JuSIN
JURNAL SISTEM INFORMASI

Website: ojs.itb-ad.ac.id/index.php/JUSIN
E-ISSN: 2797-8516

ANALISIS SENTIMEN OPINI MASYARAKAT TERKAIT PENGGUNAAN *MOBILE* JKN PADA MEDIA SOSIAL X DENGAN PERBANDINGAN ALGORITMA *K-NEAREST NEIGHBOR* (KNN) DAN *LOGISTIC REGRESSION* (LR)

Kelvin Valensio Hematang[(1)], Nengah Widya Utami[2] , I Nyoman Purnama[3]

[1] *Universitas Primakara, Bali*
[2] *Universitas Primakara, Bali*
[3] *Universitas Primakara, Bali*

Abstract

BPJS Kesehatan launched the Mobile JKN application in 2017 to enhance healthcare accessibility and simplify membership management. As of September 2024, 98.67% of Indonesia's population has joined the JKN program, highlighting the importance of public opinion in the application's development. This study analyzes public sentiment toward Mobile JKN through the social media platform X using the K-Nearest Neighbor (KNN) and Logistic Regression (LR) algorithms. From 4,777 reviews, 4,648 clean data were obtained after Preprocessing and labeled using IndoBERT, resulting in 2,066 positive reviews and 2,581 negative reviews. The Evaluation results show that KNN achieved an accuracy of 83.87%, while LR performed better with 87.85% along with higher precision, recall, and f1-score.
The findings reveal the dominance of negative sentiment in user reviews and provide insights for BPJS Kesehatan to improve Mobile JKN into a more responsive application that meets users' needs and feedback.

Kata Kunci: *Mobile JKN, sentiment analysis, social media X, K-Nearest Neighbor (KNN), Logistic Regression (LR)*

Informasi Artikel:
Dikirim : 21 September 2025
Ditelaah: 22 September 2025
Diterima: 30 September 2025
Publikasi : 23 Desember 2025

Juli – Desember 2025, Vol 6 (2) : hlm 131-146

(*) Korespondensi: kelfinvalensio@gmail.com *(Kelvin Hematang)*

PENDAHULUAN

Seiring perkembangan teknologi, layanan kesehatan di Indonesia terus berinovasi untuk mempermudah akses masyarakat. Pemerintah melalui BPJS Kesehatan mengelola program Jaminan Kesehatan Nasional–Kartu Indonesia Sehat (JKN-KIS) guna menjamin akses layanan kesehatan bagi seluruh penduduk. Hingga September 2024, jumlah peserta JKN mencapai 277.143.330 orang (BPJS, 2024). Untuk meningkatkan kualitas layanan, BPJS meluncurkan aplikasi *Mobile* JKN pada 2017 sebagai sarana digital bagi peserta dalam mengakses informasi kepesertaan dan layanan kesehatan (Putri, et al., 2022).

Cakupan kepesertaan JKN yang mencapai 98,67 persen dari penduduk Indonesia (Antaranews, 2024) menegaskan pentingnya keberlanjutan layanan digital seperti *Mobile* JKN. Analisis sentimen diperlukan untuk memahami opini masyarakat karena berpengaruh terhadap pengembangan layanan. Analisis sentimen merupakan metode untuk mengidentifikasi pandangan publik terhadap produk atau layanan melalui opini daring (Pamungkas dan Kharisudin, 2021). Media sosial X dengan 24,85 juta pengguna di Indonesia menjadi sumber data relevan karena banyak digunakan untuk menyampaikan pengalaman dan ulasan terkait layanan kesehatan (We Are Social dan Meltwater, 2024).

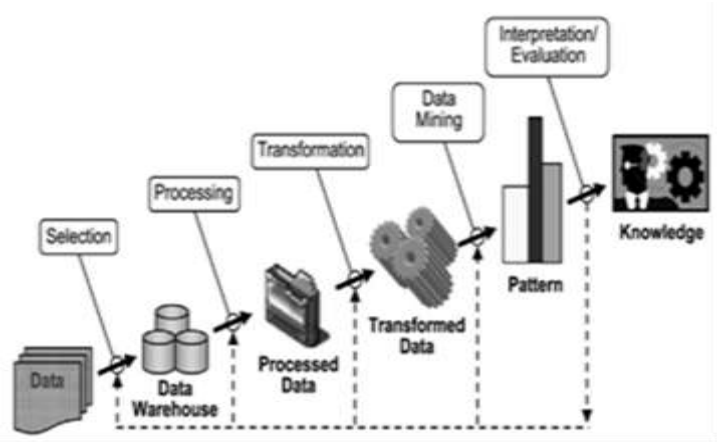
Dalam mengolah opini di media sosial, data mining berperan untuk mengekstraksi pola dari data tidak terstruktur menjadi informasi bermakna (Dewi, et al., 2022). Salah satu algoritma yang banyak digunakan adalah *K-Nearest Neighbor* (KNN), yang bekerja dengan mengelompokkan data berdasarkan kedekatan jarak antar-poin (Legito, et al., 2023). *Logistic Regression* (LR) juga umum dipakai dalam klasifikasi teks karena kemampuannya memodelkan hubungan variabel input dengan kelas target menggunakan fungsi logistik (Satviki dan Sanjaya, 2024).

Penelitian sebelumnya mendukung efektivitas metode ini. KNN mencapai akurasi 71,39% pada analisis ulasan aplikasi Tokopedia (Afdal, et al., 2022). Pada ulasan agen perjalanan online, KNN lebih unggul dibanding Naïve Bayes (Sholeha, et al., 2022). Sementara itu, penelitian sentimen terhadap film Sri Asih menunjukkan *Logistic Regression* (81%) lebih baik dibanding CNN (78%) dan KNN (61%) (Dwi Wijaya, 2023).

Berdasarkan uraian tersebut, penelitian ini menganalisis sentimen masyarakat terhadap aplikasi *Mobile* JKN melalui data media sosial X dengan membandingkan kinerja *K-Nearest Neighbor* dan *Logistic Regression*. Tujuannya untuk mengetahui opini publik, mengidentifikasi pola sentimen, serta menentukan algoritma paling optimal dalam klasifikasi sentimen.

METODE

Penelitian ini dijalankan dengan memanfaatkan metode *Knowledge Discovery from Data* (KDD). Urutan proses yang terlibat dalam KDD dapat dilihat dalam ilustrasi berikut.:



Gambar 1. Proses *Knowledge Discovery from Data*

Berikut adalah penjelasan dari gambar diatas terkait tahapan proses *Knowledge Discovery from Data* (KDD):

1.*Selection*

Selection merupakan proses pertama dalam KDD di mana perlu melakukan *Selection* data dari kumpulan data yang ada menuju tahap penggalian informasi.

2.*Preprocessing*

Preprocessing adalah tahap penting dalam proses Knowledge Discovery in Databases (KDD) yang berfokus pada pengolahan data. Tahapan ini mencakup penghapusan data duplikat, pemeriksaan inkonsistensi data, serta perbaikan kesalahan data untuk memastikan kualitas data yang optimal sebelum analisis lebih lanjut.

3.*Transformation*

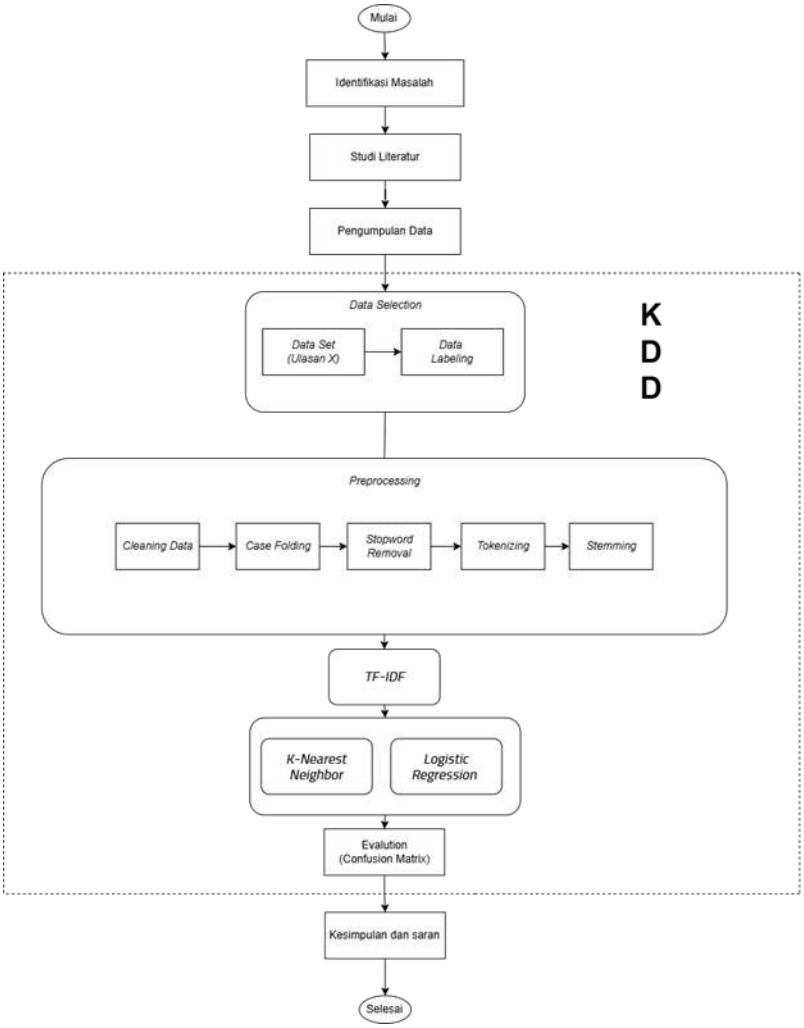
Transformation data yang telah dipilih, data yang terpilih harus diubah sesuai dengan algoritma yang akan digunakan dalam data mining.

4.Data mining

Pada tahap data mining, proses ini melibatkan pencarian pola atau informasi yang relevan dan menarik dalam data yang telah dipilih, menggunakan algoritma tertentu untuk mengungkap wawasan yang tersembunyi.

5.*Evaluation*

Pada tahap ini, pola informasi yang diperoleh dari proses data mining perlu dievaluasi. Pengetahuan yang diperoleh harus disajikan dalam bentuk yang lebih mudah dipahami oleh pengguna, seperti ringkasan atau visualisasi, untuk mempermudah pemahaman dan pengambilan keputusan.



Gambar 2. Kerangka Penelitian

Berikut di bawah ini merupakan penjelasan dari alur penelitian yang terdapat pada gambar di atas:

1. Identifikasi Masalah
   Pada tahap identifikasi masalah, peneliti melakukan proses untuk menggali permasalahan yang ada pada objek penelitian, yaitu aplikasi *Mobile* JKN. Adapun permasalahan yang ada yaitu banyak keluhan terhadap fitur maupun layanan yang ada pada aplikasi *Mobile* JKN.
2. Studi Literatur
   Setelah melakukan identifikasi masalah, peneliti melakukan pengkajian teori-teori yang ada pada jurnal yang berkaitan dengan penelitian yang akan dilaksanakan. Studi literatur terkait tentang data mining, algoritma *K-Nearest Neighbor, Logistik egression*.
3. Pengumpulan Data
4. Pengumpulan data dilakukan dengan cara *Web Scrapping* dari *Google Colabs*. Dengan jumlah data 4648 yang di ambil dari bulan Januari 2024 hingga bulan November 2024.
5. *Data Selection*
   Data Selection adalah proses memilih data relevan dari kumpulan besar agar analisis lebih efisien dan berkualitas. Pada penelitian ini, data diambil dari media sosial X

melalui web scraping berupa opini masyarakat tentang aplikasi *Mobile* JKN. Pelabelan dilakukan dengan *IndoBERT* yang dioptimalkan untuk klasifikasi sentimen Bahasa Indonesia (positif, netral, negatif). Metode ini lebih akurat dan efisien dibanding berbasis kata kunci karena *IndoBERT* mampu memahami konteks, struktur bahasa, serta makna kalimat, termasuk yang ambigu.

HASIL DAN PEMBAHASAN

Pada tahapan *Preprocessing*, dataset akan dilakukan pembersihan data yang rusak atau tidak lengkap dengan cara dihapus atau diperbaiki. Tahapan-tahapan yang dilakukan dalam *Preprocessing* adalah sebagai berikut.

1. *Cleaning* Data

   Cleaning data yang dilakukan pada penelitian ini yaitu proses penghapusan data. Pada proses ini ada 1 data yang terhapus karena tidak sesuai dengan elemen penelitian dan data ada yang kosong yang menyebabkan data tersebut dihapus.

Tabel 1 hasil *Cleaning* Data

No	Ulasan	Sentimen
0	*Mobile* jkn kenapa yu error ah rese skali	negative
1	ka tp ak ga bsa login *Mobile* jkn nya trs gmn yh	negative
2	halo kenapa saya gabisa daftar ya di aplikasi *Mobile* jkn	negative
3	ini kenapa otp *Mobile* jkn ga masuk masuk	negative
4	makin kesini makin upgrade fitur *Mobile* jkn jadi makin mudah dan membantubpjs bpjskesehatan	positive
4648	halo min izin nanya aku tadi daftar *Mobile* jkn tapi gabisa min karna sms nya ga masuk itu solusinya gimana ya min bpjs aku juga ilang jadi mau ngurus juga bingung kemana min terimakasih min	negative

2. *Case Folding*

   Case Folding dilakukan pada penelitian ini yaitu proses merubah seluruh data dari huruf besar menjadi huruf kecil yang bertujuan untuk mengkonsistensi representasi teks pada analisis data. Pada penelitian ini mengacu pada table 2.

Tabel 2 Hasil Case Folding

No	Ulasan	Sentimen
0	*Mobile* jkn kenapa yu error ah rese skali	negative
1	ka tp ak ga bsa login *Mobile* jkn nya trs gmn yh	negative
2	halo kenapa saya gabisa daftar ya di aplikasi *Mobile* jkn	negative
3	ini kenapa otp *Mobile* jkn ga masuk masuk	negative

4	makin kesini makin upgrade fitur *Mobile* jkn jadi makin mudah dan membantubpjs bpjskesehatan	positive
4648	halo min izin nanya aku tadi daftar *Mobile* jkn tapi gabisa min karna sms nya ga masuk itu solusinya gimana ya min bpjs aku juga ilang jadi mau ngurus juga bingung kemana min terimakasih min	negative

3. *Stopword Removal*

Stopword Removal pada penelitian ini dilakukan untuk menghapus kata kata yang tidak memiliki makna signifikan, yang bertujuan untuk meningkatkan kualitas analisis. Hasil dari *Stopword Removal* dapat dilihat pada tabel 3.

Tabel 3 Hasil Stopward Removal

No	Ulasan	Sentimen
0	*Mobile* jkn kenapa yu error ah rese skali	negative
1	ka tp ak ga bsa login *Mobile* jkn nya trs gmn yh	negative
2	halo gabisa daftar ya aplikasi *Mobile* jkn	negative
3	otp *Mobile* jkn ga masuk masuk	negative
4	kesini upgrade fitur *Mobile* jkn mudah membantubpjs bpjskesehatan	positive
4648	halo min izin nanya daftar *Mobile* jkn gabisa min karna sms nya ga masuk solusinya gimana ya min bpjs ilang ngurus bingung kemana min terimakasih min	negative

4. *Tokenizing*

Tokenizing dilakukan pada penelitian ini bertujuan untuk memisahkan kata-kata, frasa pada setiap kalimat menjadi lebih kecil yang bertujuan untuk melanjutkan pemrosesan data. Hasil dapat dilihat pada table 4 dibawah ini.

Tabel 4 Hasil *Tokenizing*

No	Ulasan	Sentimen
0	['*Mobile*', 'jkn', 'yu', 'error', 'ah', 'rese', 'skali']	negative
1	['ka', 'tp', 'ak', 'ga', 'bsa', 'login', '*Mobile*', 'jkn', 'nya', 'trs', 'gmn', 'yh']	negative
2	['halo', 'gabisa', 'daftar', 'ya', 'aplikasi', '*Mobile*', 'jkn']	negative
3	['otp', '*Mobile*', 'jkn', 'ga', 'masuk', 'masuk']	negative
4	['kesini', 'upgrade', 'fitur', '*Mobile*', 'jkn', 'mudah', 'membantubpjs', 'bpjskesehatan']	positive

4648	['halo', 'min', 'izin', 'nanya', 'daftar', '*Mobile*', 'jkn', 'gabisa', 'min', 'karna', 'sms', 'nya', 'ga', 'masuk', 'solusinya', 'gimana', 'ya', 'min', 'bpjs', 'ilang', 'ngurus', 'bingung', 'kemana', 'min', 'terimakasih', 'min']	negative

5. *Stemming*

Stemming merupakan proses yang bertujuan untuk merubah kata menjadi bentuk kata dasarnya, atau menghilangkan imbuhan dan juga akhiran dari kata, yang berfungsi untuk mengurangi variasi kata yang digunakan dalam analisis dan meningkatkan efisiensi dalam analisis.

Tabel 5 Hasil *Stemming*

No	Ulasan	Sentimen
0	*Mobile* jkn yu error ah rese skali	negative
1	ka tp ak ga bsa login *Mobile* jkn nya trs gmn yh	negative
2	halo gabisa daftar ya aplikasi *Mobile* jkn	negative
3	otp *Mobile* jkn ga masuk masuk	negative
4	kesini upgrade fitur *Mobile* jkn mudah membantubpjs bpjskesehatan	positive
4648	halo min izin nanya daftar *Mobile* jkn gabisa min karna sms nya ga masuk solusi gimana ya min bpjs ilang ngurus bingung mana min terimakasih min	negative

Data Mining

Pada proses data mining akan menentukan penggunaan algoritma yang akan digunakan untuk menganalisis data. Pada penelitian ini menggunakan algoritma KNN dan *Logistic Regression* , menggunakan *confusion matrix* sebagai evaluasi dan menggunakan *google colabs* untuk melakukan analisis dalam penelitian ini.

Dataset yang sudah siap digunakan selanjutnya akan dilakukan proses data mining menggunakan *google colab*. Pada proses ini memiliki bebrapa tahapan yang perlu dilakukan. Berikut tahapan-tahapan untuk melakukan klasifikasi menggunakan KNN dan *Logistic Regression* pada *google colab*.

1. *Import libraries*

Library yang diperlukan dalam melakukan klasifikasi menggunakan KNN dan *Logistic Regression* yaitu numpy, pandas, dan scikit-learn, library numpy digunakan untuk operasi matriks, pandas digunakan untuk membaca data, scikit-learn digunakan untuk KNN dan *Logistic Regression* dan untuk mengimport performa model. Berikut adalah code program untuk mengimport library yang diperlukan dapat dilihat pada tabel 6.

Tabel 6 Import Library

```
import pandas as pd
import numpy as np
from sklearn.model_Selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics import classification_report, accuracy_score, precision_score, recall_score, f1_score
from sklearn.neighbors import KNeighborsClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay
```

2. *Read* data

Dataset yang digunakan dalam penelitian ini dan sudah dilakukan *Preprocessing* disimpan dalam memory Local Disk Cyang di impor ke google colab, maka diperlukan code untuk memuat dataset dengan menggunakan method " data_labeled_roberta.csv". Pada table 7 merupakan code untuk memanggil data set.

Tabel 7 *Read* Data

```
data = pd.Read_csv("data_labeled_roberta.csv")
```

3. IndoBert

Dataset yang telah melalui *Preprocessing*, selanjutnya dilakukan proses labeling untuk memberi label pada ulasan.

Tabel 8 Indobert

```
# Load dataset
df = pd.Read_csv("data_preprocessed.csv")

# Gunakan kolom text_clean
texts = df["text_cleaned"].astype(str).tolist()

# Load tokenizer dan model RoBERTa multibahasa untuk klasifikasi sentimen
model_name = "cardiffnlp/twitter-xlm-roberta-base-sentiment"
tokenizer = AutoTokenizer.from_pretrained(model_name)
model = AutoModelForSequenceClassification.from_pretrained(model_name)

# Fungsi prediksi batched untuk efisiensi, 2 label output
def predict_batch_binary(texts, batch_size=32):
    predicted_labels = []
    for i in range(0, len(texts), batch_size):
        batch = texts[i:i + batch_size]
```

```
        inputs = tokenizer(batch, return_tensors="pt", padding=True,
    truncation=True)
        with torch.no_grad():
            outputs = model(**inputs)
            probs = torch.nn.functional.softmax(outputs.logits, dim=-1)
            for prob in probs:
                # Ambil probabilitas label positive dan negative
                positive_prob = prob[2].item()
                negative_prob = prob[0].item()
                # Pilih label berdasarkan probabilitas yang lebih tinggi
                if positive_prob >= negative_prob:
                    predicted_labels.append("positive")
                else:
                    predicted_labels.append("negative")
        return predicted_labels

    # Lakukan prediksi
    df["predicted_label_roberta"] = predict_batch_binary(texts)

    # Simpan hasilnya
    output_path = "data_labeled_roberta.csv"
    df.to_csv(output_path, index=False)

output_path
```

4. TF-IDF

   Setelah data selesai diproses, representasi data dilakukan menggunakan pembobotan *Term Frequency-Inverse Document Frequency* (TF-IDF).Dengan menggunakan *"TfidfVectorizer"* dari *"sklearn"* mengubah teks ke representasi numerik . seperti code pada tabel 9.

Tabel 9 TF-IDF

```
vectorizer = TfidfVectorizer()
X = vectorizer.fit_transform(df["text_cleaned"])
y = df["predicted_label_roberta"]
```

5. *Split* Data

   Selanjutnya, data yang telah diubah ke bentuk numerik dibagi menjadi dua bagian, yaitu data latih dan data uji. Pembagian ini dilakukan menggunakan fungsi "train_test_split" dengan proporsi 80% untuk pelatihan dan 20% untuk pengujian, dikarenkan mendapatkan akurasi yang lebih baik.

Parameter random_state=42 digunakan untuk memastikan hasil pembagian data tetap konsisten setiap kali kode dijalankan, Penggunaan parameter random_state=42 bertujuan agar proses pembagian data selalu menghasilkan hasil yang sama setiap kali kode dijalankan. Dengan demikian, eksperimen menjadi konsisten, mudah direplikasi, serta meminimalkan perbedaan hasil akibat pembagian data yang berubah-ubah. Angka 42 hanya digunakan sebagai nilai acak populer tanpa makna khusus. Seperti tabel 10.

Tabel 10 Split Data

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)
```

6. Klasifikasi algoritma KNN

*K-Nearest Neighbor*s (KNN), yaitu algoritma klasifikasi berbasis instance-based yang memprediksi kelas data baru berdasarkan kedekatan dengan data latih. Dengan n_neighbors=5, model menentukan kelas berdasarkan lima tetangga terdekat. Proses pelatihan dilakukan dengan "fit", dan prediksi terhadap data uji menggunakan "predict". Seperti tabel 11.

Tabel 11 KKN

```
# KNN
knn = KNeighborsClassifier(n_neighbors=5)
knn.fit(X_train, y_train)
y_pred_knn = knn.predict(X_test)

results['KNN'] = {
 'accuracy': accuracy_score(y_test, y_pred_knn),
 'precision': precision_score(y_test, y_pred_knn,     average='weighted',
zero_division=0),
'recall': recall_score(y_test, y_pred_knn, average='weighted'),
'f1_score': f1_score(y_test, y_pred_knn, average='weighted')
```

Hasil dari setelah mencjalankan program diatas dapat dilihat pada gambar dibawah ini:

```
=== KNN ===
              precision    recall  f1-score   support

    negative       0.82      0.89      0.85       504
    positive       0.86      0.76      0.81       426

    accuracy                           0.83       930
   macro avg       0.84      0.83      0.83       930
weighted avg       0.84      0.83      0.83       930
```

Gambar 3. Hasil dari KNN

7. *Logistic Regression*

Logistic Regression yang digunakan untuk klasifikasi multi-kelas dengan parameter multi_class='multinomial'. Model ini menggunakan solver “lbfgs” untuk optimasi dan max_iter=1000 agar pelatihan mencapai konvergensi.seperti tabel 12

Tabel 12 *Logistic Regression*

```
# Logistic Regression
logreg = LogisticRegression(multi_class='multinomial', solver='lbfgs',
max_iter=1000)
logreg.fit(X_train, y_train)
y_pred_log = logreg.predict(X_test)

results['Logistic Regression'] = {
'accuracy': accuracy_score(y_test, y_pred_log), 'precision':
precision_score(y_test, y_pred_log, average='weighted', zero_division=0),
 'recall': recall_score(y_test, y_pred_log, average='weighted'),
'f1_score': f1_score(y_test, y_pred_log, average='weighted')
```

Hasil dari setelah mencjalankan program diatas dapat dilihat pada gambar dibawah ini:

```
=== Logistic Regression ===
              precision    recall  f1-score   support

    negative       0.83      0.93      0.88       504
    positive       0.90      0.78      0.84       426

    accuracy                           0.86       930
   macro avg       0.87      0.85      0.86       930
weighted avg       0.87      0.86      0.86       930
```

Gambar 4 Hasil dari Logistik *Regression*

Confusion matrix

Untuk analisis lebih dalam terhadap hasil klasifikasi, ditampilkan *confusion matrix* dari masing-masing model. *Confusion matrix* menunjukkan jumlah prediksi benar dan salah untuk setiap kelas, sehingga kita dapat melihat pola kesalahan model. seperti tabel dibawah

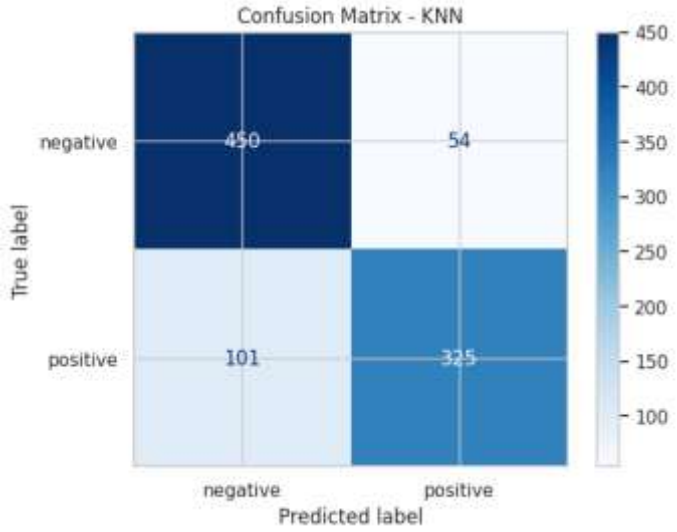
Tabel 12 Hasil *Confusion matrix*

```
# KNN
cm_knn = confusion_matrix(y_test, y_pred_knn, labels=logreg.classes_)
```

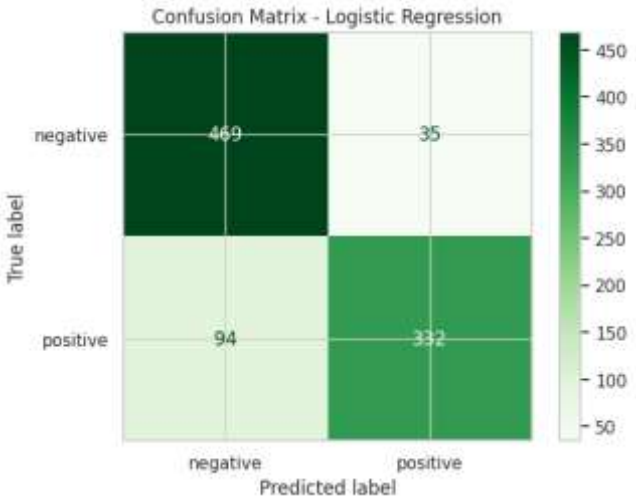
```
disp_knn = ConfusionMatrixDisplay(confusion_matrix=cm_knn, display_labels=logreg.classes_)
disp_knn.plot(cmap='Blues')
plt.title("Confusion matrix - KNN")
plt.show()

# Logistic Regression
cm_log = confusion_matrix(y_test, y_pred_log, labels=logreg.classes_)
disp_log = ConfusionMatrixDisplay(confusion_matrix=cm_log, display_labels=logreg.classes_)
disp_log.plot(cmap='Greens')
plt.title("Confusion matrix - Logistic Regression")
plt.show()
```

Hasil dari setelah menjalankan program diatas dibuat menggunakan "ConfusionMatrixDisplay" dengan pewarnaan berbeda untuk setiap model (biru untuk KNN dan hijau untuk *Logistic Regression*) dapat dilihat pada gambar dibawah ini:



Gambar 5 Hasil *Confusion matrix* KNN



Gambar 6 Hasil *Confusion matrix* Logistik *Regression*

Visualisasi

Visualisasi berfungsi untuk membuat informasi berupa topik yang paling sering muncul yang dibuat oleh pengguna aplikasi JKN *Mobile*, sehingga dari banyak teks ulasan yang ada dapat dibuat informasi yang dianggap penting. Dalam penelitian ini visualisasi dilakukan dengan cara membuat word cloud.

1. Sentimen Positif

Data sentimen positif merupakan hasil dari pelabelan yang di klasifikasi kedalam kelas positif. Berikut merupakan hasil visualisasi ulasan positif yang dibuat menggunakan word cloud pada Gambar 4.4.

Gambar 7 *Wordcloud* Sentimen Positif

Gambar 5 menunjukan bahwa pada kelas sentimen positif kata yang paling sering muncul yaitu seperti bpjs, menu info, terima kasih.

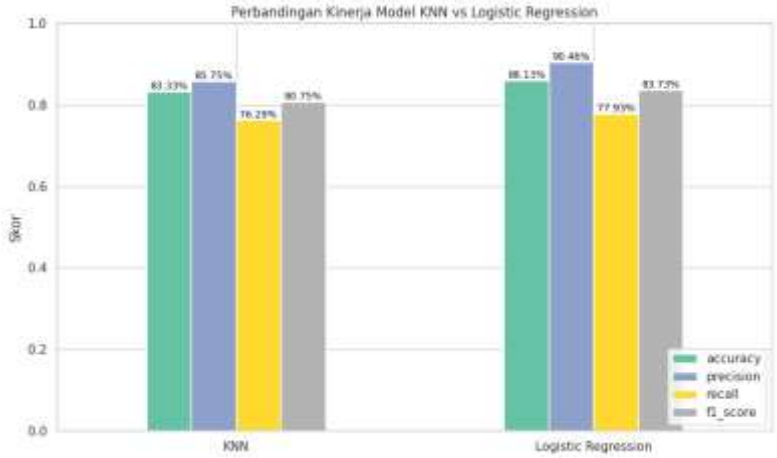
1. Sentimen Negatif

Ulasan yang memiliki kelas negatif adalah hasil pelabelan data yang termasuk dalam kelas negatif dalam analisis sentimen. Berikut adalah hasil visualisasi ulasan negatif dalam wordcloud pada Gambar 4.6.

Gambar 8 *Wordcloud* Sentimen Negatif

Gambar 6 Menunjukan bahwa pada kelas sentimen negative kata yang paing sering muncul yaitu kata-kata seperti aplikasi, daftar.

2. Perbandingan algoritma *KNN* dan *Logistik Regression*
   Evaluasi performa model *K-Nearest Neighbor* (KNN) dan Logistik *Regression* (LR) dilakukan untuk klasifikasi sentimen ulasan pengguna aplikasi JKN*Mobile*, dengan hasil metrik precision, recall, f1-score, dan accuracy.



Gambar 9 Perbandingan KNN dan Logistik *Regression*

Grafik di atas menunjukkan perbandingan kinerja dua model klasifikasi sentimen, yaitu *K-Nearest Neighbor* (KNN) dan *Logistic Regression* (LR), berdasarkan empat metrik evaluasi: accuracy, precision, recall, dan f1-score. Hasilnya, *Logistic Regression* unggul di semua metrik dengan skor accuracy 86.13%, precision 90.46%, recall 77.93%, dan f1-score 83.73%, dibandingkan KNN yang mencatat accuracy 83.33%, precision 85.75%, recall 76.29%, dan f1-score 80.75%. Perbedaan ini menunjukkan bahwa *Logistic Regression* mampu memberikan prediksi yang lebih konsisten dan akurat dalam mengklasifikasikan sentimen ulasan pengguna aplikasi JKN *Mobile*, menjadikannya pilihan model yang lebih efektif dibandingkan KNN untuk tugas ini.

Interpretasi hasil

Dataset penelitian ini berupa ulasan pengguna aplikasi JKN *Mobile* yang telah melalui *Preprocessing* dan pelabelan sentimen otomatis dengan model Roberta bahasa Indonesia. Data kemudian dikategorikan menjadi dua kelas, positif dan negatif, untuk pelatihan serta

evaluasi model klasifikasi. Dua algoritma yang digunakan adalah *K-Nearest Neighbor* (KNN) dan *Logistic Regression*. Hasil evaluasi menunjukkan *Logistic Regression* lebih unggul dengan akurasi 87,85%, sedangkan KNN memperoleh akurasi 83,87%.

KESIMPULAN

Berdasarkan hasil penelitian yang telah dilakukan maka diperoleh beberapa kesimpulan sebagai berikut:

Dari 4.777 data ulasan mengenai aplikasi *Mobile* JKN, menghasilkan 4.648 data yang bersih setelah dilakukan *Preprocessing*, selanjutnya proses pelabelan data menggunakan IndoBert. Berdasarkan pelabelan tersebut mendapatkan ulasan kelas positif sebanyak 2.066 data, dan ulasan kelas negatif sebanyak 2.581 data.

Hasil evaluasi model klasifikasi menunjukkan bahwa penerapan algoritma *K-Nearest Neighbor* (KNN) menghasilkan akurasi sebesar 83.87%, sedangkan algoritma *Logistic Regression* menghasilkan akurasi sebesar 87.85%. Dengan demikian, dapat disimpulkan bahwa *Logistic Regression* memiliki performa yang lebih baik dalam mengklasifikasikan sentimen ulasan pengguna aplikasi JKN *Mobile* dibandingkan dengan KNN, berdasarkan nilai akurasi serta metrik evaluasi lainnya seperti precision, recall, dan f1-score.

DAFTAR PUSTAKA

Adya Febriana Putri, M., Adi Sastra Wijaya, K. and Wayan Supriliyani, N. (2024) '**Efektivitas aplikasi *Mobile* Jaminan Kesehatan Nasional (JKN) dalam meningkatkan kualitas pelayanan (Studi kasus Kantor BPJS Cabang Denpasar)**', Communication and Policy Review, 1(2). Available at: https://ijespgjournal.org/index.php/shkr

Afdal, M., Rahma Elita, L. and Studi Sistem Informasi, P. (2022) '**Penerapan text mining pada aplikasi Tokopedia menggunakan algoritma *K-Nearest Neighbor***', Jurnal Ilmiah Rekayasa dan Manajemen Sistem Informasi, 8(1).

Alrajak, M.S. and Ernawati, I. (2020) '**Analisis sentimen terhadap pelayanan PT PLN di Jakarta pada Twitter dengan algoritma *K-Nearest Neighbor* (K-NN)**'.

Arifin, A.J. and Nugroho, A. (2023) '**Uji akurasi penggunaan metode KNN dalam analisis sentimen kenaikan harga BBM pada media Twitter**'.

Arifin, N., Enri, U. and Sulistiyowati, N. (2021) '**Penerapan algoritma support vector machine (SVM) dengan TF-IDF n-gram untuk text classification**'.

Auliya Agustina, D., Subanti, S. and Zukhronah, E. (2020) '**Implementasi text mining pada analisis sentimen pengguna Twitter terhadap marketplace di Indonesia menggunakan algoritma support vector machine**'.

Azis Adjie Sumanjaya, A. and Ridok, A. (2022) '**Analisis sentimen data tweets terhadap penanganan Covid-19 di Indonesia menggunakan metode Naïve Bayes dan**

pemilihan kata bersentimen menggunakan lexicon based', J-PTIIK, 6(4). Available at: http://j-ptiik.ub.ac.id

BPJS (2024) Data JKN.

Chandani, V. and Wahono, R.S. (2015) '**Komparasi algoritma klasifikasi machine learning dan feature *Selection* pada analisis sentimen review film**', Journal of Intelligent Systems, 1(1). Available at: http://journal.ilmukomputer.org

Dewi, S.P., Nurwati, N. and Rahayu, E. (2022) '**Penerapan data mining untuk prediksi penjualan produk terlaris menggunakan metode *K-Nearest Neighbor***', Building of Informatics, Technology and Science (BITS), 3(4), pp.639–648. doi:10.47065/bits.v3i4.1408

Dharmawan, L.R., Arwani, I. and Ratnawati, D.E. (2020) '**Analisis sentimen pada sosial media Twitter terhadap layanan sistem informasi akademik mahasiswa Universitas Brawijaya dengan metode *K-Nearest Neighbor***', J-PTIIK, 4(3). Available at: http://j-ptiik.ub.ac.id

Duei Putri, D., Nama, G.F. and Sulistiono, W.E. (2022) '**Analisis sentimen kinerja Dewan Perwakilan Rakyat (DPR) pada Twitter menggunakan metode Naïve Bayes Classifier'**, Jurnal Informatika dan Teknik Elektro Terapan, 10(1). doi:10.23960/jitet.v10i1.2262

Furqan, M. and Mayang Sari, S. (2022) '**Analisis sentimen menggunakan *K-Nearest Neighbor* terhadap new normal masa Covid-19 di Indonesia'**, Jurnal Teknologi Informasi, 21(1).

Hafiz, Y.A. and Sudarmilah, E. (2023) '**Implementasi web scraping pada portal berita online**'.

Harahap, E.P., Purnomo, H.D., Iriani, A., Sembiring, I. and Nurtino, T. (2024) '**Trends in sentiment of Twitter users towards Indonesian tourism: analysis with the *K-Nearest Neighbor* method'**, Computer Science and Information Technologies, 5(1), pp.19–28. doi:10.11591/csit.v5i1.pp19-28