut didapat data kelas positif sebanyak 23, Data kelas negatif 672 data, dan Data kelas netral 5 data. Total dataset yang digunakan pada akhirnya berjumlah 700 data. Dataset tersebut akan dibagi menjadi data *training* dan data *testing* dengan menggunakan perbandingan rasio 90:10, 80:20, dan 70:30. Hasilnya di evaluasi menggunakan *metrix* evaluasi yaitu *accuracy* pada Tabel 7 dan 8.

Tabel 7. Hasil Pembagian Data KNN

Rasio	Akurasi
70:30	96%
80:20	95%
90:10	95%

Tabel 8. Hasil Pembagian Data Naive Bayes

Rasio	Akurasi
70:30	96%
80:20	95%
90:10	95%

Setelah dilakukan pengujian dengan 3 model rasio yaitu 90:10, 80:20, dan 70:30 didapatkan akurasi untuk masing masing rasio sebesar 95% untuk pengujian menggunakan 90:10, 95% untuk pengujian dengan rasio 80:20, dan 96% untuk pengujian dengan rasio 70:30. Dengan berbagai model rasio pengujian tersebut, menunjukkan bahwa model KNN dan Naive Bayes yang dilatih dengan teknik teks *pre-processing* yang benar serta pembobotan TF-IDF memberikan performa yang baik.

Hasil evaluasi menunjukkan bahwa model KNN dan Naive Bayes dengan rasio pengujian 70:30 mencapai akurasi sebesar 0.9619, Ini berarti model berhasil mengklasifikasikan sekitar 96.19% data dengan benar. Pada rasio pengujian 80:20, model menunjukkan performa dengan akurasi sebesar 0.9571, mengindikasikan bahwa sekitar 95.71% data diklasifikasikan dengan benar. Dan pada rasio pengujian 90:10, akurasi model menunjukkan akurasi yang sama 0.9571, yang berarti model mampu mengklasifikasikan sekitar 95.71% data dengan benar.

Perbedaan akurasi ini menunjukkan bahwa performa model sedikit bervariasi tergantung pada rasio pembagian data latih dan data uji. Pada rasio 70:30, model menunjukkan performa yang lebih baik dibandingkan dengan rasio 90:10 dan 80:20. Hal ini dapat disebabkan oleh berbagai faktor, seperti jumlah data uji yang lebih besar pada rasio 70:30, yang memungkinkan model untuk mempelajari lebih banyak pola dari data. Sedangkan pada rasio 90:10 dan 80:20, jumlah data uji yang lebih sedikit mungkin tidak cukup untuk menangkap variasi yang ada dalam data, sehingga menyebabkan penurunan akurasi. Secara keseluruhan, hasil ini menunjukkan pemilihan rasio yang tepat dalam pembagian data latih dan data uji untuk mendapatkan performa model yang optimal. Pemilihan rasio yang baik dapat membantu model

mempelajari pola dengan lebih baik dan menghasilkan prediksi yang lebih akurat [25] Dari ketiga rasio tersebut peneliti mengambil rasio yang terbaik yaitu 70:30.

1. Metode K-Nearest Neighbor (KNN)

Pada metode KNN data yang digunakan yaitu sebanyak 700 data. Data tersebut dibagi menjadi dua kelompok yaitu data *training* (data latih) dan data *testing* (data uji) dengan menggunakan rasio 70:30 yang mendapatkan performa lebih baik. Sebanyak 490 data digunakan sebagai data *training* dan 210 data sebagai data *testing*.

Setelah menentukan rasio terbaik, dilakukan pengujian untuk mencari akurasi optimal pada berbagai nilai K dalam algoritma *K-Nearest Neighbors* (KNN). Jarak K dihitung menggunakan *Euclidean Distance*. Beberapa nilai K digunakan untuk membandingkan hasil akurasi dari masing-masing nilai. Uji ini dilakukan baik saat validasi maupun evaluasi akhir, untuk menentukan nilai K yang paling optimal di antara semua nilai yang diuji. Dengan membandingkan akurasi dari setiap nilai K, kita dapat menemukan nilai K yang memberikan kinerja terbaik untuk model tersebut [26].

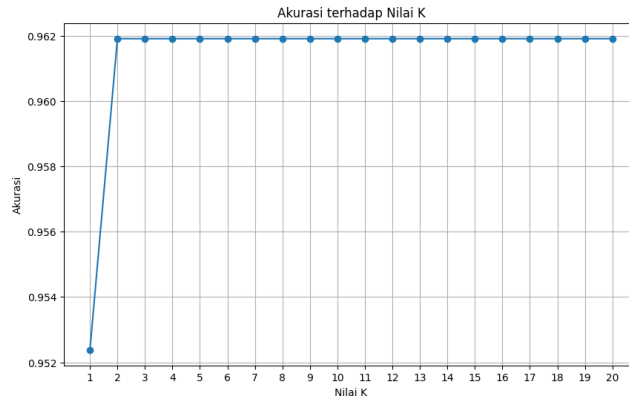
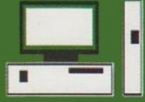
Evaluasi akurasi model *K-Nearest Neighbor* (KNN) dengan berbagai nilai K, K1 menunjukkan akurasi 0.9524 dan K2 hingga K20 menunjukkan bahwa setiap nilai K menghasilkan akurasi yang sama, yaitu 0.9619. Hal ini menunjukkan bahwa model KNN memiliki performa yang sangat baik dan stabil pada dataset yang digunakan. Perubahan nilai K tidak mempengaruhi akurasi secara signifikan, karena ketidakseimbangan kelas, di mana salah satu kelas memiliki jumlah sampel lebih banyak dibandingkan dengan kelas lainnya. Ketidakseimbangan ini bisa membuat model lebih mudah memprediksi kelas mayoritas dengan benar, sehingga meningkatkan akurasi keseluruhan tanpa benar-benar mencerminkan kemampuan model dalam memprediksi kelas minoritas dengan akurat [27] Hasilnya pada Gambar 7.

```

Akurasi untuk berbagai nilai K:
Nilai K: 1, Akurasi: 0.9524
Nilai K: 2, Akurasi: 0.9619
Nilai K: 3, Akurasi: 0.9619
Nilai K: 4, Akurasi: 0.9619
Nilai K: 5, Akurasi: 0.9619
Nilai K: 6, Akurasi: 0.9619
Nilai K: 7, Akurasi: 0.9619
Nilai K: 8, Akurasi: 0.9619
Nilai K: 9, Akurasi: 0.9619
Nilai K: 10, Akurasi: 0.9619
Nilai K: 11, Akurasi: 0.9619
Nilai K: 12, Akurasi: 0.9619
Nilai K: 13, Akurasi: 0.9619
Nilai K: 14, Akurasi: 0.9619
Nilai K: 15, Akurasi: 0.9619
Nilai K: 16, Akurasi: 0.9619
Nilai K: 17, Akurasi: 0.9619
Nilai K: 18, Akurasi: 0.9619
Nilai K: 19, Akurasi: 0.9619
Nilai K: 20, Akurasi: 0.9619
    
```

Gambar 7. Hasil Akurasi Nilai K

Grafik yang menunjukkan hubungan antara nilai K dan akurasi dalam model *K-Nearest Neighbor* (KNN) menunjukkan bahwa sumbu horizontal mewakili nilai K, yang berkisar dari 1 hingga 20, sementara sumbu vertikal mewakili akurasi model. Dari grafik tersebut, terlihat K1 mendapatkan akurasi 0.952 dan K2 hingga K20 mendapatkan akurasi yang sama 0.962 untuk setiap nilai K yang diuji. Hasil ini pada Gambar 8.



Gambar 8. Grafik Akurasi Nilai K

Hasil evaluasi akurasi dari model *K-Nearest Neighbor* (KNN) menunjukkan bahwa K1 mencapai akurasi sebesar 0.952, sedangkan K2 hingga K20 menghasilkan akurasi yang sama, yaitu 0.962. Hal ini menunjukkan bahwa model KNN menunjukkan performa yang sangat baik dan stabil pada dataset yang digunakan, di mana perubahan nilai K tidak berpengaruh secara signifikan terhadap akurasi. Stabilitas ini mengindikasikan bahwa model KNN efektif dalam memprediksi label data uji dengan tingkat ketepatan yang konsisten. Meskipun terdapat perbedaan pada K1, nilai akurasi K2 hingga K20 tetap konsisten pada 0.962. Berdasarkan hasil ini, peneliti memilih nilai K=5 sebagai parameter final untuk model KNN.

Untuk mengevaluasi kinerja suatu model klasifikasi yang memprediksi sentimen dari teks. Evaluasi ini melibatkan perhitungan nilai-nilai seperti akurasi, presisi, dan recall. Kemudian, dilakukan perhitungan *persentase* dari masing-masing kategori tersebut terhadap total jumlah data. Akurasi model dihitung berdasarkan prediksi yang dihasilkan pada data uji pada Gambar 9 dan 10.

Accuracy: 0.9619047619047619

Classification Report:

	precision	recall	f1-score	support
NEGATIF	0.96	1.00	0.98	202
NETRAL	1.00	0.00	0.00	1
POSITIF	1.00	0.00	0.00	7
accuracy			0.96	210
macro avg	0.99	0.33	0.33	210
weighted avg	0.96	0.96	0.94	210

Gambar 9. Hasil Klasifikasi KNN Expert

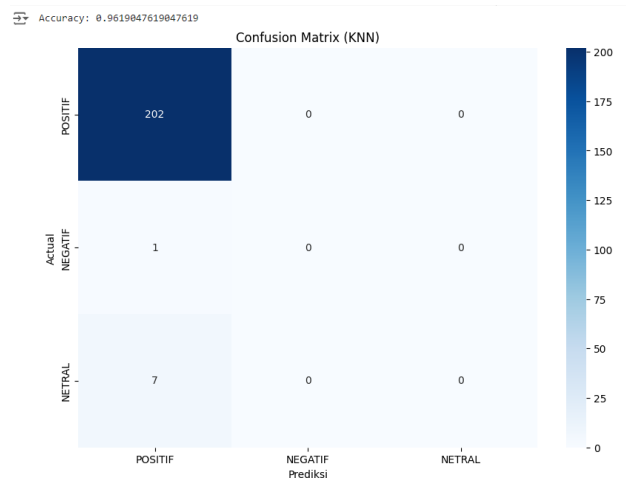
Accuracy: 0.780952380952381

Classification Report:

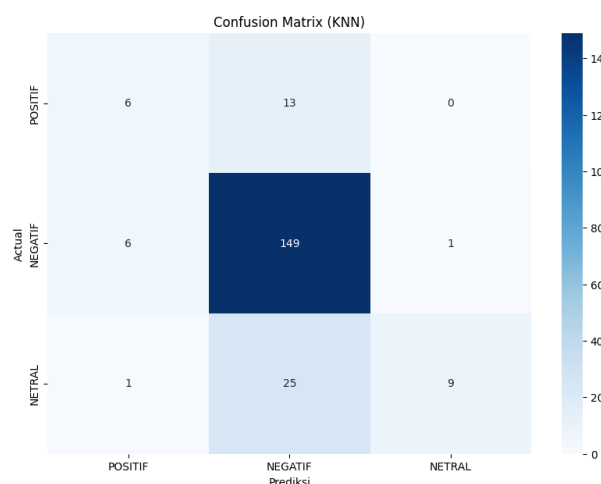
	precision	recall	f1-score	support
Negatif	0.46	0.32	0.37	19
Netral	0.80	0.96	0.87	156
Positif	0.90	0.26	0.40	35
accuracy			0.78	210
macro avg	0.72	0.51	0.55	210
weighted avg	0.78	0.78	0.75	210

Gambar 10. Hasil Klasifikasi KNN Lexicon

Akurasi, Presisi, dan Recall dari model diatas adalah hasil yang digunakan oleh peneliti menggunakan data yang diberi label oleh (*expert*) mendapatkan akurasi sebesar 0.9619 atau 96%, Presisi 96%, dan Recall 100%. Sebelumnya peneliti juga melakukan proses sampai tahap akurasi, presisi, dan recall menggunakan data yang diberi label model *lexicon* dan mendapatkan akurasi sebesar 0.7809 atau 78%, presisi 46%, dan recall 32%. Untuk tampilan *Confusion Matrix* pada Gambar 11 dan 12.



Gambar 11. Hasil Confusion Matriks KNN Expert



Gambar 12. Hasil Confusion Matriks KNN Lexicon

Adapun perhitungan manual akurasi berdasarkan hasil Gambar 11 dan 12, pada Tabel 9 dan 10.

Tabel 9. Hasil Data Uji KNN Expert

Kategori	Prediksi	Prediksi	Prediksi
	Positif	Negatif	Netral
Aktual Positif	202	0	0
Aktual Negatif	1	0	0
Aktual Netral	7	0	0

Akurasi adalah rasio dari jumlah prediksi yang benar dan membaginya terhadap total jumlah prediksi. Berikut adalah hasil perhitungan manual akurasi pada Persamaan (3.1).

$$Akurasi = \frac{202+0+0}{202+0+0+1+0+0+7+0+0} = \frac{202}{210} = 0.9619\% \quad (3.1)$$

Tabel 10. Hasil Data Uji KNN Lexicon

Kategori	Prediksi Positif	Prediksi Negatif	Prediksi Netral
Aktual Positif	6	13	0
Aktual Negatif	6	149	1
Aktual Netral	1	25	9

Berikut adalah hasil perhitungan manual akurasi pada Persamaan (3.2).

$$Akurasi = \frac{6+149+9}{6+13+0+6+149+1+1+25+9} = \frac{164}{210} = 0.7809\% \quad (3.2)$$

Hasil pengujian model menggunakan *labelling expert* menunjukkan akurasi sebesar 0.9619 atau 96% pada data uji, Berarti dari 210 data yang diuji model mampu memprediksi dengan benar 96 kasus. Dan hasil pengujian model menggunakan *lexicon* mendapatkan akurasi sebesar 0.7809 atau 78% pada data uji, Berarti dari 210 data yang diuji model mampu memprediksi dengan benar 78 kasus. Akurasi ini dihitung berdasarkan jumlah prediksi yang benar dibagi dengan total jumlah kasus.

2. Naive Bayes

Pada metode Naive Bayes data yang digunakan yaitu sebanyak 700 data. Data tersebut dibagi menjadi dua kelompok yaitu data *training* dan data *testing* dengan menggunakan rasio 70:30 yang mendapatkan performa lebih baik. Sebanyak 490 data digunakan sebagai data *training* dan 210 data sebagai data *testing*. Akurasi model dihitung berdasarkan prediksi yang dihasilkan pada data uji pada Gambar 13 dan 14.

```

Accuracy: 0.9619047619047619

Classification Report:
              precision    recall  f1-score   support

   NEGATIF      0.96      1.00      0.98         202
    NETRAL      1.00      0.00      0.00          1
   POSITIF      1.00      0.00      0.00          7

 accuracy              0.96         210
 macro avg              0.99         210
 weighted avg           0.96         210
    
```

Gambar 13. Hasil Klasifikasi Naive Bayes Expert


```

Accuracy: 0.7428571428571429

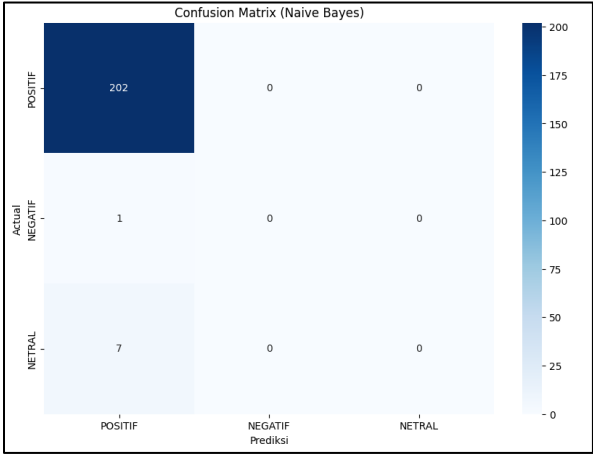
Classification Report:
      precision    recall  f1-score   support

   Negatif         1.00      0.00      0.00         19
    Netral         0.74      1.00      0.85        156
    Positif         1.00      0.00      0.00         35

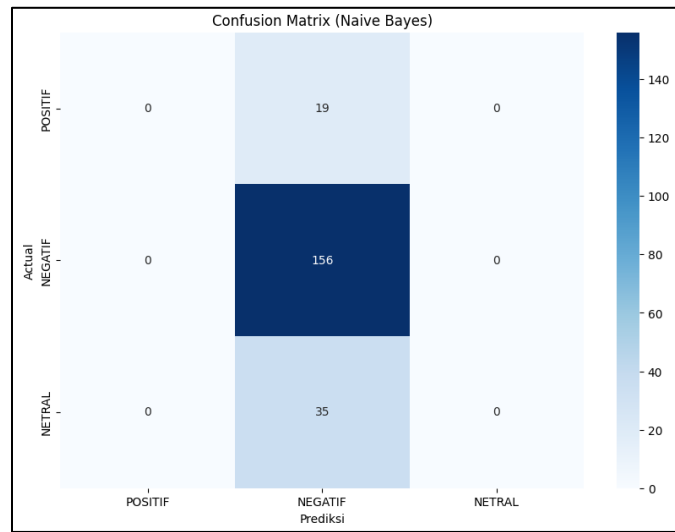
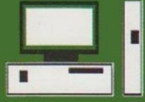
 accuracy          0.74         210
  macro avg         0.91      0.33      0.28         210
 weighted avg         0.81      0.74      0.63         210
  
```

Gambar 14. Hasil Klasifikasi Naive Bayes Lexicon

Akurasi, Presisi, dan Recall dari model di atas adalah hasil yang digunakan oleh peneliti menggunakan data yang diberi label oleh (*expert*) mendapatkan akurasi sebesar 0.9619 atau 96%, Presisi 96%, dan Recall 100%. Sebelumnya peneliti juga melakukan proses sampai tahap akurasi, presisi, dan recall menggunakan data yang diberi label model *lexicon* dan mendapatkan akurasi sebesar 0.7428 atau 74%, presisi 100%, dan recall 0%. Untuk tampilan *Confusion Matrix* pada Gambar 15 dan 16.



Gambar 15. Hasil Confusion Matriks Naive Bayes Expert



Gambar 16. Hasil Confusion Matriks Naive Bayes Lexicon

Adapun perhitungan manual akurasi berdasarkan hasil Gambar 15 dan 16, pada Tabel 11 dan 12.

Tabel 11. Hasil Data Uji Naive Bayes Expert

Kategori	Prediksi Positif	Prediksi Negatif	Prediksi Netral
Aktual Positif	202	0	0
Aktual Negatif	1	0	0
Aktual Netral	7	0	0

Akurasi adalah rasio dari jumlah prediksi yang benar dan membagnya terhadap total jumlah prediksi.

Berikut adalah hasil perhitungan manual akurasi pada Persamaan (3.3).

$$Akurasi = \frac{202+0+0}{202+0+0+1+0+0+7+0+0} = \frac{202}{210} = 0.9619\% \quad (3.3)$$

Tabel 12. Hasil Data Uji Naive Bayes Lexicon

Kategori	Prediksi Positif	Prediksi Negatif	Prediksi Netral
Aktual Positif	0	19	0
Aktual Negatif	0	156	0
Aktual Netral	0	35	0

Berikut adalah hasil perhitungan manual akurasi pada Persamaan (3.4).

$$Akurasi = \frac{0+156+0}{0+19+0+0+156+0+0+35+0} = \frac{156}{210} = 0.7428\% \quad (3.4)$$

Hasil pengujian model menggunakan *labelling expert* menunjukkan akurasi sebesar 0.9619 atau 96% pada data uji, Berarti dari 210 data yang diuji model mampu memprediksi dengan benar 96 kasus. Dan hasil pengujian model menggunakan *lexicon* mendapatkan akurasi sebesar 0.7428 atau 74% pada data uji, Berati dari 210 data yang diuji model mampu memprediksi dengan benar 74 kasus. Akurasi ini dihitung berdasarkan jumlah prediksi yang benar dibagi dengan total jumlah kasus.

Dari kedua metode diatas terdapat perbedaan akurasi jika seorang peneliti menggunakan data yang sama tetapi menggunakan *labelling* yang berbeda. jika peneliti menggunakan data yang diberi *labelling* model *lexicon*, pada metode KNN mendapatkan akurasi sebesar 78% dan metode Naive Bayes mendapatkan akurasi sebesar 74%. Dan dapat dilihat terjadi perbedaan jika dibandingkan dengan menggunakan data yang diberi label oleh (*expert*) menggunakan metode KNN dan Naive Bayes mendapatkan akurasi yang sama sebesar 96%.

Perbedaan akurasi tersebut disebabkan karena terjadinya perubahan kelas sentimen yang diberi label oleh seorang ahli bahasa (*expert*) dan label menggunakan model *lexicon*. Terdapat penambahan pada kelas positif yang diberi label *lexicon*, dari 23 data kelas positif menjadi 104 data positif. Selanjutnya penambahan pada kelas netral dari 5 data kelas negatif menjadi 520 data netral. Dan menurunnya kelas negatif yang diberi label *expert*, dari 672 data kelas negatif menjadi 76 data netral. Hal inilah yang membuat terjadi perbedaan akurasi dari kedua metode tersebut.

4. Kesimpulan

Penelitian ini membahas perbandingan akurasi metode *K-Nearest Neighbor* (KNN) dan Naive Bayes dalam klasifikasi sentimen menggunakan data yang diberi label secara ahli bahasa (*expert*) dan menggunakan model *lexicon*. Pada proses *Pre-processing* melibatkan *Case Folding*, *Cleaning*, *Tokenizing*, *Stopword Removal*, dan *Stemming* terbukti efektif dalam mempersiapkan data teks untuk analisis. Penggunaan metode TF-IDF dalam pembobotan kata membantu mengidentifikasi kata-kata kunci yang relevan dari setiap ulasan, memungkinkan model untuk lebih memahami konteks dan makna di balik teks.

Hasil penelitian menunjukkan bahwa metode KNN dan Naive Bayes memiliki akurasi yang berbeda berdasarkan sumber label data. Pada data yang diberi label model *lexicon*, akurasi KNN mencapai 78% dan Naive Bayes 74%. Namun, ketika data diberi label oleh *expert*, akurasi kedua metode meningkat menjadi 96%. Perbedaan ini disebabkan oleh perubahan kelas sentimen saat pelabelan menggunakan model *lexicon* dibandingkan dengan pelabelan oleh *expert*, dimana data positif dan netral menurun tetapi pada data negatif mengalami peningkatan. Penelitian ini menyimpulkan bahwa kualitas pelabelan data sangat mempengaruhi akurasi metode klasifikasi yang digunakan, dan pelabelan oleh ahli bahasa (*expert*) dapat meningkatkan keakuratan dalam klasifikasi sentimen.

Selain itu, penelitian ini juga meneliti pengaruh rasio pembagian data dan nilai K pada kinerja model KNN. Hasilnya menunjukkan bahwa rasio pembagian data 70:30 memberikan akurasi terbaik sebesar 96%, dibandingkan dengan rasio 90:10 dan 80:20 yang masing-masing mencapai akurasi 95%. Dalam hal penggunaan nilai K, nilai K1 mencapai akurasi 95.24%, sementara nilai K2 hingga K20 mencapai akurasi yang sama yaitu 96.19%. Berdasarkan hasil ini, nilai K=5 dipilih sebagai parameter final karena memberikan kinerja optimal tanpa penurunan akurasi yang signifikan.

Penelitian ini menyimpulkan, bahwa pentingnya pemilihan metode, rasio data, dan parameter model yang tepat dalam meningkatkan akurasi klasifikasi sentimen. penggunaan pelabelan oleh *expert* dapat meningkatkan dan berkontribusi signifikan terhadap hasil yang diperoleh. dan memberikan informasi tentang efisiensi dan efektivitas dalam memilih metode klasifikasi yang lebih efektif dalam melakukan analisis sentimen yang sesuai dengan kebutuhan penelitian.

Daftar Pustaka

- [1] E. Salehudin Basryah, A. Erfina, and C. Warman, "ANALISIS SENTIMEN APLIKASI DOMPET DIGITAL DI ERA 4.0 PADA MASA PENDEMI COVID-19 DI PLAY STORE MENGGUNAKAN ALGORITMA NAIVE BAYES CLASSIFIER," 2021.
- [2] M. Badri, "Adopsi Inovasi Aplikasi Dompot Digital di Kota Pekanbaru," 2020. [Online]. Available: www.ejournal.polbeng.ac.id/index.php/IBP



- [3] F. A. Larasati, D. E. Ratnawati, and B. T. Hanggara, “Analisis Sentimen Ulasan Aplikasi Dana dengan Metode Random Forest,” 2022. [Online]. Available: <http://j-ptiik.ub.ac.id>
- [4] Heti Palestina Yunani, “4 Alasan Aman Pakai DANA, Memberikan Kenyamanan saat Bertransaksi,” *Harian Disway*.
- [5] A. Athallah Muhammad *et al.*, “ANALISIS SENTIMEN PENGGUNA APLIKASI DANA BERDASARKAN ULASAN PADA GOOGLE PLAY MENGGUNAKAN METODE SUPPORT VECTOR MACHINE,” *Senamika*, 2022.
- [6] Shanty, “Capai Pertumbuhan Positif 2023 Hingga 23% DANA Sambut Optimis 2024.”
- [7] A. Nurian and B. Nurina Sari, “ANALISIS SENTIMEN ULASAN PENGGUNA APLIKASI GOOGLE PLAY MENGGUNAKAN NAÏVE BAYES,” *Jurnal Informatika dan Teknik Elektro Terapan*, vol. 11, no. 3, pp. 2830–7062, 2023, doi: 10.23960/jitet.v11i3%20s1.3348.
- [8] A. Z. Amrullah, A. Sofyan Anas, M. Adrian, and J. Hidayat, “Analisis Sentimen Movie Review Menggunakan Naive Bayes Classifier Dengan Seleksi Fitur Chi Square,” *Jurnal*, vol. 2, no. 1, 2020, doi: 10.30812/bite.v2i1.804.
- [9] E. Ogi, I. Pratiwi¹, and W. Yustanti², “Analisis Sentimen Kualitas Layanan Teknologi Pembayaran Elektronik pada Twitter (Studi Kasus Ovo dan Dana),” *JEISBI*, vol. 02, p. 2021, 2021.
- [10] M. K. Rifa, M. H. Totohendarto, and M. R. Muttaqin, “Analisis Sentimen Pengguna E-Wallet Dana Dan Gopay Pada Twitter Menggunakan Metode Support Vector Machine (SVM),” *IJCCS*, vol. x, No.x, pp. 1–5, 2023.
- [11] R. Puspita and A. Widodo, “Perbandingan Metode KNN, Decision Tree, dan Naïve Bayes Terhadap Analisis Sentimen Pengguna Layanan BPJS,” *Jurnal Informatika Universitas Pamulang*, vol. 5, no. 4, p. 646, Dec. 2021, doi: 10.32493/informatika.v5i4.7622.
- [12] D. Cahyanti, A. Rahmayani, and S. Ainy Husniar, “Indonesian Journal of Data and Science Analisis performa metode Knn pada Dataset pasien pengidap Kanker Payudara,” vol. 1, no. 2, pp. 39–43, 2020.
- [13] S. S. Salim and J. Mayary, “ANALISIS SENTIMEN PENGGUNA TWITTER TERHADAP DOMPET ELEKTRONIK DENGAN METODE LEXICON BASED DAN K – NEAREST NEIGHBOR,” *Jurnal Ilmiah Informatika Komputer*, vol. 25, no. 1, pp. 1–17, 2020, doi: 10.35760/ik.2020.v25i1.2411.
- [14] S. Nurul, J. Fitriyyah, N. Safriadi, E. Esyudha, and P. #3, “Analisis Sentimen Calon Presiden Indonesia 2019 dari Media Sosial Twitter Menggunakan Metode Naive Bayes,” 2019, [Online]. Available: <http://dev.twitter.com>.
- [15] W. E. Saputro, H. Yuana, and W. D. Puspitasari, “ANALISIS SENTIMEN PENGGUNA DOMPET DIGITAL DANA PADA KOLOM KOMENTAR GOOGLE PLAY STORE DENGAN METODE KLASIFIKASI SUPPORT VECTOR MACHINE,” 2023.
- [16] M. Fernanda and N. Fathoni, “Perbandingan Performa Labeling Lexicon InSet dan VADER pada Analisa Sentimen Rohingya di Aplikasi X dengan SVM,” *Jurnal Informatika dan Sains Teknologi*, vol. 1, no. 3, pp. 62–76, 2024, doi: 10.62951/modem.v1i3.112.



- [17] S. Khairunnisa, A. Adiwijaya, and S. Al Faraby, "Pengaruh Text Preprocessing terhadap Analisis Sentimen Komentar Masyarakat pada Media Sosial Twitter (Studi Kasus Pandemi COVID-19)," *JURNAL MEDIA INFORMATIKA BUDIDARMA*, vol. 5, no. 2, p. 406, Apr. 2021, doi: 10.30865/mib.v5i2.2835.
- [18] A. Dwiki, A. Putra, and S. Juanita, "Analisis Sentimen Pada Ulasan Pengguna Aplikasi Bibit Dan Bareksa Dengan Algoritma KNN," *JATISI*, vol. 8, no. 2, 2021, [Online]. Available: <http://jurnal.mdp.ac.id>
- [19] T. Fadiyah Basar, D. E. Ratnawati, and I. Arwani, "Analisis Sentimen Pengguna Twitter terhadap Pembayaran Cashless menggunakan ShopeePay dengan Algoritma Random Forest," 2022. [Online]. Available: <http://j-ptiik.ub.ac.id>
- [20] A. Putri, C. Syaficha Hardiana, E. Novfuja, F. Try Puspa Siregar, Y. Fatma, and R. Wahyuni, "Komparasi Algoritma K-NN, Naive Bayes dan SVM untuk Prediksi Kelulusan Mahasiswa Tingkat Akhir," *Institut Riset dan Publikasi Indonesia (IRPI) MALCOM: Indonesian Journal of Machine Learning and Computer Science Journal Homepage*, vol. 3, no. 1, pp. 20–26, 2023.
- [21] D. Azzahra Nasution, H. H. Khotimah, and N. Chamidah, "PERBANDINGAN NORMALISASI DATA UNTUK KLASIFIKASI WINE MENGGUNAKAN ALGORITMA K-NN," 2019.
- [22] P. Putra, A. M. H Pardede, and S. Syahputra, "ANALISIS METODE K-NEAREST NEIGHBOUR (KNN) DALAM KLASIFIKASI DATA IRIS BUNGA," *Jurnal Teknik Informatika Kaputama (JTJK)*, vol. 6, no. 1, 2022.
- [23] N. M. A. J. Astari, Dewa Gede Hendra Divayana, and Gede Indrawan, "Analisis Sentimen Dokumen Twitter Mengenai Dampak Virus Corona Menggunakan Metode Naive Bayes Classifier," *Jurnal Sistem dan Informatika (JSI)*, vol. 15, no. 1, pp. 27–29, Nov. 2020, doi: 10.30864/jsi.v15i1.332.
- [24] J. Khatib Sulaiman, M. Dzakwan Ar Rosyid, and I. Artikel Abstrak, "Klasifikasi Tingkat Risiko Kesehatan Ibu Hamil Menggunakan Algoritma Support Vector Machine Universitas AMIKOM Yogyakarta," *Indonesian Journal of Computer Science Attribution*, vol. 12, no. 5, pp. 2023–2798, 2023.
- [25] R. Kurniawan and D. Anubhakti, "Implementasi Algoritma Support Vector Machine Dalam Memprediksi Harga Saham PT. KRAKATAU STEEL TBK," vol. 2, no. 2, 2023.
- [26] I. H. Herman, D. Widiyanto, and I. Ernawati, *PENGUNAAN K-NEAREST NEIGHBOR (KNN) UNTUK MENGIDENTIFIKASI CITRA BATIK PEWARNA ALAMI DAN PEWARNA SINTETIS BERDASARKAN WARNA*. 2020.
- [27] R. Nur Ariyanti and R. Cahya Wihandika, "Identifikasi Jenis Attention Deficit Hyperactivity Disorder (ADHD) Pada Anak Usia Dini Menggunakan Metode Modified K-Nearest Neighbor (MKNN)," 2019. [Online]. Available: <http://j-ptiik.ub.ac.id>