



## *Aspect-Based Sentiment Analysis of Tourism Reviews Using Variants of the K-Nearest Neighbor Algorithm*

### **Analisis Sentimen Berbasis Aspek pada Ulasan Pariwisata Menggunakan Varian Algoritma K-Nearest Neighbor**

Anastasya Nurfitriyani Hidayat<sup>1\*</sup>, Ahmad Luky Ramdani<sup>2</sup>, Luluk Muthoharoh<sup>3</sup>

<sup>1,2,3</sup>Program Studi Sains Data, Institut Teknologi Sumatera, Indonesia

E-Mail: <sup>1</sup>anastasya.120450080@student.itera.ac.id,  
<sup>2</sup>ahmadluky@sd.itera.ac.id, <sup>3</sup>luluk.muthoharoh@sd.itera.ac.id

Received Jan 14th 2026; Revised Feb 14th 2026; Accepted Mar 12th 2026; Available Online Apr 19th 2026  
Corresponding Author: Anastasya Nurfitriyani Hidayat  
Copyright ©2026 by Authors, Published by Institut Riset dan Publikasi Indonesia (IRPI)

#### **Abstract**

*The tourism sector is a key sector in regional development and community welfare. Visitor reviews are important because they provide information that helps improve the quality of tourist attractions. This research aims to conduct aspect-based sentiment analysis on tourist attractions in Samosir Regency with a focus on three main aspects, namely Attractions, Amenity, and Accessibility, based on visitor reviews on Google Maps. Based on the results of the Attraction Aspect experiment, Modified K-Nearest Neighbor (MKNN) achieved 55.50% accuracy, 62.28% precision, 80.03% recall, and F1-Score 59.96%, while K-Nearest Neighbor (KNN) achieved 74% accuracy, 65.17% precision, 84.64% recall, and F1-Score 70.45%. For the Amenity aspect, MKNN achieved 89.50% accuracy, 87.71% precision, 85.19% recall, and an F1-Score of 86.27%, while KNN achieved 92.50% accuracy, 93.54% precision, 86.99% recall, and an F1-Score of 89.83%. In the Accessibility aspect, MKNN obtained an accuracy of 85.50% with a precision of 77.53%, recall 72.44%, F1-Score 74.71%, while KNN achieved an accuracy of 86.50%, precision 79.02%, recall 70.59%, F1-Score 74.05%, it can be seen that the KNN model still shows superior performance compared to MKNN in all three aspects analyzed.*

*Keywords: Aspect-Based Sentiment, Classification, KNN, MKNN, Reviews, Tourism Objects.*

#### **Abstrak**

Sektor pariwisata merupakan sektor kunci dalam pembangunan daerah dan kesejahteraan masyarakat. Ulasan pengunjung penting karena memberikan informasi yang membantu meningkatkan kualitas objek wisata. Penelitian ini bertujuan untuk melakukan analisis sentimen berbasis aspek pada objek wisata di Kabupaten Samosir dengan fokus pada tiga aspek utama, yaitu atraksi (*attractions*), amenitas (*amenity*), dan aksesibilitas (*accessibility*) berdasarkan ulasan pengunjung di Google Maps. Berdasarkan hasil eksperimen aspek objek wisata, Modified K-Nearest Neighbor (MKNN) mencapai akurasi 55,50%, presisi 62,28%, *recall* 80,03%, dan *F1-Score* 59,96%, sedangkan K-Nearest Neighbor (KNN) mencapai akurasi 74%, presisi 65,17%, *recall* 84,64%, dan *F1-Score* 70,45%. Untuk aspek kenyamanan, MKNN mencapai akurasi 89,50%, presisi 87,71%, *recall* 85,19%, dan *F1-Score* 86,27%, sedangkan KNN menghasilkan akurasi 92,50%, presisi 93,54%, *recall* 86,99%, dan *F1-Score* 89,83%. Pada aspek Aksesibilitas, MKNN memperoleh akurasi 85,50% dengan presisi 77,53%, *recall* 72,44%, dan *F1-Score* 74,71%, sedangkan KNN mencapai akurasi 86,50%, presisi 79,02%, *recall* 70,59%, dan *F1-Score* 74,05%. Dari sini terlihat bahwa model KNN masih menunjukkan performa yang lebih unggul dibandingkan dengan MKNN pada ketiga aspek yang dianalisis.

Kata Kunci: Klasifikasi, KNN, MKNN, Objek Wisata, Sentimen Berbasis Aspek, Ulasan.

## **1. PENDAHULUAN**

Keberadaan objek wisata merupakan salah satu elemen penting dalam sektor pariwisata yang berkontribusi signifikan terhadap pembangunan daerah, penciptaan lapangan kerja, serta promosi potensi wisata suatu wilayah [1][2]. Indonesia memiliki kekayaan alam dan budaya yang beragam, salah satunya adalah Danau Toba yang terletak di Kabupaten Samosir dan telah ditetapkan sebagai salah satu dari sepuluh destinasi pariwisata super prioritas nasional [3]. Keberagaman daya tarik wisata yang dimiliki Kabupaten



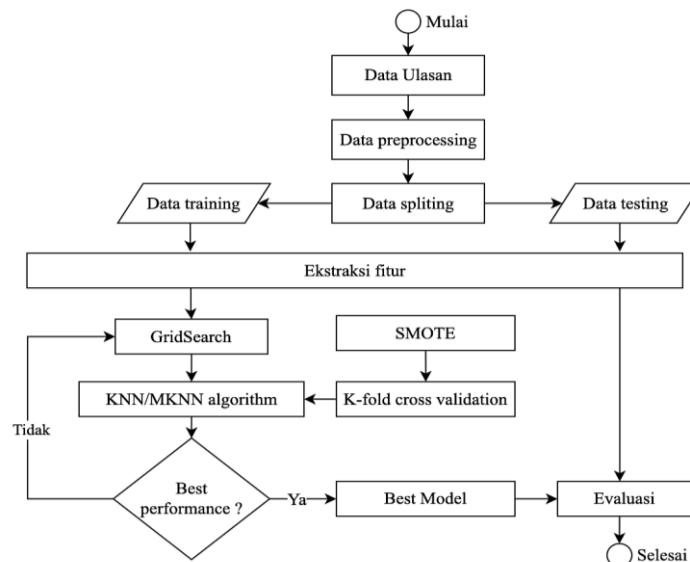
Samosir menarik kunjungan wisatawan domestik maupun mancanegara. Oleh karena itu, diperlukan upaya berkelanjutan untuk meningkatkan kualitas pengelolaan dan pengembangan objek wisata agar mampu memenuhi harapan dan kebutuhan pengunjung. Adapun salah satu cara yang dapat dilakukan adalah dengan menganalisis kepuasan wisatawan pada tempat wisata yang dikunjunginya. Metode yang berguna untuk menganalisis kepuasan wisatawan adalah menggunakan analisis sentimen.

Analisis sentimen merupakan topik dalam data sains yang mencoba untuk memahami kepuasan wisatawan dengan menganalisis ulasan pengunjung. Namun, dalam analisis sentimen perlu pemahaman terhadap aspek-aspek yang menjadi fokus dalam ulasan pengunjung. Oleh karena itu, perlu dilakukan analisis sentimen berbasis aspek (ASBA). ASBA merupakan teknik *Natural Language Processing* (NLP) untuk mengidentifikasi dan menganalisis sentimen aspek sebuah entitas. ASBA memberi pemahaman mendetail atas sentimen dengan mengelompokkan teks menjadi beberapa aspek dan menentukan sentimen dari tiap aspeknya [4]. Terdapat tiga aspek pariwisata untuk dianalisis untuk menarik calon wisatawan agar melakukan kunjungan ketiga aspek tersebut mencakup 3A yaitu atraksi, amenities dan aksesibilitas [5].

Beberapa penelitian menyatakan bahwa salah satu algoritma klasifikasi dalam metode *machine learning* populer digunakan yaitu *K-Nearest Neighbor* (KNN) karena kemudahan implementasinya [6]. Meskipun metode KNN memiliki kelebihan seperti implementasi yang mudah dan sederhana, metode KNN memiliki keterbatasan performa karena hanya mengandalkan jarak dalam proses klasifikasinya. Oleh karena itu, beberapa penelitian telah melakukan perbaikan KNN dengan mengembangkan algoritma *Modified K-Nearest Neighbor* (MKNN) yang menambah proses validitas data latih dan *weight voting* yang diharapkan mampu meningkatkan performa klasifikasi [7]. Salah satu penelitian mengklasifikasikan teks pada buku tafsir mendapatkan akurasi KNN sebesar 98,01%, MKNN sebesar 98,12% dan FKNN sebesar 88,3% dengan nilai  $K = 5$  [8]. Sementara itu, penelitian lain menunjukkan bahwa MKNN mampu meningkatkan akurasi klasifikasi dari 93,94% (KNN) menjadi 99,20% (MKNN). Beberapa penelitian menunjukkan bahwa MKNN mampu memberikan hasil klasifikasi yang lebih baik dibandingkan dengan KNN. Meskipun demikian, penerapan variasi dari algoritma KNN dalam analisis sentimen berbasis aspek masih sangat terbatas. Oleh karena itu, penelitian ini berkontribusi dengan menganalisis variasi dari KNN yaitu MKNN, dalam analisis sentimen berbasis aspek terhadap ulasan Google Maps untuk objek wisata. Selain itu, penelitian ini menganalisis *K-Fold Cross Validation* pada data tidak seimbang sebagai bagian dari proses pembelajaran *machine learning*. Hal ini dilakukan untuk meningkatkan kualitas model pada data yang tidak seimbang.

## 2. DATA DAN METODOLOGI PENELITIAN

Penelitian ini membandingkan dua metode klasifikasi, yaitu *K-Nearest Neighbor* (KNN) dan *Modified K-Nearest Neighbor* (MKNN) untuk menganalisis sentimen ulasan berbasis aspek di Kabupaten Samosir. Gambar 1 menunjukkan langkah-langkah penelitian yang dilakukan secara berurutan.



Gambar 1. Metodologi Penelitian

### 2.1. Data Ulasan

Penelitian ini menggunakan data ulasan pengunjung yang bersumber dari Google Maps<sup>1</sup> untuk objek wisata di Kabupaten Samosir. Setiap ulasan akan dilakukan proses pelabelan sentimen yang dilakukan secara

manual oleh 5 annotator menggunakan 3 kelas, yaitu positif, negatif, dan none (ulasan yang tidak mengandung aspek) untuk masing-masing aspek, yaitu atraksi (daya tarik wisata ataupun keunikan tempat wisata), amenitas (fasilitas dan pelayanan), dan aksesibilitas (akses informasi dan transportasi).

## 2.2. Preprocessing data

Tahapan ini bertujuan membersihkan dan menyiapkan data sebelum dilakukan proses analisis. Tahapan ini mencakup beberapa langkah, yaitu *case folding*, *cleaning*, normalisasi, *stemming*, *stopword removal* dan tokenisasi. *Case folding* merupakan proses mengubah semua huruf dalam teks menjadi huruf kecil untuk menyeragamkan bentuk keseluruhan teks. *Cleaning* merupakan proses penghapusan tanda baca, simbol, angka, emotikon dan spasi. Normalisasi mengubah kata-kata tidak baku menjadi kata baku dengan menggunakan kamus. *Stemming* menggunakan pustaka Sastrawi untuk mengubah kata-kata ke bentuk dasarnya dengan menghilangkan imbuhan. Tokenisasi dilakukan dengan memecah kalimat menjadi kata-kata terpisah. Tahap terakhir *stopword removal* bertujuan untuk menghapus kata-kata umum seperti “dan”, “di”, “yang”, “dengan”. Proses *stopword removal* ini menggunakan daftar *stopword* yang disediakan oleh library NLTK.

## 2.3. Pembagian data

Proses pembagian data dilakukan dengan membagi 80% data latih dan 20% data uji dengan *random\_state=42*. Tujuan untuk memastikan hasil pembagian data yang konsisten dan *stratifikasi* digunakan untuk menjaga proporsi distribusi kelas dalam data latih dan data uji tetap sama dengan distribusi dataset asli. Pembagian data tidak ada panduan yang jelas tentang rasio yang terbaik atau optimal untuk kumpulan data tertentu. Pembagian data 80%:20% didasarkan pada prinsip pareto [9]. Namun aturan praktis yang umumnya dilakukan oleh peneliti *machine learning*

## 2.4. Ekstraksi fitur

Ekstraksi fitur menggunakan *term frequency-inverse document frequency* (TF-IDF). Proses ini terdiri dari fit transform pada data latih untuk membangun *vocabulary* dan menghitung nilai IDF menggunakan Persamaan 3, sedangkan pada data uji hanya dilakukan *transform* IDF yang sama untuk mencegah kebocoran data. Penelitian ini juga menerapkan *sublinear scaling* untuk mengurangi dominasi kata-kata dengan frekuensi sangat tinggi dalam dokumen sehingga distribusi bobot antar kata menjadi lebih seimbang menggunakan Persamaan 4 [10].

$$\text{TF-IDF}_{(t,d)} = \text{tf}_{(t,d)} \times \text{IDF}_t \quad (1)$$

$$\text{tf}_{(t,d)} = \frac{f(t,d)}{N(d)} \quad (2)$$

$$\text{IDF}_t = \log\left(\frac{N}{\text{DF}_t}\right) \quad (3)$$

$$\text{TF}_{\text{scaled}}(t,d) = 1 + \log(\text{TF}(t,d)) \quad (4)$$

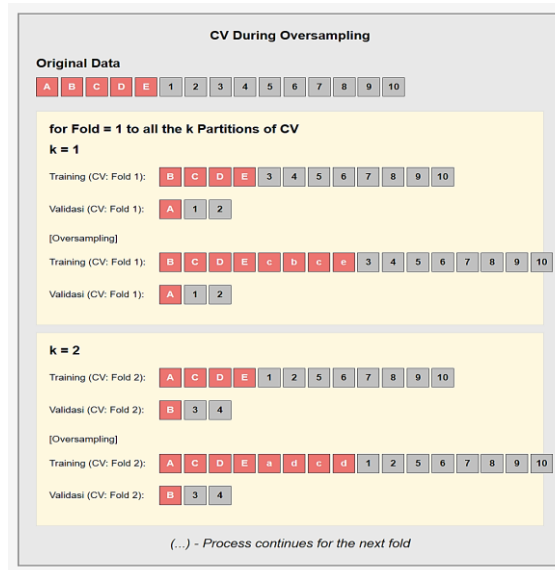
$\text{TF-IDF}_{(t,d)}$  pada Persamaan 1, merupakan nilai yang berkaitan dengan bobot kata ke- $t$  dalam dokumen ke- $d$ . Sedangkan TF (*term frequency*) untuk mengetahui seberapa banyak suatu kata yang muncul dalam suatu dokumen  $f(t,d)$  adalah frekuensi kemunculan kata  $t$  dalam dokumen.  $\text{DF}_t$  adalah jumlah dokumen yang mengandung kata  $t$ . Sedangkan  $N$  adalah jumlah total dokumen keseluruhan. *Sublinear scaling* diterapkan menggunakan Persamaan 4.

## 2.5. Synthetic Minority Over-Sampling Technique (SMOTE)

SMOTE merupakan algoritma untuk mengatasi masalah ketidakseimbangan kelas melalui proses oversampling pada kelas minoritas dengan tujuan menghasilkan dataset yang seimbang [11]. Penelitian ini menerapkan teknik SMOTE dari pustaka *imbalanced-learn*<sup>2</sup>. SMOTE diterapkan pada setiap subset data latih dalam proses *K-Fold Cross Validation* untuk menghindari data leakage. SMOTE menggunakan jarak Euclidean untuk mencari tetangga terdekat dari sampel minoritas yang dihitung. Setelah tetangga terdekat ditemukan, SMOTE menghasilkan data sintesis baru menggunakan Persamaan 5.

$$x_{\text{sintesis}} = x_i + \lambda \cdot (x_j - x_i) \quad (5)$$

dengan  $x_{\text{sintesis}}$  adalah sampel sintesis yang dihasilkan,  $x_i$  adalah sampel asli dari kelas minoritas,  $x_j$  adalah Tetangga terdekat dari  $x_i$  dalam kelas minoritas dan  $\lambda$  adalah bilangan acak antara 0 dan 1. Ilustrasi SMOTE dalam skema *k-fold cross* dapat dilihat pada Gambar 2.



Gambar 2. SMOTE pada K-Fold Cross Validation [10]

Berdasarkan ilustrasi pada Gambar 2, proses *K-Fold Cross Validation* dilakukan dengan membagi dataset menjadi *k* partisi (*fold*) yang saling lepas. Pada setiap iterasi, satu *fold* digunakan sebagai data validasi, sedangkan *K-1 fold* lainnya digunakan sebagai data pelatihan. Untuk mengatasi permasalahan ketidakseimbangan kelas, metode SMOTE diterapkan hanya pada data pelatihan di setiap *fold*, sementara data validasi dibiarkan dalam kondisi asli tanpa modifikasi. Sebagaimana ditunjukkan pada *fold* ke-1, data minoritas pada himpunan pelatihan diperbanyak melalui pembentukan sampel sintesis berdasarkan kedekatan antar tetangga (*k-nearest neighbors*). Proses ini menghasilkan tambahan data minoritas pada data pelatihan tanpa melibatkan data pada *fold* validasi. Selanjutnya, pada *fold* ke-2 dan *fold* berikutnya, mekanisme yang sama diterapkan secara independen, di mana komposisi data pelatihan dan validasi berubah sesuai dengan partisi *cross validation*, namun prinsip penerapan SMOTE tetap konsisten hanya pada data pelatihan.

Pendekatan ini bertujuan untuk mencegah terjadinya *data leakage*, yaitu kondisi di mana informasi dari data validasi secara tidak langsung digunakan dalam proses pelatihan. Dengan menjaga agar data validasi tidak terpengaruh oleh proses oversampling, evaluasi kinerja model menjadi lebih objektif dan merepresentasikan kemampuan generalisasi model pada data yang belum pernah dilihat sebelumnya.

### 2.6. K-Nearest Neighbor (KNN)

Algoritma KNN untuk melakukan klasifikasi terhadap objek (data uji) berdasarkan jumlah tetangga terdekat data pembelajaran (data latih) yang jaraknya paling dekat dengan objek tersebut [12]. Teknik pencarian tetangga terdekat dalam KNN menggunakan pengukuran jarak umum yang digunakan pada KNN yaitu jarak *euclidean* pada Persamaan 6 [13].

$$d_{(x,y)} = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \tag{6}$$

$d_{(x,y)}$  adalah jarak antara dua titik *x* dan *y*,  $x_i$  adalah fitur ke-*i* dari vektor *x*,  $y_i$  adalah Fitur ke-*i* dari vektor *y* dan *n* adalah jumlah fitur. Prediksi kelas data uji dilakukan berdasarkan *weight uniform* (bobot sama untuk semua tetangga) dan *distance* (bobot berdasarkan jarak menggunakan Persamaan 7 [14]).

$$W_i = \frac{1}{d_{(x,y)}} \tag{7}$$

$W_i$  adalah bobot dari tetangga ke-*i* dan  $d_{(x,y)}$  adalah jarak antara dua titik *x* dan *y*. Setelah bobot dihitung untuk setiap tetangga, jumlah bobot untuk setiap kelas dihitung. Kelas yang memiliki total bobot terbesar akan dipilih sebagai hasil klasifikasi pada data uji.

### 2.7. Modified K-Nearest Neighbor (MKNN)

MKNN adalah pengembangan dari algoritma KNN dengan menambahkan dua tahapan baru yaitu tahap validitas data latih serta *weight voting*.

### 2.7.1. Menghitung Nilai Validitas Data Latih

Nilai validitas dalam MKNN digunakan untuk mengukur tingkat kesamaan setiap data latih dengan tetangga-tetangga terdekatnya. Proses perhitungan nilai validitas data latih dimulai dengan mencari K tetangga terdekat untuk setiap data latih, kemudian dilakukan perbandingan antara kelas data latih tersebut dengan kelas dari masing-masing tetangganya menggunakan fungsi kesamaan S dalam Persamaan 8.

$$S(a,b) = \begin{cases} 1 & \text{jika } a=b \\ 0 & \text{jika } a \neq b \end{cases} \quad (8)$$

S adalah *similarity*/kesamaan/kemiripan  $a$  adalah kelas dari data latih  $x$  dan  $b$  adalah kelas dari tetangga ke- $i$  yang terdekat dengan  $x$ . Fungsi S memberikan nilai 1 jika label kelas  $a$  dan  $b$  sama menandakan bahwa titik data latih dan tetangga terdekatnya berada dalam kelas yang sama. Sebaliknya jika nilai 0 diberikan ketika label kelas  $a$  dan  $b$  berbeda, hal ini menunjukkan bahwa titik data latih dan tetangga terdekatnya berada dalam kelas yang berbeda. Setelah semua perbandingan selesai, nilai-nilai kesamaan ini dijumlahkan dan dibagi dengan K untuk mendapatkan nilai validitas data latih menggunakan Persamaan 9.

$$\text{Validitas}(x) = \frac{1}{K} \sum_{i=1}^K S(\text{lbl}(x), \text{lbl}(N_i(x))) \quad (9)$$

K adalah jumlah tetangga terdekat,  $\text{lbl}(x)$  adalah label kelas  $x$  dari titik data latih  $x$ ,  $\text{lbl}(N_i(x))$  adalah label kelas dari tetangga ke- $i$  yang terdekat dengan  $x$

### 2.7.2. Menghitung Weight Voting

Dalam MKNN, perhitungan *weight voting* atau pembobotan langkah untuk menentukan kelas atau hasil klasifikasi dari data uji. *Weight voting* digunakan untuk memberikan bobot pada setiap data latih berdasarkan kedekatannya dengan data uji serta validitasnya. Perhitungan bobot atau *weight voting* menggunakan persamaan 10.

$$W_{(i,j)} = \text{Validitas}_{(j)} \times \frac{1}{d_{e(i,j)} + 0.5} \quad (10)$$

$W_{(i,j)}$  adalah nilai *weight voting* untuk data uji  $i$  dan data latih  $j$ ,  $\text{Validitas}_{(j)}$  adalah nilai validitas data latih ke- $j$  dan  $d_{e(i,j)}$  adalah jarak antara data uji  $i$  dan data latih  $j$ . Setelah menghitung *weight voting* atau bobot, langkah berikutnya mengurutkan nilai *weight voting* dari yang terbesar hingga terkecil sebanyak K. Nilai *weight voting* yang telah diurutkan kemudian dijumlahkan berdasarkan kelasnya, dimana kelas dengan jumlah *weight voting* terbesar akan dipilih sebagai hasil klasifikasi untuk data uji.

## 2.8. Evaluasi Model

Pada tahap ini, evaluasi dilakukan untuk menilai seberapa baik performa model MKNN dan KNN dalam melakukan klasifikasi. Metrik evaluasi seperti *precision*, *recall*, *F1-Score* dan akurasi yang dihitung menggunakan pustaka dari *scikit-learn*<sup>3</sup> dengan menggunakan pendekatan *matrix confusion* [15]. *Matrix confusion* yang digunakan merupakan matriks yang menggunakan *multi-class* pada tiga kelas sentimen yaitu kelas sentimen positif, negatif, dan none. Tabel *confusion matrix* dapat dilihat pada Tabel 1.

**Tabel 1.** *Matrix Confusion* dengan Tiga Kelas Sentimen

Aktual	Prediksi		
	Positif	Negatif	None
Positif	<i>True Positive</i> (TP)	<i>False Negative</i> 1 (FN1)	<i>False None</i> 1 (FNo1)
Negatif	<i>False Positive</i> 1 (FP1)	<i>True Negative</i> (TN)	<i>False None</i> (FNo)
None	<i>False Positive</i> 2 (FP2)	<i>False Negative</i> 2 (FN2)	<i>True None</i> (TNo)

### 2.8.1. Precision

*Precision* ini menunjukkan dari keseluruhan ulasan yang diprediksi model untuk kelas tertentu berapa banyak yang prediksinya sesuai dengan kelas aktualnya.

$$\text{Precision Positif} = \frac{TP}{TP + FP1 + FP2} \quad (11)$$

$$\text{Precision Negatif} = \frac{TNg}{TN1 + TN + FN2} \tag{12}$$

$$\text{Precision None} = \frac{TNo}{FN01 + FN02 + TNo} \tag{13}$$

**2.8.2. Recall**

*Recall* mengukur kemampuan model dalam mengenali seluruh ulasan yang seharusnya termasuk dalam suatu kelas tertentu.

$$\text{Recall Positif} = \frac{TP}{TP + FN1 + FN01} \tag{14}$$

$$\text{Recall Negatif} = \frac{TN}{FP1 + TN + FN02} \tag{15}$$

$$\text{Recall None} = \frac{TN}{FP2 + FN2 + TNo} \tag{16}$$

**2.8.3. F1-Score**

*F1-Score* adalah metrik evaluasi yang mengukur performa setiap kelas dalam model dengan menghitung rata-rata harmonik dari presisi dan *recall*.

$$F1\text{-Positif} = 2 \times \frac{\text{Precision Positif} \times \text{Recall Positif}}{\text{Presisi Positif} + \text{Recall Positif}} \tag{17}$$

$$F1\text{-Negatif} = 2 \times \frac{\text{Precision Negatif} \times \text{Recall Negatif}}{\text{Presisi Negatif} + \text{Recall Negatif}} \tag{18}$$

$$F1\text{-None} = 2 \times \frac{\text{Precision None} \times \text{Recall None}}{\text{Presisi None} + \text{Recall None}} \tag{19}$$

**2.8.4. Akurasi**

Akurasi adalah metrik evaluasi yang digunakan untuk mengukur tingkat keberhasilan model dalam memprediksi kelas yang tepat dari keseluruhan kelas. Untuk menghitung akurasi, hal ini ditunjukkan pada Persamaan 20.

$$\text{Akurasi} = \frac{TP + TN + TNo}{TP + FN1 + \dots + FN2 + TNo} \tag{20}$$

**3. HASIL DAN PEMBAHASAN**

**3.1. Pengumpulan dan distribusi data ulasan**

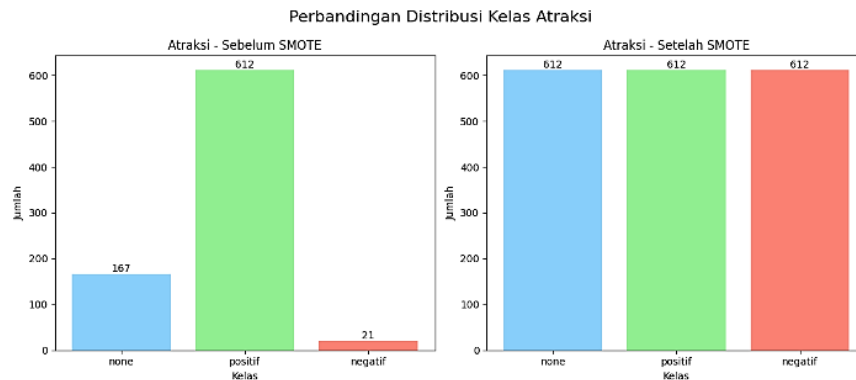
Data ulasan berasal dari aplikasi Google Maps dengan menggunakan serpapi<sup>4</sup>. Hasil proses pengumpulan data didapatkan sebanyak 791 ulasan. Selanjutnya, dilakukan proses pelabelan sentimen secara manual oleh annotator. Proses pelabelan sentimen berdasarkan aspek secara manual terhadap setiap ulasan. Pelabelan dilakukan dengan mengklasifikasikan ulasan ke dalam tiga kelas, yaitu positif, negatif, dan none, untuk masing-masing aspek yang diteliti, meliputi atraksi, amenitas, dan aksesibilitas, seperti yang ditunjukkan pada Tabel 2. Proses pelabelan dilakukan berdasarkan [16].

**Table 2.** Contoh Data Hasil Pelabelan Data

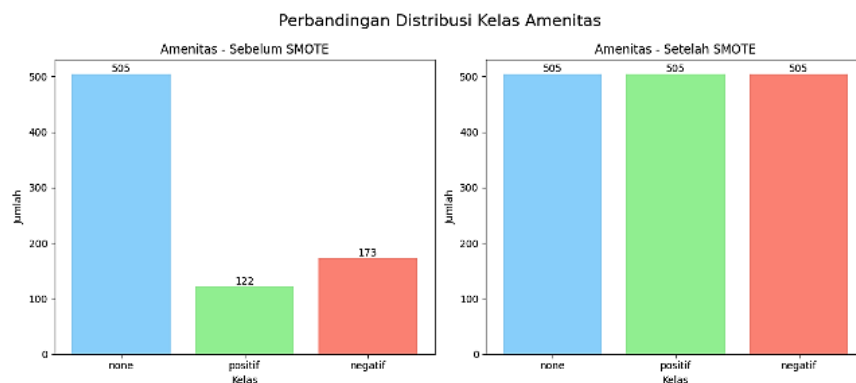
Ulasan	Atraksi	Amenitas	Aksesibilitas
Keindahan alam serta panorama bukit yang menjadi daya tarik utama di kawasan ini, ditambah dengan akses perjalanan yang mudah dan jalur yang baik menuju puncak bukit.	positif	none	positif

Pada penelitian ini ditemukan adanya ketidakseimbangan distribusi kelas yang signifikan pada ketiga aspek target yaitu atraksi, amenitas dan aksesibilitas. Berdasarkan analisis awal data, aspek atraksi menunjukkan distribusi yang sangat tidak seimbang dengan kelas positif sebanyak 612 ulasan, kelas *none* sebanyak 167 ulasan dan kelas negatif hanya 21 ulasan. Kondisi serupa juga terjadi pada aspek amenitas dengan kelas *none* mendominasi 505 ulasan, diikuti kelas negatif 173 ulasan dan kelas positif 122 ulasan. Sementara pada aspek aksesibilitas, kelas *none* memiliki 600 ulasan, kelas positif 18 ulasan dan kelas negatif 82 ulasan. Untuk mengatasi masalah ketidakseimbangan data ini, diterapkan teknik SMOTE pada data latih. SMOTE bekerja dengan menghasilkan data sintesis untuk kelas minoritas sehingga distribusi data menjadi

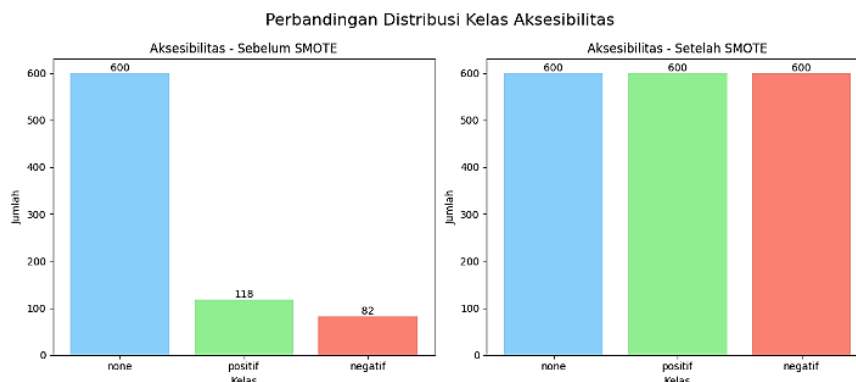
seimbang. Dalam proses *K-Fold Cross Validation*, SMOTE diterapkan untuk subset data latih tiap *fold* agar tidak terjadi kebocoran informasi ke data validasi. Hasil penerapan SMOTE pada data latih di penelitian ini dapat dilihat pada Gambar 3, 4, dan 5.



**Gambar 1.** Distribusi Data Sebelum (Kiri) dan Sesudah (Kanan) Smote pada Aspek Atraksi



**Gambar 2.** Distribusi Data Sebelum (Kiri) dan Sesudah (Kanan) Smote pada Aspek Amenitas



**Gambar 3.** Distribusi Data Sebelum (Kiri) dan Sesudah (Kanan) Smote pada Aspek Aksesibilitas

### 3.2. Pra-pemrosesan data

Pra-pemrosesan data terdiri dari beberapa tahapan yaitu case folding, pembersihan data, normalisasi, stemming, tokenisasi dan *stopword*. Case folding bertujuan mengubah seluruh komentar menjadi huruf kecil semua. Misalnya, kata “PELAYANANNYA” menjadi “pelayanannya” dan “BUKIT HOLBUNG” menjadi “bukit holbung”. Proses pembersihan data bertujuan untuk menghilangkan komponen yang tidak diperlukan dalam analisis teks, seperti tanda baca, simbol, angka dan spasi berlebih. Proses normalisasi teks bertujuan untuk mengubah kata-kata yang tidak baku termasuk istilah asing, kesalahan pengetikan, maupun singkatan menjadi bentuk standar dan formal yang sesuai dalam bahasa Indonesia. Proses *stemming* ini bertujuan untuk mengubah sebuah kata ke bentuk dasar. Proses ini menghilangkan imbuhan dari kata untuk mengurangi variasi kata yang berasal dari bentuk yang sama. Contoh proses *stemming* seperti kata “pelayanannya” menjadi kata dasarnya yaitu “layan”. Tokenisasi merupakan proses untuk memecah atau memotong teks menjadi unit-unit yang lebih kecil yang disebut dengan token. Hasil tahapan tokenisasi mengubah setiap

kalimat yang sebelumnya dalam bentuk teks utuh, dipisahkan menjadi bagian kata-kata yang terpisah sebagai unit-unit yang lebih kecil. Proses *stopword* dilakukan penghapusan terhadap kata-kata umum yang sering muncul dalam teks seperti kata penghubung. Proses ini bertujuan untuk lebih fokus pada kata-kata yang benar-benar bermakna. Tabel 3 memperlihatkan hasil seluruh tahapan pra-pemrosesan data.

### 3.2. Ekstraksi fitur

Setelah melalui tahap pra-pemrosesan data, dilakukan proses ekstraksi fitur menggunakan metode TF-IDF untuk mengubah data teks menjadi representasi numerik. Penelitian ini, digunakan *sublinear scaling* untuk mengurangi dominasi kata-kata dengan frekuensi kemunculan sangat tinggi untuk menghasilkan bobot yang lebih seimbang dengan mengurangi perbedaan yang terlalu besar antara kata yang sangat sering muncul dan kata yang jarang muncul, implementasi *sublinear scaling* ini seperti saran penelitian [17]. Pada Tabel 4 menampilkan TF-IDF data latih, dimana setiap baris (D1 hingga D800) mewakili satu dokumen atau ulasan dan setiap kolom merepresentasikan kata unik dalam dataset. Pada D6 kata “wisata” memiliki bobot 0.160899 dan kata “yesus” memiliki bobot 0.177595, sedangkan pada D799 kata “air” memiliki bobot 0.266778. Banyaknya nilai 0 dalam tabel menandakan kata tersebut tidak muncul di ulasan tersebut.

**Tabel 3.** Tahapan Pra-Pemrosesan Data

	Sebelum	Sesudah
<i>Case Folding</i>	BUKIT HOLBUNG dgn daya tarik bukit yg indah, dan PELAYANANNYA friendly	bukit holbung dgn daya tarik bukit yg indah, dan pelayanannya friendly
Pembersihan data	bukit holbung dgn daya tarik bukit yg indah, dan pelayanannya friendly	bukit holbung dgn daya tarik bukit yg indah dan pelayanannya friendly
Normalisasi	bukit holbung dgn daya tarik bukit yg indah dan pelayanannya friendly	bukit holbung dengan daya tarik bukit yang indah dan pelayanannya ramah
Steaming	bukit holbung dengan daya tarik bukit yang indah dan pelayanannya ramah	bukit holbung dengan daya tarik bukit yang indah dan layan ramah
Tokenisasi	bukit holbung dengan daya tarik bukit yang indah dan pelayanannya ramah	['bukit', 'holbung', 'dengan', 'daya', 'tarik', 'bukit', 'yang', 'indah', 'dan', 'pelayanannya', 'ramah']
Stopword	['bukit', 'holbung', 'dengan', 'daya', 'tarik', 'bukit', 'yang', 'indah', 'dan', 'pelayanannya', 'ramah']	['bukit', 'holbung', 'daya', 'tarik', 'bukit', 'indah', 'pelayanannya', 'ramah']

**Tabel 4.** Hasil TF-IDF Data Latih

D	air	...	wisata	wisatawan	yesus
1	0		0	0	0
4	0		0	0	0
5	0		0	0	0
6	0		0.160899	0	0.177595
....	....	...	...	...	...
795	0		0	0	0
799	0.266778		0	0	0
800	0		0	0	0

Tabel 5 menampilkan hasil TF-IDF data uji dalam bentuk matriks berukuran 200 x 984, dimana setiap baris mewakili satu dokumen uji dan kolomnya sama dengan *vocabulary* yang diperoleh dari data latih. Pada data uji, dapat dilihat bahwa kata “air” memiliki bobot 0.351283 di D4, kata “wisata” memiliki bobot 0.10581 pada D6 dan 0.488973 pada D795, sedangkan kata “yesus” memiliki bobot 0.323075 pada D796.

**Tabel 5.** Hasil TF-IDF Data Uji

D	air	...	wisata	wisatawan	yesus
3	0	...	0	0	0
4	0.351283	...	0	0	0
5	0	...	0	0	0
6	0	...	0.10581	0	0
....	....	...	...	...	...
195	0	...	0.488973	0	0
196	0	...	0	0	0.323075
197	0	...	0	0	0
200	0	...	0	0	0

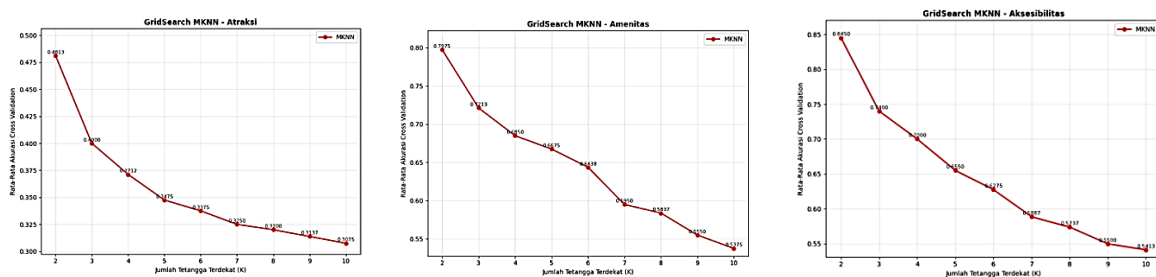
### 3.3. Grid Search dengan K-Fold Cross Validation

Penentuan nilai hyperparameter terbaik untuk model dilakukan menggunakan metode *grid search* dengan *K-Fold Cross-Validation*. *Hyperparameter* yang dioptimalkan dalam penelitian ini meliputi jumlah tetangga terdekat atau *K* yang diuji dalam rentang 2 hingga 10 mengacu pada penelitian sebelumnya yang membandingkan performa KNN dan MKNN [18]. Selain itu, untuk model KNN metode pembobotan yang diuji terdiri dari dua jenis yaitu *uniform* dimana semua tetangga memiliki bobot yang sama dan *distance* tetangga yang lebih dekat memiliki bobot lebih besar dalam pengambilan keputusan klasifikasi. Penelitian ini menggunakan *10-fold* berdasarkan rekomendasi literatur yang menyatakan bahwa  $k=10$  umum digunakan terutama untuk masalah klasifikasi karena memberikan keseimbangan yang baik antara efisiensi komputasi dan estimasi kinerja model [19]. Nilai *hyperparameter* optimal ini kemudian digunakan untuk melatih model *final* yang akan digunakan dalam tahap pengujian.

#### 3.4.1. Hasil Grid Search dengan K-Fold Cross Validation Model MKNN

Hasil *grid search* model MKNN untuk ketiga aspek Atraksi, Amenitas dan Aksesibilitas seperti yang terlihat pada Gambar 6, 7 dan 8 menunjukkan bahwa  $K=2$  konsisten memberikan performa terbaik dalam *10-fold cross-validation*. Aspek atraksi (Gambar 6 kiri) performa tinggi dicapai pada  $K=2$  dengan rata-rata akurasi 48.13%. Aspek amenitas (Gambar 6 tengah) juga menunjukkan bahwa nilai  $K=2$  memberikan rata-rata akurasi terbaik sebesar 79.75%. Demikian pula aspek Aksesibilitas (Gambar 6 tengah),  $K=2$  menghasilkan rata-rata akurasi tertinggi sebesar 84.50%.

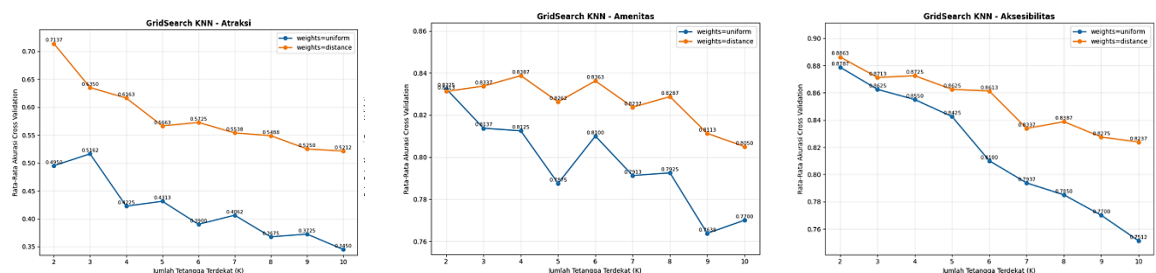
Penurunan akurasi terlihat jelas pada ketiga gambar tersebut dimana semakin besar nilai *K*, semakin rendah akurasinya. Hal ini menunjukkan bahwa penambahan jumlah tetangga berdampak buruk terhadap performa model MKNN. Hasil ini sesuai dengan penelitian sebelumnya yang menyatakan bahwa nilai *K* yang kecil lebih optimal dalam klasifikasi karena ketika *K* terlalu besar, model cenderung kehilangan kemampuan menangkap pola dan mempertimbangkan tetangga yang tidak relevan sehingga mengganggu pengambilan Keputusan [20]. Oleh karena itu,  $K = 2$  merupakan hyperparameter optimal untuk model MKNN pada semua aspek dalam dataset penelitian ini.



Gambar 4. Hasil Grid Search MKNN Aspek Atraksi (Kiri), Amenitas (Tengah) dan Aksesibilitas (Kanan)

#### 3.4.2. Hasil Grid Search dengan K-Fold Cross Validation model KNN

Hasil *grid search* model KNN untuk ketiga aspek atraksi, amenitas dan aksesibilitas seperti yang ditunjukkan Gambar 7. Pada Gambar 7 (kiri) aspek atraksi, terlihat performa tertinggi pada  $K=2$  dengan *weights distance* yang menghasilkan akurasi 71.37%. Gambar 7 aspek amenitas (tengah) menunjukkan hyperparameter optimal dikonfigurasi  $K=4$  dan *weights distance* dengan rata-rata akurasi 83.87% dan aspek aksesibilitas di Gambar 7 (kanan) diperoleh kombinasi hyperparameter terbaik di  $K = 2$  dan *weights distance* sebesar rata-rata akurasi 88.63% dalam *10-fold cross validation*.



Gambar 7. Hasil Grid Search KNN Aspek Atraksi (Kiri), Amenitas (Tengah) dan Aksesibilitas (Kanan)

### 3.5. Analisis Evaluasi Model MKNN

Evaluasi model MKNN dilakukan dengan menggunakan data uji untuk menguji kemampuan model dalam memprediksi kelas untuk setiap aspek Atraksi, Amenitas dan Aksesibilitas, setelah model dilatih

menggunakan data latih yang telah dioptimalkan dengan SMOTE dan *hyperparameter* terbaik, kemudian model digunakan untuk melakukan prediksi pada data uji untuk setiap aspek. Dapat dilihat Tabel 6 hasil evaluasi peraspek model MKNN menggunakan metrik presisi, *recall*, *F1-Score* dan akurasi.

**3.6. Analisis Evaluasi Model KNN**

Evaluasi model KNN dilakukan untuk menilai seberapa baik model dapat mengklasifikasikan data uji setelah dilakukan proses pengoptimalan. Evaluasi ini menggunakan metrik seperti presisi, *recall*, *F1-Score* dan akurasi yang dihasilkan di Tabel 7.

**3.7. Analisis Perbandingan Model MKNN dan KNN**

Setelah mengevaluasi hasil presisi, *recall*, *F1-Score* dan akurasi dari model MKNN dan KNN untuk ketiga aspek, langkah selanjutnya adalah melakukan perbandingan terhadap kedua model tersebut. Berdasarkan Tabel X dan Tabel XI terlihat perbedaan performa antara model MKNN dan KNN. Hasil ini sejalan dengan penelitian Karo et.al yang membandingkan kedua algoritma tersebut dalam penelitiannya, KNN menunjukkan performa yang lebih baik daripada MKNN dengan selisih rata-rata 25% didata pengujian, selain itu waktu eksekusi algoritma KNN juga lebih cepat dibanding dengan MKNN [20]. Pada aspek Atraksi, menunjukkan akurasi yang cukup rendah setelah diterapkan SMOTE, hal ini terjadi karena jumlah data kelas minoritas sangat kecil sehingga SMOTE menghasilkan terlalu banyak data sintesis yang kurang representatif [21].

Berdasarkan analisis perbandingan performa model MKNN dan KNN yang telah diterapkan pada dataset yang sudah diseimbangkan dengan metode SMOTE, dapat dilihat perbedaan antara kedua model tersebut. Aspek Atraksi, model KNN mencapai akurasi sebesar 74.00% jauh lebih tinggi dibandingkan dengan model MKNN yang hanya mencapai 55.50% dengan selisih yang cukup besar yaitu 18.50%, hal ini menunjukkan bahwa KNN memiliki kemampuan lebih baik dalam mengklasifikasikan sentimen pada aspek Atraksi. Untuk aspek Amenitas, model KNN juga menunjukkan performa yang lebih baik dengan akurasi 92.50% sedangkan MKNN mencapai 89.50% dengan selisih lebih kecil yaitu 3%. Meskipun selisihnya tidak sebesar pada aspek Atraksi, model KNN tetap konsisten menunjukkan performa yang lebih baik. Pada aspek Aksesibilitas, performa KNN mencapai akurasi 86.50% sedangkan MKNN 85.50% dengan selisih 1%. Secara keseluruhan, meskipun telah dilakukan penyeimbangan data menggunakan metode SMOTE, model KNN tetap menunjukkan performa lebih unggul dibandingkan MKNN pada ketiga aspek yang dianalisis. Hasil ini menarik karena bertolak belakang dengan asumsi bahwa MKNN sebagai pengembangan dari KNN seharusnya memberikan performa yang lebih baik. Namun seperti yang dijelaskan dalam penelitian [20], MKNN tidak selalu unggul terutama pada dataset dengan karakteristik tertentu. Secara keseluruhan, meskipun MKNN telah dimodifikasi untuk meningkatkan performa KNN, hasil penelitian ini menunjukkan bahwa pada kasus tertentu, algoritma KNN masih dapat memberikan hasil yang lebih baik dengan waktu komputasi yang lebih efisien [20].

**Tabel 6.** Hasil Evaluasi Peraspek Model MKNN

Aspek	Kelas sentimen	Presisi	Recall	F1-Score	Akurasi
Atraksi	positif	0.9848	0.4248	0.5936	0.5550
	negatif	0.5556	1.0000	0.7143	
	none	0.3280	0.9762	0.4910	
	Macro average	0.6228	0.8003	0.5996	
Amenitas	positif	0.7500	0.7742	0.7619	0.8950
	negatif	0.9730	0.8372	0.9000	
	none	0.9084	0.9444	0.9261	
	Macro average	0.8771	0.8519	0.8627	
Aksesibilitas	positif	0.7500	0.6207	0.6792	0.8550
	negatif	0.6842	0.6190	0.6500	
	none	0.8917	0.9333	0.9121	
	Macro average	0.7753	0.7244	0.7471	

**Tabel 7.** Hasil Evaluasi Peraspek Model KNN

Aspek		Presisi	Recall	F1-Score	Akurasi
Atraksi	positif	0.9391	0.7059	0.8060	0.7400
	negatif	0.5556	0.1000	0.7143	
	none	0.4605	0.8333	0.5932	
	Macro average	0.6517	0.8464	0.7045	
Amenitas	positif	0.9200	0.7419	0.8214	0.9250
	negatif	0.9744	0.8837	0.9268	
	none	0.9118	0.9841	0.9466	
	Macro average	0.9354	0.8699	0.8983	

Aspek		Presisi	Recall	F1-Score	Akurasi
Aksesibilitas	positif	0.8095	0.5862	0.6800	0.8650
	negatif	0.6667	0.5714	0.6154	
	none	0.8944	0.9600	0.9260	
	Macro average	0.7902	0.7059	0.7405	

#### 4. KESIMPULAN

Hasil penerapan metode MKNN dalam mengklasifikasikan sentimen berdasarkan aspek pada ulasan objek wisata di Kabupaten Samosir diperoleh melalui beberapa tahap pemodelan. Dimulai dengan pra-pemrosesan data yang meliputi case folding, pembersihan data, normalisasi teks, stemming, tokenisasi dan stopword removal. Selanjutnya, data dibagi menjadi data latih dan uji dengan proporsi 80:20. Pada data latih dilakukan ekstraksi fitur menggunakan TF-IDF, menghasilkan daftar kata unik dan nilai IDF dari data latih digunakan untuk mengubah data uji menjadi representasi vektor TF-IDF. Untuk mengatasi ketidakseimbangan data, diterapkan metode SMOTE pada data latih dan untuk mendapatkan model optimal, dilakukan pencarian hyperparameter menggunakan Grid Search dengan 10-fold cross-validation. Model kemudian dilatih kembali menggunakan hyperparameter terbaik dan dievaluasi menggunakan data uji untuk masing-masing aspek.

Penelitian ini mengevaluasi perbandingan metode MKNN dan KNN dalam skema pembelajaran yang melibatkan data yang tidak seimbang. Hasil penelitian menunjukkan bahwa KNN memberikan performa yang lebih baik secara keseluruhan dengan memperoleh akurasi mencapai 74%, presisi 65.17%, recall 84.64% dan F1-Score 70.45% pada aspek atraksi. Untuk aspek amenities, KNN menghasilkan akurasi 92.50%, presisi 93.54%, recall 86.99%, F1-Score 89.83%. Pada aspek aksesibilitas, KNN mencapai akurasi 86.50%, presisi 79.02%, recall 70.59%, F1-Score 74.05%.

#### REFERENCES

- [1] F. J. Muhammad, I. Irawati, dan M. Halimah, "Implementasi Kebijakan Program Pengembangan Pariwisata Berkelanjutan di Kota Manado," *JIIIP J. Ilm. Ilmu Pemerintah.*, vol. 7, no. 1, hlm. 63–83, 2022.
- [2] M. Prisma, N. Wahyu, dan U. N. Surabaya, "JURNAL GEOGRAFI Pengembangan Wilayah Kabupaten Lamongan di bidang Pariwisata berbasis Community Based Tourism," *Geogr. Dan Pengajarannya*, vol. xx, hlm. 0–8, 2020.
- [3] N. Nusantoro, I. M. C. Mandira, H. S. Ghaour, dan A. J. S. Runturambi, "Peran Tentara Nasional Indonesia (Kodam I Bukit Barisan) Dalam Mendukung Pengembangan Pariwisata Berkelanjutan di Danau Toba," *Altasia J. Pariwisata Indones.*, vol. 5, no. 2, hlm. 1–15, 2023.
- [4] "Penerapan Aspect-Based Sentiment Analysis untuk Identifikasi Masalah Kualitas Layanan Aplikasi Mobile iPusnas berdasarkan Persepsi Pengguna," *J. Pengemb. Teknol. Inf. Dan Ilmu Komput.*, vol. 9, no. 7, 24 2025, Diakses: 2 Januari 2026. [Daring]. Tersedia pada: <https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/15039>
- [5] P. Wirawan Eka dan I. M. Semara Trisna, *Pengantar Pariwisata*. Unit Penerbit dan Publikasi Institut Pariwisata dan Bisnis Internasional, 2021.
- [6] S. Zhang, "Challenges in KNN Classification," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 10, hlm. 4663–4675, 2022.
- [7] G. A. Rosso, "Milton," *William Blake Context*, no. September, hlm. 184–191, 2019.
- [8] A. Nur, Mustakim, S. Syarifandi, dan S. Amin, "Implementation of text mining classification as a model in the conclusion of Tafsir Bil Ma'tsur and Bil Ra'yi contents," *Int. J. Eng. Adv. Technol.*, vol. 9, no. 1, hlm. 2789–2795, 2019.
- [9] V. R. Joseph, "Optimal ratio for data splitting," *Stat. Anal. Data Min. ASA Data Sci. J.*, vol. 15, no. 4, hlm. 531–538, 2022, doi: <https://doi.org/10.1002/sam.11583>.
- [10] Z. Hadi, M. Zulpahmi, A. Asrory, dan R. Forest, "Detecting Fake Reviews Using BERT and Sublinear \_ TF Methods on Hotel Reviews in the Lombok Tourism Area," vol. 8, no. 2, hlm. 550–556, 2024.
- [11] D. Dablain, B. Krawczyk, dan N. V Chawla, "DeepSMOTE : Fusing Deep Learning and SMOTE for Imbalanced Data," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 9, hlm. 6390–6404, 2023.
- [12] T. Waryono, *Fundamental of Phyton for machine learning*. Yogyakarta: Gava Media, 2018.
- [13] A. E. Pramudit dan M. B. Akbar, "Absensi Dengan Pengenalan Wajah Menggunakan Convolutional Neural Network (CNN) Dan Euclidean Distance Attendance with face recognition using Convolutional Neural Network (CNN) and Euclidean Distance," *J. Info Digit*, vol. 2, no. 2, hlm. 616–631, 2024.
- [14] K. U. Syaliman, D. Gunawan, T. Informatika, dan P. C. Riau, "Jurnal Testing dan Implementasi Sistem Informasi Implementation Of Local Mean Distance Weighting K- Nearest Neighbor In

- Determining Vocational High School Majors In Pekanbaru Penerapan Pembobotan Local Mean Distance K-Nearest,” Vol. 2, No. 1, Hlm. 59–68, 2024.
- [15] A. E. S. Saputro, K. A. Notodiputro, dan I. A., “Study of Sentiment of Governor’s Election Opinion in 2018,” *Int. J. Sci. Res. Sci. Eng. Technol.*, no. December, hlm. 231–238, 2018.
- [16] D. I. Af’idah, P. D. Anggraeni, M. Rizki, A. B. Setiawan, dan S. F. Handayani, “Aspect-Based Sentiment Analysis for Indonesian Tourist Attraction Reviews Using Bidirectional Long Short-Term Memory,” *JUITA J. Inform.*, vol. 11, no. 1, hlm. 27, 2023.
- [17] R. Yi, L. Guo, S. Wei, A. Zhou, S. Wang, dan S. Member, “EdgeMoE : Empowering Sparse Large Language Models on Mobile Devices,” *IEEE Trans. Mob. Comput.*, hlm. 1–16, 2025.
- [18] I. Gazalba, N. Gayatri, dan I. Reza, “Comparative Analysis of *K-Nearest Neighbor* and *Modified K-Nearest Neighbor* Algorithm for Data Classification,” hlm. 294–298, 2017.
- [19] L. A. Yates, Z. Aandahl, S. A. Richards, dan B. W. Brook, “Cross validation for model selection: A review with examples from ecology,” *Ecol. Monogr.*, vol. 93, no. 1, hlm. 1–36, 2023.
- [20] N. Zhang, W. Karimoune, L. Thompson, dan H. Dang, “A between-class overlapping coherence-based algorithm in KNN classification,” *2017 IEEE Int. Conf. Syst. Man Cybern. SMC 2017*, vol. 2017-Janua, hlm. 572–577, 2017.