

EVALUASI KINERJA *LOGISTIC REGRESSION* DAN *MULTINOMIAL NAÏVE BAYES* DALAM KLASIFIKASI SENTIMEN *MOBILE BANKING*

Agil Aryanusa^{1*}, Ronny Makhfuddin Akbar², Yanuarini Nur S³

^{1,2,3}Universitas Islam Majapahit; Jl. Raya Jabon No.KM.0.7, Gayaman, Kec. Mojoanyar, Kab Mojokerto, Jawa Timur 61364; (0321) 399474

Keywords:

Logistic Regression, Multinomial Naïve Bayes Sentiment Analysis, Mobile Banking

Correspondent Email:

akuagil3@gmail.com

Abstrak. Dengan pesatnya perkembangan teknologi digital, aplikasi *mobile banking* telah menjadi kebutuhan utama bagi masyarakat modern. Namun, keluhan pengguna terkait *bug* dan gangguan teknis sering kali muncul, yang dapat menurunkan kualitas layanan. Penelitian ini bertujuan untuk mengevaluasi sentimen pengguna terhadap fitur-fitur aplikasi *mobile banking* dengan membandingkan performa algoritma *Logistic Regression* dan *Multinomial Naïve Bayes*. Analisis sentimen ini penting untuk memahami persepsi publik secara objektif. Data ulasan pengguna dari empat aplikasi *mobile banking* populer (blu by BCA, BRImo, BNI Mobile Banking, dan Livin' by Mandiri) dikumpulkan dari Google Play Store sebanyak 50.000 data per-aplikasi, dengan total 200.000 data. Proses penelitian meliputi pengumpulan data, preprocessing (*cleaning, case folding, tokenization, normalization, stopword removal, dan stemming*), pelabelan sentimen menggunakan *lexicon-based*, serta klasifikasi dan evaluasi model. Hasil penelitian menunjukkan bahwa sebagian besar ulasan pengguna bersifat negatif, terutama terkait fitur login, transaksi, dan verifikasi, yang mengindikasikan ketidakpuasan terhadap keandalan teknis. Dalam perbandingan performa, model *Logistic Regression* jauh lebih unggul dengan akurasi 92-93%, dibandingkan dengan *Multinomial Naïve Bayes* yang hanya mencapai 71-74%. Hal ini membuktikan bahwa *Logistic Regression* lebih efektif dalam klasifikasi sentimen ulasan aplikasi *mobile banking*.



Copyright © [JITET](http://www.jitet.org) (Jurnal Informatika dan Teknik Elektro Terapan). This article is an open access article distributed under terms and conditions of the Creative Commons Attribution (CC BY NC)

Abstract. With the rapid development of digital technology, mobile banking applications have become a primary need for modern society. However, user complaints regarding bugs and technical glitches often arise, which can reduce service quality. This study aims to evaluate user sentiment toward mobile banking application features by comparing the performance of *Logistic Regression* and *Multinomial Naïve Bayes* algorithms. This sentiment analysis is important for understanding public perception objectively. User review data from four popular mobile banking applications (blu by BCA, BRImo, BNI Mobile Banking, and Livin' by Mandiri) were collected from the Google Play Store, with 50,000 data points per application, totaling 200,000 data points. The research process included data collection, preprocessing (*cleaning, case folding, tokenization, normalization, stopword removal, and stemming*), sentiment labeling using a *lexicon-based* approach, as well as model classification and evaluation. The research results showed that most user reviews were negative, particularly regarding login features, transactions, and verification, indicating dissatisfaction with technical reliability. In terms of performance comparison, the *Logistic Regression* model was far superior with an accuracy of 92-93%, compared to the *Multinomial Naïve Bayes* model, which only achieved 71-74%. This proves

that Logistic Regression is more effective in classifying the sentiment of mobile banking app reviews.

1. PENDAHULUAN

Dengan pesatnya perkembangan teknologi digital, aplikasi *mobile banking* telah menjadi kebutuhan utama dalam kehidupan masyarakat modern. Aplikasi ini memungkinkan pengguna melakukan berbagai transaksi keuangan secara cepat dan efisien, mulai dari transfer dana, pembayaran tagihan, hingga pengelolaan investasi tanpa harus datang ke bank secara langsung. Popularitas ini didukung oleh penetrasi *smartphone* dan akses internet yang luas, terutama di Indonesia [1]. Berdasarkan survei Populix, Bank BCA menempati posisi teratas dalam penggunaan aplikasi *mobile banking* dengan 60% pengguna, diikuti oleh Bank BRI (26%), Bank Mandiri (25%), dan Bank BNI (23%) [2], [3]. Aplikasi tersebut dapat diunduh melalui Google Play Store, yang juga menyediakan kolom ulasan pengguna. Ulasan ini menjadi sumber data yang penting karena sering digunakan sebagai tolak ukur dalam menilai suatu produk secara cepat dan efisien [4].

Namun, *bug* dan gangguan teknis pada aplikasi *mobile banking* sering kali menimbulkan keluhan dari pengguna. Hal ini dapat menurunkan kualitas layanan dan mengganggu kenyamanan pengguna, seperti yang terjadi pada aplikasi BNI *Mobile Banking* pada Februari 2023. Saat itu, banyak pengguna melaporkan kesulitan dalam melakukan verifikasi perubahan rekening akibat masalah teknis [5]. Keluhan-keluhan semacam ini mencerminkan ketidakpuasan pengguna terhadap kestabilan dan performa aplikasi, yang pada gilirannya dapat memengaruhi persepsi dan kepercayaan nasabah. Oleh karena itu, penting untuk melakukan analisis sentimen terhadap ulasan-ulasan tersebut untuk menggambarkan secara objektif persepsi publik terhadap fitur aplikasi *mobile banking*. Analisis sentimen merupakan proses otomatisasi untuk memahami, mengekstrak, dan mengklasifikasi opini publik dari data teks tidak terstruktur [4], serta dapat membantu dalam mengevaluasi aspek positif maupun negatif dari sebuah produk atau layanan digital.

Penelitian ini bertujuan untuk mengevaluasi sentimen pengguna terhadap fitur aplikasi *mobile banking* dengan membandingkan performa algoritma *Logistic Regression* dan *Multinomial Naïve Bayes*. Rumusan masalah dalam penelitian ini mencakup: (1) bagaimana hasil analisis sentimen dapat digunakan untuk mengevaluasi tingkat kepuasan pengguna terhadap kinerja fitur aplikasi *mobile banking*, dan (2) seberapa akurat metode *Logistic Regression* dan *Multinomial Naïve Bayes* dalam menentukan sentimen pengguna terhadap fitur aplikasi *mobile banking*. Penelitian ini dibatasi pada empat aplikasi *mobile banking* populer: blu by BCA, BRImo, BNI Mobile Banking, dan Livin' by Mandiri, dengan sumber data berupa ulasan pengguna dari Google Play Store. Fokus analisis diarahkan pada fitur-fitur utama seperti keamanan data, kecepatan transaksi, serta kemudahan akses informasi rekening. Sebanyak 50.000 data ulasan dari masing-masing aplikasi digunakan sebagai data penelitian [6], [7], [8], [9], [10].

2. TINJAUAN PUSTAKA

2.1 Analisis Sentimen

Analisis sentimen adalah metode untuk mengumpulkan informasi dari evaluasi terhadap masalah tertentu atau program aplikasi. Dalam kegiatan yang berfokus pada masalah, analisis sentimen menganalisis keyakinan, perasaan, dan gagasan seseorang terkait suatu topik tertentu. [11]. Algoritma analisis teks digunakan dalam proses analisis sentimen untuk memproses, memahami, dan mengklasifikasikan opini positif atau negatif dalam data teks. Analisis sentimen telah menjadi sangat populer akibat meningkatnya permintaan dari individu atau perusahaan untuk memahami sudut pandang seseorang terhadap suatu isu. Tergantung pada kumpulan data yang digunakan, analisis sentimen akan ditangani secara berbeda [12].

2.2 Mobile Banking

Layanan perbankan *mobile* sering diunduh melalui aplikasi, pemberitahuan SMS, atau komunikasi suara. Tujuan dari layanan ini adalah untuk memudahkan akses nasabah ke

layanan keuangan kapan saja dan dari mana saja. Peningkatan daya saing perbankan, perluasan basis nasabah, dan peningkatan kinerja keuangan bank sangat bergantung pada layanan perbankan *mobile* [13].

2.3 Python

Bahasa pemrograman tingkat tinggi *Python* diciptakan oleh *Guido Van Rossum* dan pertama kali dirilis pada tahun 1991. Karena *python* adalah bahasa pemrograman yang serbaguna yang dapat digunakan untuk perhitungan, visualisasi, dan pembelajaran mesin yang mendalam, bahasa ini juga menjadi sangat populer saat ini [14].

2.4 Lexicon Based

Teks berupa kalimat yang mengandung istilah positif dan negatif dari kamus *lexicon* digunakan sebagai dasar untuk prosedur ekstraksi nilai. Jumlah istilah dalam kamus *lexicon* bergantung pada jumlah kata dalam setiap teks atau kalimat. Hal ini dapat dipahami dengan menerapkan rumus di bawah ini [15].

$$S_{positive} = \sum_{i \in t}^{n} positive\ score_i \quad (1)$$

$$S_{negative} = \sum_{i \in t}^{n} negative\ score_i \quad (2)$$

Dapat dijelaskan bahwa nilai bobot kalimat ($S_{positive}$) adalah jumlah nilai polaritas dari n kata yang menyatakan opini positif. Nilai bobot kalimat ($S_{negative}$) adalah jumlah skor polaritas dari n kata yang beropini negatif. Atau, dapat dipastikan dengan membandingkan jenis nilai netral, negatif, dan positif.

2.5 Logistic Regression

Hubungan antara variabel yang berbeda, X dan D , yang memiliki karakteristik variabel *dependen dikotomis*, berfungsi sebagai contoh metodologi model matematika yang dikenal sebagai *Logistic Regression*. Sebuah variabel harus memiliki salah satu dari dua nilai potensial saat melakukan pengamatan atau pengukuran jika variabel tersebut merupakan variabel *dependen dikotomis* [16]. Dapat dirumuskan:

$$f(y_i) = \pi_i^{y_i} (1 - \pi_i)^{1-y_i} \quad (3)$$

dimana :

π_i = peluang kejadian ke- i

y_i = perubah acak ke- i yang terdiri dari 0 dan 1

2.6 Multinomial Naïve Bayes

Multinomial Naïve Bayes adalah algoritma *Naïve Bayes* lain yang berkinerja lebih baik dengan data hitungan atau frekuensi, seperti dalam pemrosesan bahasa alami, di mana kami memeriksa frekuensi kemunculan kata dalam dokumen [17]. Probabilitas suatu kelas dokumen ditentukan menggunakan metode *Naive Bayes* dialokasikan ke kelas dengan probabilitas tertinggi [18].

$$P(C|X) = \frac{P(X|C) \cdot P(C)}{P(X)} \quad (4)$$

Dimana :

$P(C|X)$: probabilitas kelas C setelah fitur X

$P(X|C)$: probabilitas fitur X diberi kelas C

$P(C)$: probabilitas prior dari kelas C

$P(X)$: probabilitas fitur X .

2.7 Confusion Matrix

Confusion Matrix adalah salah satu metode yang sering digunakan dalam *text mining*. Ketika menilai kinerja pengklasifikasi, metode ini berkinerja lebih baik. Untuk menilai performa *classifier* menggunakan *Confusion Matrix*, jumlah kelas X yang diklasifikasikan sebagai kelas Y perlu ditentukan [19].

Tabel 1. Confusion Matrix

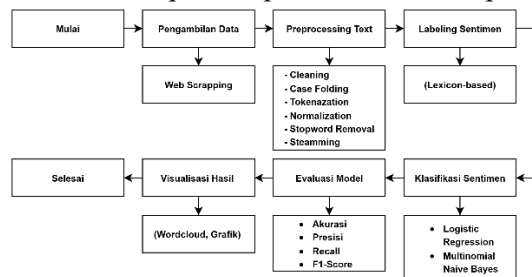
	Positive	Negative
Positive	TP	FP
Negative	FN	TN

Pada tabel 1, hasil positif dapat diprediksi secara akurat. Data yang seharusnya negatif tetapi diberi label positif dikenal sebagai FP. FN adalah data yang diklasifikasikan sebagai negatif meskipun seharusnya positif. Data yang diantisipasi dengan benar adalah TN.

3. METODE PENELITIAN

Penelitian ini menggunakan metodologi eksperimental dan pendekatan kuantitatif untuk mengevaluasi efektivitas algoritma klasifikasi *Multinomial Naïve Bayes* dan *Logistic Regression* dalam menganalisis sentimen pengguna aplikasi perbankan *mobile*. Pengumpulan data, prapemrosesan, penandaan,

penimbangan, klasifikasi, evaluasi model, dan visualisasi hasil merupakan langkah-langkah awal dalam proses penelitian bertahap ini.



Gambar 1. Metode Penelitian Sentimen

Pada gambar 1 di atas, penelitian ini dimulai dengan mengumpulkan informasi tentang ulasan pengguna dari ke-empat aplikasi perbankan yaitu blu by BCA, BRImo, BNI Mobile Banking, dan Livin' by Mandiri dari platform Google Play Store menggunakan teknik pengumpulan *web scrapping*. Dengan jumlah total ulasan yang berhasil dikumpulkan sebanyak 200.000 data. Data kemudian dibersihkan dari duplikasi dan *noise* sebelum diproses lebih lanjut. Data yang telah dikumpulkan kemudian dilakukan tahap *preprocessing* untuk mengubah teks mentah menjadi format yang siap dianalisis [20]. Tahapan *preprocessing* terdiri dari:

- 1) *Cleaning* adalah proses menghilangkan karakter khusus, tanda baca, dan simbol.
- 2) Mengubah semua huruf menjadi huruf kecil dikenal sebagai *case folding*.
- 3) Tokenisasi adalah proses memisahkan kata-kata menjadi kalimat.
- 4) Normalisasi adalah proses mengubah bentuk kata slang dan bentuk kata non-standar lainnya menjadi bentuk standar.
- 5) *Stopword Removal* adalah proses menghilangkan istilah yang sering digunakan namun tidak memberikan kontribusi signifikan bagi penelitian.
- 6) *Stemming* adalah proses menerapkan metode Nazief & Adriani untuk mengembalikan kata ke bentuk dasarnya.

Dengan menggunakan metode berbasis *lexicon based*, algoritma pelabelan sentimen mencocokkan kata-kata dalam ulasan dengan kamus sentimen bahasa Indonesia. Hasil pelabelan digunakan untuk mengidentifikasi kelas data, dan setiap kata diberi skor sentimen

(positif atau negatif) [21]. Setelah pelabelan, ulasan yang telah diproses dikonversi ke bentuk numerik menggunakan metode *Term Frequency-Inverse Document Frequency* (TF-IDF). TF-IDF membantu memberikan bobot yang sesuai pada kata-kata penting dan relevan dalam dokumen [22].

Data yang telah diberi bobot diklasifikasikan menggunakan dua algoritma, yaitu *Logistic Regression* sebagai model utama dan *Multinomial Naive Bayes* sebagai pembanding. Data dibagi menjadi 80% untuk pelatihan dan 20% untuk pengujian guna mengevaluasi efektivitas dan akurasi masing-masing model dalam klasifikasi sentimen. Setelah proses klasifikasi selesai, evaluasi model dilakukan menggunakan *Confusion Matrix* dan beberapa metrik performa untuk menilai kualitas hasil. Metrik yang digunakan meliputi akurasi (persentase prediksi benar), presisi (kemampuan mengklasifikasikan data positif secara tepat), *recall* (kemampuan mendeteksi seluruh data positif), dan *F1-Score* (rata-rata harmonik presisi dan recall). Untuk memudahkan interpretasi, hasil dievaluasi secara visual melalui *wordcloud* dan grafik batang yang menampilkan frekuensi kata dominan serta proporsi sentimen pada masing-masing aplikasi.

4. HASIL DAN PEMBAHASAN

Penyajiaan hasil dari proses analisis sentimen yang dilakukan terhadap ulasan pengguna dari aplikasi mobile banking. Proses analisis mencakup pelabelan data menggunakan pendekatan *lexicon-based*, klasifikasi menggunakan algoritma *Logistic Regression*, serta evaluasi performa model menggunakan *Confusion Matrix*. Visualisasi dalam bentuk *wordcloud* dan grafik juga digunakan untuk memperkuat interpretasi data.

4.1 Crawling Data

Halaman aplikasi Google Play Store digunakan untuk melakukan proses perayapan data dengan menggunakan teknologi *web scraping Python*. Gunakan ID aplikasi (parameter id pada URL) sebagai tujuan utama pengambilan data untuk aplikasi *mobile banking* yang dikenal, data disimpan dalam bentuk format csv agar lebih mudah di proses lebih lanjut.

The screenshot shows a list of URLs and their corresponding content, likely from a scraping process. The URLs are listed in a table with columns for 'URL' and 'Content'. The content is a list of URLs, some of which are repeated. The table is titled 'Hasil Proses Scrapping'.

Gambar 2. Hasil Proses Scrapping

Pada gambar 2, memperlihatkan hasil dari proses scrapping ulasan pengguna pada aplikasi *mobile banking* yang ditargetkan. Setelah data berhasil dikumpulkan melalui proses ini, dilakukan rekapitulasi jumlah data yang diperoleh dari masing-masing aplikasi. Rincian jumlah data dari setiap aplikasi yang berhasil diambil dapat dilihat pada Tabel 1 berikut.

Tabel 2. Rincian pengambilan data

No	Nama Aplikasi	Jumlah Data
1	BRImo	50.000
2	blu by BCA	50.000
3	BNI Mobile	50.000
4	Livin' by Mandiri	50.000
Total		200.000

Tabel.2 di atas menampilkan volume data secara keseluruhan telah terkumpul 200.000 data. Data ini dapat dianalisis untuk berbagai hal seperti menilai sentimen pengguna dan kinerja aplikasi.

4.2 Preprocessing Text

4.2.1 Cleaning Text

Langkah pembersihan data dalam NLP untuk menstandarisasi teks. Dengan menghapus elemen yang tidak penting. Seperti tautan, angka, simbol, tagar, dan sebutan, agar data menjadi lebih sederhana dan siap untuk dianalisis sentimen.

```
def cleaningText(text):
    text = re.sub(r'[A-Za-z0-9]+', '', text)
    text = re.sub(r'RT|S+', '', text)
    text = re.sub(r'http|S+', '', text)
    text = re.sub(r'[0-9]+', '', text)
    text = re.sub(r'[^\w\s]', '', text)
    text = text.replace('\n', '')
    text = text.translate(str.maketrans('', '', string.punctuation))
    text = text.strip(' ')
    return text
```

Gambar 3. Metode Cleaning Text

Pada gambar 3, proses pembersihan ini mencakup penghapusan mention, hashtag, simbol tautan (URL), angka, karakter non-

alfanumerik, dan tanda baca yang tidak relevan. Selain itu, fungsi ini juga menghapus karakter *newline* serta spasi di awal dan akhir kalimat.

	content	text_clean
0	setelah di-update malah nggak bisa berjalan se...	setelah diupdate malah nggak bisa berjalan sep...
1	Livin mandiri lagi kenapa ini ya???? Mau buka ...	Livin mandiri lagi kenapa ini ya Mau buka apli...
2	Bener kata orang ² , apk ini sering ada kendala ...	Bener kata orang ² apk ini sering ada kendala t...

Gambar 4. Hasil Cleaning Data

Dalam gambar 4, dapat di lihat bahwa kolom *content* dan *text_clean* memiliki perbedaan seperti “Livin mandiri lagi kenapa ini ya???? Mau buka...” lalu mengalami perubahan menjadi “Livin mandiri lagi kenapa ini ya Mau buka apli...”.

4.2.2 Case Folding

Case Folding sangat penting dalam analisis teks karena membantu mencegah ambiguitas makna yang disebabkan oleh huruf besar. Hasilnya, proses selanjutnya menjadi lebih tepat dan dapat diandalkan.

```
def casefoldingText(text):
    text = text.lower()
    return text
```

Gambar 5. Metode Case Folding

Pada gambar 5, Tersebut kode `text.lower()` digunakan untuk mengubah seluruh huruf dalam teks menjadi huruf kecil (*lowercase*).

	content	text_clean	text_casefoldingText
0	setelah di-update malah nggak bisa berjalan se...	setelah diupdate malah nggak bisa berjalan sep...	setelah diupdate malah nggak bisa berjalan sep...
1	Livin mandiri lagi kenapa ini ya???? Mau buka ...	Livin mandiri lagi kenapa ini ya Mau buka apli...	livin mandiri lagi kenapa ini ya mau buka apli...
2	Bener kata orang ² , apk ini sering ada kendala ...	Bener kata orang ² apk ini sering ada kendala t...	bener kata orang ² apk ini sering ada kendala t...

Gambar 6. Hasil Proses Case Folding

Dalam gambar 6, dapat di lihat bahwa kolom *text_clean* dan *text_casefoldingText* memiliki perbedaan seperti “Livin mandiri lagi kenapa ini ya Mau buka apli...” lalu mengalami perubahan menjadi “livin mandiri lagi kenapa ini ya mau buka apli...”.

4.2.3 Slangword

Slangword menggunakan glosarium istilah-istilah gaul tertentu, yang biasanya digunakan dalam konteks informal dan santai, atau dengan menerjemahkan teks ke dalam bahasa yang lebih diterima secara luas.

```
def fix_slangwords(text):
    words = text.split()
    fixed_words = []

    for word in words:
        if word.lower() in slangwords:
            fixed_words.append(slangwords[word.lower()])
        else:
            fixed_words.append(word)

    fixed_text = ' '.join(fixed_words)
    return fixed_text
```

Gambar 7. Metode Slangword

Pada gambar 7, setiap kata dalam teks diperiksa melalui perulangan. Jika kata (dalam huruf kecil) terdapat dalam *dictionary slangwords*, maka kata tersebut diganti dengan padanannya. Jika tidak, kata asli tetap digunakan.

text_clean	text_casefoldingText	text_slangwords
setelah diupdate malah nggak bisa berjalan sep...	setelah diupdate malah nggak bisa berjalan sep...	setelah diupdate bahkan tidak bisa berjalan se...
livin mandiri lagi kenapa ini ya mau buka apli...	livin mandiri lagi kenapa ini ya mau buka apli...	livin mandiri lagi kenapa ini iya mau buka apli...
Bener kata orang ² apk ini sering ada kendala t...	bener kata orang ² apk ini sering ada kendala t...	benar kata orang ² apk ini sering ada kendala t...

Gambar 8. Hasil Proses Slangword

Dalam gambar 8, dapat di lihat bahwa kolom *text_casefoldingText* dan *text_Slangword* memiliki perbedaan seperti “livin mandiri lagi kenapa ini ya mau buka apli...” lalu mengalami perubahan menjadi “livin mandiri lagi kenapa ini iya mau buka apli...”.

4.2.4 Tokenazition

Metode ini memecah tulisan yang panjang menjadi potongan-potongan yang dapat dicerna, seperti kata atau frasa. Hal ini memudahkan proses pembuatan model atau penggunaan token-token ini untuk analisis teks tambahan.

```
def tokenizingText(text):
    text = word_tokenize(text)
    return text
```

Gambar 9. Metode Tokenazition

Proses *tokenazition* pada gambar 9, dilakukan menggunakan fungsi *word_tokenize()* dari pustaka NLP seperti NLTK. Teks yang telah di

tokenisasi kemudian dikembalikan sebagai output fungsi dalam bentuk list kata.

text_casefoldingText	text_slangwords	text_tokenizingText
setelah diupdate malah nggak bisa berjalan sep...	setelah diupdate bahkan tidak bisa berjalan se...	[setelah, diupdate, bahkan, tidak, bisa, berja...
livin mandiri lagi kenapa ini ya mau buka apli...	livin mandiri lagi kenapa ini iya mau buka apli...	[livin, mandiri, lagi, kenapa, ini, iya, mau, ...
bener kata orang ² apk ini sering ada kendala t...	benar kata orang ² apk ini sering ada kendala t...	[benar, kata, orang ² , apk, ini, sering, ada, k...

Gambar 10. Hasil Proses Tokenazition

Dalam gambar 10, dapat di lihat bahwa kolom *text_Slangword* dan *text_tokenizingText* memiliki perbedaan seperti “livin mandiri lagi kenapa ini iya mau buka apli...” lalu mengalami perubahan menjadi “[livin, mandiri, lagi, kenapa, ini, mau, buka, ...]”.

4.2.5 Stopword Removal

Teknik *Stopword Removal* dilakukan untuk kata penghubung, kata ganti, dan kata depan yan sebagian besar kata tersebut dihilangkan.

```
def filteringText(text):
    listStopwords = set(stopwords.words('indonesian'))
    listStopwords1 = set(stopwords.words('english'))
    listStopwords.update(listStopwords1)
    listStopwords.update(['iya', 'yaa', 'gak', 'nya', 'na', 'sih', 'ku', 'di'])
    filtered = []
    for txt in text:
        if txt not in listStopwords:
            filtered.append(txt)
    text = filtered
    return text
```

Gambar 11. Metode Stopword Removal

Pada gambar 11, memiliki fungsi *filteringText* berfungsi untuk menghapus kata-kata yang tidak memiliki makna penting (*stopwords*) dari sebuah teks. Fungsi ini menggabungkan daftar pustaka *stopwords* dari NLTK, lalu menambahkan beberapa kata tidak baku lainnya seperti “yaa”, “gak”, “sih”, dan sebagainya ke dalam daftar tersebut. Selanjutnya, fungsi memeriksa setiap kata dalam teks dan hanya menyimpan kata-kata yang tidak termasuk dalam *stopwords* ke dalam list baru. Hasilnya berupa teks yang telah difilter dari kata-kata umum yang tidak relevan.

text_slangwords	text_tokenizingText	text_stopword
setelah diupdate bahkan tidak bisa berjalan se...	[setelah, diupdate, bahkan, tidak, bisa, berja...	[diupdate, berjalan, aplikasi, kompatibel, dg,...
livin mandiri lagi kenapa ini iya mau buka apl...	[livin, mandiri, lagi, kenapa, ini, iya, mau, ...	[livin, mandiri, buka, aplikasi, loading, lama...
benar kata orang ² apk ini sering ada kendala t...	[benar, kata, orang ² , apk, ini, sering, ada, k...	[orang ² , apk, kendala, transaksi, mengalami, t...

Gambar 12. Hasil Proses Slangword

Dalam gambar 12, dapat di lihat bahwa kolom `text_tokenizingText` dan `text_stopword` memiliki perbedaan seperti “[livin, mandiri, lagi, kenapa, ini, mau, buka, ...]” lalu mengalami perubahan menjadi “[livin, mandiri, buka, aplikasi, loading, lama...]”.

4.2.6 Stemming

Dengan menghilangkan imbuhan atau akhiran, proses ini berusaha mengembalikan varian kata ke bentuk dasar atau kata dasarnya, sehingga meningkatkan akurasi dan keefektifan analisis teks.

```
def stemmingText(text):
    factory = StemmerFactory()
    stemmer = factory.create_stemmer()
    words = text.split()
    stemmed_words = [stemmer.stem(word) for word in words]
    stemmed_text = ' '.join(stemmed_words)
    return stemmed_text
```

Gambar 13. Metode Stemming

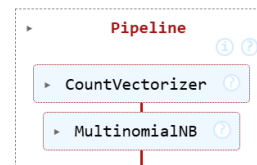
Pada gambar 13, fungsi ini menggunakan pustaka *stemmer* dari Sastrawi, dengan membuat objek *stemmer* melalui *StemmerFactory*. Teks input terlebih dahulu dipecah menjadi kata-kata menggunakan *split()*, lalu setiap kata diolah agar menjadi bentuk dasarnya menggunakan metode *stem()*.

text_tokenizingText	text_stopword	text_akhir
[setelah, diupdate, bahkan, tidak, bisa, berja...	[diupdate, berjalan, aplikasi, kompatibel, dg,...	diupdate berjalan aplikasi kompatibel dg handp...
[livin, mandiri, lagi, kenapa, ini, iya, mau, ...	[livin, mandiri, buka, aplikasi, loading, lama...	livin mandiri buka aplikasi loading lamaaaa pd...
[benar, kata, orang ² , apk, ini, sering, ada, k...	[orang ² , apk, kendala, transaksi, mengalami, t...	orang ² apk kendala transaksi mengalami transak...

Gambar 14. Hasil Proses Stemming

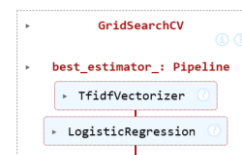
Pada gambar 14, dapat di lihat bahwa kolom `ext_stopword` dan `text_akhir` memiliki perbedaan seperti “[livin, mandiri, buka, aplikasi, loading, lama...]” lalu mengalami perubahan menjadi “[livin mandiri buka aplikasi loading lama...]”.

4.3 Klasifikasi Text



Gambar 15. Klasifikasi Multinomial Naive Bayes

Sesuai dengan gambar 15, tersebut menunjukkan *pipeline* untuk pemrosesan teks dan klasifikasi menggunakan algoritma *Multinomial Naive Bayes*. *Pipeline* ini terdiri dari dua komponen, yaitu *CountVectorizer* untuk mengubah teks menjadi data numerik, dan *Multinomial Naive Bayes* sebagai model klasifikasi.



Gambar 16. Klasifikasi Logistic Regression

Sesuai dengan gambar 16, tersebut menunjukkan hasil *GridSearchCV* yang menghasilkan *pipeline* terbaik untuk klasifikasi teks, terdiri dari *TfidfVectorizer* dan *Logistic Regression*. *TfidfVectorizer* mengubah teks menjadi data numerik berbobot, lalu diklasifikasikan oleh *Logistic Regression*. Dengan *GridSearchCV*, kombinasi parameter terbaik dipilih secara otomatis untuk meningkatkan akurasi model.

4.3.1 Pengujian Aplikasi blu by BCA

Logistic Regression						Naive Bayes					
Akurasi: 0.93						Akurasi: 0.74					
Laporan Klasifikasi:						**Laporan Klasifikasi:**					
precision	recall	f1-score	support			precision	recall	f1-score	support		
Negatif	0.97	0.93	0.95	3189		Negatif	0.72	0.89	0.80	3189	
Netral	0.88	0.94	0.91	2664		Netral	0.78	0.61	0.69	2664	
Positif	0.93	0.92	0.92	2295		Positif	0.76	0.70	0.72	2295	
accuracy			0.93	8148		accuracy			0.74	8148	
macro avg	0.93	0.93	0.93	8148		macro avg	0.75	0.73	0.74	8148	
weighted avg	0.93	0.93	0.93	8148		weighted avg	0.75	0.74	0.74	8148	

Gambar 17. Hasil Klasifikasi blu by BCA

Dapat dilihat pada gambar 17, berdasarkan ulasan pengguna aplikasi Blu by BCA, model *Logistic Regression* mengungguli *Multinomial Naive Bayes* dalam hal akurasi. Dibandingkan dengan *Naive Bayes*, yang hanya memprediksi sentimen 74% dari waktu, *Logistic Regression* memiliki akurasi 93%.

4.3.2 Pengujian Aplikasi BNI Mobile

Logistic Regression					Naive Bayes				
Akurasi: 0.92					Akurasi: 0.72				
Laporan Klasifikasi:					**Laporan Klasifikasi:**				
precision	recall	f1-score	support		precision	recall	f1-score	support	
Negatif	0.96	0.94	0.95	2887	Negatif	0.66	0.96	0.78	2887
Netral	0.87	0.94	0.90	2225	Netral	0.81	0.55	0.66	2225
Positif	0.93	0.86	0.89	1408	Positif	0.86	0.48	0.62	1408
accuracy	0.92 6520				accuracy	0.72 6520			
macro avg	0.92	0.91	0.92	6520	macro avg	0.78	0.66	0.68	6520
weighted avg	0.93	0.92	0.92	6520	weighted avg	0.75	0.72	0.70	6520

Gambar 18. Hasil Klasifikasi BNI Mobile

Dapat dilihat pada gambar 18, berdasarkan penilaian pengguna aplikasi BNI Mobile, model *Logistic Regression* mengungguli model *Multinomial Naïve Bayes* dengan selisih yang cukup signifikan. *Multinomial Naïve Bayes* hanya dapat mengklasifikasikan sentimen dengan akurasi 72%, sedangkan *Logistic Regression* dapat melakukannya hingga 92%.

4.3.3 Pengujian Aplikasi BRImo

Logistic Regression					Naive Bayes				
Akurasi: 0.93					Akurasi: 0.71				
Laporan Klasifikasi:					**Laporan Klasifikasi:**				
precision	recall	f1-score	support		precision	recall	f1-score	support	
Negatif	0.95	0.93	0.94	3335	Negatif	0.67	0.90	0.77	3335
Netral	0.88	0.91	0.89	2182	Netral	0.72	0.29	0.41	2182
Positif	0.95	0.95	0.95	3295	Positif	0.76	0.78	0.77	3295
accuracy	0.93 8812				accuracy	0.71 8812			
macro avg	0.93	0.93	0.93	8812	macro avg	0.71	0.66	0.65	8812
weighted avg	0.93	0.93	0.93	8812	weighted avg	0.71	0.71	0.68	8812

Gambar 19. Hasil Klasifikasi BRImo

Dapat dilihat pada gambar 19, penelitian sentimen pengguna aplikasi BRImo mengungkapkan bahwa model *Logistic Regression* mengungguli *Multinomial Naïve Bayes* dengan selisih yang signifikan. Dibandingkan dengan *Multinomial Naïve Bayes* yang hanya mencapai akurasi 71%, *Logistic Regression* mencapai 93%.

4.3.4 Pengujian Aplikasi Livin' by Mandiri

Logistic Regression					Naive Bayes				
Akurasi: 0.93					Akurasi: 0.71				
Laporan Klasifikasi:					**Laporan Klasifikasi:**				
precision	recall	f1-score	support		precision	recall	f1-score	support	
Negatif	0.96	0.94	0.95	3363	Negatif	0.66	0.91	0.77	3363
Netral	0.88	0.91	0.89	2310	Netral	0.74	0.46	0.57	2310
Positif	0.93	0.93	0.93	2418	Positif	0.79	0.66	0.72	2418
accuracy	0.93 8091				accuracy	0.71 8091			
macro avg	0.93	0.93	0.93	8091	macro avg	0.73	0.68	0.68	8091
weighted avg	0.93	0.93	0.93	8091	weighted avg	0.72	0.71	0.70	8091

Gambar 20. Hasil Klasifikasi Livin' by Mandiri

Dapat dilihat pada gambar 20, penelitian sentimen pengguna aplikasi Livin' by Mandiri

mengungkapkan bahwa model *Logistic Regression* mengungguli *Multinomial Naïve Bayes* dengan selisih yang signifikan. Dibandingkan dengan *Multinomial Naïve Bayes* yang hanya mencapai akurasi 71%, *Logistic Regression* mencapai 93%.

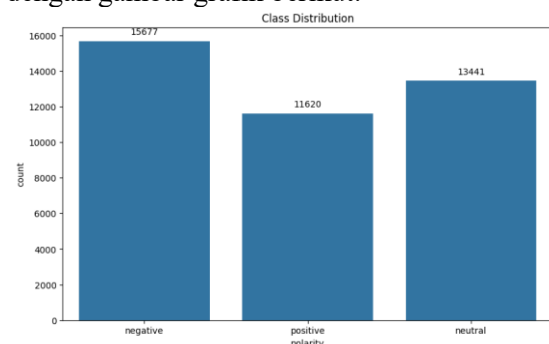
4.4 Visualisasi Data

4.4.1 Visualisasi Aplikasi blu by BCA



Gambar 21. World Cloud blu by BCA

Secara garis besar pada gambar 21 di atas, tersebut condong ke arah keluhan atau sentimen negatif pengguna terhadap aplikasi blu by BCA. Hal ini terlihat dari dominasi kata-kata seperti "login", "update", "ribet", "susah", "tolong", dan "verifikasi", yang menunjukkan banyaknya masalah teknis atau kesulitan dalam penggunaan aplikasi. Meskipun ada kata positif seperti "terima kasih", porsinya jauh lebih kecil dibanding kata-kata bermuatan keluhan. Dapat dibuktikan dengan gambar grafik berikut.



Gambar 22. Class Distribusi blu by BCA

Pada Gambar 22 di atas, distribusi kelas sentimen dari ulasan pengguna terhadap aplikasi, yang dipisahkan menjadi tiga kategori-negatif, positif, dan netral-digambarkan dalam grafik di atas. Dengan 15.677 ulasan, kategori negatif unggul, diikuti oleh kategori netral dengan 13.441 ulasan dan kategori baik dengan 11.620 ulasan. Distribusi ini menunjukkan ketidakpuasan, bahwa

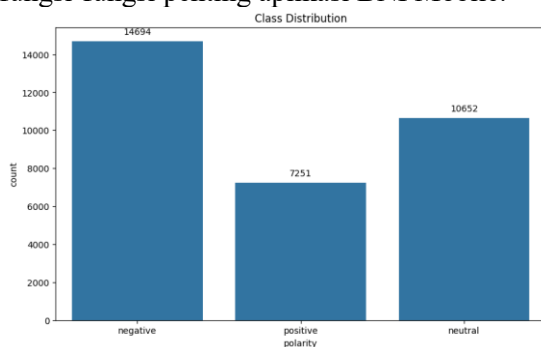
persepsi aplikasi ini masih memerlukan perbaikan yang berkaitan dengan pengalaman pengguna.

4.4.2 Visualisasi Aplikasi BNI Mobile



Gambar 23. World Cloud BNI Mobile

Pada gambar 23 di atas, kata-kata seperti “login”, “kode OTP”, “aplikasi”, ‘update’, dan “transaksi” sangat dominan dan menunjukkan bahwa konsumen terutama peduli dengan transaksi, verifikasi OTP, dan prosedur login aplikasi. Selain itu, penggunaan istilah seperti “error”, ‘rumit’, dan “sulit” menunjukkan bahwa keluhan yang disampaikan adalah mengenai kesulitan teknis. Hasilnya, kumpulan kata tersebut menggambarkan banyaknya pengguna yang mengalami kesulitan dalam mengakses dan menggunakan fungsi-fungsi penting aplikasi BNI Mobile.



Gambar 24. Class Distribusi BNI Mobile

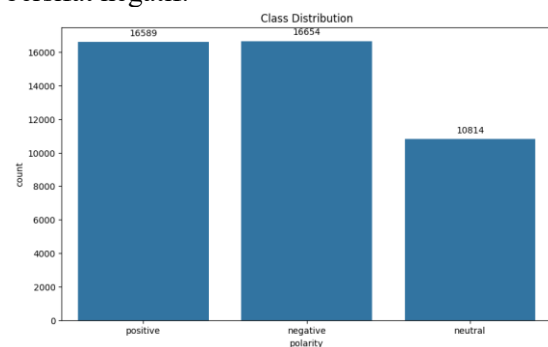
Gambar 24 di atas menyatakan bahwa, dengan total 14.694 ulasan grafik distribusi sentimen untuk aplikasi BNI Mobile menunjukkan bahwa sebagian besar ulasan pengguna bersifat negatif. Ulasan dengan sentimen positif hanya berjumlah 7.251, sementara ulasan dengan sentimen netral menempati peringkat kedua dengan 10.652 ulasan. Perbedaan ini menunjukkan bahwa lebih banyak pengguna tidak puas dengan aplikasi BNI Mobile.

4.4.3 Visualisasi Aplikasi BRImo



Gambar 25. World Cloud BRImo

Istilah “aplikasi,” “brimo,” dan ‘transaksi’ adalah yang paling sering digunakan dalam ulasan pengguna, seperti yang terlihat pada Gambar 25 di atas. Masalah terkait transfer, pembaruan, dan login sering disebutkan. Penggunaan kata-kata seperti “tolong,” “kesalahan,” dan “perbaiki” yang sering muncul menunjukkan bahwa masalah teknis menjadi perhatian utama. Hal ini sejalan dengan studi sebelumnya yang menemukan bahwa sebagian besar ulasan bersifat negatif.



Gambar 26. Class Distribusi BRImo

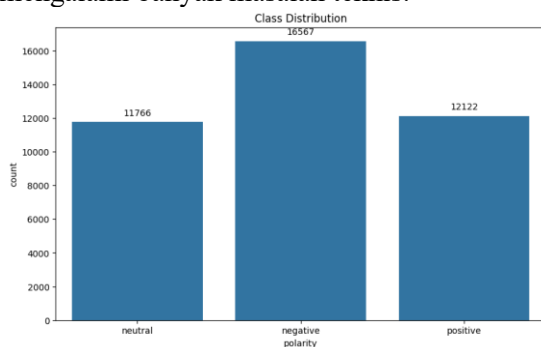
Distribusi sentimen pengguna terhadap aplikasi BRIImo ditampilkan dalam grafik pada gambar 26 di atas. Sentimen netral lebih rendah (10.814 ulasan), sementara sentimen negatif sedikit lebih tinggi (16.654 ulasan) dibandingkan sentimen positif (16.589 ulasan). Data menunjukkan bahwa meskipun sentimen netral lebih rendah, ulasan tentang aplikasi BRIImo didominasi oleh sentimen positif dan negatif dalam proporsi yang hampir sama. Hal ini menunjukkan bahwa meskipun pengalaman pengguna berbeda-beda, banyak yang masih mengalami masalah.

4.4.4 Visualisasi Aplikasi Livin' by Mandiri



Gambar 27. World Cloud Livin' by Mandiri

Istilah-istilah yang paling sering muncul dalam ulasan pengguna aplikasi Livin' by Mandiri ditampilkan dalam *word cloud* pada gambar 27 di atas. Kata-kata seperti “aplikasi,” “transaksi,” “pembaruan,” “login,” dan ‘transfer’ sering muncul dan menunjukkan bahwa pengguna fokus pada fitur utama aplikasi. Kata-kata seperti “gk,” “kesalahan,” “rumit,” dan “tolong diperbaiki” juga terdapat, menunjukkan bahwa pengguna masih mengalami banyak masalah teknis.



Gambar 28. Class Distribusi Livin' by Mandiri

Pada gambar 28 di atas, distribusi sentimen dalam ulasan aplikasi Livin' by Mandiri ditampilkan dalam grafik. Dengan 16.567 ulasan, sentimen negatif mendominasi, diikuti oleh sentimen positif (12.122 ulasan) dan sentimen netral (11.766 ulasan). Fakta bahwa sebagian besar ulasan untuk aplikasi Livin' by Mandiri bersifat negatif menunjukkan bahwa banyak pelanggan masih mengalami masalah dengan aplikasi tersebut.

5. KESIMPULAN

- a. Analisis sentimen terhadap 200.000 ulasan dari empat aplikasi *mobile banking* menunjukkan bahwa sebagian besar pengguna masih merasa tidak puas terhadap performa aplikasi, terutama

terkait fitur login, transaksi, dan verifikasi. Sentimen negatif mendominasi ulasan, seperti terlihat pada aplikasi blu by BCA, BNI Mobile, dan Livin' by Mandiri. Hal ini mencerminkan bahwa fitur-fitur utama belum sepenuhnya memenuhi ekspektasi pengguna dan memerlukan peningkatan dalam aspek keandalan teknis dan kemudahan penggunaan.

- b. Dari hasil pengujian, metode *Logistic Regression* menunjukkan tingkat akurasi yang lebih tinggi dibandingkan *Multinomial Naïve Bayes* dalam klasifikasi sentimen. *Logistic Regression* mencapai akurasi 92–93%, sedangkan *Multinomial Naïve Bayes* hanya 71–74%. Ini membuktikan bahwa *Logistic Regression* lebih efektif dan dapat diandalkan untuk mengidentifikasi sentimen pengguna secara akurat dalam konteks ulasan aplikasi *mobile banking*.
- c. Penggunaan metode *lexicon-based* untuk pelabelan sentimen dan pembobotan TF-IDF terbukti efektif dalam mendeteksi dan mengklasifikasi opini pengguna secara otomatis. Proses *preprocessing* yang lengkap (*cleaning, case folding, tokenization, slangword normalization, stopword removal, dan stemming*) turut meningkatkan keakuratan hasil klasifikasi dan visualisasi data, seperti *word cloud* dan grafik distribusi sentimen.

UCAPAN TERIMA KASIH

Kami ingin mengucapkan terima kasih kepada seluruh pihak di Universitas Islam Majapahit atas kontribusi mereka dalam penelitian ini, khususnya para dosen program studi informatika. Selain itu, kami juga berterima kasih karena penelitian ini diterbitkan oleh Jurnal Informatika dan Teknik Elektro Terapan (JITET).

DAFTAR PUSTAKA

- [1] N. Fibriyanti Arminda, N. Sulistiyowati, and T. Nur Padilah, "Implementasi Algoritma Multinomial Naive Bayes Pada Analisis Sentimen Terhadap Ulasan Pengguna Aplikasi Brimo," *JATI (Jurnal Mhs. Tek. Inform.*, vol. 7, no. 3, pp. 1817–1822, 2023, doi: 10.36040/jati.v7i3.7012.
- [2] D. Angelia, "Aplikasi Mobile Banking Paling Banyak Digunakan Masyarakat Indonesia 2022." [Online]. Available:

- <https://goodstats.id/article/aplikasi-mobile-banking-paling-banyak-digunakan-masyarakat-indonesia-2022-Vb18i>
- [3] L. Sitohang and Agustiawan, "Upaya Meningkatkan Pemahaman Dan Keamanan Dalam Penggunaan Aplikasi Brimo Kepada Nasabah Di Pt Bank Rakyat Indonesia (Bri) Di Kantor Kcp Ahmad Yani," *SEMAR J. Sos. dan Pengabd. Masy.*, vol. 1, no. 01, pp. 36–44, 2023, doi: 10.59966/semar.v1i01.15.
 - [4] A. Rahman, E. Utami, and S. Sudarmawan, "Sentimen Analisis Terhadap Aplikasi pada Google Playstore Menggunakan Algoritma Naïve Bayes dan Algoritma Genetika," *J. Komtika (Komputasi dan Inform.)*, vol. 5, no. 1, pp. 60–71, 2021, doi: 10.31603/komtika.v5i1.5188.
 - [5] A. Dwi, "Selain BSI, Ini 5 Kasus Gangguan Layanan M-Banking yang Sempat Ramai Disorot," *Tempo.co*. Accessed: Feb. 20, 2025. [Online]. Available: <https://www.tempo.co/ekonomi/selain-bsi-ini-5-kasus-gangguan-layanan-m-banking-yang-sempat-ramai-disorot--189107>
 - [6] R. Yusuf, K. Bahumatra, N. Komaria, E. A. Aqma, and L. Cahyani, "Analisis Sentimen Terhadap Aplikasi Google Meet Berdasarkan Komentar Pengguna Menggunakan Metode Logistic Regression," vol. 11, no. 1, pp. 53–64, 2024.
 - [7] N. A. Amalia, I. T. Utami, and Y. Wilandari, "Analisis Sentimen Kebijakan Penyelenggara Sistem Elektronik Lingkup Privat Menggunakan Penalized Logistic Regression Dan Support Vector Machine," *J. Gaussian*, vol. 12, no. 4, pp. 560–569, 2024, doi: 10.14710/j.gauss.12.4.560-569.
 - [8] Sutarman, R. Siringoringo, D. Arisandi, E. Kurniawan, and E. B. Nababan, "Model Klasifikasi Dengan Logistic Regression Dan Recursive Feature Elimination Pada Data Tidak Seimbang," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 11, no. 4, pp. 735–742, 2024, doi: 10.25126/jtiik.1148198.
 - [9] F. Rizal, A. Wijaya, and F. Hasyim, "Analisis Sentimen Masyarakat Indonesia Terhadap Aplikasi TikTok Menggunakan Algoritma Logistic Regression," *J. homepage AKIRATECH J. Comput. Electr. Eng.*, vol. 1, no. 2, pp. 57–65, 2024, [Online]. Available: <https://journal.ajbnews.com/index.php/akiratech>
 - [10] A. G. Budianto, A. Trisno, E. Suryo, and G. Rudi, "Perbandingan Performa Algoritma Support Vector Machine (SVM) dan Logistic Regression untuk Analisis Sentimen Pengguna Aplikasi Retail di Android," vol. 10, no. November, pp. 1–10, 2024, doi: 10.34128/jsi.v10i2.911.
 - [11] F. Zamachsari, G. Vangeran Saragih, and W. Gata, "Terakreditasi SINTA Peringkat 2 Analisis Sentimen Pemindahan Ibu Kota Negara dengan Feature Selection Algoritma Naive Bayes dan Support Vector Machine," *Masa Berlaku Mulai*, vol. 4, no. 3, pp. 504–512, 2020.
 - [12] W. Widayat, "Analisis Sentimen Movie Review menggunakan Word2Vec dan metode LSTM Deep Learning," *J. Media Inform. Budidarma*, vol. 5, no. 3, p. 1018, 2021, doi: 10.30865/mib.v5i3.3111.
 - [13] M. Ayuningtyas and L. Sufina, "Pengaruh Penggunaan Mobile Banking, Internet Banking, dan Atm terhadap Kinerja Keuangan Perbankan (Studi Kasus Sektor Bank Konvensional yang Terdaftar di Bursa Efek Indonesia) Tahun 2017- 2021," *J. Keuang. dan Perbank.*, vol. 19, no. 2, pp. 119–130, 2023, doi: 10.35384/jkp.v19i2.394.
 - [14] M. R. S. Alfarizi, M. Z. Al-farish, M. Taufiqurrahman, G. Ardiansah, and M. Elgar, "Penggunaan Python Sebagai Bahasa Pemrograman untuk Machine Learning dan Deep Learning," *Karya Ilm. Mhs. Bertauhid (KARIMAH TAUHID)*, vol. 2, no. 1, pp. 1–6, 2023.
 - [15] A. R. Ismail and Raden Bagus Fajriya Hakim, "Implementasi Lexicon Based Untuk Analisis Sentimen Dalam Menentukan Rekomendasi Pantai Di DI Yogyakarta Berdasarkan Data Twitter," *Emerg. Stat. Data Sci. J.*, vol. 1, no. 1, pp. 37–46, 2023, doi: 10.20885/esds.vol1.iss.1.art5.
 - [16] E. Darmaja, V. C. Mawardi, and N. J. Perdana, "Review Sentimen Analisis Aplikasi Sosial Media Di Google Playstore Menggunakan Metode Logistic Regression," *Pros. Serina*, pp. 513–520, 2021, [Online]. Available: <https://journal.untar.ac.id/index.php/PSERINA/article/view/17504%0Ahttps://journal.untar.ac.id/index.php/PSERINA/article/download/17504/9467>
 - [17] K. Kevin, M. Enjeli, and A. Wijaya, "Analisis Sentimen Penggunaan Aplikasi Kinemaster Menggunakan Metode Naive Bayes," *J. Ilm. Comput. Sci.*, vol. 2, no. 2, pp. 89–98, 2024, doi: 10.58602/jics.v2i2.24.
 - [18] D. Prasetia, N. Rahaningsih, R. D. Dana, and C. L. Rohmat, "Analisis Sentimen Pengguna Aplikasi Mybluebird Dengan

- Algoritma Naïve Bayes Di Playstore,” *J. Inform. dan Tek. Elektro Terap.*, vol. 13, no. 1, 2025, doi: 10.23960/jitet.v13i1.5687.
- [19] P. R. Alvita Wagiswari D, I. Susilawati, and A. Witanti, “Analisis Sentimen pada Komentar Aplikasi MyPertamina dengan Metode Multinomial Naive Bayes,” *ForAI J. Informatics Artif. Intell. J.*, vol. 1, no. 1, pp. 10–19, 2023, [Online]. Available: <https://jurnal.forai.or.id/index.php/forai/article/view/4>
- [20] M. I. Putri and I. Kharisudin, “Analisis Sentimen Pengguna Aplikasi Marketplace Tokopedia Pada Situs Google Play Menggunakan Metode Support Vector Machine (SVM), Naïve Bayes, dan Logistic Regression,” *Prism. Pros. Semin. Nas. Mat.*, vol. 5, pp. 759–766, 2022, [Online]. Available: <https://journal.unnes.ac.id/sju/index.php/prisma/>
- [21] S. Anggina, N. Y. Setiawan, and F. A. Bachtar, “Analisis Ulasan Pelanggan Menggunakan Multinomial Naïve Bayes Classifier dengan Lexicon-Based dan TF-IDF Pada Formaggio Coffee and Resto,” *is Best Account. Inf. Syst. Inf. Technol. Bus. Enterp. this is link OJS us*, vol. 7, no. 1, pp. 76–90, 2022, doi: 10.34010/aisthebest.v7i1.7072.
- [22] 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.