Jurnal Informatika Terpadu

https://journal.nurulfikri.ac.id/index.php/JIT
ISSN ONLINE : 2460-8998

PENERAPAN ALGORITMA *NAIVE BAYES* DALAM ANALISIS SENTIMEN ULASAN APLIKASI KITALULUS DI GOOGLE PLAY STORE

Dina Siti Nurrochmah[1], Nining Rahaningsih[2], Raditya Danar Dana[3], Cep Lukman Rohmat[4]

[1] Program Studi Teknik Informatika, STMIK IKMI Cirebon
[2] Program Studi Komputerisasi Akutansi, STMIK IKMI Cirebon
[3] Program Studi Manajemen Informatika, STMIK IKMI Cirebon
[4] Program Studi Rekayasa Perangkat Lunak, STMIK IKMI Cirebon
Kota Cirebon, Jawa Barat, Indonesia 45143
dinasitinurrochmah@gmail.com, niningr157@yahoo.co.id, radith_danar@yahoo.com, ceplukmanrohmat@gmail.com

Abstract

Online job search applications are proliferating and are crucial for job seekers in Indonesia. As seen in Google Play Store reviews, KitaLulus, a leading platform, faces technical issues, unresponsive services, and limited job postings. This study analyzes user sentiment using the Naive Bayes algorithm. Data was collected from 1,000 reviews through web scraping between September and November 2024. The pre-processing steps included text cleaning, tokenization, stopword removal, and stemming. It classified reviews into positive, neutral, and negative sentiments. A confusion matrix evaluated the model using accuracy, precision, recall, and F1-score. Results showed positive reviews, but some users reported performance issues and limited features. The Naive Bayes model achieved 88% accuracy, 87% precision, 88% recall, and an 85% F1 score. This method efficiently processes extensive text data with lower computational costs than KNN and SVM. This research helps improve application development, enhance service quality, and expand sentiment analysis studies in IT. The findings will guide the creation of innovative strategies to benefit the community.

Keywords: *Google Play Store, KitaLulus, Naive Bayes, Sentiment Analysis, User Reviews*

Abstrak

Aplikasi pencarian kerja daring semakin berkembang dan memegang peranan penting dalam memenuhi kebutuhan pencari kerja di Indonesia. Aplikasi KitaLulus, salah satu platform unggulan, menghadapi tantangan berupa ulasan pengguna di Google Play Store yang mengindikasikan adanya kendala teknis, ketidakresponsifan layanan, dan kurangnya informasi lowongan pekerjaan. Penelitian ini bertujuan untuk menganalisis sentimen ulasan pengguna terhadap aplikasi KitaLulus dengan menerapkan algoritma *Naive Bayes*. Data dikumpulkan melalui metode *web scraping* dari 1.000 ulasan yang diperoleh antara September hingga November 2024. Proses pra-pemrosesan data meliputi pembersihan teks, *tokenisasi*, penghapusan *stopword*, dan *stemming*, sehingga ulasan dapat diklasifikasikan ke dalam tiga kategori sentimen: positif, netral, dan negatif. Evaluasi model dilakukan menggunakan *confusion matrix* dengan metrik *akurasi*, *presisi*, *recall*, dan *F1-score*. Hasil penelitian menunjukkan mayoritas ulasan bersifat positif, meskipun terdapat sejumlah ulasan negatif yang mengeluhkan performa aplikasi dan keterbatasan fitur. Model *Naive Bayes* mencapai *akurasi* 88%, *presisi* 87%, *recall* 88%, dan *F1-score* 85%. Keunggulan metode ini terletak pada efisiensi pengolahan data teks berukuran besar dengan komputasi yang lebih ringan dibandingkan metode lain seperti *KNN* dan *SVM*. Penelitian ini memberikan kontribusi signifikan bagi pengembangan aplikasi, meningkatkan kualitas layanan, dan memperkaya literatur analisis sentimen dalam bidang teknologi informasi. Temuan penelitian ini diharapkan menjadi dasar utama pengembangan strategi inovatif bagi masyarakat.

Kata kunci: Analisis Sentimen, Google Play Store, KitaLulus, *Naive Bayes*, Ulasan Pengguna

1. PENDAHULUAN

Seiring dengan kemajuan teknologi digital, perangkat pencarian kerja daring telah muncul sebagai sumber utama bagi para pencari kerja yang mencari peluang yang sesuai dengan kualifikasi mereka. KitaLulus adalah platform terkenal di Indonesia yang menawarkan akses ke berbagai lowongan kerja beserta fitur-fitur tambahan seperti persiapan wawancara dan pembuatan riwayat hidup[1]. Namun, evaluasi pengguna di Google Play Store menunjukkan bahwa meskipun penggunaannya meluas,

Diterima 13 Desember 2024, Direvisi 22 Februari 2025, Diterima untuk publikasi 29 Maret 2025

aplikasi ini masih memiliki sejumlah masalah, seperti kemampuan yang terbatas, keterbatasan teknis, dan ketidakpuasan terhadap layanan yang ditawarkan. Masalah-masalah ini dapat memengaruhi keberlanjutan aplikasi di pasar yang sangat kompetitif, serta pengalaman pengguna[2].

Analisis sentimen merupakan metode yang relevan untuk memeriksa tren ulasan pengguna guna memahami bagaimana pengguna mengevaluasi aplikasi KitaLulus. Dengan menggunakan teknik ini, masukan pengguna dapat dikategorikan sebagai baik, netral, atau negatif, sehingga memberikan lebih banyak informasi kepada pengembang untuk membantu mereka meningkatkan kualitas aplikasi. *Naive Bayes* merupakan metode yang populer dalam analisis sentimen berbasis teks karena efektivitasnya dalam memproses data teks dalam jumlah besar dengan *overhead* pemrosesan yang minimal. Dengan demikian, tujuan dari penelitian ini adalah untuk menggunakan metode *Naive Bayes* guna menganalisis sentimen evaluasi pengguna terhadap aplikasi KitaLulus di Google Play Store.

Berikut ini adalah tujuan utama dari penelitian ini: (1) menganalisis dan mengategorikan ulasan pengguna terhadap aplikasi KitaLulus menggunakan teknik *Naive Bayes* guna memahami bagaimana pengguna menginterpretasikan aplikasi, (2) menilai efektivitas algoritma *Naive Bayes* dalam prosedur pengategorian ulasan pengguna terhadap aplikasi KitaLulus, dan (3) mengidentifikasi elemen-elemen yang memengaruhi tingkat kepuasan pengguna berdasarkan temuan analisis sentimen.

Secara teori, penelitian ini memajukan pembuatan model analisis sentimen berbasis *Naive Bayes* untuk klasifikasi teks dalam aplikasi yang terkait dengan pencarian kerja. Selain itu, penelitian ini menambah pengetahuan tentang analisis sentimen, khususnya terkait aplikasi yang berfokus pada layanan ketenagakerjaan di Indonesia. Diharapkan bahwa temuan penelitian ini akan membantu pengembang program KitaLulus dalam menentukan fitur mana, berdasarkan sentimen pengguna, yang memerlukan peningkatan guna meningkatkan kebahagiaan dan loyalitas pengguna dalam jangka panjang.

Metodologi berbasis analisis sentimen dalam studi ini diharapkan dapat memberikan pemahaman yang lebih menyeluruh tentang persepsi pengguna terhadap aplikasi KitaLulus dan menjadi dasar bagi pilihan strategis yang bertujuan untuk meningkatkan fitur dan layanannya. Lebih jauh, studi ini diharapkan dapat menciptakan peluang baru untuk kemajuan dalam penggunaan algoritma pembelajaran mesin untuk analisis sentimen di berbagai industri layanan digital.

2. METODE PENELITIAN

Algortima *Naive Bayes* untuk analisis sentimen digunakan dalam metodologi eksperimen kuantitatif penelitian ini. Metode ini dipilih karena kemampuannya untuk mengekstrak sentimen dari data tekstual, khususnya evaluasi pengguna aplikasi. Untuk mengungkap pola sentimen yang menunjukkan tingkat kebahagiaan pengguna, penelitian ini juga menggunakan metode komputasi untuk mengumpulkan dan memeriksa data ulasan pengguna untuk aplikasi KitaLulus di Google Play Store.

2.1. Metode Pengumpulan Data dan Analisis Data

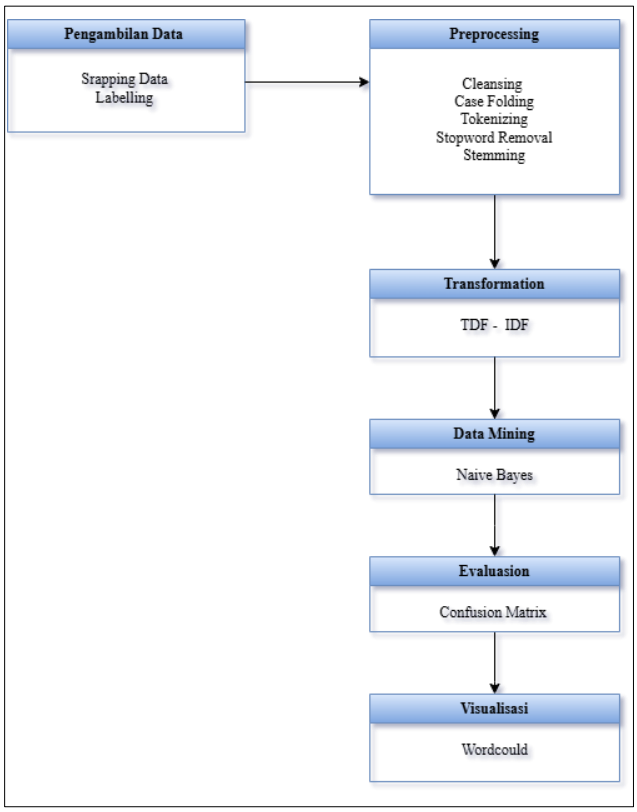
Menggunakan teknik *web scraping* dari Google Play Store, yang memungkinkan pengumpulan otomatis sejumlah besar data, data ulasan dikumpulkan untuk memeriksa sentimen pengguna terhadap aplikasi KitaLulus. Untuk menjamin relevansi data, metode ini mengakses ulasan aplikasi menggunakan bahasa pemrograman Python dan modul *google-play-scraper*. Ulasan kemudian difilter menurut bahasa dan negara. Informasi yang dikumpulkan, yang mencakup ulasan, peringkat, dan atribut terkait, disimpan dalam format *CSV*. Setelah pengumpulan data, seorang ahli bahasa secara manual memberi label ulasan ke dalam tiga kategori sentimen: positif, netral, dan negatif. Label ini berfungsi sebagai set pelatihan untuk model *Naive Bayes*. Tujuan dari teknik *scraping* ini adalah untuk mengumpulkan data terstruktur yang dapat digunakan untuk mengukur sentimen pengguna dan menawarkan wawasan komprehensif tentang bagaimana pengguna melihat aplikasi KitaLulus [3].

Proses pengolahan data melibatkan teknik *Natural Language Processing* (NLP) seperti *tokenization*, *stopword removal*, dan *stemming* untuk menyiapkan data sebelum dianalisis menggunakan metode *Naive Bayes* [4]. *Naive Bayes* merupakan algoritma *machine learning* berbasis probabilistik yang banyak digunakan dalam klasifikasi sentimen karena kemampuannya menangani data teks dengan efisien dan kecepatan komputasi yang ringan dibandingkan metode lain seperti KNN dan SVM[5]. Metode *Naive Bayes* telah banyak digunakan dalam penelitian tentang analisis sentimen evaluasi pengguna aplikasi, dengan hasil yang menunjukkan tingkat akurasi yang baik di berbagai konteks. Dengan menggunakan pemisahan data 80:20, Agustina et al. [6] mencapai akurasi 83% dalam penelitian mereka sebelumnya menggunakan *Naive Bayes* untuk ulasan Shopee di Google Play Store. Rifaldi et al. [7] menggunakan *Naive Bayes* untuk menganalisis *tweet* ChatGPT dengan *recall* 89,47%, presisi 80,95%, dan akurasi 80%. Layanan yang disediakan oleh Grab Indonesia dinilai oleh Hasan dan Dwijayanti [8], yang menemukan bahwa sentimen positif memiliki tingkat akurasi 97% dan skor f1 96%. Meskipun masih ada ruang untuk perbaikan, hasil ini menunjukkan bahwa *Naive Bayes* lebih unggul dalam klasifikasi data teks. Untuk meningkatkan kepuasan pengguna dan kualitas layanan aplikasi berbasis pendidikan, penelitian ini bermaksud menerapkan *Naive Bayes* untuk menganalisis ulasan

pelanggan KitaLulus dengan penekanan pada persiapan teks dan pengoptimalan parameter.

Dalam penelitian ini, metode *Naive Bayes* digunakan untuk menganalisis sentimen ulasan pengguna aplikasi KitaLulus di Google Play Store, di mana data dikumpulkan melalui *web scraping* dan diklasifikasikan ke dalam tiga kategori: positif, netral, dan negatif. Keunggulan utama dari metode ini adalah kemampuannya dalam menangani *dataset* besar dengan tingkat akurasi tinggi serta kemudahan implementasi dalam berbagai skenario klasifikasi teks. Proses *Knowledge Discovery in Database (KDD)* diterapkan dalam penelitian ini, yang meliputi pengumpulan data, *preprocessing*, transformasi dengan TF-IDF, *data mining* menggunakan *Naive Bayes*, dan evaluasi model menggunakan metrik seperti *accuracy*, *precision*, *recall*, dan *skor F1* dari *Confusion Matrix* [9]. Hasil evaluasi ini akan memberikan wawasan penting dalam mengidentifikasi salah klasifikasi dan meningkatkan akurasi model di masa mendatang. *Word Cloud* juga digunakan untuk memvisualisasikan kata-kata yang paling sering muncul, sehingga memberikan gambaran umum tentang sentimen yang terkandung dalam data *review*[10].

2.2. Tahapan Penelitian



Gambar 1. Tahapan Metode Penelitian

Adapun tahapan metode penelitian dapat dilihat pada Gambar 1 sebagai berikut.

A. Pengambilan Data

Data ulasan pengguna aplikasi KitaLulus dikumpulkan menggunakan pustaka Google Play Scraper, dengan maksimum 1000 ulasan dari periode September hingga November 2024. Data disimpan dalam format *CSV* untuk pemrosesan lebih lanjut.

B. Pelabelan Data

Data diberi label secara manual sebagai positif, netral, atau negatif oleh seorang ahli bahasa, memastikan data berkualitas tinggi untuk algoritma *supervised learning*.

C. *Preprocessing*

Tahap *preprocessing* dilakukan sebelum *dataset* dari *text mining* dimasukkan ke dalam model. Proses ini mencakup analisis sintaksis untuk memastikan struktur teks yang tepat dan analisis semantik untuk menjaga makna yang sesuai[11]. Tujuan utamanya adalah mengubah teks menjadi data berkualitas tinggi yang siap untuk analisis lebih lanjut. Langkah-langkah yang dilakukan meliputi pembersihan data, tokenisasi, *case folding*, dan *stemming:*

a) Pembersihan Data (*Data Cleansing*)

Untuk meningkatkan kualitas data teks dan menjamin bahwa hanya informasi penting yang disimpan, karakter yang tidak relevan seperti emotikon, angka, URL, dan tanda baca dihilangkan. Agar model dapat berkonsentrasi pada kata-kata yang penting dan relevan untuk memahami sentimen atau makna teks, prosedur pembersihan ini berupaya menghilangkan gangguan dari analisis. Data dibuat lebih terstruktur dan disiapkan untuk pemrosesan tambahan dengan menghilangkan komponen-komponen ini [12].

b) Case Folding

Untuk menjaga keseragaman dalam analisis, semua teks diubah menjadi huruf kecil, menyamakan istilah dengan berbagai ejaan, seperti "Pendidikan" dan "pendidikan." Karena algoritma tidak akan membedakan antara kata-kata yang hanya berbeda dalam penggunaan huruf kapital, prosedur ini sangat penting untuk mencegah pengulangan kata, yang dapat berdampak pada temuan analisis. Karena kata-kata yang sama dianggap sebanding meskipun ditulis secara berbeda, langkah ini meningkatkan akurasi analisis.

c) Tokenizing

Tahapan tokenisasi teks melibatkan pembagian teks menjadi potongan-potongan yang lebih kecil, seperti kata, frasa, atau simbol, sehingga algoritma dapat memeriksanya lebih lanjut. Setiap teks atau pernyataan dibagi menjadi unit-unit yang lebih kecil, biasanya kata-kata[13]. Misalnya, model dapat menentukan frekuensi kemunculan kata-kata dalam kumpulan data dengan memecah kalimat "Aplikasi *Kitalulus* sangat

membantu" menjadi token "Aplikasi", "Kitalulus", "sangat", dan "membantu".

d) Stopword Removal

Teks tersebut menghilangkan istilah-istilah umum yang sering digunakan dalam bahasa tersebut, termasuk "dan," "atau," dan "di," serta kata-kata lain yang tidak memiliki banyak konotasi sentimental. Untuk lebih berkonsentrasi pada istilah-istilah yang benar-benar membantu dalam memahami sentimen pengguna seperti kata-kata yang mencirikan perasaan, atribut, atau sifat yang terkait dengan subjek yang diteliti prosedur ini berupaya untuk mengecualikan kata-kata yang tidak relevan dengan analisis.

e) Stemming

Konversi kata ke bentuk dasarnya, seperti "menjari" menjadi "ajari," mengurangi varian kata dan mempermudah analisis data. Prosedur ini meningkatkan akurasi model dalam mengenali pola atau perasaan dalam teks dengan memungkinkannya berkonsentrasi pada makna esensial kata tersebut, terlepas dari berbagai bentuk kata.

D. *Transformasi* (TF-IDF)

Dalam penelitian ini, teknik reduksi variabel yang digunakan pada langkah ekstraksi kumpulan data disebut *Term Frequency–Inverse Document Frequency*, atau TF-IDF [14]. TF-IDF digunakan untuk memberi bobot lebih pada kata-kata relevan dalam analisis sentimen, mengurangi pengaruh istilah yang sering muncul tetapi kurang bermakna.

E. *Data Mining*

Algoritma *Naive Bayes* diterapkan untuk melatih model klasifikasi sentimen, memprediksi apakah ulasan termasuk kategori positif, netral, atau negatif berdasarkan pola kata. Dalam hal kalkulasi probabilitas, metode *Naive Bayes* lebih baik daripada algoritma lain karena lebih efisien.

F. Evaluasi Model

Confusion matrix adalah alat evaluasi penting dalam klasifikasi model yang menunjukkan hasil prediksi terhadap data yang sesungguhnya [15]. Model dinilai menggunakan metrik seperti *akurasi*, *presisi*, *recall*, dan *skor F1*. *Confusion Matrix* digunakan untuk mengidentifikasi kekuatan dan kelemahan klasifikasi, yang membantu perbaikan model.

G. Visualisasi

Wordcloud digunakan untuk menampilkan kata-kata yang paling sering muncul dalam ulasan, memberikan wawasan visual intuitif tentang aspek apa yang dianggap penting oleh pengguna. Dengan menyorot kata-kata yang sering muncul, *wordcloud* memungkinkan identifikasi cepat terhadap tema atau fitur yang disukai atau dikeluhkan pengguna, baik yang terkait dengan kualitas, kinerja, atau fungsionalitas aplikasi. Selain itu, ukuran kata-kata dalam awan kata mencerminkan frekuensi kemunculannya, sehingga kata-kata yang lebih besar menunjukkan fokus utama pengguna, baik dalam konteks positif maupun negatif. Hal ini memudahkan pengembang atau pemangku kepentingan untuk memahami persepsi pengguna secara keseluruhan.

3. HASIL DAN PEMBAHASAN

3.1 *Scraping* Data

Dengan menggunakan modul *google-play-scraper* dalam bahasa pemrograman Python, data ulasan pengguna aplikasi KitaLulus di Google Play Store dikumpulkan secara otomatis untuk penelitian ini. Dengan menggunakan parameter seperti ID aplikasi *(com.kitalulus.app)*, pengaturan bahasa *(lang='id')*, tempat asal *(country='id')*, dan pengurutan berdasarkan ulasan terbaru *(Sort.NEWEST)*, pengumpulan data difokuskan pada ulasan dalam bahasa Indonesia untuk menjaga relevansi analisis. Dari September hingga November 2024, total 1.000 ulasan berhasil dikumpulkan. Kemudian, data yang berisi atribut seperti sentimen, nama pengguna, peringkat, dan tanggal disimpan dalam format *Comma-Separated Values* (CSV). Prosedur selanjutnya, seperti pembersihan data, klasifikasi sentimen, dan pelatihan model *Naive Bayes*, difasilitasi dengan penyimpanan dalam format *CSV*. Ringkasan ulasan aplikasi KitaLulus dapat ditemukan dalam data hasil pengikisan yang ditampilkan pada Gambar 2.

	userName	score	at	content
0	Si alpin 1	1	28/10/2024 10:10	Apk cacat, daftarnya terlalu ribet. Kalo penga...
1	Linda Febriana	5	28/10/2024 08:45	bagus
2	Yamet Pekanbaru Tampan	5	28/10/2024 08:25	aplikasi nya mantap, sangat membantu para rekr...
3	Tono Anton	5	28/10/2024 07:38	propesional
4	Bang Tass	5	28/10/2024 07:04	Pelayanan cepat dan supportif
...	...	...	...	...
995	Suryani uya	5	11/09/2024 03:14	bagus
996	Muhamad Junaidi	4	11/09/2024 01:58	bintang 4 dulu ya min
997	Hafidatus Zahrah	4	11/09/2024 01:44	aplikasi KitaLulus sangat membantu mencari pek...
998	hardi ansyah	5	11/09/2024 01:39	bagus sekali
999	Ryan Andri	4	10/09/2024 17:32	baik

Gambar 2. Hasil *Scraping Data*

3.2 *Labelling* Data

Pada tahap sebelumnya telah didapatkan berupa data mentah yang dimana data tersebut belum dilabeli. Untuk hasil yang lebih akurat *labelling* data dilakukan oleh seorang pakar/ahli bahasa yaitu, seorang guru Bahasa Indonesia yang mengajar di SMK Negeri 1 Cirebon. Data yang telah dikumpulkan dikategorikan secara manual ke dalam *sentiment* positif, negatif, dan netral. Langkah ini bertujuan untuk menyiapkan data sebagai *training set* dalam pelatihan model Naïve Bayes.

3.3 *Preprocessing*

Proses analisis sentimen dimulai dengan mengubah data ulasan dalam format CSV yang tidak terstruktur menjadi data yang lebih terstruktur. Langkah persiapan teks meliputi pembersihan data melalui *case folding*, *stemming*, *tokenisasi*, dan penghapusan *stopword* untuk mempermudah analisis.

a) *Case Folding*

Case folding adalah langkah awal dalam persiapan data teks dengan mengubah semua huruf menjadi huruf kecil untuk konsistensi analisis. Proses ini, dilakukan menggunakan pustaka Python seperti *re*. Proses Case Folding dapat dilihat pada Gambar 3.

Proses perubahan huruf besar menjadi huruf kecil terlihat pada kolom *content*, misalnya "Aplikasi" diubah menjadi "aplikasi" di kolom *text_clean*, dan perubahan serupa terjadi pada setiap baris berikutnya. Pada Gambar 4 menunjukkan hasil *output* tahapan *Case Folding*.

```
[ ] pip install pandas

Requirement already satisfied: pandas in /usr/local/lib/python3.10/dist-packages (2.2.2)
Requirement already satisfied: numpy>=1.22.4 in /usr/local/lib/python3.10/dist-packages (from pandas) (1.26.4)
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.10/dist-packages (from pandas) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (from pandas) (2024.2)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.10/dist-packages (from pandas) (2024.2)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.8.2->pandas) (1.16.0)

[ ] import pandas as pd
    data = pd.read_csv('/content/DataKitaLulus1.csv', delimiter=';')  # Ganti ';' dengan delimiter yang benar
    data.head(10)
```

Gambar 3. Proses *Case Folding*

	userName	score	at	content	Label	text_clean
0	Si alpin 1	1	28/10/2024 10:10	Apk cacat, daftarnya terlalu ribet. Kalo penga...	Negatif	apk cacat daftarnya terlalu ribet kalo pengala...
1	Linda Febriana	5	28/10/2024 08:45	bagus	Positif	bagus
2	Yamet Pekanbaru Tampan	5	28/10/2024 08:25	aplikasi nya mantap, sangat membantu para rekr...	Positif	aplikasi nya mantap sangat membantu para rekru...
3	Tono Anton	5	28/10/2024 07:38	propesional	Positif	propesional
4	Bang Tass	5	28/10/2024 07:04	Pelayanan cepat dan supportif	Positif	pelayanan cepat dan supportif
5	Bagas 253 253	5	28/10/2024 05:51	mntp apk nya	Positif	mntp apk nya
6	Muhammad Syahrudy	5	28/10/2024 04:41	Aplikasi membantu buat cari pekerjaan terimaka...	Positif	aplikasi membantu buat cari pekerjaan terimaka...
7	Rusna Saleh	5	28/10/2024 04:37	sangat membantu	Positif	sangat membantu
8	Nur Huda	5	28/10/2024 03:52	sangat puas	Positif	sangat puas
9	Edi Tenong	5	27/10/2024 16:46	trabaik	Positif	trabaik
10	Arshal Grx	4	27/10/2024 16:33	sangat membantu	Positif	sangat membantu
11	Yosep Sopian017	5	27/10/2024 14:24	mantap banget tapi sayang udah lama gak dapat ...	Positif	mantap banget tapi sayang udah lama gak dapat ...
12	Deva Lino	5	27/10/2024 13:34	baguss	Positif	baguss
13	Lenta Hasibuan	1	27/10/2024 13:25	baik	Negatif	baik

Gambar 4. Hasil *Case Folding*

b) *Stopword Removal*

Kata-kata yang dianggap *stopword*, seperti "dan," "atau," "di," "ke," "dengan," dan "yang," dihilangkan dalam proses ini karena tidak memiliki makna sentimen yang kuat. Proses *Stopword Removal* dapat dilihat pada Gambar 5 di bawah ini.

Proses *Stopword Removal* dimulai dengan mengimpor library Python, dan hasilnya dapat dilihat pada kolom *text_stopword* pada Gambar 6 di mana kata "dan" dalam kalimat "pelayanan cepat dan supportif" dihilangkan, sehingga menjadi "pelayanan cepat supportif".

```
import nltk.corpus
nltk.download('stopwords')
from nltk.corpus import stopwords
stop = stopwords.words('indonesian')

[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.

[ ] data_clean['Text_StopWord'] = data_clean['text_clean'].apply(lambda x:' '.join([word for word in x.split() if word not in (stop)]))

[ ] data_clean.head(50)
```

Gambar 5. Proses *Stopword Removal*

	content	Label	text_clean	Text_StopWord
0	Apk cacat, daftarnya terlalu ribet. Kalo penga...	Negatif	apk cacat daftarnya terlalu ribet kalo pengala...	apk cacat daftarnya ribet kalo pengalaman ga u...
1	bagus	Positif	bagus	bagus
2	aplikasi nya mantap, sangat membantu para rekr...	Positif	aplikasi nya mantap sangat membantu para rekru...	aplikasi nya mantap membantu rekruter pencari ...
3	propesional	Positif	propesional	propesional
4	Pelayanan cepat dan supportif	Positif	pelayanan cepat dan supportif	pelayanan cepat supportif
5	mntp apk nya	Positif	mntp apk nya	mntp apk nya
6	Aplikasi membantu buat cari pekerjaan terimaka...	Positif	aplikasi membantu buat cari pekerjaan terimaka...	aplikasi membantu cari pekerjaan terimakasih l...
7	sangat membantu	Positif	sangat membantu	membantu
8	sangat puas	Positif	sangat puas	puas

Gambar 6. Hasil *Stopword Removal*

c) *Tokenizing*

Sebagai komponen dasar analisis sentimen, *tokenisasi* adalah proses memecah teks menjadi beberapa bagian yang dikenal sebagai *token*, seperti kata, frasa, atau simbol. Tujuan dari prosedur ini adalah untuk mengatur data teks mentah yang tidak terstruktur sehingga algoritme pembelajaran mesin dapat memprosesnya. Token yang dihasilkan memfasilitasi klasifikasi atau pengelompokan sentimen yang lebih akurat dan efisien dengan memungkinkan algoritma untuk mengidentifikasi pola, hubungan, dan makna dalam data teks.

Pada Gambar 7 ini menunjukkan, data yang telah melalui tahap penghapusan *stopword* (di kolom *text_stopword*) diproses menggunakan *library* NLTK, menghasilkan token-token yang siap untuk analisis lebih lanjut.

Hasil proses *Tokenizing* pada Gambar 8 menunjukkan kolom *text_stopword*, kalimat "aplikasi nya mantap membantu rekruter pencari.." diubah menjadi token terpisah seperti [aplikasi, nya, mantap, membantu, rekruter, pencari..] di kolom *text_tokens*.

```
import nltk
nltk.download('punkt')
from nltk.tokenize import sent_tokenize, word_tokenize
data_clean['text_tokens'] = data_clean['Text_StopWord'].apply(lambda x: word_tokenize(x))
data_clean.head()
```

Gambar 7. Proses *Tokenizing*

```
/root/nltk_data...
zip.
```

	content	Label	text_clean	Text_StopWord	text_tokens
0	Apk cacat, daftarnya terlalu ribet. Kalo penga...	Negatif	apk cacat daftarnya terlalu ribet kalo pengala...	apk cacat daftarnya ribet kalo pengalaman ga u...	[apk, cacat, daftarnya, ribet, kalo, pengalama...
1	bagus	Positif	bagus	bagus	[bagus]
2	aplikasi nya mantap, sangat membantu para rekr...	Positif	aplikasi nya mantap sangat membantu para rekru...	aplikasi nya mantap membantu rekruter pencari ...	[aplikasi, nya, mantap, membantu, rekruter, pe...
3	propesional	Positif	propesional	propesional	[propesional]
4	Pelayanan cepat dan supportif	Positif	pelayanan cepat dan supportif	pelayanan cepat supportif	[pelayanan, cepat, supportif]

Gambar 8. Hasil *Tokenizing*

d) *Stemming*

Stemming dalam pemrosesan bahasa alami (NLP) adalah teknik untuk mengembalikan kata ke bentuk dasar atau akarnya, yang membantu meningkatkan akurasi model analisis sentimen dengan mengurangi variasi kata. Pada Gambar 9 merupakan proses *stemming* dilakukan.

Dalam proses ini, *library Python Sastrawi* digunakan untuk melakukan *stemming* pada *dataset* Bahasa Indonesia, menghasilkan kamus yang memetakan kata asli ke bentuk dasarnya. Seperti yang ditunjukkan pada Gambar 10 kata "membantu" diubah menjadi "bantu" dalam kolom *text_steamindo*. Proses ini mengurangi keragaman kata yang tidak perlu saat analisis lebih lanjut.

```
pip install Sastrawi
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
factory = StemmerFactory()
stemmer = factory.create_stemmer()
```

```
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
#import swifter

# create stemmer
factory = StemmerFactory()
stemmer = factory.create_stemmer()

# stemmed
def stemmed_wrapper(term):
    return stemmer.stem(term)

term_dict = {}
hitung=0

for document in data_clean['text_tokens']:
    for term in document:
        if term not in term_dict:
            term_dict[term] = ' '

print(len(term_dict))
print("------------------------")
for term in term_dict:
    term_dict[term] = stemmed_wrapper(term)
    hitung+=1
    print(hitung,":",term,":" ,term_dict[term])

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

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

Gambar 9. Proses *Stemming*

content	Label	text_clean	Text_StopWord	text_tokens	text_steamindo
Apk cacat, daftarnya terlalu ribet. Kalo penga...	Negatif	apk cacat daftarnya terlalu ribet kalo pengala...	apk cacat daftarnya ribet kalo pengalaman ga u...	[apk, cacat, daftarnya, ribet, kalo, pengalama...	apk cacat daftar ribet kalo alam ga umbar bsa ...
bagus	Positif	bagus	bagus	[bagus]	bagus
aplikasi nya mantap, sangat membantu para rekr...	Positif	aplikasi nya mantap sangat membantu para rekru...	aplikasi nya mantap membantu rekruter pencari ...	[aplikasi, nya, mantap, membantu, rekruter, pe...	aplikasi nya mantap bantu rekruter cari kerja
propesional	Positif	propesional	propesional	[propesional]	propesional
Pelayanan cepat dan supportif	Positif	pelayanan cepat dan supportif	pelayanan cepat supportif	[pelayanan, cepat, supportif]	layan cepat supportif
mntp apk nya	Positif	mntp apk nya	mntp apk nya	[mntp, apk, nya]	mntp apk nya
Aplikasi membantu buat cari pekerjaan terimaka...	Positif	aplikasi membantu buat cari pekerjaan terimaka...	aplikasi membantu cari pekerjaan terimakasih l...	[aplikasi, membantu, cari, pekerjaan, terimaka...	aplikasi bantu cari kerja terimakasih lulus mo...
sangat membantu	Positif	sangat membantu	membantu	[membantu]	bantu
sangat puas	Positif	sangat puas	puas	[puas]	puas
trabaik	Positif	trabaik	trabaik	[trabaik]	trabaik

Gambar 10. Hasil *Stemming*

3.4 *Trasnformation* (TF-IDF)

Dengan menggunakan pendekatan *Naïve Bayes* untuk analisis sentimen, transformasi data dimulai dengan membagi kumpulan data menjadi dua kategori utama: data pelatihan dan data pengujian. Bagian ini berupaya menjamin bahwa model dapat mengidentifikasi pola dalam data pelatihan, yang mencakup 80% dari kumpulan data, dan bahwa data pengujian, yang mencakup 20% dari kumpulan data, digunakan untuk mengevaluasi model secara tidak memihak. Rasio 80:20 sering digunakan karena menawarkan rasio pembelajaran dan pengujian yang seimbang, yang memungkinkan model untuk belajar cukup banyak sambil mengevaluasi kinerjanya pada data yang sebelumnya tidak terlihat. Data pengujian digunakan untuk menilai daya prediktif model pada data baru, sementara data pelatihan digunakan untuk melatih model untuk mengenali pola sentimen. Gambar 11 di bawah ini memberikan penjelasan terperinci tentang prosedur ini, yang merupakan langkah penting dalam mempersiapkan penyelidikan tambahan.

Pendekatan TF-IDF *(Term Frequency-Inverse Document Frequency)* kemudian digunakan untuk mengubah teks menjadi data numerik. Dengan membandingkan frekuensi kemunculan kata dalam dokumen tertentu dengan frekuensinya di seluruh *korpus*, pendekatan ini menentukan bobot *TF-IDF* untuk setiap kata. Metode ini mengurangi dampak kata-kata yang sering muncul tetapi tidak terlalu informatif sekaligus membuat model lebih sensitif terhadap kata-kata yang khusus atau penting dalam konteks tertentu. Prosedur ini dilakukan pada Gambar 12 dengan memproses data pelatihan dengan *TF-IDF Vectorizer*, yang menghasilkan matriks fitur numerik yang

mewakili kata-kata dalam dokumen. Model kemudian dilatih menggunakan matriks ini, dan untuk menjamin konsistensi dalam analisis, data uji ditangani menggunakan metodologi yang sama. Metode ini menjamin bahwa data disiapkan untuk mendeteksi pola sentimen dengan teknik pembelajaran mesin seperti *Naive Bayes*.

CountVectorizer berfungsi mengubah teks menjadi representasi numerik berdasarkan frekuensi kemunculan kata, dengan membuat vektor dari kata-kata dalam data pelatihan. Proses ini mempersiapkan data agar dapat digunakan oleh model pembelajaran mesin, seperti *Naive Bayes*.

```
[ ] #membagi data menjadi data training dan testing dengan test_size = 0.20 dan random state nya 0
    from sklearn.model_selection import train_test_split
    X_train, X_test, y_train, y_test = train_test_split(data_clean['content'], data_clean['Label'],
                                                        test_size = 0.20,
                                                        random_state = 0)
```

Gambar 11. Proses *Spliting Data*

```
from sklearn.feature_extraction.text import TfidfVectorizer
tfidf_vectorizer = TfidfVectorizer()
tfidf_train = tfidf_vectorizer.fit_transform(X_train)
tfidf_test = tfidf_vectorizer.transform(X_test)

print(X_train.shape)
print(y_train.shape)
print(X_test.shape)
print(y_test.shape)

(800,)
(800,)
(200,)
(200,)

from sklearn.feature_extraction.text import CountVectorizer

vectorizer = CountVectorizer()
vectorizer.fit(X_train)

CountVectorizer
CountVectorizer()
```

Gambar 12. Proses *TF-IDF*

3.5 Implementasi Algoritma *Naïve Bayes*

Tahap pertama dalam penerapan klasifikasi sentimen menggunakan *Naive Bayes* adalah kompilasi model dengan mengimpor modul dari sklearn. Selanjutnya, model *Multinomial Naive Bayes* diimplementasikan, dimulai dengan mengimpor MultinomialNB dan membuat instansi model. Model kemudian dilatih menggunakan data pelatihan yang telah diproses dengan TF-IDF melalui fungsi *fit(tfidf_train, y_train)*. Setelah itu, prediksi dilakukan pada data uji menggunakan fungsi *predict(tfidf_test)*, dan hasil prediksi dibandingkan dengan label sebenarnya untuk mengukur akurasi model menggunakan *accuracy_score(y_test, y_pred)*. Proses ini memungkinkan evaluasi kinerja model berdasarkan akurasi prediksi yang dihasilkan. Pada Gambar 13 menunjukkan proses metode *Naive Bayes.*

```
[ ] from sklearn.naive_bayes import MultinomialNB

    nb = MultinomialNB()
    nb.fit(tfidf_train, y_train)

    MultinomialNB
    MultinomialNB()

[ ] X_train.toarray()

    array([[0, 0, 0, ..., 0, 0, 0],
           [0, 0, 0, ..., 0, 0, 0],
           [0, 0, 0, ..., 0, 0, 0],
           ...,
           [0, 0, 0, ..., 0, 0, 0],
           [0, 0, 0, ..., 1, 0, 0],
           [0, 0, 0, ..., 0, 0, 0]])

    y_pred = nb.predict(tfidf_test)

    from sklearn.metrics import accuracy_score

    accuracy = accuracy_score(y_test, y_pred)
```

Gambar 13. Implementasi Naïve Bayes

3.6 Evaluasi Model

Confusion Matrix digunakan untuk mengevaluasi kinerja model yang telah dilatih dan diuji. Akurasi, presisi, *recall*, dan F1-score yang ditentukan oleh nilai *true positive* (TP), *true negative* (TN), *false positive* (FP), dan *false negative* (FN) diukur untuk melakukan evaluasi. Evaluasi model dalam penelitian ini dilakukan dengan perhitungan matematis yang diimplementasikan menggunakan *software. Software* yang digunakan adalah *Python*, dengan bantuan pustaka *scikit-learn* (sklearn) untuk melakukan perhitungan metrik evaluasi model. *Scikit-learn* adalah pustaka dalam Python yang digunakan untuk *machine learning*, termasuk klasifikasi, regresi, dan klasterisasi.

```
import pandas as pd
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix

clf = MultinomialNB()
clf.fit(X_train, y_train)
predicted = clf.predict(X_test)

print("MultinomialNB Accuracy:", accuracy_score(y_test,predicted))
print("MultinomialNB Precision:", precision_score(y_test,predicted, average="weighted"))
print("MultinomialNB Recall:", recall_score(y_test,predicted, average="weighted"))
print("MultinomialNB f1_score:", f1_score(y_test,predicted, average="weighted"))

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

# Load dataset
data_clean = pd.read_csv('hasil_TextPreProcessingKitaLulus.csv')
```

Gambar 14. Proses Evaluasi Model

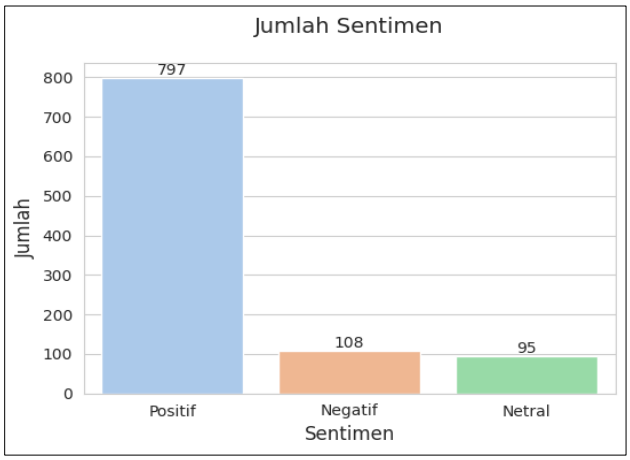
Proses evaluasi dimulai pada Gambar 14 dengan mengimpor pustaka *pandas* dan *sklearn* untuk mendukung pemrosesan data serta penilaian kinerja model klasifikasi teks. Teks diubah menjadi representasi numerik menggunakan pendekatan *TF-IDF* melalui *TfidfVectorizer*, kemudian model *Multinomial Naive Bayes* diterapkan untuk melakukan klasifikasi. Proses pelatihan dilakukan pada data latih (*X_train* dan *y_train*) menggunakan *clf.fit*, diikuti dengan prediksi pada data uji (*X_test*) menggunakan *clf.predict*, dengan hasil prediksi disimpan dalam variabel tertentu. Kinerja model dievaluasi menggunakan metrik seperti *akurasi*, *presisi*, *recall*, dan *skor F1*. Matriks kebingungan (*confusion_matrix*) membantu mengidentifikasi prediksi yang tepat maupun yang keliru, sementara laporan klasifikasi (*classification_report*) memberikan ringkasan komprehensif untuk setiap kelas, memungkinkan analisis mendalam terhadap kemampuan model dalam klasifikasi teks.

```
MultinomialNB Accuracy: 0.875
MultinomialNB Precision: 0.8695558659217878
MultinomialNB Recall: 0.875
MultinomialNB f1_score: 0.8493411306680626
confusion_matrix:
 [[ 11   1   4]
 [  2   5  16]
 [  2   0 159]]
=================================================

              precision    recall  f1-score   support

     Negatif       0.73      0.69      0.71        16
      Netral       0.83      0.22      0.34        23
     Positif       0.89      0.99      0.94       161

    accuracy                           0.88       200
   macro avg       0.82      0.63      0.66       200
weighted avg       0.87      0.88      0.85       200
```

Gambar 15. Hasil Evaluasi Model

Hasil evaluasi pada Gambar 15 menunjukkan bahwa model *Multinomial Naive Bayes* memiliki *akurasi* 88%, *presis*i 87%, *recall* 88%, dan *F1-score* 85%. Untuk sentimen negatif, model memiliki *presisi* 73%, *recall* 69%, *dan F1-score* 71%; untuk sentimen netral, *presisi* 83%, *recall* 22%, dan *F1-score* 34%; dan untuk sentimen positif, *presisi* 89%, *recall* 99%, dan *F1-score* 94%.

3.7 Visualisasi



Gambar 16. Diagram Batang

Hasil visualisasi ditampilkan sebagai diagram batang pada Gambar 16. yang menampilkan distribusi sikap dari data ulasan aplikasi KitaLulus. Mayoritas pengguna merasa senang dengan aplikasi ini, sebagaimana dibuktikan oleh 797 ulasan dengan sentimen positif yang ditampilkan dalam diagram batang ini. Di sisi lain, terdapat 108 evaluasi yang menyatakan kekecewaan atau kekhawatiran dari sebagian kecil orang. Lebih jauh, 95 ulasan memiliki sikap netral, yang menunjukkan bahwa komentar pengguna pada umumnya objektif atau tidak jelas. Berdasarkan kategorisasi sentimen, visualisasi ini menawarkan gambaran yang jelas tentang bagaimana pengguna memandang program ini.

Gambar 17. *Wordcloud*

Wordcloud adalah alat visualisasi data yang berguna untuk memeriksa seberapa sering kata muncul dalam sebuah dokumen. Teknik ini menyajikan istilah yang sering muncul dalam evaluasi pengguna dalam gaya grafis yang menarik, di mana besarnya kata menunjukkan frekuensi kemunculannya. *Wordcloud* pada Gambar 17 menampilkan istilah yang paling umum ditemukan dalam ulasan aplikasi KitaLulus, termasuk "aplikasi," "bantuan," "pekerjaan," dan seterusnya. Dengan memberikan gambaran umum kepada pengembang tentang bagaimana perasaan pengguna terhadap program tersebut, visualisasi ini memungkinkan mereka untuk mengidentifikasi istilah yang paling relevan dan sering digunakan dalam evaluasi pengguna. Ini membantu dalam menentukan fitur mana yang menurut orang menarik atau tidak menarik.

4. KESIMPULAN

Penelitian ini berhasil mengklasifikasikan sentimen ulasan pengguna aplikasi KitaLulus menggunakan algoritma *Naive Bayes* dengan tingkat akurasi 88%, yang menunjukkan dominasi ulasan positif meskipun terdapat beberapa keluhan terkait *bug* dan keterbatasan fitur. Implikasi penelitian ini dapat digunakan oleh pengembang aplikasi untuk meningkatkan layanan berdasarkan analisis sentimen, serta memperluas pemahaman akademis tentang efektivitas *Naive Bayes* dalam klasifikasi teks. Namun, penelitian ini memiliki keterbatasan dalam jumlah data yang terbatas pada 1.000 ulasan dan belum membandingkan performa algoritma dengan metode lain seperti LSTM atau SVM. Penelitian lanjutan dapat mempertimbangkan *dataset* yang lebih luas, analisis mendalam terhadap sentimen netral, serta eksplorasi model klasifikasi lain untuk meningkatkan akurasi. Kontribusi penelitian ini bagi masyarakat adalah membantu peningkatan kualitas aplikasi KitaLulus, sehingga dapat memberikan pengalaman pencarian kerja yang lebih baik bagi pengguna.

DAFTAR PUSTAKA

[1] KitaLulus., "Platform Pencarian Kerja Berorientasi Komunitas Terbesar dan Teraman di Indonesia," 2024. https://id.kitalulus.com/tentang-kitalulus (accessed Oct. 15, 2024).

[2] E. A. H. Ernianti Hasibuan, "Analisis Sentimen pada Ulasan Aplikasi Amazon Shopping di Google Play," *J. Tek. dan Sci.*, vol. 1, no. 3, pp. 13–24, 2022.

[3] B. Irawan, A. Bahtiar, P. Studi, T. Informatika, K. Cirebon, and A. Adakami, "Penggunaan Algoritma Naïve Bayes dalam Menganalisis Sentimen Ulasan Aplikasi Adakami di Google Play Store," *JATI (Jurnal Mhs. Tek. Inform.*, vol. 7, no. 6, pp. 3879–3885, 2023.

[4] Y. A. Singgalen, "Analisis Sentimen Wisatawan Melalui Data Ulasan Candi Borobudur di Tripadvisor Menggunakan Algoritma Naïve Bayes Classifier," *Build. Informatics, Technol. Sci.*, vol. 4, no. 3, 2022, doi: 10.47065/bits.v4i3.2486.

[5] A. Perdana, A. Hermawan, and D. Avianto, "Analisis Sentimen Terhadap Isu Penundaan Pemilu di Twitter Menggunakan *Naive Bayes* Classifier," *J. SISFOKOM*, vol. 11, pp. 195–200, 2022.

[6] A. R. K. Nurhaliza Agustina C.A, Desy Herlina Citra, Wido Purnama, Chairun Nisa, "Implementasi Algoritma *Naive Bayes* untuk Analisis Sentimen Ulasan Shopee pada Google Play Store," *MALCOM Indones. J. Mach. Learn. Comput. Sci.*, vol. 2, no. April, pp. 47–54, 2022.

[7] I. J. Muhamad Ilmar Rifaldi, Yudhi Raymond Ramadhan., "Analisis Sentimen terhadap Aplikasi ChatGPT pada Twitter Menggunakan Algoritma Naïve Bayes," *J. Sains Komput. Inform.*, vol. 7, no. September, pp. 802–814, 2023.

[8] F. N. Hasan and M. Dwijayanti, "Analisis Sentimen Ulasan Pelanggan terhadap Layanan Grab Indonesia Menggunakan Multinominal Naïve Bayes Classifier," *J. Linguist. komputasional*, vol. 4, no. 2, pp. 52–58, 2021.

[9] S. M. Siroj, I. Arwani, and D. E. Ratnawati, "Analisis Sentimen Opini Publik pada Twitter terhadap Efek Pembelajaran Daring di Universitas Brawijaya menggunakan Metode K-Nearest Neighbor," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 5, no. 7, pp. 3131–3140, 2021.

[10] R. Sistem *et al.*, "Analisis Sentimen dan Pemodelan Topik Pariwisata Lombok," *J. RESTI*, vol. 1, no. 10, pp. 123–131, 2021.

[11] T. Krisdiyanto, E. Maricha, and O. Nurharyanto, "Analisis Sentimen Opini Masyarakat Indonesia Terhadap Kebijakan PPKM pada Media Sosial Twitter Menggunakan Naïve Bayes Clasifiers," *J. CoreIT*, vol. 7, no. 1, pp. 32–37, 2021.

[12] A. Kusneti, L., Ratu, A., Wijaya, "Analisis Sentimen Ulasan Aplikasi Linkedin Dalam Google Play Store Dengan Model Naïve Bayes," *Djtechno J. Teknol. Inf.*, vol. 4, no. 2, pp. 374–385, 2023, doi: 10.46576/djtechno.

[13] E. Indrayuni and M. Informatika, "Komparasi

Algoritma *Naive Bayes* dan Support Vector Machine Untuk Analisa Sentimen Review Film," vol. 14, no. 2, pp. 175–180, 2018.

[14] F. Sidik, I. Suhada, A. H. Anwar, and F. N. Hasan, "Analisis Sentimen Terhadap Pembelajaran Daring dengan Algoritma Naïve Bayes Classifier," *J. Linguist. komputasional*, vol. 5, no. 1, pp. 34–43, 2022.

[15] P. Yuniar, "Analisis Sentimen Ulasan pada Gojek Menggunakan Metode Naïve Bayes," *Statistika*, vol. 23, no. 2, pp. 164–175, 2023.