Department of Digital Business
Journal of Artificial Intelligence and Digital Business (RIGGS)
Homepage: https://journal.ilmudata.co.id/index.php/RIGGS
Vol. 5 No. 2 (2026) pp: 26424-26432
P-ISSN: 2963-9298, e-ISSN: 2963-914X

Perbandingan Kinerja Naïve Bayes dan Logistic Regression pada Analisis Sentimen Access by KAI

Nadya Mayla Putri Rahmadani, Muhamad Rizaludin
Prodi Teknologi Informasi, Fakultas Sains dan Teknologi, Institut Teknologi dan Sains Nadhlatul Ulama Pekalongan
nadyamaylaptrr@gmail.com*, rizal.lonly@gmail.com

Abstrak

Penelitian ini membandingkan kinerja algoritma Naive Bayes dan Logistic Regression dalam analisis sentimen ulasan pengguna aplikasi Access By KAI yang diperoleh dari Google Play Store. Data dikumpulkan melalui metode web scraping pada periode 30 Oktober 2025 hingga 30 Desember 2025, kemudian dilakukan tahapan preprocessing yang meliputi cleaning, case folding, tokenizing, stopword removal, stemming, dan pembobotan TF-IDF. Dataset diklasifikasikan ke dalam dua kelas sentimen, yaitu positif dan negatif, lalu dievaluasi menggunakan RapidMiner dengan metrik confusion matrix yang mencakup accuracy, precision, recall, dan F1-score. Hasil penelitian menunjukkan bahwa Naive Bayes memperoleh akurasi sebesar 100, sedangkan Logistic Regression memperoleh akurasi sebesar 93,91. Pada model Logistic Regression, seluruh ulasan positif berhasil diklasifikasikan dengan benar, namun masih terdapat beberapa ulasan negatif yang salah diklasifikasikan sebagai positif. Berdasarkan hasil tersebut, Naive Bayes menunjukkan performa yang lebih baik pada dataset ini dan dapat direkomendasikan sebagai metode yang lebih efektif untuk klasifikasi sentimen ulasan pengguna Access By KAI.

Kata kunci: *Analisis Sentimen, Access By KAI, Naive Bayes, Logistic Regression.*

1. Latar Belakang

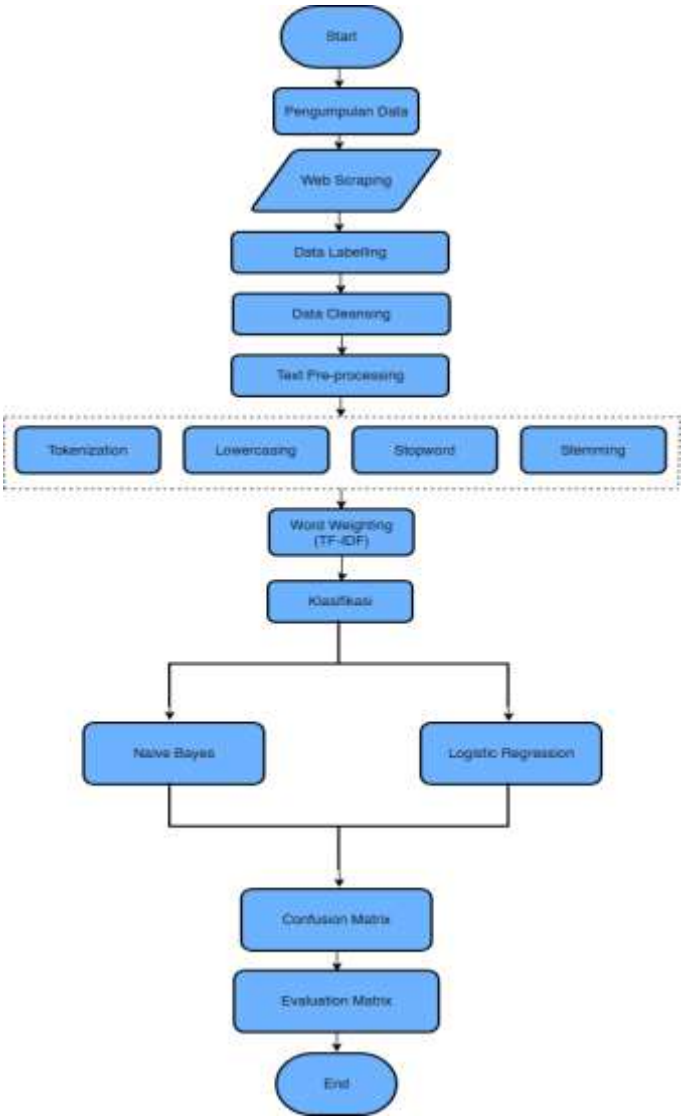
Perkembangan teknologi digital telah mendorong transformasi layanan publik, termasuk sektor transportasi, ke arah layanan berbasis aplikasi yang lebih cepat dan efisien [1]. Dalam konteks ini, Access By KAI menjadi kanal utama pemesanan tiket dan akses informasi perjalanan bagi pengguna kereta api di Indonesia [2]. Ulasan pengguna pada Google Play Store menjadi sumber informasi penting untuk mengevaluasi kualitas layanan karena memuat pengalaman langsung pengguna terhadap stabilitas sistem, kemudahan penggunaan, dan keandalan transaksi [3].

Skripsi sumber artikel ini menjelaskan bahwa Access By KAI masih menerima berbagai keluhan, seperti kegagalan transaksi, error saat login, masalah pembayaran, dan ketidakstabilan aplikasi [4]. Kondisi tersebut memperlihatkan bahwa persepsi pengguna terhadap layanan digital belum sepenuhnya positif dan perlu dianalisis secara sistematis melalui pendekatan *text mining* dan analisis sentimen [5]. Analisis sentimen memungkinkan pengelompokan opini pengguna ke dalam kelas sentimen tertentu sehingga dapat digunakan sebagai dasar evaluasi kualitas layanan aplikasi [6], [7].

Penelitian ini berfokus pada perbandingan dua algoritma klasifikasi, yaitu *Naive Bayes* dan *Logistic Regression*, untuk mengidentifikasi metode yang lebih baik dalam mengklasifikasikan sentimen ulasan pengguna Access By KAI [8]. Kedua algoritma dipilih karena umum digunakan pada data teks, relatif efisien, dan memiliki performa yang kompetitif pada klasifikasi sentimen berbahasa Indonesia [9], [10]. Tujuan penelitian adalah membandingkan nilai *accuracy, precision, recall*, dan *F1-score* dari kedua algoritma pada dataset ulasan pengguna aplikasi Access By KAI.

2. Metode Penelitian

Metode penelitian pada artikel ini menjelaskan tahapan yang dilakukan untuk mengolah ulasan pengguna aplikasi Access By KAI menjadi data yang siap diklasifikasikan [11]. Tahapan tersebut meliputi pengumpulan data dari Google Play Store melalui *web scraping*, pelabelan sentimen, prapemrosesan teks, pembobotan fitur menggunakan TF-IDF, serta pemodelan dan evaluasi menggunakan algoritma *Naive Bayes* dan *Logistic Regression*[12], [13]. Perhatikan Gambar 1.



Gambar 1 *Diagram Flowchart* Alur Penelitian

Dengan susunan metode ini, proses penelitian dapat dijelaskan secara sistematis dan hasil perbandingan kinerja kedua algoritma dapat dianalisis secara objektif.

2.1. Pendekatan Penelitian

Penelitian ini menggunakan pendekatan kuantitatif dengan analisis sentimen berbasis *machine learning*. Fokus penelitian adalah mengklasifikasikan ulasan pengguna ke dalam dua kelas, yaitu sentimen positif dan negatif, kemudian membandingkan performa *Naive Bayes* dan *Logistic Regression* menggunakan metrik evaluasi klasifikasi.

2.2. Sumber dan Pengumpulan Data

Sumber data penelitian adalah ulasan pengguna aplikasi Access By KAI yang diambil dari Google Play Store. Pengumpulan data dilakukan menggunakan teknik *web scraping* dengan bantuan *library google-play-scraper* [14].

Dari proses scraping awal diperoleh 116.160 data ulasan, kemudian data difilter sesuai periode penelitian 30 Oktober 2025 sampai 30 Desember 2025 sehingga diperoleh 2.741 data ulasan yang relevan..

Pada tahap *filtering*, hanya atribut yang relevan untuk analisis yang dipertahankan, yaitu tanggal ulasan, isi ulasan, dan rating. Hasil *filtering* ini kemudian disimpan ke dalam format *CSV* untuk memudahkan proses pengolahan lanjutan pada tahap pelabelan dan pemodelan [15].

DOI: https://doi.org/10.31004/riggs.v5i2.13244
Lisensi: Creative Commons Attribution 4.0 International (CC BY 4.0)

2.3. Pelabelan Data

Pelabelan sentimen dilakukan berdasarkan *rating* yang diberikan pengguna. *Rating* 4 dan 5 dikategorikan sebagai sentimen positif, *rating* 3 sebagai sentimen netral, sedangkan *rating* 1 dan 2 dikategorikan sebagai sentimen negatif. Skema pelabelan ini digunakan agar proses klasifikasi memiliki target kelas yang jelas dan konsisten.

Setelah proses pelabelan, data netral dihapus agar penelitian berfokus pada klasifikasi biner antara sentimen positif dan negatif. Setelah penghapusan data netral, jumlah data yang digunakan pada tahap berikutnya menjadi 2.554 ulasan berlabel.

2.4. Prapemrosesan Text

Sebelum dilakukan pemodelan, data teks menjalani tahapan pembersihan dan normalisasi. Tahap ini meliputi *cleaning* untuk menghapus URL, tanda baca, angka, emoji, dan spasi berlebih, kemudian *case folding* untuk mengubah seluruh huruf menjadi huruf kecil. Tahap awal ini bertujuan agar data menjadi lebih seragam dan mengurangi *noise.*

Tahap berikutnya adalah *tokenizing*, *stopword removal*, dan *stemming. Tokenizing* memecah teks menjadi kata-kata individual, *stopword removal* menghapus kata-kata umum yang kurang informatif, sedangkan *stemming* mengubah kata berimbuhan menjadi bentuk dasar.

2.5. Pembobotan Fitur TF-IDF

Setelah *preprocessing* selesai, teks diubah menjadi vektor numerik menggunakan metode *TF-IDF*. Metode ini memberikan bobot lebih tinggi pada kata yang penting dalam dokumen tetapi tidak terlalu sering muncul di seluruh korpus.

Proses ini mencatat sepuluh kata dengan bobot *TF-IDF* tertinggi. Bobot ini menjadi fitur utama yang dipakai oleh model *Naive Bayes* dan *Logistic Regression*

2.6. Pemodelan dan Evaluasi

Pemodelan dilakukan menggunakan RapidMiner dengan alur *Retrieve*, *Process Documents from Data*, *Set Role*, *Validation*, *Apply Model*, dan *Performance*. Kedua model dibangun menggunakan fitur yang sama agar hasil evaluasi dapat dibandingkan secara adil. Pada Proses ini data dibagi menjadi data latih dan data uji dengan proporsi 80:20. Evaluasi model dilakukan menggunakan *confusion matrix* dan *metrik accuracy*, *precision*, *recall*, serta *F1-score*. Dengan pendekatan ini, performa masing-masing model dapat diamati secara rinci pada kedua kelas sentimen.

3. Hasil dan Diskusi

3.1. Hasil Pengumpulan dan Seleksi Data

Hasil *scraping* awal menghasilkan 116.160 ulasan pengguna Access By KAI dari Google Play Store. Proses ditunjukan pada Gambar 2.

```
reviews = reviews_all(
    'com.kai.kaiticketing',
    sleep_milliseconds=0,
    lang='id',
    country='id',
    sort=Sort.NEWEST
)

print("Jumlah review berhasil diambil:", len(reviews))
print(reviews[:2])

df = pd.DataFrame(reviews)
print(df.columns.tolist())
df.head()
```

Gambar 2. Proses Pengumpulan Data

Setelah itu data di *filter* berdasarkan periode penelitan. Perhatikan Gambar 3 dibawah ini.

DOI: https://doi.org/10.31004/riggs.v5i2.13244
Lisensi: Creative Commons Attribution 4.0 International (CC BY 4.0)

```
df_filtered = df[
    (df['at'] >= '2025-10-30') &
    (df['at'] < '2025-12-31')
]

df_filtered = df_filtered[['at', 'content', 'score']]
df_filtered.head()
```

```
print("Jumlah data setelah filter tanggal:", len(df_filtered))
```

```
Jumlah data setelah filter tanggal: 2741
```

Gambar 3. Proses *Filter Data*

Setelah *filtering* berdasarkan periode penelitian, jumlah data yang digunakan menjadi 2.741 ulasan. Tahap ini penting karena memastikan data yang dianalisis relevan dengan batas waktu penelitian yang telah ditetapkan. Selanjutnya dilakukan proses *labelling* data. Perhatikan Gambar 4.

```
# Buat label sentimen
def label_sentimen(score):
    if score >= 4:
        return 'positif'
    elif score == 3:
        return 'netral'
    else:
        return 'negatif'

df_filtered['sentiment'] = df_filtered['score'].apply(label_sentimen)

# Simpan ke CSV
df_filtered.to_csv('ulasan_access_by_kai_labeled.csv', index=False)

print("CSV berhasil disimpan")
print(df_filtered.head())
```

```
CSV berhasil disimpan
```

Gambar 4. Proses *Labelling Data*

Selanjutnya dilakukan proses menghapus data netral karena yang dipakai hanya data positif dan negatif, untuk prosesnya perhatikan Gambar 5.

```
# Menghapus data dengan sentimen netral
df_binary = df_filtered[df_filtered['sentiment'] != 'netral'].copy()

# Simpan ke CSV
df_binary.to_csv('ulasan_access_by_kai_binary.csv', index=False)

print("CSV berhasil disimpan")
print(df_binary.head())
```

Gambar 5. Proses Menghapus Data Netral

Setelah proses pelabelan sentimen dan penghapusan data netral pada Gambar 4 dan Gambar 5 diatas, jumlah data akhir yang digunakan dalam pemodelan menjadi 2.554 ulasan. Data inilah yang kemudian diproses lebih lanjut melalui *cleaning*, *preprocessing*, pembobotan *TF-IDF*, dan klasifikasi model.

3.2. Hasil Prapemrosesan dan TF-IDF

Tahap *preprocessing* menghasilkan data teks yang lebih bersih dan seragam. Langkah awal dilakukan proses *cleaning* data untuk menghapus URL, tanda baca, angka, emoji, dan spasi berlebih, dan *case folding.* Perhatikan Gambar 6.

DOI: https://doi.org/10.31004/riggs.v5i2.13244
Lisensi: Creative Commons Attribution 4.0 International (CC BY 4.0)

```python
import pandas as pd
import re

df = pd.read_csv('ulasan_access_by_kai_labeled.csv')

def clean_text(text):
    text = str(text).lower()
    text = re.sub(r'http\\S+|www\\S+', ' ', text)
    text = re.sub(r'[^a-zA-Z\\s]', ' ', text)
    text = re.sub(r'\\s+', ' ', text).strip()
    return text

df['clean_content'] = df['content'].apply(clean_text)

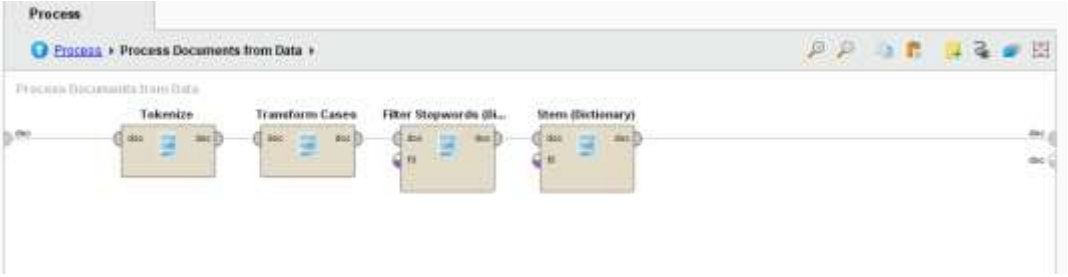
df = df[df['clean_content'].str.strip() != '']

df.to_csv('ulasan_access_by_kai_cleaned.csv', index=False)

print(df[['content', 'clean_content']].head())
print("Jumlah data setelah cleaning:", len(df))
```

Gambar 6. Proses *Cleaning Data*

Setelah *cleaning data*, dilakukan *PreProcessing* Teks di Rapidminer meliputi *tokenizing*, *transform case*, *stopword removal*, dan *stemming.* Perhatikan Gambar 7.



Gambar 7. Proses *PreProcessing* Teks

Pada contoh *preprocessing*, kalimat ulasan yang semula panjang dan penuh kata umum berhasil diringkas menjadi token dasar yang lebih relevan untuk analisis. Perhatikan Tabel 1 dibawah ini.

Tabel 1. Contoh *PrePocessing*

Tahap	Sebelum	Sesudah
Data asli	kenapa setiap pindah aplikasi selalu dianggap keluar jd harus daftar dr awal. sangat merepotkan, baiknya kasih jeda waktu karena tidak semua calon penumpang sudah terdaftar.	kenapa setiap pindah aplikasi selalu dianggap keluar jd harus daftar dr awal. sangat merepotkan, baiknya kasih jeda waktu karena tidak semua calon penumpang sudah terdaftar.
Cleaning / case folding	kenapa setiap pindah aplikasi selalu dianggap keluar jd harus daftar dr awal. sangat merepotkan, baiknya kasih jeda waktu karena tidak semua calon penumpang sudah terdaftar.	kenapa setiap pindah aplikasi selalu dianggap keluar jd harus daftar dr awal sangat merepotkan baiknya kasih jeda waktu karena tidak semua calon penumpang sudah terdaftar
Tokenizing	kenapa setiap pindah aplikasi selalu dianggap keluar jd harus daftar dr awal sangat merepotkan baiknya kasih jeda waktu karena tidak semua calon penumpang sudah terdaftar	kenapa, setiap, pindah, aplikasi, selalu, dianggap, keluar, jd, harus, daftar, dr, awal, sangat, merepotkan, baiknya, kasih, jeda, waktu, karena, tidak, semua, calon, penumpang, sudah, terdaftar
Stopword removal	kenapa, setiap, pindah, aplikasi, selalu, dianggap, keluar, jd, harus, daftar, dr, awal, sangat, merepotkan, baiknya, kasih, jeda, waktu, karena, tidak, semua, calon, penumpang, sudah, terdaftar	pindah, aplikasi, dianggap, keluar, daftar, awal, merepotkan, kasih, jeda, waktu, calon, penumpang, terdaftar
Stemming	pindah, aplikasi, dianggap, keluar, jd, daftar, dr, awal, merepotkan, kasih, jeda, waktu, calon, penumpang, terdaftar	pindah, aplikasi, anggap, keluar, jd, daftar, dr, awal, repot, kasih, jeda, waktu, calon, penumpang, daftar

DOI: https://doi.org/10.31004/riggs.v5i2.13244
Lisensi: Creative Commons Attribution 4.0 International (CC BY 4.0)

Hasil ini menunjukkan bahwa *preprocessing* berperan penting dalam mengurangi noise dan meningkatkan kualitas representasi teks.

Pada tahap pembobotan TF-IDF, kata-kata seperti “bagus”, “sangat”, “aplikasi”, “mantap”, “mudah”, dan “baik” muncul sebagai fitur dominan. Seperti yang ditunjukan pada Gambar 8 dan Tabel 2 dibawah ini.



Gambar 8. *Wordcloud*

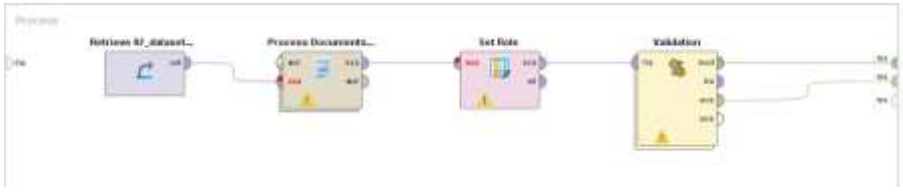
Tabel 2. Hasil Pembobotan Fitur *TF-IDF*

Kata	Nilai
bagus	0.053826
sangat	0.045388
aplikasi	0.042205
mantap	0.042096
mudah	0.038149
bantu	0.037925
baik	0.033381
buka	0.032843
tiket	0.029176
pesan	0.027766

Dominasi kata-kata tersebut memperlihatkan bahwa ulasan pengguna banyak memuat kata evaluatif yang kuat, sehingga *TF-IDF* menjadi representasi yang sesuai untuk klasifikasi sentimen pada penelitian ini.

3.3. Hasil Klasifikasi Naïve Bayes

Pemodelan *Naïve Bayes* di Rapidminer ditunjukan pada Gambar 9.



Gambar 9. Pemodelan *Naïve bayes* di Rapidminer

DOI: https://doi.org/10.31004/riggs.v5i2.13244
Lisensi: Creative Commons Attribution 4.0 International (CC BY 4.0)

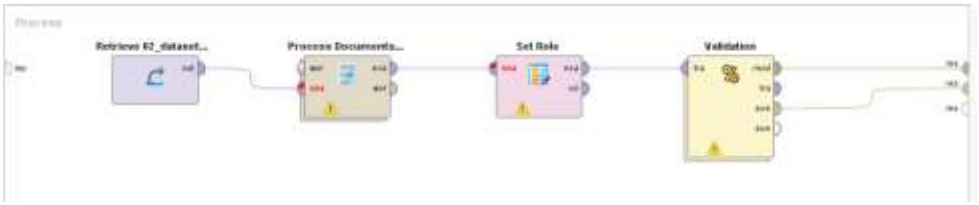
Pada model *Naive Bayes*, hasil *confusion matrix* menunjukkan bahwa seluruh 323 ulasan positif diprediksi dengan benar sebagai positif dan seluruh 186 ulasan negatif diprediksi dengan benar sebagai negatif. Dengan hasil tersebut, model ini memperoleh *accuracy* 100 serta nilai *precision*, *recall*, dan *F1-score* yang sempurna pada data uji. Perhatikan Gambar 10.



Gambar 10. Hasil *Confusion Matrix Naïve Bayes*

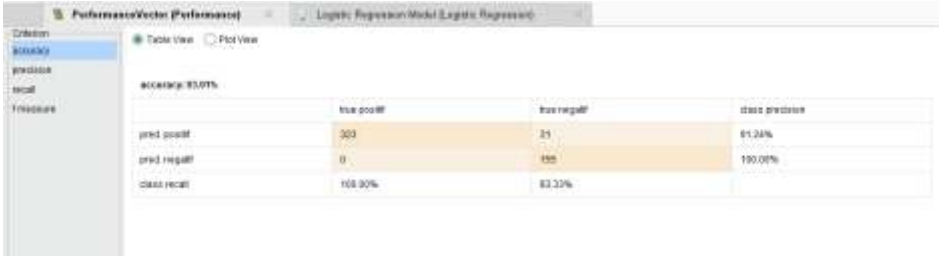
3.4. Hasil Klasifikasi Logistic Regression

Pemodelan *Logistic Regression* di Rapidminer ditunjukan pada Gambar 11.



Gambar 11. Pemodelan *Logistic Reggression* di Rapidminer

Model *Logistic Regression* juga menunjukkan performa yang tinggi, tetapi masih lebih rendah dibanding *Naive Bayes*. Hasil *confusion matrix* menunjukkan bahwa seluruh 323 ulasan positif diprediksi benar sebagai positif, sedangkan dari 186 ulasan negatif terdapat 155 ulasan yang diprediksi benar sebagai negatif dan 31 ulasan yang salah diklasifikasikan sebagai positif. Perhatikan Gambar 12.



Gambar 12. Hasil *Confusion Matrix Logistic Regression*

Akibatnya, *Logistic Regression* memperoleh *accuracy* 93,91. *Recall* kelas positif mencapai 100, tetapi *recall* kelas negatif turun menjadi 83,33 karena masih terdapat ulasan negatif yang gagal dikenali dengan benar. Temuan ini menunjukkan bahwa model lebih baik dalam mengenali sentimen positif dibanding sentimen negatif pada dataset penelitian ini

3.5. Perbandingan Kinerja Model

Perbandingan hasil evaluasi kedua model dapat diringkas sebagaimana tabel 3 dibawah ini.

DOI: https://doi.org/10.31004/riggs.v5i2.13244
Lisensi: Creative Commons Attribution 4.0 International (CC BY 4.0)

Tabel 3. Perbandingan Hasil Confusion Matrix Pada Kedua Algoritma

Metrik	*Naïve Bayes*	*Logistic Regression*
Akurasi	100,00%	93,91%
Presisi kelas positif	100,00%	91,24%
***Recall* kelas positif**	100,00%	100,00%
Presisi kelas negatif	100,00%	100,00%
***Recall* kelas negatif**	100,00%	83,33%
***F1-score* kelas negatif**	100,00%	90,91%

Berdasarkan tabel tersebut, *Naive Bayes* lebih unggul dibanding *Logistic Regression* pada seluruh metrik utama, terutama pada kemampuan mengenali kelas negatif. Keunggulan ini menunjukkan bahwa kombinasi *preprocessing, TF-IDF*, dan mekanisme probabilistik *Naive Bayes* lebih sesuai dengan karakteristik data ulasan Access By KAI pada penelitian ini

3.6. Pembahasan Hasil

Secara metodologis, hasil penelitian memperlihatkan bahwa kualitas data dan tahapan *preprocessing* memiliki peran besar dalam performa model. *Cleaning, tokenizing, stopword removal, stemming,* dan *TF-IDF* membantu mengubah ulasan mentah menjadi fitur yang lebih representatif, sehingga model dapat mengenali pola sentimen dengan lebih baik.

Dari sisi praktis, hasil ini menunjukkan bahwa *Naive Bayes* dapat dijadikan model awal yang efisien untuk membangun sistem pemantauan sentimen otomatis pada aplikasi layanan publik seperti Access By KAI. Meskipun *Logistic Regression* tetap menunjukkan performa tinggi, model ini pada penelitian ini masih cenderung kurang kuat dalam mendeteksi sebagian ulasan negatif

4. Kesimpulan

Berdasarkan hasil penelitian terhadap ulasan pengguna aplikasi *Access by KAI* periode 30 Oktober 2025 sampai 30 Desember 2025, metode *Naïve Bayes* dan *Logistic Regression* sama-sama dapat digunakan untuk klasifikasi sentimen ulasan pengguna, namun *Naïve Bayes* menunjukkan performa terbaik dengan *accuracy* 100% dan hasil evaluasi yang lebih seimbang pada seluruh metrik, sedangkan *Logistic Regression* memperoleh *accuracy* 93,91% dan masih mengalami kesalahan klasifikasi pada sebagian ulasan negatif, sehingga *Naïve Bayes* lebih direkomendasikan sebagai model utama pada penelitian ini. Untuk penelitian selanjutnya, disarankan menggunakan data ulasan yang lebih terbaru dengan periode pengambilan yang lebih luas serta membandingkan metode *Naïve Bayes* dan *Logistic Regression* dengan algoritma lain, seperti *K-Nearest Neighbor (KNN), Random Forest*, dan *Support Vector Machine (SVM)*, agar diperoleh hasil analisis sentimen yang lebih komprehensif dan akurat.

Referensi

[1] J. Yumame, "Analisis Kualitas Pelayanan Publik di Era Digital: Implementasi dan Tantangan," 2024. [Online]. Available: https://attractivejournal.com/index.php/bce/

[2] Ariyanti and Susanti, "Identifikasi Pengguna Aplikasi Transportasi Access by KAI dengan Ulasan dan Rating Menggunakan Analisis Sentimen," 2024.

[3] Sulaeman and Sukarsa, "mrizal1,+79.+Penelitian+S5+-+Pengembangan+Aplikasi+Mobile+Berbasis+Kecerdasan+Buatan+untuk+Meningkatkan+Efisiensi+Proses+Bisnis," 2024.

[4] R. Damanhuri and V. A. Husein, "Analisis Sentimen pada Ulasan Aplikasi Access by KAI Berbahasa Indonesia Menggunakan Word-Embedding dan Classical Machine Learning," *Jurnal Masyarakat Informatika*, vol. 15, no. 2, pp. 97–106, Nov. 2024, doi: 10.14710/jmasif.15.2.62383.

[5] M. R. Hanafi and R. K. R, "Analisis Sentimen pada Ulasan Aplikasi Sirekap di Google Play Menggunakan Algoritma Naive Bayes," *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, vol. 4, no. 4, pp. 1578–1586, Oct. 2024, doi: 10.57152/malcom.v4i4.1693.

[6] B. Agustian, D. N. Hadimin, U. Sangga, B. Ypkp, and K. Bandung, "Deni Nurdyana Hadimin / Analisis Efektivitas Penerapan Mobile Application KAI Access Di PT Kereta Api Indonesia (Persero) 19," 2022.

[7] S. Fauziah *et al.*, "KOMPARASI METODE FEATURE SELECTION TEXT MINING PADA PERMASALAHAN KLASIFIKASI KELUHAN PELANGGAN INDUSTRI TELEKOMUNIKASI MENGGUNAKAN SMOTE DAN NAÏVE BAYES COMPARISON OF TEXT MINING FEATURE SELECTION METHODS IN THE CLASSIFICATION OF CUSTOMER COMPLAINTS IN THE TELECOMMUNICATION INDUSTRY USING SMOTE AND NAÏVE BAYES," 2023.

[8] neal Efrata Madao, A. Irsyad, and M. Rivani Ibrahim, "ANALISIS SENTIMEN PADA ULASAN PENGGUNA APLIKASI JAMSOSTEK MOBILE DENGAN MENGGUNAKAN NAIVE BAYES DAN LOGISTIC REGRESSION," *Jurnal Teknologi Informasi*, vol. 6, no. 2, 2025, doi: 10.46576/djtechno.

[9] M. Rifqi Mikatama and M. Reza, "ANALISIS SENTIMEN TERHADAP WARGA NET PADA LAGA AFC U23 INDONESIA VS KOREA SELATAN," 2024.

DOI: https://doi.org/10.31004/riggs.v5i2.13244
Lisensi: Creative Commons Attribution 4.0 International (CC BY 4.0)

[10] KHUSNU NOVA LUTHFI, "ANALISIS SENTIMEN PENGGUNA APLIKASI ACCESS BY KAI," 2025.
[11] V. B. Lestari and C. A. Hutagalung, "J-KOMA Journal of Computer Science and Applications Evaluation of TF-IDF Extraction Techniques in Sentiment Analysis of Indonesian-Language Marketplaces Using SVM, Logistic Regression, and Naive Bayes," pp. 22–2025, 2023, doi: 10.21009/j.
[12] R. Marta Dinata, E. Rayhana, V. Hadi, and U. Alkaf, "ANALISIS KOMPREHENSIF KINERJA MODEL KLASIFIKASI SENTIMEN: EVALUASI LINTAS METRIK PADA DATASET TWEET FILM BAHASA INDONESIA," *Jurnal Rekayasa Informasi*, vol. 14, no. 1, p. 2025, 2024.
[13] S. A. H. Bahtiar, C. K. Dewa, and A. Luthfi, "Comparison of Naïve Bayes and Logistic Regression in Sentiment Analysis on Marketplace Reviews Using Rating-Based Labeling," *Journal of Information Systems and Informatics*, vol. 5, no. 3, pp. 915–927, Aug. 2023, doi: 10.51519/journalisi.v5i3.539.
[14] R. Vikri, A. E. Al, R. V. Alif, and A. H. Hasugian, "Analisis Sentimen Ulasan Pelanggan Aplikasi Hokben Di Google Playstore Menggunakan Metode Support Vector Machine," vol. 07, no. 02, 2025, doi: 10.54209/jatilima.v7i02.1501.
[15] N. Zeliana Putri, A. Rinaldi Dikananda, and A. Rifa, "Jurnal Informatika Terpadu ANALISIS SENTIMEN APLIKASI SEABANK DENGAN ALGORITMA NAIVE BAYES UNTUK OPTIMALISASI PELAYANAN," *Jurnal Informatika Terpadu*, vol. 11, no. 1, pp. 55–62, 2025, [Online]. Available: https://journal.nurulfikri.ac.id/index.php/JIT

DOI: https://doi.org/10.31004/riggs.v5i2.13244
Lisensi: Creative Commons Attribution 4.0 International (CC BY 4.0)