Jurnal Ilmu Komputer dan Informatika (JIKI)
Vol. 4, No. 1, Juni 2024, Hal. 31-42
https://jiki.jurnal-id.com
DOI: https://doi.org/10.54082/jiki.132
P-ISSN: 2807-6664
E-ISSN: 2807-6591

Efektifitas SMOTE dalam Mengatasi Imbalanced Class Algoritma K-Nearest Neighbors pada Analisis Sentimen terhadap Starlink

Candra[*1], Kelvin William Chandra[2], Hafiz Irsyad[3]

[1,2,3]Informatika, Fakultas Ilmu Komputer dan Rekayasa, Universitas Multi Data Palembang, Indonesia
Email: [1]candra_2007@mhs.mdp.ac.id, [2]kelvin.william@mhs.mdp.ac.id, [3]hafizirsyad@mdp.ac.id

Abstrak

Imbalanced class menjadi salah satu tantangan utama dalam melakukan analisis sentimen, khususnya antara ulasan positif dan negatif yang dapat menyebabkan bias prediksi. Penelitian ini bertujuan untuk mengetahui lebih lanjut apakah permasalahan *imbalanced class* pada algoritma *K-Nearest Neighbors* (KNN) dapat diatasi dengan menggunakan SMOTE (*Synthetic Minority Over-sampling Technique*) pada analisis sentimen terhadap layanan *Starlink*. SMOTE akan digunakan untuk menciptakan sampel sintetis dari kelas minoritas untuk mengatasi ketidakseimbangan data. Hasil penelitian menunjukkan bahwa sebelum menggunakan SMOTE performa model KNN menghasilkan 69,97% (*Accuracy*), 68,47% (*Precission*), 64,99% (*Recall*), 65,41% (*F1-Score*). Namun, setelah menggunakan SMOTE performa model KNN mengalami peningkatan menjadi 76,74% (*Accuracy*), 79,15% (*Precission*), 75,91% (*Recall*), 75,82% (*F1-Score*). Dengan peningkatan ini, dapat disimpulkan bahwa kombinasi antara algoritma KNN dan SMOTE berhasil meningkatkan akurasi dan keandalan model KNN dalam mengatasi *imbalanced class* pada analisis sentimen terhadap layanan *Starlink.*

Kata kunci: *Analisa Sentimen, Imbalanced Class, K-Nearest Neighbors, Synthetic Minority Oversampling Technique, Starlink*

Abstract

Imbalanced class is one of the main challenges in conducting sentiment analysis, especially between positive and negative reviews which can cause prediction bias. This research aims to find out more whether the imbalanced class problem in the K-Nearest Neighbors (KNN) algorithm can be overcome by using SMOTE (Synthetic Minority Over-sampling Technique) in sentiment analysis of Starlink services. SMOTE will be used to create synthetic samples from minority classes to address data imbalance. The research results show that before using SMOTE the performance of the KNN model was 69.97% (Accuracy), 68.47% (Precision), 64.99% (Recall), 65.41% (F1-Score). However, after using SMOTE the performance of the KNN model increased to 76.74% (Accuracy), 79.15% (Precission), 75.91% (Recall), 75.82% (F1-Score). With this improvement, it can be concluded that the combination of the KNN and SMOTE algorithms has succeeded in increasing the accuracy and reliability of the KNN model in overcoming the imbalanced class in sentiment analysis of Starlink services..

Keywords: *Imbalanced Class, K-Nearest Neighbors, Synthetic Minority Oversampling Technique, Sentiment Analysis, Starlink*

1. PENDAHULUAN

Dalam era digital saat ini, berbagai informasi dapat menyebar dengan sangat cepat salah satunya melalui *platform* media sosial seperti *Instagram*, *Youtube*, *Facebook* dan *Twitter*. Media sosial telah menjadi alat yang sangat kuat dalam membentuk opini publik dan mempengaruhi persepsi masyarakat terhadap berbagai isu. Setiap hari, jutaan pengguna dari berbagai negara membagikan pendapat, pengalaman, dan perasaan mereka di platform-platform ini, sehingga menciptakan lautan data dan informasi yang berharga. Namun, untuk memahami informasi yang disampaikan tersebut bersifat positif atau negatif perlu dilakukannya analisis sentimen.

Analisis sentimen merupakan suatu teknik pengekstrak data teks yang digunakan untuk memperoleh informasi mengenai sentimen positif maupun negatif dari pengguna internet [1]. Dalam melakukan analisis sentimen, terdapat berbagai macam algoritma yang sering diterapkan diantaranya

menggunakan algoritma *Support Vector Machine (SVM), Naive Bayes*, dan *K-Nearest Neighbor (KNN)* [2]. Dari penelitian sebelumnya oleh Pertiwi dalam Analisis Sentimen Opini Publik mengenai Sarana dan Transportasi Mudik Tahun 2019 pada *Twitter* menggunakan algoritma *Naive Bayes, Neural Network* dan KNN dan SVM, didapatkan bahwa algoritma KNN memiliki akurasi yang lebih tinggi yaitu 90,76%, dibandingkan algoritma SVM (89,03%), *Naive Bayes* (78,16%) dan *Neural Network* (52,73%) [3]. Namun, hal ini berbanding terbalik pada penelitian yang dilakukan oleh Nurmalasari *et al*. pada analisis sentimen aplikasi *kredivo*, algoritma KNN mendapatkan akurasi sebesar 84%, dimana nilai akurasi ini lebih rendah dibandingkan dengan SVM (88%) dan *Naive Bayes* (89%) [4].

Hal tersebut kemungkinan besar disebabkan oleh salah satu tantangan utama dalam analisis sentimen yaitu ketidakseimbangan kelas (*imbalanced class*), dimana jumlah antara ulasan positif dan negatif tidak seimbang [5]. Ketidakseimbangan ini dapat mengakibatkan bias dalam model prediksi, sehingga model lebih cenderung memprediksi kelas mayoritas.

Salah satu metode yang sering digunakan untuk menangani ketidakseimbangan data dan dapat meningkatkan performa algoritma klasifikasi adalah metode *Synthetic Minority Over-sampling Technique* (SMOTE) [6]. SMOTE merupakan teknik *oversampling* yang bertujuan untuk mengatasi ketidakseimbangan kelas dengan cara menciptakan sampel-sampel sintetis dari kelas minoritas melalui interpolasi [7]. Pada penelitian sebelumnya oleh Kurnia *et al*., penggunaan SMOTE dalam analisis sentimen dengan metode *Naïve Bayes*, SMOTE dan *Adaboost* pada *Twitter Bank* BTN dapat meningkatkan akurasi klasifikasi menjadi 87,05% dari awalnya dibawah 78% [8]. Selain itu, pada penelitian Implementasi *Random Forest* Menggunakan SMOTE untuk Analisis Sentimen Ulasan Aplikasi *Sister for Students* UNEJ yang dilakukan oleh Anjani *et al*., juga menyimpulkan bahwa SMOTE yang digunakan pada model dapat meningkatkan performa *accuracy* klasifikasi mencapai 98,9% (*testing*) dan mengurangi kesalahan prediksi [9].

Pada penelitian ini, penulis akan menggunakan algoritma *K-Nearest Neighbor* (KNN) yang dikombinasikan dengan *Synthetic Minority Over-sampling Technique* (SMOTE) untuk meneliti lebih lanjut apakah penerapan SMOTE dapat meningkatkan performa algoritma KNN dalam analisis sentimen publik terhadap *Starlink* dari salah satu *channel Youtube* yaitu *GadgetIn*. Secara sederhana, *Starlink* merupakan layanan internet satelit yang diperkenalkan oleh *SpaceX* dan telah mendapatkan banyak perhatian publik dan berbagai ulasan di media sosial. Tujuan dan manfaat dari penelitian ini adalah untuk mengetahui sentimen dari publik terhadap sebuah layanan internet berbasis satelit yang bernama *Starlink.* Selain itu, hasil akhir perhitungan klasifikasi dari analisis sentimen ini juga dapat digunakan oleh *SpaceX* sebagai bahan evaluasi pada produknya, yaitu *Starlink* untuk meningkatkan kualitas produk dan kepuasan para pelanggan.

2. METODE PENELITIAN

Penelitian ini menggunakan *Google Collaboratory* dan menggunakan bahasa pemrograman *Python*. Metode yang digunakan yaitu *K-Nearest Neighbor* yang dikombinasikan dengan metode SMOTE untuk analisis sentimen pada *Starlink.*

2.1. *K-Nearest Neighbor (KNN)*

Algoritma *K-Nearest Neighbor* adalah suatu metode klasifikasi yang didasarkan pada analogi, yaitu membandingkan data uji dengan data pelatihan yang berada dekat dengan dan memiliki kemiripan dengan data uji tersebut [10]. Algoritma *K-Nearest Neighbor* mengklasifikasikan data baru yang belum diketahui kelasnya dengan memilih k data yang berada paling dekat dengan data baru tersebut. Frekuensi kelas terbanyak dari sejumlah k data terdekat akan dipilih sebagai kelas prediksi untuk data baru [11]. Terdapat dua pendekatan perhitungan jarak yang umum digunakan untuk atribut yang bertipe numerik, yaitu [10]:

a. Euclidean Distance

Berikut adalah bentuk persamaan untuk *Euclidean Distance* :

DOI: https://doi.org/10.54082/jiki.132
P-ISSN: 2807-6664
E-ISSN: 2807-6591

$$d(x_1, x_2) = \sqrt{\sum_{i=1}^{n} (x_{1i} - x_{2i})^2} \quad (1)$$

Keterangan :
- n : jumlah keseluruhan data
- x_1 : data uji
- x_2 : data pembelajaran

b. Manhattan Distance

Berikut adalah bentuk persamaan untuk *Manhattan Distance* :

$$Dist(p_i(an), p_i(nc)) = \frac{p_i(an) - p_i(nc)}{(max\ \_dist\_i)} \quad (2)$$

Keterangan :
- p_i : atribut ke-*i*
- an : data pembelajaran
- nc : data uji

Hasil *accuracy*, *precission* dan *recall* pada proses akhir testing merupakan bagian yang sangat penting untuk melihat performa dari algoritma KNN. Tingkat akurasi yang semakin baik dan hasil sentimen yang sesuai prediksi kelasnya, maka hasil akurasi akan baik dan sebaliknya. Berikut adalah rumus dari accuracy, precission dan recall :

1) Accuracy

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

Keterangan :
- TP : *True Positive*
- FP : *False Positive*
- TN : *True Negative*
- FN : *False Negative*

2) Precission

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

Keterangan :
- *TP : True Positive*
- *FP : False Positive*

3) Recall

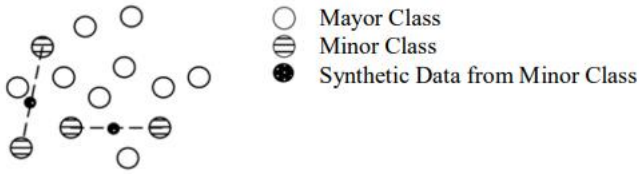
$$Recall = \frac{TP}{TP + FN} \quad (5)$$

Keterangan :
- *TP : True Positive*
- *FN : False Negative*

DOI: https://doi.org/10.54082/jiki.132
P-ISSN: 2807-6664
E-ISSN: 2807-6591

2.2. *Synthetic Minority Oversampling Technique (SMOTE)*

Salah satu pendekatan untuk menangani kasus kelas yang tidak seimbang adalah *Synthetic Minority Oversampling Technique (SMOTE)*, yang pertama kali diperkenalkan oleh *Nithes V. Chawla*. *Synthetic Minority Oversampling Technique* (SMOTE) merupakan salah satu metode resampling dengan tujuan untuk menyamakan distribusi kelas yang bekerja dengan cara mengambil sampel kelas minoritas dan melakukan penyisipan sampel sintesis untuk menambahkan jumlah data pada kelas minoritas [12]. Metode SMOTE bekerja dengan mencari k tetangga terdekat (yaitu k data terdekat sebanyak k) untuk setiap data dalam kelas minoritas, setelah itu data sintetis dibuat sebanyak persentase duplikasi yang diinginkan antara minoritas dan k tetangga terdekat yang dipilih secara acak [11]. Ilustrasi distribusi data setelah menerapkan SMOTE dapat dilihat pada Gambar 1.



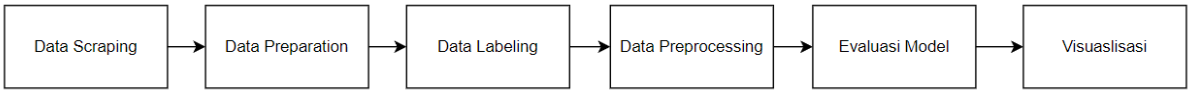
Gambar 1. Ilustrasi distribusi data dengan SMOTE

Rumus SMOTE :

$$synthetic = x_i + rand(0,1) \times \left(x_{neighbor} - x_i\right) \quad (6)$$

Keterangan:
- x_i adalah contoh minoritas asli.
- $x_{neighbor}$ adalah tetangga terdekat yang dipilih.
- $rand(0,1)$ adalah nilai acak antara 0 dan 1.

Dalam penelitian ini, terdapat beberapa tahapan meliputi *scraping* data, *cleaning & cleansing data*, *labeling data*, dan *preprocessing data*, *klasifikasi*, *evaluasi model*, dan visualisasi *word cloud*. *Flowchat* dari penelitian yang dilakukan dapat dilihat pada Gambar 2.



Gambar 2. *Flowchart* Penelitian

Keterangan :
a. Data *Scraping* : proses ekstraksi dan penggabungan konten yang menarik secara sistematis dan menyusun konten tersebut sesuai keinginan [13].
b. Data *Preparation* : proses persiapan data sebelum diolah, di mana pada tahap ini, data akan dibaca dan diproses dalam *workspace*.
c. Data *Labeling* : proses pemberian label pada data berdasarkan makna dalam komentar tersebut. Pada tahap ini, proses labelling data akan dilakukan menggunakan library Transformer. Transformer dengan cepat menjadi arsitektur dominan untuk pemrosesan bahasa alami, melampaui model neural alternatif seperti jaringan neural konvolusi dan jaringan neural rekuren dalam kinerja untuk tugas-tugas pemahaman bahasa alami dan generasi bahasa alami. Arsitektur ini dapat diskalakan dengan data pelatihan dan ukuran model, memfasilitasi pelatihan paralel yang efisien, dan menangkap fitur urutan jangka panjang [14].
d. Data *Preprocessing* : proses untuk mempersiapkan data yang telah diberi label agar dapat diterima oleh model untuk menghindari ambiguitas dan kesalahan yang dapat terjadi pada proses evaluasi model. Data *preprocessing* pada penelitian ini memiliki beberapa tahapan diantaranya :

DOI: https://doi.org/10.54082/jiki.132
P-ISSN: 2807-6664
E-ISSN: 2807-6591

1) *Casefolding*, teknik untuk mengurangi semua huruf kata menjadi huruf kecil yang memungkinkan perbandingan case-insensitive, semua teks yang ada dalam huruf besar didalam teks dikonversi ke bentuk huruf kecil [15].
2) *Cleaning* & *Cleansing*, data *Cleaning* adalah proses untuk mengenali dan memperbaiki (atau menghapus) catatan yang rusak atau tidak akurat dari satu set catatan, tabel, atau basis data [16]. Data *Cleansing* adalah operasi yang dilakukan pada data yang ada untuk menghilangkan anomali dan memperoleh kumpulan data yang merupakan representasi akurat dan unik, proses ini melibatkan menghilangkan kesalahan, menyelesaikan ketidakkonsistenan, dan mentransformasikan data ke dalam format yang seragam [17].
3) *Tokenizing*, proses pemisahan urutan karakter tertentu dalam suatu dokumen tertentu menjadi unit yang lebih kecil yang disebut dengan token dengan menghilangkan karakter unik seperti tanda baca [18].
4) *Normalize*, proses pengubahan kata-kata pada komentar yang belum menggunakan bahasa baku diubah menjadi menggunakan bahasa baku [19].
5) *Stopword Removal*, proses penghapusan kata yang sering muncul dan tidak memiliki arti apapun, contoh *Stopwords* dalam bahasa Indonesia seperti "yang", "di", "untuk", dan "dari" [20].
6) *Stemming*, mengubah kata menjadi kata dasar dengan menghapus awalan frasa [20].

e. Evaluasi Model : proses ini merupakan proses akhir untuk mengecek seberapa baik model yang dikembangkan dapat mengolah data yang diberikan. Pada proses dilakukan uji coba menggunakan confusion matrix. *Confusion matrix* adalah sebuah metode yang biasa digunakan untuk perhitungan akurasi. Dalam pengujian keakuratan hasil pencarian akan dievaluasi nilai *recall*, *precision*, *accuracy*, dan *error rate* [21].
f. Visualisasi : proses ini menampilkan hasil analisa sentimen yang dilakukan oleh model yang telah di kembangkan. Tampilan hasil analisa sentimen akan ditampilkan dalam bentuk *Word Cloud*. *Word cloud* merupakan kumpulan kata yang dihasilkan dari badan teks yang dapat berfungsi sebagai titik awal untuk analisis yang lebih dalam, word cloud dapat membantu menilai apakah teks yang diberikan relevan untuk kebutuhan informasi tertentu [22].

3. HASIL DAN PEMBAHASAN

Pada tahapan ini, penulis melakukan proses implementasi tahapan penelitian, dimulai dari pengumpulan data komentar pada platform Youtube hingga tahap evaluasi model menggunakan confusion matrix.

a. Tahap *Data Scraping*
Tahapan *data scraping* atau pengumpulan data dilakukan dengan mengumpulkan data komentar pada platform *Youtube* yang berkaitan dengan topik *Starlink* dengan menggunakan kata kunci dan rentang waktu yang telah ditentukan sebelumnya. Berdasarkan tahapan pengumpulan data, didapatkan 11,288 komentar dari salah satu channel *GadgetIn* terkait topik *Starlink*. Namun, saat melakukan *Scraping* data menggunakan *Netlytic API*, data komentar hanya terbatas pada 2500 data komentar, sehingga data yang akan digunakan pada tahapan selanjutnya berjumlah 2500 data komentar.
b. Tahap Data *Preparation*
Dari hasil data yang telah dikumpulkan sebanyak 2500 data komentar, selanjutnya akan dilakukan proses *preparation* data yaitu membaca dataset pada *Google Collaboratory* untuk kemudian akan dilakukan pengolahan.
c. Tahap *Labeling Data*
Secara sederhana, tahapan *labeling* adalah tahapan pemberian label atau tanda pada kolom sentimen berupa *neutral*, *negative*, ataupun *positive* pada masing-masing setiap baris komentar. Pada tahapan ini, kami melakukan proses *labeling* menggunakan salah satu library *Python* berupa *Huggingface Transformer*. Dari total 2500 komentar yang diperoleh, didapatkan sentimen *neutral* berjumlah 982 data, sentimen *negative* berjumlah 972 data dan sentimen *positive* berjumlah 546 data.

DOI: https://doi.org/10.54082/jiki.132
P-ISSN: 2807-6664
E-ISSN: 2807-6591

d. Tahap *Preprocessing* Data

Tahap preprocessing data terdiri dari beberapa proses yang terdiri dari *casefolding*, *cleansing*, *labeling*, *tokenizing*, *normalize*, *stopward removal* dan *stemming*.

1) Proses *Casefolding*,

Berikut adalah proses *casefolding* pada data komentar :

Tabel 1. Proses *Casefolding*

Sebelum	Sesudah
Harganya 7,8 juta dan biaya layanan 750rbu apakah bisa di pakai 10 user/perangkat? mohon infonya	harganya 7,8 juta dan biaya layanan 750rbu apakah bisa di pakai 10 user/perangkat? mohon infonya

Pada Tabel 1, kolom ”Sebelum” merupakan data *text* asli dari salah satu komentar pengguna yang masih mengandung kombinasi huruf besar dan kecil. Sedangkan, kolom ”Sesudah” merupakan data dalam bentuk *text* yang dilakukan proses *casefolding*, dimana semua huruf telah diubah menjadi huruf kecil (*lowercase*).

2) Proses *Cleaning & Cleansing*,

Berikut adalah proses *cleaning & cleansing* pada data komentar :

Tabel 2. Proses *Cleansing & Cleaning*

Sebelum	Sesudah
harganya 7,8 juta dan biaya layanan 750rbu apakah bisa di pakai 10 user/perangkat? mohon infonya	harganya 78 juta dan biaya layanan 750rbu apakah bisa di pakai 10 userperangkat mohon infonya

Pada Tabel 2, kolom ”Sebelum” merupakan *text* komentar pengguna yang telah melalui proses *casefolding*, namun masih mengandung karakter-karakter spesial seperti tanda baca. Sedangkan kolom ”Sesudah” merupakan *text* komentar yang dilakukan proses *cleaning & cleansing*, dimana karakter spesial yang berada pada kolom tersebut telah dihilangkan sehingga *text* komentar yang diolah akan memiliki format yang seragam.

3) Proses *Tokenizing*,

Berikut adalah proses *tokenizing* pada data komentar :

Tabel 3. Proses *Tokenizing*

Sebelum	Sesudah
harganya 78 juta dan biaya layanan 750rbu apakah bisa di pakai 10 userperangkat mohon infonya	[‘harganya’, ‘78’, ‘juta’, ‘dan’, ‘biaya’, ‘layanan’, ‘750rbu’, ‘apakah’, ‘bisa’, ‘di’, ‘pakai’, ‘10’, ‘userperangkat’, ‘mohon’, ‘infonya’]

Pada Tabel 3, kolom ”Sebelum” merupakan *text* komentar pengguna yang telah melalui proses *casefolding* dan *cleaning* & *cleansing*. Sedangkan, pada kolom ”Sesudah” merupakan *text* komentar yang dilakukan proses pemecahan karakter menjadi bentuk token (*tokenizing*).

4) Proses *Normalize*,

Berikut adalah proses *normalize* pada data komentar :

Tabel 4. Proses *Normalize*

Sebelum	Sesudah
[‘harganya’, ‘78’, ‘juta’, ‘dan’, ‘biaya’, ‘layanan’, ‘750rbu’, ‘apakah’, ‘bisa’, ‘di’, ‘pakai’, ‘10’, ‘userperangkat’, ‘mohon’, ‘infonya’]	[‘harganya’, ‘78’, ‘juta’, ‘dan’, ‘biaya’, ‘layanan’, ‘750rbu’, ‘apakah’, ‘mampu, ‘di’, ‘pakai’, ‘10’, ‘userperangkat’, ‘mohon’, ‘infonya’]

DOI: https://doi.org/10.54082/jiki.132
P-ISSN: 2807-6664
E-ISSN: 2807-6591

Pada Tabel 4, kolom ”Sebelum” merupakan *text* komentar yang telah melalui proses *casefolding*, *cleaning & cleansing*, dan *tokenizing*. Sedangkan, kolom ”Sesudah” merupakan *text* komentar yang dilakukan proses *normalize*, dimana *text* komentar dengan kata yang belum menggunakan bahasa baku akan diubah menjadi bahasa baku.

5) Proses *Stopword Removal*,
Berikut adalah proses *stopword removal* pada data komentar :

Tabel 5. Proses *Stopward Removal*

Sebelum	Sesudah
[‘harganya’, ‘78’, ‘juta’, ‘dan’, ‘biaya’, ‘layanan’, ‘750rbu’, ‘apakah’, ‘mampu’, ‘di’, ‘pakai’, ‘10’, ‘userperangkat’, ‘mohon’, ‘infonya’]	[‘harganya’, ‘biaya’, ‘layanan’, ‘mampu’, ‘pakai’, ‘userperangkat’, ‘mohon’, ‘infonya’]

Pada Tabel 5, kolom ”Sebelum” merupakan *text* komentar yang telah melalui proses *casefolding*, *cleaning & cleansing*, *tokenizing*, dan *normalize*. Sedangkan, kolom ”Sesudah” merupakan text komentar yang dilakukan proses *stopword removal*, dimana pada proses ini kata yang tidak memiliki arti akan dihilangkan dari *text* komentar.

6) Proses *Stemming*,
Berikut adalah proses *stemming* pada data komentar :

Tabel 6. Proses *Stemming*

Sebelum	Sesudah
[‘harganya’, ‘biaya’, ‘layanan’, ‘mampu’, ‘pakai’, ‘userperangkat’, ‘mohon’, ‘infonya’]	[‘harga’, ‘biaya’, ‘layan’, ‘mampu’, ‘pakai’, ‘userperangkat’, ‘mohon’, ‘info’]

Pada Tabel 6, kolom ”Sebelum” merupakan *text* komentar yang telah melalui proses *casefolding*, *cleaning & cleansing*, *tokenizing*, *normalize*, dan *stopword removal*. Sedangkan, kolom ”Setelah” merupakan text komentar yang dilakukan proses *stemming*, dimana pada proses ini kata dalam *text* komentar yang mengandung awalan frasa akan dihilangkan.

e. Evaluasi Model

Pada tahap ini, pengujian model akan dilakukan dengan menggunakan dua skenario yaitu metode KNN dan metode KNN yang dioptimasi menggunakan metode SMOTE, dimana masing-masing dari model akan diuji sebanyak 8 kali dengan menggunakan teknik *K-Fold Cross Validation*. Pengujian ini dilakukan terhadap parameter k pada metode *K-Nearest Neighbors* yang bervariasi untuk menemukan nilai parameter yang optimal dari rentang awal k = 3 hingga k = 10.

Berikut merupakan hasil performa skenario pertama yaitu model *K-Nearest Neighbor* (KNN):

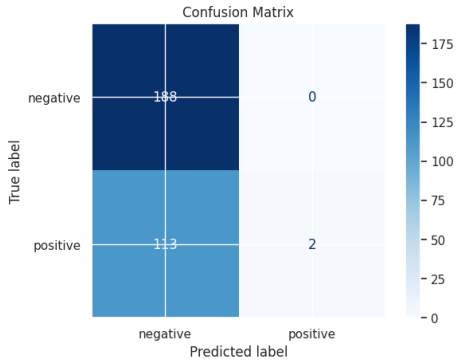
Tabel 7. Hasil Performa Model KNN

K	*Accuracy (Macro)*	*Precission(Macro)*	*Recall (Macro)*	*F1-Score (Macro)*
3	0,6337	0,8144	0,5174	0,4197
4	0,6271	0,8123	0,5087	0,4016
5	0,6271	0,8123	0,5087	0,4016
6	0,6271	0,8123	0,5087	0,4016
7	0,6799	0,6610	0,6238	0,6249
8	0,6601	0,6366	0,5944	0,5890
9	0,6931	0,6730	0,6514	0,6556
10	**0,6997**	**0,6847**	**0,6499**	**0,6541**

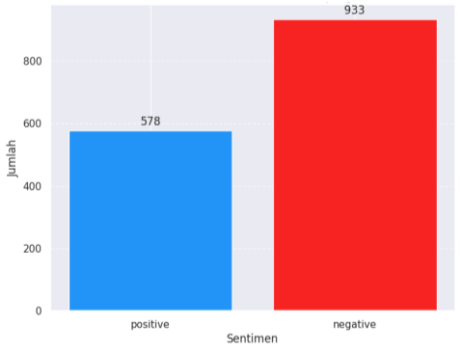
Dari hasil pengujian yang telah dilakukan model KNN terlampir pada Tabel 7, parameter k yang memiliki nilai yang optimal berada pada k=10 dengan nilai akurasi mencapai 69,97%, *precission (macro)* 68,47%, *recall(macro)* 64,99%, *F1-Score* 65,41%. Perhitungan *Confusion*

DOI: https://doi.org/10.54082/jiki.132
P-ISSN: 2807-6664
E-ISSN: 2807-6591

Matrix pada parameter k=10 dapat dilihat pada Gambar 3, sedangkan distribusi data sebelum dilakukan *resampling* dapat dilihat pada Gambar 4.



Gambar 3. *Confusion Matrix* model KNN



Gambar 4. Distribusi Sentimen sebelum *Resampling*

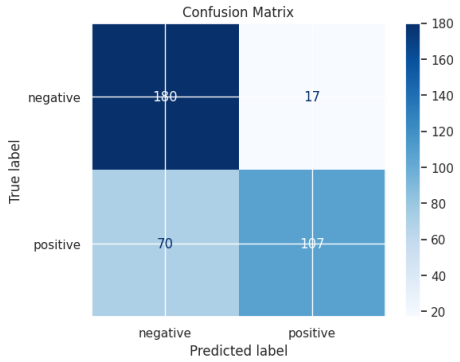
Kemudian, pada skenario kedua akan dilakukan pengujian pada metode KNN yang dioptimasi menggunakan metode SMOTE untuk dilakukan penyeimbangaan antara value *positive* dan *negative* pada kolom sentimen data. Berikut hasil performa skenario kedua yaitu model *K-Nearest Neighbor* (KNN) + SMOTE.

Tabel 8. Hasil Performa Model KNN + SMOTE

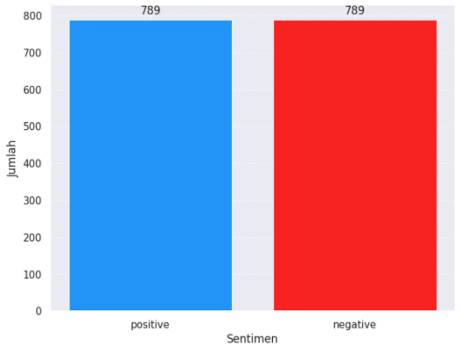
K	*Accuracy (Macro)*	*Precission(Macro)*	*Recall (Macro)*	*F1-Score (Macro)*
3	0,6390	0,7507	0,6511	0,6042
4	0,5455	0,7526	0,5431	0,4249
5	**0,7674**	**0,7915**	**0,7591**	**0,7582**
6	0,6257	0,7481	0,6014	0,5468
7	0,5401	0,6215	0,5464	0,4601
8	0,5107	0,6597	0,5277	0,3960
9	0,5267	0,6987	0,5364	0,4124
10	0,5642	0,7302	0,5596	0,4620

Dari hasil pengujian kedua yang telah dilakukan model KNN + SMOTE terlampir pada Tabel 8, parameter k yang memiliki nilai yang optimal berada pada k=5 dengan nilai akurasi mencapai 76,74%, *precission (macro)* 79,15%, *recall(macro)* 75,91%, *F1-Score* 75,82%. Perhitungan *Confusion Matrix* pada parameter k=5 dapat dilihat pada Gambar 5, sedangkan distribusi data setelah dilakukan *resampling* menjadi seimbang yakni dengan jumlah 789 data. Distribusi tersebut dapat dilihat pada Gambar 6.

DOI: https://doi.org/10.54082/jiki.132
P-ISSN: 2807-6664
E-ISSN: 2807-6591



Gambar 5. *Confusion Matrix* model KNN + SMOTE



Gambar 6. Distribusi Sentimen setelah *Resampling* dengan SMOTE

Secara keseluruhan, perbandingan antara performa model KNN dengan model KNN + SMOTE dapat dilihat pada Tabel 9.

Tabel 9. Perbandingan Performa model KNN dengan model KNN + SMOTE

Performance	**Model KNN**	**Model KNN + SMOTE**	**Jumlah Peningkatan**
Accuracy(Macro)	69,97%	76,74%	6,77%
Precission (Macro)	68,47%	79,15%	10,68%
Recall (Macro)	64,99%	75,91%	10,92%
F1-Score (Macro)	65,41%	75,82%	10,41%

Tabel 9 menunjukkan bahwa terjadi peningkatan performa setelah menggunakan model KNN + SMOTE. Nilai *Accuracy (Macro)* meningkat sebesar 6,77% dari awalnya 69,97% menjadi 76,74%. Nilai *Precission (Macro)* meningkat sebesar 10,68% dari awalnya 68,47% menjadi 79,15%. Nilai *Recall(Macro)* meningkat sebesar 10,92% dari awalnya 64,99% menjadi 75,91%. Nilai *F1-Score(Macro)* meningkat sebesar 10,41% dari awalnya 65,41% menjadi 75,82%.

f. Visualisasi *Word Cloud*

Visualisasi dalam bentuk *word cloud* pada penelitian ini menggunakan data sentimen *positive* dan *negative*. Word Cound tersebut dapat dilihat pada Gambar 7.

DOI: https://doi.org/10.54082/jiki.132
P-ISSN: 2807-6664
E-ISSN: 2807-6591

Gambar 7. Visualisasi Sentimen

Kata dengan ukuran *font* besar pada Gambar 7 merupakan kata yang sering muncul (dominan) pada sentimen, sedangkan kata dengan ukuran *font* kecil merupakan kata yang memiliki kemungkinan jarang muncul pada sentimen. Tulisan berwarna biru pada Gambar 7 merupakan visualisasi *word cloud* dengan sentimen *positive*, dimana kata yang sering muncul didominasi oleh kata *starlink*, cocok, banget, bang, internet, murah. Tulisan berwarna merah pada pada Gambar 7 merupakan visualisasi *word cloud* dengan sentimen *negative*, dimana kata yang sering muncul didominasi oleh kata bang, internet, *starlink*, mahal, dan sinyal.

4. KESIMPULAN

Penelitian ini membuktikan bahwa penggunaan SMOTE efektif dalam mengatasi ketidakseimbangan data pada analisis sentimen menggunakan algoritma *K-Nearest Neighbors* (KNN). Hasil penelitian menunjukkan peningkatan signifikan dalam performa model KNN setelah menggunakan SMOTE. Nilai *Accuracy (Macro)* meningkat sebesar 6,77% dari 69,97% menjadi 76,74%. Nilai *Precission (Macro)* meningkat sebesar 10,68% dari 68,47% menjadi 79,15%. Nilai *Recall (Macro)* meningkat sebesar 10,92% dari 64,99% menjadi 75,91%. Nilai *F1-Score (Macro)* meningkat sebesar 10,41% dari 65,41% menjadi 75,82%. Implementasi SMOTE membantu menciptakan distribusi data yang lebih seimbang, yang pada gilirannya meningkatkan kemampuan model dalam memprediksi kelas minoritas dengan lebih baik. Temuan ini mengindikasikan bahwa teknik *oversampling* seperti SMOTE dapat menjadi alat yang sangat berguna dalam analisis sentimen, terutama dalam konteks ulasan layanan yang bervariasi seperti *Starlink*. Namun, terdapat hal-hal yang perlu dipertimbangkan lagi ketika melakukan penelitian analisa sentimen menggunakan SMOTE dan algoritma KNN, yaitu ketika pada tahap data *labeling* perlu mempertimbangkan kembali penggunaan *Transformer* untuk memberikan label otomatis pada data sentimen, hal ini perlu diperhatikan untuk meningkatkan akurasi dan peforma pada model. Penggunaan *Transformer* untuk *labeling* data yang berbasis bahasa Indonesia, masih belum cukup stabil karena terdapat beberapa sentimen yang keliru pada saat proses *labeling* data.

Penelitian ini memberikan kontribusi penting dalam bidang analisis sentimen, menawarkan solusi praktis untuk mengatasi masalah ketidakseimbangan kelas (*imbalanced class*) dan meningkatkan akurasi model prediksi. Temuan ini dapat membantu penyedia layanan seperti *Starlink* dalam memahami opini publik dan meningkatkan kualitas layanan berdasarkan ulasan pengguna.

DAFTAR PUSTAKA

[1] T. Walasary, "Survey Paper tentang Analisis Sentimen," *KONSTELASI Konvergensi Teknol. dan Sist. Inf.*, vol. 2, no. 1, pp. 201–206, 2022, doi: 10.24002/konstelasi.v2i1.5378.

[2] J. W. Iskandar and Y. Nataliani, "Perbandingan Naïve Bayes, SVM, dan k-NN untuk Analisis Sentimen Gadget Berbasis Aspek," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 5, no. 6, pp. 1120–1126, 2021, doi: 10.29207/resti.v5i6.3588.

[3] M. W. Pertiwi, "Analisis Sentimen Opini Publik Mengenai Sarana Dan Transportasi Mudik Tahun 2019 Pada Twitter Menggunakan Algoritma Naïve Bayes, Neural Network, Knn Dan Svm," *Inti Nusa Mandiri*, vol. 14, no. 1, pp. 27–32, 2019, [Online]. Available:

DOI: https://doi.org/10.54082/jiki.132
P-ISSN: 2807-6664
E-ISSN: 2807-6591

http://www.nusamandiri.ac.id

[4] D. Nurmalasari, T. I. Hermanto, and I. M. Nugroho, "Perbandingan Algoritma SVM , KNN dan NBC Terhadap Analisis Sentimen Aplikasi Loan Service," *J. Media Inform. Budidarma*, vol. 7, no. 3, pp. 1521–1530, 2023, doi: 10.30865/mib.v7i3.6427.

[5] K. Karfindo, R. Turaina, and R. Saputra, "Optimalisasi Klasifikasi Umpan Balik Mahasiswa Terhadap Layanan Kampus dengan Sinergi Random Forest dan Smote," *J. Nas. Komputasi dan Teknol. Inf.*, vol. 6, no. 6, pp. 820–827, 2024, doi: 10.32672/jnkti.v6i6.7269.

[6] E. Sutoyo and M. A. Fadlurrahman, "Penerapan SMOTE untuk Mengatasi Imbalance Class dalam Klasifikasi Television Advertisement Performance Rating Menggunakan Artificial Neural Network," *J. Edukasi dan Penelit. Inform.*, vol. 6, no. 3, pp. 379–385, 2020, doi: 10.26418/jp.v6i3.42896.

[7] A. Surya Firmansyah, A. Aziz, and M. Ahsan, "Optimasi K-Nearest Neighbor Menggunakan Algoritma Smote Untuk Mengatasi Imbalance Class Pada Klasifikasi Analisis Sentimen," *JATI (Jurnal Mhs. Tek. Inform.*, vol. 7, no. 6, pp. 3341–3347, 2024, doi: 10.36040/jati.v7i6.7257.

[8] Kurnia, I. Purnamasari, and D. D. Saputra, "Analisis Sentimen Dengan Metode Naïve Bayes, SMOTE Dan Adaboost Pada Twitter Bank BTN," *J. JTIK (Jurnal Teknol. Inf. dan Komunikasi)*, vol. 7, no. 2, pp. 235–242, 2023, doi: 10.35870/jtik.v7i3.707.

[9] A. F. Anjani, D. Anggraeni, and I. M. Tirta, "Implementasi Random Forest Menggunakan SMOTE untuk Analisis Sentimen Ulasan Aplikasi Sister for Students UNEJ," *J. Nas. Teknol. dan Sist. Inf.*, vol. 9, no. 2, pp. 163–172, 2023, doi: 10.25077/teknosi.v9i2.2023.163-172.

[10] A. Y. Permana and M. M. Effendi, "Analisis Sentimen pada Teks Opini Penilaian Kinerja Dosen dengan Pendekatan Algoritma KNN," *J. Ilm. Komputasi*, vol. 19, no. 1, pp. 39–50, 2020, doi: 10.32409/jikstik.19.1.154.

[11] A. G. Pertiwi, N. Bachtiar, R. Kusumaningrum, I. Waspada, and A. Wibowo, "Comparison of performance of k-nearest neighbor algorithm using smote and k-nearest neighbor algorithm without smote in diagnosis of diabetes disease in balanced data," *J. Phys. Conf. Ser.*, vol. 1524, no. 1, pp. 1–9, 2020, doi: 10.1088/1742-6596/1524/1/012048.

[12] A. Nurhopipah and C. Magnolia, "Perbandingan Metode Resampling Pada Imbalanced Dataset Untuk Klasifikasi Komentar Program Mbkm," *J. Publ. Ilmu Komput. dan Multimed.*, vol. 1, no. 2, pp. 9–22, 2022, doi: 10.55606/jupikom.v2i1.862.

[13] D. Glez-Peña, A. Lourenço, H. López-Fernández, M. Reboiro-Jato, and F. Fdez-Riverola, "Web scraping technologies in an API world," *Brief. Bioinform.*, vol. 15, no. 5, pp. 788–797, 2013, doi: 10.1093/bib/bbt026.

[14] T. Wolf *et al.*, "Transformers: State-of-the-Art Natural Language Processing," *EMNLP 2020 - Conf. Empir. Methods Nat. Lang. Process. Proc. Syst. Demonstr.*, vol. 5, pp. 38–45, 2020, doi: 10.18653/v1/2020.emnlp-demos.6.

[15] S. Bauskar, V. Badole, P. Jain, and M. Chawla, "Natural Language Processing based Hybrid Model for Detecting Fake News Using Content-Based Features and Social Features," *Int. J. Inf. Eng. Electron. Bus.*, vol. 11, no. 4, pp. 1–10, 2019, doi: 10.5815/ijieeb.2019.04.01.

[16] T. Tarannum, "Cleaning of Web Scraped Data with Python," Brac University, 2019.

[17] P. Jeatrakul, K. W. Wong, and C. C. Fung, "Data Cleaning for Classification Using Misclassification Analysis," *IWACIII 2009 - Int. Work. Adv. Comput. Intell. Intell. Informatics*, vol. 14, no. 3, pp. 297–302, 2010.

[18] K. N. Sridevi and S. Prakasha, "Doc-To-Tokens based Pre-Processing in Information Retrieval System," *Webology*, vol. 18, no. Special Issue, pp. 570–579, 2021, doi: 10.14704/WEB/V18SI05/WEB18247.

[19] A. Guswandri, R. P. Cahyono, S. I. Akutansi, and T. Komputer, "Penerapan Sentimen Analis Menggunakan Metode Naïve Bayes Dan Svm," *Ilmudata.org*, vol. 2, no. 12, pp. 1–16, 2022.

DOI: https://doi.org/10.54082/jiki.132
P-ISSN: 2807-6664
E-ISSN: 2807-6591

[20] A. W. Pradana and M. Hayaty, "The Effect of Stemming and Removal of Stopwords on the Accuracy of Sentiment Analysis on Indonesian-language Texts," *Kinet. Game Technol. Inf. Syst. Comput. Network, Comput. Electron. Control*, vol. 4, no. 4, pp. 375–380, 2019, doi: 10.22219/kinetik.v4i4.912.

[21] Melita Ria, "Penerapan Metode Term Frequency Inverse Document Frequency (TF-IDF) dan Cosine Similarity pada Sistem Temu Kembali Informasi Untuk Mengetahui Syarah Hadits Berbasis Web," Universitas Islam Negeri Syarif Hidayatullah Jakarta, 2018.

[22] F. Heimerl, S. Lohmann, S. Lange, and T. Ertl, "Word cloud explorer: Text analytics based on word clouds," *Proc. Annu. Hawaii Int. Conf. Syst. Sci.*, pp. 1833–1842, 2014, doi: 10.1109/HICSS.2014.231.