Jurnal KomtekInfo

https://jkomtekinfo.org/ojs

2024 Vol. 11 No. 4 Hal: 371-379 e-ISSN: 2502-8758

Implementasi K-Nearest Neighbor Berbasis Particle Swarm Optimization untuk Analisis Sentimen Generasi Sandwich

Salma Nofri Yanti✉, Yuhandri, Sumijan

Fakultas Ilmu Komputer, Universitas Putra Indonesia YPTK, Padang, 25221, Indonesia

salmanofriyanti47@gmail.com

Abstract

The Sandwich Generation phenomenon refers to individuals who have to care for aging parents while raising their children, creating a double burden that has a significant impact on their social, emotional, and economic lives. In modern society, demographic changes such as increasing life expectancy and decreasing birth rates, as well as economic challenges such as increasing healthcare costs, further strengthen the relevance of this phenomenon. Therefore, understanding public perceptions of the Sandwich Generation has become increasingly important, especially through sentiment analysis on social media that reflects broader societal opinions. This study proposes a novel approach by implementing the K-Nearest Neighbor (KNN) algorithm optimized using Particle Swarm Optimization (PSO) for sentiment analysis related to the Sandwich Generation. KNN was chosen because of its ability to classify data based on the proximity between data points, while PSO was used to optimize the selection of KNN parameters to improve model accuracy. The data used in this study included 565 tweets containing the keyword "Sandwich Generation" categorized into three sentiments: 124 positive, 345 negative, and 96 neutral. The results of the study showed that in testing with 113 documents, the KNN model optimized with PSO achieved an accuracy of 79.6%, with a precision of 14.41%, a recall of 88.89%, and an F1-score of 24.81%. The implementation of PSO-based KNN has proven effective in improving the accuracy of sentiment analysis on the Sandwich Generation phenomenon, and the resulting web application has the potential to be widely used for further research, social strategy development, and better public policy advocacy.

Keywords: Sandwich Generation, Sentiment Analysis, K-Nearest Neighbor, Particle Swarm Optimization, Twitter

Abstrak

Fenomena Generasi Sandwich merujuk pada individu yang harus merawat orang tua yang menua sekaligus mengasuh anak-anak mereka, menciptakan beban ganda yang berdampak signifikan pada kehidupan sosial, emosional, dan ekonomi mereka. Dalam masyarakat modern, perubahan demografis seperti peningkatan harapan hidup dan menurunnya angka kelahiran, serta tantangan ekonomi seperti biaya perawatan kesehatan yang meningkat, semakin memperkuat relevansi fenomena ini. Oleh karena itu, memahami persepsi publik terhadap Generasi Sandwich menjadi semakin penting, terutama melalui analisis sentimen di media sosial yang mencerminkan opini masyarakat yang lebih luas. Penelitian ini mengusulkan pendekatan baru dengan mengimplementasikan algoritma K-Nearest Neighbor (KNN) yang dioptimalkan menggunakan Particle Swarm Optimization (PSO) untuk analisis sentimen terkait Generasi Sandwich. KNN dipilih karena kemampuannya dalam mengklasifikasikan data berdasarkan kedekatan antar titik data, sementara PSO digunakan untuk mengoptimalkan pemilihan parameter KNN guna meningkatkan akurasi model. Data yang digunakan dalam penelitian ini mencakup 565 tweet yang mengandung kata kunci "Generasi Sandwich" yang dikategorikan menjadi tiga sentimen: 124 positif, 345 negatif, dan 96 netral. Hasil penelitian menunjukkan bahwa pada pengujian dengan 113 dokumen, model KNN yang dioptimalkan dengan PSO mencapai akurasi sebesar 79,6%, dengan precision sebesar 14,41%, recall sebesar 88,89%, dan F1-score sebesar 24,81%. Implementasi KNN berbasis PSO terbukti efektif dalam meningkatkan akurasi analisis sentimen pada fenomena Generasi Sandwich, dan aplikasi web yang dihasilkan berpotensi digunakan secara luas untuk penelitian lanjutan, pengembangan strategi sosial, dan advokasi kebijakan publik yang lebih baik.

Kata kunci: Generasi Sandwich, Analisis Sentimen, K-Nearest Neighbor, Particle Swarm Optimization, Twitter

1. Pendahuluan

Penggunaan internet semakin meluas dengan banyaknya platform yang menyediakan berbagai layanan kepada masyarakat untuk mengakses internet[1]. Media sosial adalah salah satu perkembangan teknologi yang memberikan kemudahan bagi kita untuk untuk mencari dan menyebarkan informasi secara online tanpa batas , salah satunya menggunakan media sosial Twitter [2]. Analisis sentimen menjadi penting dalam memahami persepsi publik terhadap fenomena tertentu yang dapat mempengaruhi hasilnya. Analisis sentimen dapat membantu dalam pengambilan keputusan strategis dan komunikasi yang lebih efektif[3]. Meningkatnya penggunaan media sosial, sentimen publik terhadap fenomena "generasi sandwich" dapat dianalisis melalui data yang tersedia di platform-platform tersebut.

Diterima: 28-08-2024 | Revisi: 28-09-2024 | Diterbitkan: 31-12-2024 | doi: 10.35134/komtekinfo.v11i4.586

Analisis sentimen ini penting untuk memahami pandangan masyarakat dan dapat digunakan sebagai dasar untuk pengambilan keputusan dalam kebijakan sosial dan ekonomi [4]. K-Nearest Neighbor (KNN) adalah algoritma klasifikasi yang populer digunakan dalam analisis sentiment, sedangkan PSO adalah algoritma optimasi yang digunakan untuk menemukan solusi yang terbaik dalam berbagai aplikasi. K-Nearest Neighbor (KNN) telah digunakan secara luas dalam analisis sentimen untuk mengklasifikasikan sentimen dari teks [5]. Penerapan analisis sentimen salah satunya dilakukan oleh S. Dyah Fritama, dkk untuk mengklasifikasikan ulasan acne spot treatment menjadi Positif atau negatif. Penelitian ini digunakan metode klasifikasi K-Nearest Neighbor karena memiliki konsep sederhana yang mudah diaplikasikan dan dimengerti. Hasil sentimen didapatkan ulasan yang paling banyak mengandung sentimen positif adalah Whitelab sebanyak 1.190 ulasan dan yang paling banyak mengandung sentimen negatif adalah Skin Game sejumlah 173 ulasan. Hasil klasifikasi menggunakan KNN akurasi terbaik adalah Whitelab sebesar 97%, kemudian Skin Game memperoleh akurasi 81%, dan nilai akurasi paling rendah adalah ERHA sebesar 75% [6].

Penelitian berikutnya dilakukan oleh S. Setianingsih, dkk membahas implementasi particle swarm optimization pada algoritma K- Nearest Neighbor untuk optimasi penentuan klasifikasi penyakit Hepatitis C. Adanya implementasi tersebut diharapkan mampu meningkatkan nilai akurasi dalam klasifikasi dan mengatasi solusi untuk kelemahan pada algoritma K-Nearest Neighbor tersebut. Dari hasil pengujian K-Nearest Neighbor diperoleh nilai akurasi 97,24% pada K=5 dan K=3. Adapun untuk hasil pengujian implementasi Particle Swarm Optimization pada K-Nearest Neighbor terjadi peningkatan nilai akurasi sebesar 2,07% menjadi 99,31%. Pengujian ini menunjukkan bahwa implementasi PSO mampu mengatasi kekurangan KNN dan model ini dapat dijadikan sebagai solusi terbaik untuk menentukan klasifikasi penyakit Hepatitis C [7].

Penelitian selanjutnya oleh M. Furqan, dkk yaitu analisis sentimen menggunakan K-Nearest Neighbor terhadap New Normal masa Covid-19 di indonesia adalah untuk memprediksi komentar ataupun opini masyarakat yang kecenderungan beropini positif maupun negatif. Preprocessing data menggunakan cleaning, case folding, normalisasi, stemming, filtering, dan tokenizing. Pada normalisasi kata bertujuan memperbaiki kesalahan penulisan kata berdasarkan KBBI dan TF-IDF sebagai metode pembobotan kata. Data yang digunakan terdiri dari 1000 tweet. Metode klasifikasi opini menggunakan metode K - Nearest Neighbor dan melakukan pengujian agar mendapatkan hasil akurasi yang paling terbaik serta mengevaluasi menggunakan confusion matrix. Hasil dari pelabelan untuk sentimen positif berjumlah 811 dan 189 untuk sentimen negatif. Klasifikasi K-NN dengan nilai k = 1 menghasilkan pengujian use training set dengan akurasiy sebesar 100%, 92,60% untuk 10-fold cross-validation dan 94,50% untuk 80% percentage split [8].

Penelitian lainnya oleh A. D. Adhi Putra, dkk juga menganalisa sentimen pada ulasan pengguna aplikasi investasi online yaitu bibit dan bareksa. Jumlah ulasan yang akan digunakan pada penelitian ini sebanyak 998 yang terdiri dari 484 sentimen positif dan 514 sentimen negatif untuk aplikasi bareksa sedangkan untuk aplikasi bibit menggunakan 1063 data yang terdiri dari 541 sentimen positif dan 522 sentimen negatif. Data tersebut juga melewati tahapan preprocessing dan modelling. Pada penelitian ini menggunakan model CRISP-DM (Cross Industry Standard Process for Data Mining) dan algoritma yang digunakan pada penelitian ini adalah K-Nearest Neighbors. Berdasarkan hasil yang diperoleh dari tahapan modelling dengan menggunakan algoritma k- nearest neighbors dan perbandingan 60:40 untuk data training dan data testing, maka nilai akurasi precision dan recall yang dihasilkan dari tiap aplikasi yaitu untuk bibit 85,14% , 91,91%, dan 76,44% sedangkan untuk bareksa yaitu 81,70% , 87,15%, 75,73% [9].

Penelitian oleh R. S. Amardita, dkk membahas sistem analisis sentimen terhadap ulasan Paris Van Java Resort Lifestyle Place di Kota Bandung dengan menggunakan algoritma KNN (K-Nearest Neighbor) berhasil mengklasifikasikan ulasan pada Google Review berupa sentimen positif dan sentimen negatif. Dari hasil pengujian pertama diperoleh bahwa hasil performa terbaik pada analisis sentimen terhadap ulasan Paris Van Java Resort Lifestyle Place di Kota Bandung dengan menggunakan algoritma KNN (K-Nearest Neighbor), yaitu pada penggunaan Term Frequency-Inverse Document Frequency (TF-IDF) Unigram dengan seluruh nilai persentase rasio data yang digunakan dengan nilai perfoma terbaik sebesar 88.29%. Dapat dikatakan bahwa pengaruh N-gram sangat berpengaruh karena hasil yang diperoleh pada pengujian mempunyai perbedaan yang signifikan. Selain itu, dari hasil pengujian yang kedua, metode penghitungan jarak dengan seluruh nilai persentase rasio data yang digunakan dengan performa terbaik adalah Euclidean Distance yang dimana mencapai akurasi sebesar 88.29%. Penggunaan metode penghitungan jarak dapat mempengaruhi keakuratan algoritma KNN (K-Nearest Neighbor) dalam analisis sentimen terhadap ulasan Paris Van Java Resort Lifestyle Place di Kota Bandung [10].

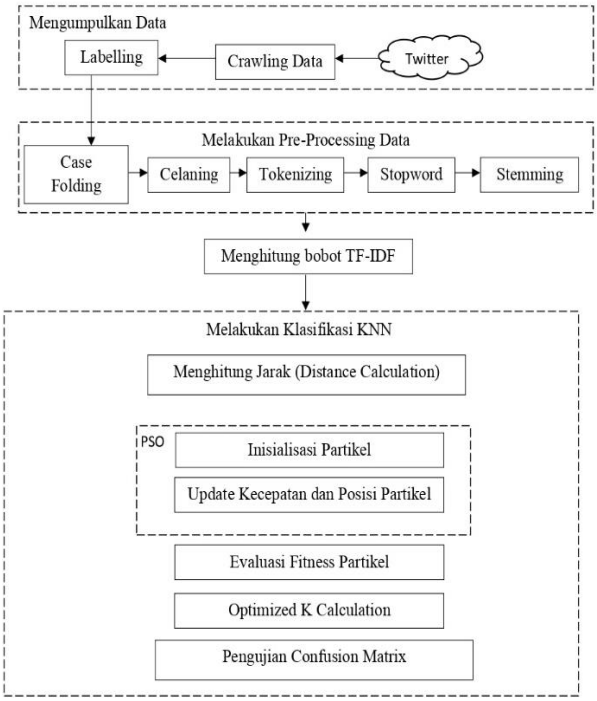
Berdasarkan kajian literatur, Penelitian ini dilakukan untuk mengatasi tantangan yang sering dihadapi dalam analisis sentimen, khususnya dalam hal akurasi dan efisiensi model. Analisis sentimen adalah teknik krusial dalam memahami opini dan perasaan pengguna terhadap produk atau layanan, yang semakin relevan dalam konteks generasi sandwich—yaitu konsumen yang dipengaruhi oleh tren dan pendapat di media sosial. K-Nearest Neighbors (KNN) merupakan metode

klasifikasi yang populer karena kemampuannya dalam membuat keputusan berbasis kedekatan data. Namun, performa KNN sangat bergantung pada pemilihan parameter yang tepat, seperti jumlah tetangga (K) dan metrik jarak. Particle Swarm Optimization (PSO) dapat memberikan solusi dengan mengoptimalkan parameter-parameter ini secara efisien, yang berpotensi meningkatkan akurasi dan efektivitas model KNN.

Tujuan utama dari penelitian ini adalah untuk mengevaluasi dampak penggunaan PSO terhadap akurasi model KNN dalam analisis sentimen. Penelitian ini bertujuan untuk menentukan apakah optimasi parameter KNN dengan PSO dapat meningkatkan akurasi dibandingkan dengan parameter yang tidak dioptimalkan. Selain itu, penelitian ini akan mengidentifikasi parameter KNN yang paling optimal setelah dioptimalkan dengan PSO, serta membandingkan efisiensi PSO dengan metode optimasi lain seperti grid search atau random search dalam hal kecepatan konvergensi dan kualitas solusi. Penelitian ini juga bertujuan untuk memahami bagaimana KNN yang dioptimalkan dengan PSO mempengaruhi hasil analisis sentimen dalam konteks generasi sandwich, serta mengidentifikasi tantangan yang mungkin timbul selama penerapan PSO, termasuk kebutuhan komputasi dan waktu pemrosesan. Dengan hasil penelitian ini, diharapkan dapat memberikan wawasan baru dalam penerapan metode optimasi untuk pembelajaran mesin dan meningkatkan efektivitas analisis sentimen, khususnya dalam memahami opini konsumen yang terpengaruh oleh media sosial dan tren saat ini.

2. Metodologi Penelitian

Metodologi Penelitian merupakan rancangan dan tahapan yang menjadi acuan dan diterapkan pada suatu penelitian yang dilakukan oleh peneliti untuk mencapai tujuan penelitian. Tujuan dari penelitian ini adalah mengimplementasi K-Nearest Neighbor berbasis Particle Swarm Optimization untuk analisis sentimen fenomena Generasi Sandwich. Metodologi penelitian ini terdiri dari beberapa tahap yang dimulai dengan pengumpulan data dari Twitter menggunakan teknik crawling untuk mendapatkan komentar yang relevan dengan fenomena Generasi Sandwich. Data yang dikumpulkan kemudian melalui proses pra-pemrosesan yang meliputi case folding, tokenizing, penghapusan stopword, dan stemming untuk membersihkan dan menyiapkan data agar siap digunakan dalam model klasifikasi. Selanjutnya, bobot kata dihitung menggunakan metode TF-IDF, dan algoritma K-Nearest Neighbor (KNN) diterapkan dengan parameter yang dioptimalkan menggunakan Particle Swarm Optimization (PSO) untuk analisis sentimen. Adapun tahapan-tahan dalam kerangka kerja penelitian pada Gambar 1



Gambar 1. Tahapan Penelitian

Gambar 1. merupakan proses klasifikasi data dari Twitter menggunakan algoritma K-Nearest Neighbors (KNN) yang dioptimalkan dengan Particle Swarm Optimization (PSO). Dimulai dari pengumpulan data melalui crawling dan pelabelan, kemudian dilakukan pra-pemrosesan data seperti pembersihan, tokenisasi, penghapusan stopword, dan stemming. Setelah itu, dilakukan perhitungan bobot TF-IDF, klasifikasi KNN dengan penghitungan jarak, serta pengoptimalan parameter K dengan PSO dan evaluasi menggunakan matriks kebingungan.

2.1 Mengumpulkan Data

Tahapan ini meurpakan tahap data dikumpulkan melalui teknik crawling, yaitu proses mengunduh data secara otomatis dari web menggunakan skrip Python. Data yang akan digunakan yaitu data *tweet* dan komentar pengguna media sosial *Twitter* dengan menggunakan kata kunci Generasi Sandwich. Data terdiri dari opini positif , netral dan opini negatif. Penulis mengkategorikan tweet dan komentar positif berdasarkan adanya kata-kata yang mengandung makna positif, seperti: bagus, baik, lanjutkan, semangat, keren, cerdas, pintar, dan sebagainya. Sedangkan opini negatif yang dikumpulkan berdasarkan adanya kata-kata yang memiliki makna tidak baik, kasar atau melecehkan, seperti: bodoh, stress, bego, kampret, bloon, lamban, tolol, goblok, cebong dan sebagainya. Setelah data terkumpul, pelabelan dilakukan secara manual oleh tim ahli yang memiliki pemahaman mendalam tentang bahasa dan konteks untuk memastikan bahwa setiap teks diberi label sentimen yang sesuai,

2.2 Melakukan Pre-Processing

Pre-processing merupakan tahapan sangat penting dalam melakukan proses klasifikasi data teks. Tujuan dilakukannya text preprocessing yaitu untuk menghilangkan noise, menyeragamkan bentuk kata dan mengurangi volume kata[11]. Tahapan-tahapan preprocessing yang dapat dilakukan dalam teks Bahasa Indonesia yaitu:

a. Case Folding dan Clealing

Tahapan case folding di mana mengubah keseluruhan bentuk huruf pada sebuah teks dokumen ke dalamhuruf kecil. Proses cleaning bertujuan untuk menghapus atau menghilangkan link nama pengguna Twitter (username) yang biasanya ditandai dengan simbol "@" yang terdapat pada dataset. Pada proses ini juga penulis masih menggunakan fasilitas yang disediakan oleh gataframework.com.

b. Stopwords Removal

Filter stopwords removal adalah proses menghilangkan kata-kata yang sering muncul namun tidak memiliki pengaruh apapun dalam ekstraksi sentimen suatu review. Kata yang termasuk seperti kata penunjuk waktu, kata tanya.

c. Tokenize

Tokenize merupakan proses untuk memisah-misahkan kata. Potongan kata tersebut disebut dengan token atau term. Proses memotong setiap kata dalam teks dan mengubah huruf dalam dokumen menjadi huruf kecil. Hanya huruf yang diterima, sedangkan karakter khusus atau tanda baca akan dihilangkan. Jadi hasil dari proses tokenize adalah kata-kata yang merupakan penyusun kalimat atau string yang dimasukan tanpa ada tanda baca.

d. Stemming

Proses Stemming digunakan untuk mengganti bentuk dari suatu kata menjadi kata dasar dari kata tersebut yang sesuai dengan struktur morfologi Bahasa Indonesia yang baik dan benar. Kata yang memiliki imbuhan awal dan akhiran seperti me, mem, meny, meng, di, per, ber, an, kan, i, nya dll., akan diubah menjadi kata dasar dengan menghilangkan kata imbuh tersebut.

2.3 Menghitung bobot TF-IDF

Pembobotan TF-IDF dilakukan Sebelum mengimplementasikan metode KNN dan setelah preprocessing data, dilakukan terlebih dahulu pembobotan kata menggunakan TF-IDF[12]. Hasil dari TF-IDF kemudian digunakan untuk pengimplemen-tasian metode KNN. TF-IDF merupakan proses pembobotan dari setiap kata (term) yang terdapat dalam dokumen. TF-IDF akan melakukan perhitungan tf, df, idf, dan TF-IDF. Menghitung TF-IDF dilakukan dengan Persamaan 1 & 2.

$$w_{dt} = TF_{dt} \times IDF_{ft} \quad (1)$$

$$idf_t = \log\left(\frac{N}{df_t}\right) \quad (2)$$

w_{dt} merupakan bobot TF-IDF atau nilai TF-IDF yang akan dicari. TF_{dt} merupakanjumlah frekuensi katayang muncul dalam sebuah dokumen. IDF_{ft} merupakan jumlah inverse frekuensi dokumen tiap kata. Dftmerupakan jumlah frekuensi dokumen tiap kata dan merupakan jumlah total dokumen.

2.4 K-Nearest Neighbor(KNN) dan PSO

Proses klasifikasi data, KNN bekerja dengan cara mengklasifikasikan titik data baru berdasarkan kedekatan atau jaraknya dengan titik data yang sudah ada dalam set pelatihan. Setiap titik data dalam set pelatihan memiliki label kelas, dan KNN mengidentifikasi K tetangga terdekat dari titik data yang diuji dengan mengukur jarak menggunakan metrik seperti Euclidean atau Manhattan. Kinerja KNN dapat dilihat sebagai berikut:

a. Menghitung Jarak

Pada tahap ini, KNN memulai proses dengan menghitung jarak antara data titik yang diuji dan data pelatihan. Jarak ini biasanya dihitung menggunakan metrik seperti Euclidean atau Manhattan, yang memungkinkan model KNN untuk menentukan tetangga terdekat dari titik data yang diuji. Metrik jarak ini menjadi fundamental dalam KNN karena kinerja model sangat bergantung pada akurasi pengukuran jarak ini.

b. Inisialisasi Partikel

Inisialisasi partikel adalah langkah awal dalam PSO, di mana partikel diatur dengan nilai parameter acak untuk memulai proses optimasi. Setiap partikel dalam PSO memiliki posisi dan kecepatan yang diatur secara acak pada awalnya. Posisi partikel ini mewakili nilai parameter KNN, seperti jumlah tetangga (K) dan metrik jarak, yang perlu dioptimalkan.

c. Update Kecepatan dan Posisi Partikel

Selama proses optimasi, PSO memperbarui kecepatan dan posisi setiap partikel berdasarkan hasil evaluasi dan informasi dari partikel terbaik

serta informasi global terbaik. Kecepatan partikel menentukan seberapa cepat perubahan parameter yang diusulkan, sementara posisi partikel mengacu pada nilai parameter KNN yang sedang dieksplorasi. Proses pembaruan ini membantu PSO untuk menjelajahi ruang parameter dan mencari konfigurasi yang optimal.

d. Evaluasi Fitnes Partikel

Setiap konfigurasi parameter yang diusulkan oleh partikel diuji dengan melatih model KNN pada data pelatihan. Kinerja model KNN kemudian dievaluasi menggunakan fungsi objektif, seperti akurasi pada set validasi. Evaluasi ini menentukan "fitness" dari setiap partikel, yaitu seberapa baik konfigurasi parameter yang diusulkan dalam meningkatkan kinerja model.

e. Optimixe K Calculation

Selama proses ini, PSO berusaha mengoptimalkan nilai K (jumlah tetangga) dan metrik jarak yang digunakan dalam model KNN. Dengan menguji berbagai konfigurasi dan mengevaluasi hasilnya, PSO membantu menemukan kombinasi parameter KNN yang memberikan hasil terbaik berdasarkan kriteria evaluasi yang ditetapkan.

2.5 Pengujian Confusion Matrix

Setelah parameter KNN dioptimalkan melalui PSO, model KNN yang terlatih menggunakan parameter terbaik diterapkan pada data pengujian. Hasil prediksi kemudian dianalisis menggunakan confusion matrix, yang memberikan gambaran tentang kinerja model dalam hal akurasi, presisi, recall, dan F1-score. Confusion matrix membantu dalam mengevaluasi bagaimana model KNN membedakan antara kelas-kelas yang berbeda dan mengidentifikasi potensi area perbaikan.Rumus dari metode evaluasi disajikan dalam Persamaan 3-6.

$$accuracy = \frac{TP+TN}{TP+FP+TN+FN} \quad (3)$$

$$precision = \frac{TP}{TP+FP} \quad (4)$$

$$recall = \frac{TP}{TP+FN} \quad (5)$$

$$f1\ score = \frac{2\times(recall\times precision)}{(recall\times precision)} \quad (6)$$

Persamaan 3 hingga 6 digunakan untuk mengevaluasi performa model klasifikasi. Accuracy (Persamaan 3) mengukur proporsi prediksi yang benar, Precision (Persamaan 4) menilai ketepatan prediksi positif, dan Recall (Persamaan 5) mengevaluasi kemampuan model menemukan semua sampel positif. F1 Score (Persamaan 6) adalah rata-rata harmonis antara precision dan recall, digunakan saat ada ketidakseimbangan kelas untuk memberikan gambaran lebih baik tentang performa model.

Confusion Matrix memiliki empat elemen yaitu True Positive (TP), False Positive(FP), True Negative (TN), dan False Negative (FN). True Positive (TP) yang menunjukkan jumlah prediksi positif yang benar, dan False Positive (FP) yang mencatat jumlah prediksi positif yang salah. True Negative (TN) menggambarkan jumlah prediksi negatif yang benar, sedangkan False Negative (FN) menunjukkan jumlah prediksi negatif yang salah. Keempat elemen ini digunakan untuk menghitung metrik evaluasi seperti accuracy, precision, recall, dan F1 score guna menilai kinerja model klasifikasi tersaji pada Tabel 1.

Tabel 1.Tabel Confusion Matrix

		Aktual	
Prediksi		TP (True Positive)	FP (False Positive)
		TN (True Negative)	FN (False Negatvie)

Tabel 1 Confusion Matrix menggambarkan hubungan antara hasil prediksi model dan kondisi aktual data. Kolom True Positive (TP) dan True Negative (TN) menunjukkan jumlah prediksi yang benar untuk kelas positif dan negatif, sementara False Positive (FP) dan False Negative (FN) mencatat jumlah kesalahan prediksi masing-masing untuk kelas positif dan negatif. Dengan memanfaatkan tabel ini, kita dapat mengevaluasi akurasi dan efektivitas model klasifikasi dalam mengidentifikasi kelas target.

3. Hasil dan Pembahasan

Rangkaian hasil penelitian berdasarkan urutan/susunan logis untuk membentuk sebuah cerita. Isinya menunjukan fakta/data dan jangan diskusikan hasilnya. Dapat menggunakan Tabel dan Angka tetapi tidak menguraikan secara berulang terhadap data yang sama dalam gambar, tabel dan teks. Untuk lebih memperjelas uraian, dapat mengunakan sub judul.

Pembahasan adalah penjelasan dasar, hubungan dan generalisasi yang ditunjukkan oleh hasil. Uraiannya menjawab pertanyaan penelitian. Jika ada hasil yang meragukan maka tampilkan secara objektif.

3.1 Preprocessing Data

Dataset yang diperoleh dari proses crawling biasanya berupa teks yang tidak terstruktur dan tidak beraturan. Agar data ini menjadi lebih bersih dan analisis sentimen menjadi lebih akurat, diperlukan proses preprocessing menggunakan Python. Tahapan preprocessing dalam penelitian ini meliputi:

a. Case Folding

Case Folding adalah tahap dalam pra-pemrosesan data teks di mana semua huruf dalam sebuah tweet diubah menjadi huruf kecil untuk memastikan konsistensi.

Proses ini membantu mengurangi variasi kata yang disebabkan oleh perbedaan penggunaan huruf besar dan kecil, seperti "Data" dan "data" yang dianggap sama. Case folding penting untuk meningkatkan akurasi analisis teks karena mengurangi kompleksitas dan jumlah fitur yang harus diproses oleh model.

Tabel 2. Tabel Hasil Case Folding

Sebelum	Sesudah
Cung yg capek jd generasi sandwich?😆🤪	cung yg capek jd generasi sandwich?😆🤪
Jangan tanya tabungan ama yang generasi sandwich, beli barang yang dia mau juga udah syukur	jangan tanya tabungan ama yang generasi sandwich, beli barang yang dia mau juga udah syukur

Tabel 2 hasil Case Folding menunjukkan perbandingan teks sebelum dan sesudah proses normalisasi huruf. Pada kolom "Sebelum", beberapa teks menggunakan huruf besar, sementara pada kolom "Sesudah", semua huruf diubah menjadi huruf kecil untuk konsistensi analisis. Proses ini membantu mengurangi perbedaan dalam data yang diakibatkan oleh variasi penggunaan huruf besar dan kecil, sehingga meningkatkan akurasi model dalam pemrosesan teks.

b. Cleaning

Celaning merupakan tahapan untuk menghapus atau menghilangkan emoji dan tanda baca seperti [!”#$%&’()*+,-./:;=>?@[] ^_`{|}~] yang tidak diperlukan dalam analisis teks. Proses ini bertujuan untuk mengurangi noise dalam data, sehingga model dapat fokus pada konten utama yang lebih relevan. Dengan membersihkan teks dari karakter-karakter yang tidak signifikan, akurasi dan efektivitas analisis sentimen dapat ditingkatkan, karena mengurangi kompleksitas data yang akan diproses pada Tabel 3.

Tabel 3. Tabel Hasil Cleaning

Sebelum	Sesudah
cung yg capek jd generasi sandwich?😆🤪	cung yg capek jd generasi sandwich
jangan tanya tabungan ama yang generasi sandwich, beli barang yang dia mau juga udah syukur	jangan tanya tabungan ama yang generasi sandwich beli barang yang dia mau juga udah syukur

Tabel 3 Hasil Cleaning menunjukkan perubahan teks sebelum dan sesudah proses pembersihan untuk menghilangkan karakter yang tidak relevan seperti emoji dan tanda baca. Pada kolom "Sebelum," teks masih mengandung emoji dan tanda tanya, sementara pada kolom "Sesudah," karakter-karakter tersebut telah dihapus untuk memastikan data lebih bersih dan siap untuk analisis lebih lanjut. Proses ini membantu meningkatkan akurasi model dengan menghilangkan elemen yang tidak memberikan kontribusi signifikan terhadap pemahaman sentimen dalam teks.

c. Tokenaizing

Tokenizing berfungsi memotong kalimat menjadi kata, karakter, simbol, atau tanda baca, sehingga setiap elemen teks dapat dianalisis secara individual. Proses ini dilakukan dengan menggunakan fungsi Regexp Tokenizer() dari library nltk untuk memisahkan kata-kata dalam teks berdasarkan pola tertentu, seperti spasi atau tanda baca. Dengan memecah teks menjadi unit-unit yang lebih kecil, tokenizing memungkinkan model untuk mengolah dan memahami setiap bagian dari teks dengan lebih efektif, meningkatkan akurasi analisis sentimen atau teks lainnya tersaji pada Tabel 4.

Tabel 4. Tabel Hasil Tokenizing

Sebelum	Sesudah
cung yg capek jd generasi sandwich	['cung', 'yg', 'capek', 'jd', 'generasi', 'sandwich']
jangan tanya tabungan ama yang Generasi Sandwich beli barang yang dia mau juga udah syukur	['jangan', 'tanya', 'tabungan', 'ama', 'yang', 'generasi', 'sandwich', 'beli', 'barang', 'yang', 'dia', 'mau', 'juga', 'udah', 'syukur']

Tabel 4 Hasil Tokenizing memperlihatkan proses pemecahan teks menjadi kata-kata individual yang siap untuk dianalisis lebih lanjut. Pada kolom "Sebelum," teks masih berbentuk kalimat utuh, sementara pada kolom "Sesudah," teks telah diubah menjadi daftar kata yang terpisah, seperti 'cung', 'yg', 'capek', dan seterusnya. Proses ini memudahkan model dalam menganalisis setiap kata secara terpisah, yang penting untuk memahami konteks dan sentimen dalam analisis teks.

d. Stopword Removal

Stopword Removal dilakukan untuk menghilangkan kata-kata yang tidak penting atau tidak memiliki arti signifikan dalam konteks analisis, seperti "dan," "yang," "atau," dan sejenisnya. Proses ini menggunakan fungsi dari library Sastrawi, yaitu StopWordRemoverFactory, untuk secara otomatis mendeteksi dan menghapus kata-kata umum yang tidak memberikan nilai informasi penting. Dengan menghapus stopwords, model dapat lebih fokus pada kata-kata yang benar-benar relevan dengan sentimen atau topik yang dianalisis, sehingga meningkatkan efisiensi dan akurasi proses analisis teks disajikan pada Tabel 5.

Tabel 5. Tabel Hasil Stopword Removal

Sebelum	Sesudah
['cung', 'yg', 'capek', 'jd', 'generasi', 'sandwich']	['cung', 'yg', 'capek', 'jd', 'generasi', 'sandwich']
['jangan', 'tanya', 'tabungan', 'ama', 'yang', 'generasi', 'sandwich', 'beli', 'barang', 'yang', 'dia', 'mau', 'juga', 'udah', 'syukur']	['tabungan', 'ama', 'generasi', 'sandwich', 'beli', 'barang', 'udah', 'syukur']

Tabel 5 Hasil Stopword Removal menunjukkan penghapusan kata-kata yang dianggap tidak penting dalam analisis teks. Pada kolom "Sebelum," daftar kata masih mengandung kata-kata umum seperti "jangan," "yang," dan "dia," yang tidak memiliki nilai informasi signifikan. Setelah proses stopword removal, kolom "Sesudah" hanya menyisakan kata-kata yang lebih relevan dengan analisis, seperti "tabungan," "generasi," dan "sandwich," sehingga meningkatkan efisiensi

model dalam memahami dan menginterpretasi konteks teks.

e. Stemming

Stemming bertujuan untuk menghilangkan kata imbuhan. Proses ini menggunakan library Sastrawi. Sastrawi digunakan untuk mengubah kata berimbuhan bahasa Indonesia menjadi bentuk dasarnya.

Tabel 6. Tabel Hasil Stemming

Sebelum	Sesudah
['cung', 'yg', 'capek', 'jd', 'generasi', 'sandwich']	['cung', 'yg', 'capek', 'jd', 'generasi', 'sandwich']
['tabungan', 'ama', 'generasi', 'sandwich', 'beli', 'barang', 'udah', 'syukur']	['tabung', 'ama', 'generasi', 'sandwich', 'beli', 'barang', 'udah', 'syukur']

Tabel 6 Hasil Stemming menunjukkan proses pengembalian kata-kata ke bentuk dasar atau akarnya. Pada kolom "Sebelum," terdapat kata-kata seperti "tabungan," yang setelah proses stemming berubah menjadi bentuk dasar "tabung" di kolom "Sesudah." Proses ini penting untuk menyederhanakan teks dan memastikan bahwa variasi morfologis dari kata yang sama dianggap identik, sehingga meningkatkan akurasi analisis teks oleh model.

3.2 Pembobotan TF-IDF

Pembobotan TF-IDF (Term Frequency-Inverse Document Frequency) adalah metode yang digunakan untuk memberikan bobot pada kata-kata dalam sebuah teks berdasarkan frekuensi kemunculannya dan seberapa penting kata tersebut dalam keseluruhan kumpulan dokumen. Term Frequency (TF) mengukur seberapa sering sebuah kata muncul dalam sebuah dokumen, sementara Inverse Document Frequency (IDF) menghitung seberapa jarang kata tersebut muncul di seluruh dokumen. Dengan menggabungkan kedua metrik ini, TF-IDF membantu model untuk mengidentifikasi kata-kata yang paling relevan dan informatif dalam analisis teks, sekaligus mengurangi pengaruh kata-kata umum yang sering muncul namun tidak memiliki nilai signifikan untuk analisis.

a. Perhitungan Nilai tf dan df

Nilai tf, didapat dengan melihat frekuensi atau banyaknya sebuah atau kata muncul di dalam sebuah dokumen. Apabila sebuah kata muncul maka dilambangkan dengan angka 1, jika kata tersebut tidak ada pada dokumen maka dilambangkan dengan angka 0. Nilai df, didapat dengan menghitung jumlah dokumen di mana termtersebut muncul dengan minimal satu dokumen pada Tabel 7.

Tabel 7. Perhitungan Nilai tf dan df pada Data Training dan Testing

Term	TF								DF
	Data Training					Data Testing			
	D1	D2	D3	D4	D5	U1	U2	U3	
mumpung	1	0	0	0	0	0	0	0	1
muda	1	0	0	0	0	0	0	0	1
badan	1	0	0	0	0	0	0	0	1
sehat	1	0	0	0	0	0	0	0	1
kuat	1	0	1	0	0	0	0	0	2

Tabel 7 menyajikan hasil perhitungan nilai Term Frequency (TF) dan Document Frequency (DF) untuk analisis teks menggunakan 8 data sampel, yang terdiri dari 5 data latih dan 3 data uji. Perhitungan dilakukan hingga term ke-6, di mana setiap term dinilai berdasarkan frekuensi kemunculannya dalam masing-masing dokumen. Nilai TF mencerminkan seberapa sering suatu kata muncul di setiap dokumen, sementara DF menunjukkan jumlah dokumen yang mengandung term tersebut, yang kemudian digunakan dalam pembobotan TF-IDF untuk mengidentifikasi kata-kata yang paling relevan dalam analisis sentimen..

b. Perhitungan TF normalisasi

Perhitungan TF normalisasi dilakukan untuk memperbaiki nilai tfdengan menghilangkan anomali yang disebabkan oleh perbedaan panjang dokumen. TF normalisasi dilakukan dengan frekuensi kemunculan termdibagi dengan panjang dokumen. Panjang dokumen, yaitu total kata yang ada pada dokumen (d). Perhitungan idf, dilakukan dengan Persamaan 2 disajikan pada Tabel 8.

Tabel 8. Nilai TF Normalisasi dan idf pada Data Training dan Testing

TF								IDF
Data Training					Data Testing			
D1	D2	D3	D4	D5	U1	U2	U3	
0,0625	0	0	0	0	0	0	0	0,90309
0,0625	0	0	0	0	0	0	0	0,90309
0,0625	0	0	0	0	0	0	0	0,90309
0,0625	0	0	0	0	0	0	0	0,90309
0,0625	0	0,0625	0	0	0	0	0	0,06020

Tabel 8 menunjukkan nilai TF Normalisasi (Term Frequency Normalized) dan IDF (Inverse Document Frequency) pada data training dan testing untuk analisis teks. Kolom "Data Training" menampilkan frekuensi term yang dinormalisasi di masing-masing dokumen pelatihan (D1 hingga D5), sementara kolom "Data Testing" menunjukkan frekuensi term di dokumen uji (U1 hingga U3). Nilai IDF mengukur seberapa jarang setiap term muncul di seluruh kumpulan dokumen, dengan nilai yang lebih tinggi menunjukkan term yang lebih spesifik dan jarang. Tabel ini membantu menentukan pentingnya setiap term dalam konteks keseluruhan data untuk meningkatkan akurasi analisis teks.

c. Perhitungan TF-IDF

Perhitungan TF-IDF dilakukan dengan mengalikan nilai TF normalisasi (Term Frequency Normalized) dengan nilai IDF (Inverse Document Frequency), seperti yang dijelaskan dalam Persamaan 1. Proses ini menghasilkan bobot yang mencerminkan pentingnya setiap term dalam konteks dokumen tertentu, relatif

terhadap seluruh kumpulan dokumen. Hasil dari perhitungan ini dapat dilihat pada Tabel 9, yang menunjukkan nilai TF-IDF untuk masing-masing term dalam data training dan testing, memberikan gambaran tentang kontribusi setiap term terhadap analisis teks yang dilakukan tersaji pada Tabel 9.

Tabel 9. Nilai TF Normalisasi dan idf pada Data Training dan Testing

TF							
Data Training					Data Testing		
D1	D2	D3	D4	D5	U1	U2	U3
0,56443	0	0	0	0	0	0	0
0,56443	0	0	0	0	0	0	0
0,56443	0	0	0	0	0	0	0
0,56443	0	0	0	0	0	0	0
0,56443	0	0,56443	0	0	0	0	0

Tabel 9 menampilkan hasil perhitungan nilai TF-IDF untuk data training dan testing setelah mengalikan nilai TF normalisasi dengan IDF. Pada kolom "Data Training," terlihat bahwa dokumen D1 memiliki nilai TF-IDF sebesar 0,56443 untuk beberapa term, sementara dokumen lainnya seperti D2, D3, D4, dan D5 sebagian besar memiliki nilai 0, kecuali untuk beberapa term di D3. Kolom "Data Testing" menunjukkan bahwa semua term di dokumen uji (U1, U2, U3) memiliki nilai 0, yang mengindikasikan term-term tersebut tidak memberikan kontribusi signifikan dalam dokumen uji ini. Hasil ini membantu dalam memahami seberapa penting setiap term dalam konteks analisis sentimen yang dilakukan.

3.3 Convusion matrix

Langkah-langkah untuk membuat Confusion Matrix yaitu mentukan prediksi untuk setiap titik data,gunakan nilai k untuk membuat prediksi untuk setiap titik data dalam dataset, selanjutnya bandingkan prediksi dengan label sebenarnya untuk setiap titik data dan buat tabel Confusion Matrix untuk menentukan jumlah True Positives (TP), True Negatives (TN), False Positives (FP), dan False Negatives (FN) dengan ketentuan sebagai berikut:

True Positives (TP): Jumlah titik data dengan label "Positif" yang diklasifikasikan dengan benar sebagai "Positif".
True Negatives (TN): Jumlah titik data dengan label "Negatif" yang diklasifikasikan dengan benar sebagai "Negatif".

False Positives (FP): Jumlah titik data dengan label "Negatif" yang salah diklasifikasikan sebagai "Positif".
False Negatives (FN): Jumlah titik data dengan label "Positif" yang salah diklasifikasikan sebagai "Negatif" tersaji pada Tabel 10.

Tabel 10. Confusion Matrix Data Testing

	Prediksi Positif	Prediksi Netral	Prediksi Negatif
Label Positif	16 (TP)	0 (FN)	0 (FN)
Label Netral	0 (FP)	12(TN)	0 (FN)
Label Negatif	0 (FP)	0 (FP)	85(TN)

Total Dokumen = 113
Akurasi:

$$Akurasi = \frac{(90)}{(113)} = 79.6\%$$

Tabel 10 menunjukkan Confusion Matrix untuk data uji, yang menggambarkan hasil klasifikasi model terhadap tiga kategori: positif, netral, dan negatif. Model berhasil mengklasifikasikan 16 dokumen dengan benar sebagai positif, 12 dokumen sebagai netral, dan 85 dokumen sebagai negatif, tanpa adanya kesalahan prediksi pada semua kategori. Dengan total 90 prediksi yang benar dari 113 dokumen uji, model mencapai akurasi sebesar 79.6%, menunjukkan kinerja yang cukup baik dalam analisis sentimen ini.

4. Kesimpulan

Berdasarkan hasil klasifikasi menggunakan algoritma K-Nearest Neighbor (KNN) model mencapai akurasi sebesar 79.6% pada dataset yang diberikan. Dari total 113 dokumen, terdapat 90 prediksi benar dan 23 prediksi salah, yang menunjukkan bahwa model KNN mampu memprediksi dengan benar sebagian besar data, namun masih ada sejumlah prediksi yang salah. Secara rinci, model menghasilkan 16 prediksi positif, 85 prediksi negatif, dan 12 prediksi netral. Nilai evaluasi model menunjukkan precision sebesar 14.41%, recall sebesar 88.89%, dan F1-score sebesar 24.81%. Meskipun model memiliki tingkat recall yang tinggi, nilai precision dan F1-score relatif rendah, menunjukkan bahwa model sering salah mengklasifikasikan data negatif sebagai positif. Optimasi parameter KNN menggunakan Particle Swarm Optimization (PSO) menunjukkan bahwa nilai k=7 memberikan akurasi terbaik sebesar 79.6%. Proses optimasi ini penting karena menunjukkan bagaimana pemilihan parameter yang tepat dapat mempengaruhi kinerja model secara signifikan. Meskipun akurasi keseluruhan belum mencapai 100%, optimasi PSO menunjukkan potensi peningkatan performa model. Namun, hasil evaluasi juga mengindikasikan adanya beberapa keterbatasan. Nilai precision yang rendah mengisyaratkan bahwa model cenderung menghasilkan banyak prediksi positif yang salah, yang mungkin disebabkan oleh ketidakseimbangan dalam distribusi data atau kekurangan dalam fitur yang digunakan untuk klasifikasi. Oleh karena itu, penelitian lebih lanjut diperlukan untuk meningkatkan model, termasuk menyeimbangkan dataset atau mengeksplorasi metode pemrosesan fitur yang lebih canggih. Secara keseluruhan, metode KNN dengan nilai k=7 cukup efektif untuk dataset ini dan mampu menangkap sebagian besar pola dalam data, tetapi perlu perbaikan lebih lanjut untuk meningkatkan akurasi dan keandalan prediksi. Aplikasi dari metode ini dalam skenario nyata dapat membantu dalam analisis sentimen publik, namun disarankan untuk menggunakan teknik tambahan atau

kombinasi model lain untuk mengatasi kekurangan yang ada. Hasil penelitian ini menyoroti pentingnya pemilihan parameter yang tepat dalam model pembelajaran mesin dan perlunya pendekatan optimasi yang adaptif seperti PSO untuk mendapatkan hasil yang optimal. Ini juga membuka peluang bagi eksplorasi metode lain yang dapat lebih efektif dalam menangani masalah klasifikasi dengan data yang lebih kompleks dan beragam.

Daftar Rujukan

[1] M. H. Al-Areef and K. Saputra S, "Analisis Sentimen Pengguna Twitter Mengenai Calon Presiden Indonesia Tahun 2024 Menggunakan Algoritma LSTM," *J. SAINTIKOM (Jurnal Sains Manaj. Inform. dan Komputer)*, vol. 22, no. 2, p. 270, 2023, doi: 10.53513/jis.v22i2.8680.

[2] C. Noventa, I. Soraya, and A. Muntazah, "Pemanfaatan Media Sosial Instagram BuddyKu Sebagai Sarana Informasi Terkini," *JKOMDIS J. Ilmu Komun. Dan Media Sos.*, vol. 3, no. 3, pp. 626–635, 2023, doi: 10.47233/jkomdis.v3i3.1124.

[3] R. I. Alhaqq, I. M. K. Putra, and Y. Ruldeviyani, "Analisis Sentimen terhadap Penggunaan Aplikasi MySAPK BKN di Google Play Store," no. January, 2023.

[4] I. Ahmad, S. Samsugi, and Y. Irawan, "Implementasi Data Mining Sebagai Pengolahan Data," *J. Teknoinfo*, vol. 16, no. 1, p. 46, 2022, [Online]. Available: http://portaldata.org/index.php/portaldata/article/view/107

[5] A. R. Isnain, J. Supriyanto, and M. P. Kharisma, "Implementation of K-Nearest Neighbor (K-NN) Algorithm For Public Sentiment Analysis of Online Learning," *IJCCS (Indonesian J. Comput. Cybern. Syst.*, vol. 15, no. 2, p. 121, 2021, doi: 10.22146/ijccs.65176.

[6] S. Dyah Fritama, Y. Raymond Ramadhan, and M. Andayani Komara, "Analisis Sentimen Review Produk Acne Spot Treatment di Female Daily Menggunakan Algoritma K-Nearest Neighbor," *Media Online)*, vol. 4, no. 1, pp. 134–143, 2023, doi: 10.30865/klik.v4i1.1070.

[7] S. Setianingsih, M. U. Chasanah, Y. I. Kurniawan, and L. Afuan, "Implementation of Particle Swarm Optimization in K-Nearest Neighbor Algorithm As Optimization Hepatitis C Classification," *J. Tek. Inform.*, vol. 4, no. 2, pp. 457–465, 2023, doi: 10.52436/1.jutif.2023.4.2.980.

[8] M. Furqan, S. Sriani, and S. M. Sari, "Analisis Sentimen Menggunakan K-Nearest Neighbor Terhadap New Normal Masa Covid-19 Di Indonesia," *Techno.Com*, vol. 21, no. 1, pp. 51–60, 2022, doi: 10.33633/tc.v21i1.5446.

[9] A. D. Adhi Putra, "Analisis Sentimen pada Ulasan pengguna Aplikasi Bibit Dan Bareksa dengan Algoritma KNN," *JATISI (Jurnal Tek. Inform. dan Sist. Informasi)*, vol. 8, no. 2, pp. 636–646, 2021, doi: 10.35957/jatisi.v8i2.962.

[10] R. S. Amardita, A. Adiwijaya, and M. D. Purbolaksono, "Analisis Sentimen terhadap Ulasan Paris Van Java Resort Lifestyle Place di Kota Bandung Menggunakan Algoritma KNN," *JURIKOM (Jurnal Ris. Komputer)*, vol. 9, no. 1, p. 62, 2022, doi: 10.30865/jurikom.v9i1.3793.

[11] D. D. E. Manurung, N. H. Matondang, and ..., "Analisis Sentimen pada Ulasan Aplikasi Jakarta Terkini (JAKI) di Google Play Store Menggunakan Metode Support Vector Machine," *... Bid. Ilmu Komput. ...*, pp. 158–167, 2022, [Online]. Available: https://conference.upnvj.ac.id/index.php/senamika/article/view/2149%0Ahttps://conference.upnvj.ac.id/index.php/senamika/article/download/2149/1649

[12] I. Verawati and B. S. Audit, "Algoritma Naïve Bayes Classifier Untuk Analisis Sentiment Pengguna Twitter Terhadap Provider By.u," *J. Media Inform. Budidarma*, vol. 6, no. 3, p. 1411, 2022, doi: 10.30865/mib.v6i3.4132.