

## **Analisis Sentimen Pengguna Instagram terhadap Timnas Indonesia U-23 pada Piala AFC menggunakan Algoritma K-Nearest Neighbor (K-NN) dengan *Synthetic Minority Over-sampling Technique* (SMOTE)**

### ***Sentiment Analysis of Instagram User's towards the Indonesia U-23 National Team in the AFC Cup using the K-Nearest Neighbor (K-NN) Algorithm with Synthetic Minority Over-sampling Technique (SMOTE)***

**Surya Pratama Tanzil\*, Muhammad Rizky Pribadi**

Universitas Multi Data Palembang

Jl. Rajawali No.14, 9 Ilir, Kec. Ilir Tim. II, Kota Palembang, Sumatera Selatan 30113

Email\*: suryapratamatanzil@mhs.mdp.ac.id

**Abstrak** - Timnas Indonesia U-23 adalah tim nasional sepakbola Indonesia dengan usia pemainnya berada dibawah 23. Sentimen masyarakat Indonesia terhadap Timnas Indonesia U-23 meningkat dengan pesat selama periode piala AFC 2024. Hal tersebut dapat dilihat pada salah satu platform sosial media *online* bernama instagram. Dengan menggunakan instagram, beragam informasi bisa didapatkan berdasarkan yang diberikan oleh masyarakat. Penelitian ini bertujuan untuk mengetahui sentimen masyarakat instagram terhadap Timnas Indonesia U-23 selama periode piala AFC berdasarkan komentar yang diambil pada platform sosial media *online* Instagram. Tujuan selanjutnya adalah untuk mengetahui tingkat akurasi analisis sentimen terhadap Timnas Indonesia U-23 menggunakan metode K-NN dengan SMOTE. Hasil penelitian ini adalah ketika menangani *dataset* yang didapatkan dari teknik *scraping* memiliki kondisi data yang tidak seimbang, maka teknik SMOTE dapat meningkatkan akurasi model sebesar 5% hingga 23%. Berdasarkan eksperimen yang dilakukan, kesimpulan penelitian ini adalah hasil akurasi terbaik diperoleh ketika SMOTE diterapkan dalam pengimplementasian k-NN dengan akurasi 63% untuk  $k = 5$ .

**Kata kunci:** k-NN, SMOTE, sentimen, akurasi.

**Abstract** - The Indonesia U-23 National Team is Indonesia's national football team with players under 23 years old. Indonesian public sentiment towards the Indonesia U-23 National Team increased significantly during the 2024 AFC Cup period. This can be observed on the online social media platform Instagram. By using Instagram, various information can be obtained based on public input. This study aims to determine the sentiment of Instagram users towards the Indonesia U-23 National Team during the AFC Cup period based on comments taken from the Instagram platform. The next goal is to determine the accuracy level of sentiment analysis towards the Indonesia U-23 National Team using the K-Nearest Neighbor (k-NN) method with SMOTE. The results of this study indicate that when handling a dataset obtained through scraping techniques, which has an imbalance condition, the SMOTE technique can improve the model's accuracy by 5% to 23%. Based on the experiments conducted, the conclusion of this study is that the best accuracy was achieved when SMOTE was applied in the implementation of k-NN, with an accuracy of 63% for  $k = 5$ .

**Keywords :** k-NN, SMOTE, sentiment, accuracy.

## **I. PENDAHULUAN**

Timnas Indonesia U-23 adalah tim nasional sepakbola Indonesia dengan usia pemainnya

berada dibawah 23[1]. Sepakbola merupakan aktifitas olahraga yang termasuk sebagai salah satu olahraga paling terkenal di dunia. Sepakbola sudah

mengalami banyak transformasi yang membuat sepakbola saat ini sudah berkembang jauh, dibanding sebelumnya yang masih menggunakan peraturan yang tradisional hingga saat ini olahraga sepakbola menjadi dicintai oleh banyak orang[2]. Sentimen masyarakat Indonesia terhadap Timnas Indonesia U-23 meningkat dengan pesat selama periode piala AFC 2024. Hal tersebut dapat dilihat pada salah satu platform sosial media *online* bernama instagram. Instagram termasuk sebagai platform sosial media *online* yang banyak digunakan masyarakat pada era saat ini[3]. Untuk memudahkan Timnas Indonesia U-23 mengetahui bagaimana respon masyarakat terhadap Timnas Indonesia U-23 selama periode piala AFC dalam platform instagram, analisis sentimen menjadi sebuah teknik yang bernilai.

Data-data sentimen yang diambil dari platform sosial media untuk menjadi dataset yang dapat di analisis dan diklasifikasikan dengan metode *machine learning*. Metode *machine learning* yang dapat digunakan meliputi *Naïve Bayes* (NB), *K-Nearest Neighbor* (K-NN), *Support Vector Machine* (SVM), *Random Forest*, dan *Decision Tree* (DT)[4]. Algoritma-algoritma tersebut memanfaatkan berbagai metode dalam mempelajari data latih serta memprediksi mengenai *sample* baru. Dalam menilai kinerja algoritma klasifikasi, biasanya data dibagi menjadi data latih dan data uji. Data latih digunakan untuk *training* model, sedangkan data uji digunakan untuk mengukur performanya. Evaluasi dijalankan melalui *matrix* seperti akurasi, presisi, dan *recall*. Sehingga, dapat diketahui seberapa tepat model dalam memprediksi yang benar.

Pada penelitian terdahulu yang melakukan analisa sentimen mengenai relokasi ibukota Nusantara menggunakan metode *Naïve Bayes* dan *K-NN*. Berdasarkan penelitian tersebut, didapatkan akurasi *Naïve Bayes* sebesar 82,27% dan akurasi *K-NN* sebesar 88,12% [5]. Lalu, pada penelitian lainnya yang membahas perbandingan antara metode klasifikasi *Random Forest*, *Naïve Bayes*, dan *K-NN* baik untuk data yang seimbang maupun tidak seimbang dengan menggunakan metode *Adaptive Synthetic* pada dataset LCMS tanaman keladi tikus, didapatkan akurasi *Random Forest* sebesar 99%, *KNN* sebesar 98,5%, dan *Naïve Bayes* sebesar 98,5%, untuk data yang bersifat tidak seimbang. Namun, ketika data bersifat seimbang diperoleh akurasi *Random Forest* sebesar 89,6%, *KNN* sebesar 95,8%, dan *Naïve Bayes* sebesar 48,5%[6]. Adapula, penelitian yang membahas mengenai klasifikasi air menggunakan *KNN*, *Naïve Bayes*, dan *Decision Tree*, didapat akurasi *Naïve*

*Bayes* sebesar 63,60%, *KNN* sebesar 86,88% dan *Decision Tree* sebesar 80,84% [7]. Kemudian, pada penelitian yang membahas mengenai analisis sentimen pada kegiatan trading menggunakan klasifikasi *SVM*, *KNN*, dan *Random Forest* diperoleh perbandingan akurasi untuk *SVM* 99,2%, *k-NN* 99,9% dan *Random Forest* 99,4%[8]. Dari penelitian terdahulu, klasifikasi *K-Nearest Neighbor* (K-NN) sering digunakan karena efisien dalam mengklasifikasikan data berdasarkan jarak. Namun, K-NN memiliki kekurangan dalam perhitungan jarak yang lambat pada data berdimensi tinggi dan kelas yang tidak seimbang[9]. Ketidakseimbangan kelas terjadi ketika jumlah label data berbeda, dengan label negatif lebih besar dari label positif. Untuk mengatasi masalah ini, digunakan metode *oversampling SMOTE* (*Synthetic Minority Oversampling Technique*) yang membangkitkan data baru untuk menyeimbangkan kelas [10].

Penelitian ini bertujuan untuk mengetahui sentimen masyarakat instagram terhadap Timnas Indonesia U-23 selama periode piala AFC. Kemudian, tujuan selanjutnya adalah untuk mengetahui tingkat akurasi analisis sentimen terhadap Timnas Indonesia U-23 menggunakan metode K-NN dengan SMOTE.

## II. METODOLOGI

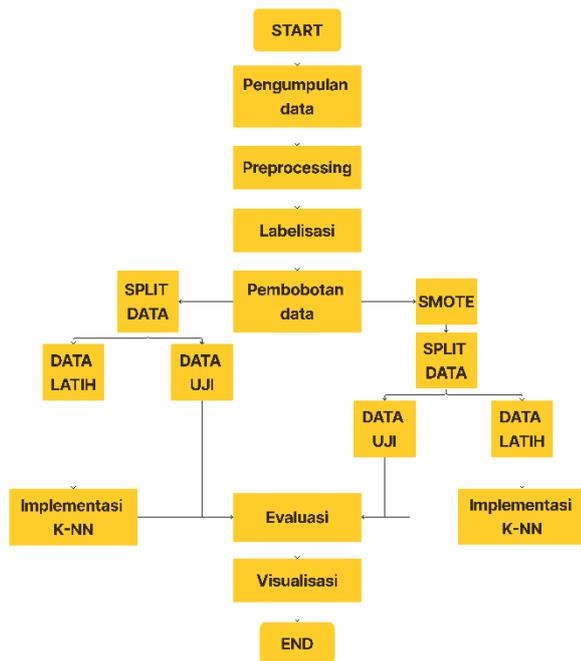
Dalam penelitian ini, penulis akan melakukan dua eksperimen dalam mengklasifikasikan sentimen. Penelitian ini akan menggunakan algoritma k-NN dengan SMOTE dan tanpa menggunakan SMOTE. Tahapan pengerjaan ini meliputi pengumpulan data, *preprocessing*, pemilihan algoritma, dan evaluasi model yang berada pada **Gambar 1**.

### A. Pengumpulan Data

Di tahap ini akan dilakukan pengumpulan data dari instagram terkait komentar masyarakat Indonesia mengenai timnas Indonesia di aplikasi instagram. Pengumpulan data ini akan melalui dua tahap yakni, "*scraping data*" dan "*cleansing*". *scraping* data adalah proses yang dilakukan dengan mengambil sejumlah komentar[11].

Setelah data terkumpul akan dilakukan proses *cleansing*. *Cleansing* adalah proses membersihkan kata-kata dengan menghapus tanda koma, titik, dan tanda baca lainnya[11]. Termasuk menghapus *mention*, *hashtags*, *RT*, dan *URL*. Setelah proses *cleansing*, tahap selanjutnya akan dilakukan *casefolding* agar setiap kalimat menjadi huruf kecil. *Casefolding* adalah proses pra-pemrosesan yang akan memproses seluruh suku kata yang

bertipe string menjadi suku kata kecil dari suku kata a hingga huruf z [5].



Gambar 1. Tahapan Pengerjaan

## B. Preprocessing

*Preprocessing* adalah hal pertama yang dilakukan dengan tujuan membersihkan *data text* dari simbol, emoji, ataupun *hashtag*. *Preprocessing* juga berfungsi dalam proses normalisasi *data text*[9]. Pada bagian ini akan dilakukan pengolahan *dataset* sehingga menghasilkan data yang jelas[12]. Tahap *preprocessing* ini terdiri dari normalisasi, *stopword*, *tokenize* dan *stemming*. Tahap pertama adalah normalisasi. Pada tahap normalisasi, setiap penyingkatan yang ada dalam teks komentar akan di normalisasi sehingga menjadi kalimat yang baik.

Tahap berikutnya adalah *stopword*. *Stopword* adalah langkah preprocessing yang menghilangkan kata-kata umum dari suatu bagian kalimat yang dianggap tidak memiliki makna ataupun dianggap kasar sehingga tidak layak untuk dilihat. Tahap *stopword* ini bisa menghentikan kata yang biasa digunakan sebagai bentuk negatif. *Stopword* ini biasanya diisi dengan “tidak”, “jelek”, dan lain-lain yang biasa muncul dalam kalimat negatif. Kata-kata pada *stopword* akan dibuang dari dataset kemudian dilakukan pemrosesan sehingga set data bisa dianalisa. Hal ini disebabkan mereka tidak ada banyak tambahan makna kalimat dari set teks yang digunakan dan mungkin meningkatkan dimensi data secara tidak perlu. Menghapus kata-kata dalam *stopwords* akan mengurangi beban komputasi pada pengolah kata dan meningkatkan kinerja beberapa algoritma pembelajaran mesin.

Kemudian, dilakukan *tokenize*. *Tokenize* melibatkan pembagian teks menjadi kata individual. Tokenisasi adalah salah satu tahapan dalam *preprocessing* yang umum ketika bekerja dengan data teks, karena memfasilitasi pemrosesan dan analisis teks. Tahap *tokenize* menggunakan operator *tokenize* seperti pada Rapid Miner. Tahap *tokenize* merupakan langkah penting dalam pemrosesan teks, khususnya untuk mengubah teks menjadi format yang dapat dianalisis oleh algoritma *machine learning*. Operator ini bekerja dengan cara memecah teks menjadi unit-unit yang lebih kecil, yaitu kata-kata atau token. Ini digunakan untuk mengiris proses data yang sebelumnya terdiri dari kalimat menjadi kata-kata[5]. Lalu, tahap berikutnya adalah *stemming*. *Stemming* adalah tahap dimana kata-kata yang mengandung imbuhan diubah menjadi kata dasar. Proses ini berfokus pada pengubahan kata-kata yang mengandung imbuhan menjadi bentuk dasarnya. Pada tahap ini Python dan Sastrawi akan digunakan untuk mencari kata dasar[13].

Dalam *preprocessing*, akurasi ketika akan mengevaluasi model dataset sentimen dapat ditingkatkan dengan menerapkan teknik-teknik preprocessing pada dataset. Teknik-teknik ini mampu mengurangi jumlah fitur dalam data, sehingga algoritma yang melatih model analisis sentimen lebih mudah mengidentifikasi pola-pola yang berkaitan dengan label data. Dengan lebih sedikit fitur, model dapat fokus pada informasi yang benar-benar penting, mengurangi kebisingan yang tidak relevan. Di sisi lain, dari sudut pandang ekstraksi fitur, penggunaan *Term Frequency-Inverse Document Frequency* (TF-IDF) dapat meningkatkan akurasi model analisis sentimen. TF-IDF memberikan bobot yang berbeda untuk setiap fitur berdasarkan frekuensinya dalam dokumen dan seberapa umum fitur tersebut di seluruh korpus. Bobot ini membantu algoritma pemodelan dalam mengenali fitur-fitur yang paling relevan dan signifikan dalam data yang digunakan[14].

## C. Labelisasi

Pada tahap ini, akan dilakukan proses pemberian label secara manual pada komentar-komentar Instagram. Dalam proses pelabelan ini, komentar yang diberi label positif berarti mendukung Timnas Indonesia, menunjukkan dukungan atau apresiasi terhadap tim tersebut. Komentar yang diberi label netral berarti isinya tidak berhubungan dengan Timnas Indonesia, tidak menunjukkan dukungan maupun kritik. Sedangkan, komentar yang diberi label negatif

berarti tidak mendukung Timnas Indonesia, mengandung kritik atau pandangan yang kurang positif terhadap tim.

Proses pemberian label ini akan dilakukan dengan hati-hati dan teliti oleh pakar Bahasa Indonesia yang berkompeten dalam memahami nuansa bahasa dan konteks sosial. Pakar tersebut akan menilai setiap komentar secara individual untuk memastikan bahwa label yang diberikan akurat dan mencerminkan sentimen yang sebenarnya dari komentar tersebut. Dengan menggunakan keahlian mereka, pakar Bahasa Indonesia dapat mengidentifikasi subtleties dalam bahasa dan konteks, yang mungkin tidak terlihat jelas pada pandangan pertama. Melalui proses ini, *dataset* yang dihasilkan akan memiliki label yang tepat dan dapat diandalkan, yang selanjutnya akan digunakan untuk analisis sentimen dan berbagai aplikasi *machine learning* lainnya. Dengan melakukan hal ini dapat membuat model menghasilkan nilai akurasi yang lebih baik.

#### D. Pembobotan Data

Pembobotan data adalah teknik pemberian nilai untuk teks-teks di dalam *dataset*. Cara umum yang cenderung digunakan dalam tahap pembobotan data adalah TF-IDF (*Term Frequency-Inverse Document Frequency*)[9]. TF-IDF merupakan metode *feature extraction* yang menggunakan pendekatan *word2vec*, TF-IDF memberikan bobot pada fitur berdasarkan seberapa sering fitur tersebut muncul dalam dataset tertentu (misalnya twitter, instagram dan youtube) dibandingkan dengan frekuensi kemunculannya dalam seluruh dataset.

Penggunaan TF-IDF bertujuan untuk memastikan bahwa fitur-fitur umum dalam percakapan sehari-hari tidak mendominasi dan mengurangi dampak fitur-fitur lain dalam analisis sentimen[14]. Fungsi pembobotan memainkan peran penting dalam situasi di mana fitur-fitur tertentu memiliki tingkat kepentingan atau relevansi yang lebih tinggi terhadap tugas yang sedang dihadapi. Penggunaan bobot ini bisa sangat bermanfaat untuk meningkatkan performa berbagai algoritma pembelajaran mesin, karena bobot memberikan model informasi yang lebih banyak mengenai relatifitas dari masing-masing fitur. Dengan menekankan fitur-fitur yang lebih penting, model dapat membuat prediksi yang lebih akurat dan efisien.

Namun, memilih bobot yang tepat adalah sebuah tantangan yang signifikan. Pilihan bobot yang salah dapat berdampak negatif pada performa model secara keseluruhan. Oleh karena itu, sangat

penting dalam menilai skema pembobotan dengan detil untuk mencegah kesalahan pemberian bobot, memastikan bahwa bobot yang dipilih cocok dengan karakteristik data dan masalah spesifik yang ingin diselesaikan. Hal ini mungkin melibatkan eksperimen dan penyesuaian berulang kali untuk menemukan kombinasi bobot yang optimal, yang pada akhirnya akan menghasilkan model dengan kinerja terbaik.

Selain itu, penting juga untuk memperhitungkan metode untuk menetapkan bobot, apakah melalui teknik statistik, domain *knowledge*, atau algoritma otomatis seperti pengoptimalan hyperparameter. Pendekatan yang sistematis dan berbasis data akan membantu memastikan bahwa bobot yang diberikan benar-benar mencerminkan kepentingan relatif fitur-fitur dalam konteks masalah yang ada.[5].

#### E. Implementasi k-NN

Proses klasifikasi dilanjutkan dengan mengimplementasikan *K-Nearest Neighbor* (k-NN). Klasifikasi adalah teknik penting dalam pembelajaran mesin yang bertujuan untuk memprediksi label kelas dari sampel berdasarkan fitur-fiturnya. Dengan memanfaatkan data tentang atribut seperti usia, jenis kelamin, dan pendapatan, model klasifikasi dapat dilatih untuk membuat prediksi yang akurat mengenai berbagai aspek, termasuk kemungkinan seseorang memiliki pekerjaan dengan gaji tinggi atau tidak. Proses ini melibatkan pemilihan fitur yang relevan, pelatihan model dengan data yang memadai, dan evaluasi kinerja model untuk memastikan keakuratan prediksi.[5]. Pada tahap ini akan dilakukan *split data* untuk membagi *data* latihan yang akan dilatih dalam model k-NN dan *data test* untuk *testingnya*.

*K-Nearest Neighbor* (KNN) didasarkan pada konsep "belajar dengan analogi." Dalam konteks pembelajaran mesin, atribut numerik dari dataset sering direpresentasikan dalam ruang n-dimensi. Setiap titik dalam ruang ini, yang mewakili data pembelajaran, diberi label kelas tertentu, misalnya c. Ketika ada data query baru yang tidak berlabel dimasukkan, algoritma *K-Nearest Neighbor* (k-NN) akan mencari sejumlah k titik data pelatihan yang paling dekat dengan data query tersebut dalam ruang n-dimensi.

Untuk menemukan titik-titik terdekat ini, algoritma k-NN mengukur jarak antara *data query* dan setiap data pelatihan. Jarak ini biasanya dihitung menggunakan metrik jarak Euclidean. Jarak Euclidean merupakan jarak garis lurus antara dua titik dalam ruang n-dimensi dan dihitung menggunakan rumus akar kuadrat dari jumlah

kuadrat perbedaan setiap koordinat yang disesuaikan dari kedua titik tersebut.

Lebih detailnya, algoritma k-NN bekerja dengan cara sebagai berikut: pertama, data query dan data pelatihan dipetakan ke dalam ruang n-dimensi. Dalam ruang ini, setiap dimensi merepresentasikan satu fitur dari data tersebut. Misalnya, jika data memiliki tiga fitur, maka data tersebut akan dipetakan dalam ruang tiga dimensi. Algoritma kemudian menghitung jarak Euclidean antara data query dan setiap titik data pelatihan untuk menentukan seberapa dekat mereka satu sama lain. Dengan demikian, algoritma k-NN memberikan pendekatan intuitif dan langsung untuk klasifikasi dengan memanfaatkan konsep kedekatan dalam ruang fitur n-dimensi, yang membuatnya sangat berguna dalam berbagai aplikasi pembelajaran mesin.

Setelah jarak antara data query dan semua titik data pelatihan dihitung, algoritma k-NN akan memilih k titik data pelatihan yang jaraknya paling dekat dengan data query. Label kelas dari data query kemudian ditentukan berdasarkan mayoritas label kelas dari k titik terdekat ini. Dengan kata lain, data query akan diberi label kelas yang paling banyak muncul di antara k tetangganya. Misalnya, jika data memiliki tiga fitur, maka titik-titik tersebut akan berada dalam ruang tiga dimensi. Algoritma kemudian menghitung *Euclidean Distance* dengan formula memiliki formula (1).

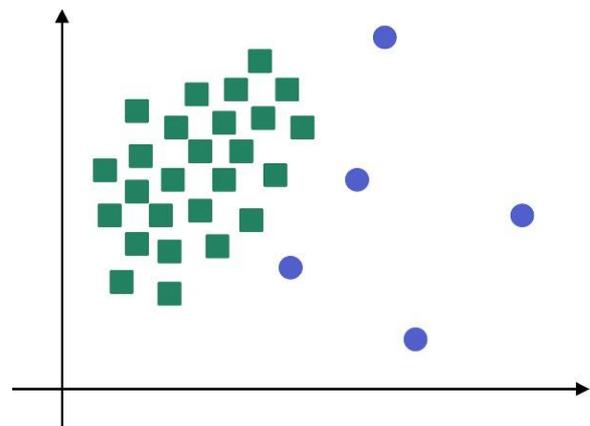
$$dist = \sum_{i=1}^p \sqrt{(x_2 + x_1)^2} \quad \dots (1)$$

*Dist* mengukur jarak antara skalar dua vektor melalui matriks berdimensi d, dengan x1 sebagai data latih dan x2 sebagai data uji. Semakin besar nilai *dist*, semakin jauh jarak antara dua individu, dan sebaliknya. Algoritma k-NN menghitung jarak antara *data query* dan *data latih*, lalu mengidentifikasi k terdekat dengan jarak terpendek. Pada tahap klasifikasi, jarak dihitung antara data baru dan seluruh data latih, kemudian dipilih k data terdekat, dan klasifikasi ditentukan berdasarkan mayoritas klasifikasi dari k data terdekat tersebut. Pemilihan nilai k yang optimal sangat penting dalam algoritma ini dan sangat bergantung pada data yang digunakan. Secara umum, memilih nilai k yang lebih besar dapat membantu mengurangi efek *noise* pada hasil klasifikasi. Namun, ini juga dapat membuat batas antara setiap klasifikasi menjadi lebih kabur, sehingga mungkin mengurangi akurasi klasifikasi di sekitar area batas keputusan. Untuk menemukan nilai k yang optimal, teknik optimasi parameter seperti *cross-validation* dapat diterapkan. *Cross-*

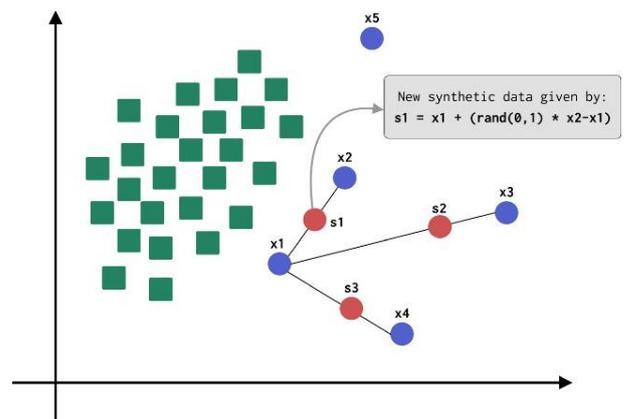
*validation* dilakukan untuk menemukan berbagai nilai k dan memilih yang memberikan hasil terbaik berdasarkan kinerja pada data validasi. Dengan demikian, dapat ditemukan nilai k dengan keseimbangan yang tepat antara mengurangi noise dan mempertahankan kejelasan batas klasifikasi[15].

### F. SMOTE

*SMOTE* berguna dalam membuat data menjadi *balance* baik dalam kelas positif dan negatif. Cara kerja algoritma *SMOTE* dilakukan dengan menambahkan sampel sintesis berdasarkan tetangga yang berdekatan dengan cara mencari selisih jarak masing-masing tetangga[9]. Sehingga, jumlah data yang akan dihasilkan berjumlah sama. Pada **Gambar 2** dan **Gambar 3** adalah penggambaran kelas ketika tidak seimbang dengan kondisi kelas setelah diterapkan *SMOTE*[16].



Gambar 2. Data sebelum seimbang



Gambar 3. Ketika menyeimbangkan data

Dengan demikian, melalui penambahan *SMOTE* tidak hanya membantu dalam menyeimbangkan data tetapi juga memperkuat representasi kelas minoritas. Ini sangat penting dalam berbagai aplikasi nyata, seperti diagnosis medis, deteksi anomali, dan analisis risiko, di mana kesalahan

dalam mendeteksi kejadian minoritas bisa berakibat serius.

### G. Implementasi k-NN dengan SMOTE

Selanjutnya, lakukan pengujian dengan menambahkan SMOTE (*Synthetic Minority Over-sampling Technique*). Pada tahap ini, data sintetis akan diciptakan untuk menyeimbangkan kelas yang sebelumnya tidak seimbang. Dengan menggunakan SMOTE, masalah ketidakseimbangan kelas dapat diatasi. Proses SMOTE bekerja dengan mengambil sampel dari kelas minoritas dan menggunakan teknik interpolasi untuk menciptakan contoh baru yang berada di antara sampel yang ada. Sampel baru ini bukan salinan langsung dari sampel yang ada, tetapi variasi yang dihasilkan dengan menggabungkan fitur-fitur dari beberapa sampel yang ada. Setelah data sintetis ditambahkan, dataset menjadi lebih seimbang. Model pembelajaran mesin yang dilatih pada dataset yang seimbang ini cenderung lebih baik dalam mengenali pola dari kelas minoritas, sehingga meningkatkan kemampuan model untuk mendeteksi kejadian yang jarang tetapi penting.

### H. Evaluasi

Pada tahap evaluasi ini, *confusion matrix* akan digunakan untuk mendapatkan nilai *accuracy*, *precision*, *recall*, dan *F1 Score*. *Confusion matrix* adalah alat yang sangat berguna dalam memeriksa kinerja algoritma klasifikasi, karena membantu melihat secara *detail* bagaimana hasil klasifikasi dibandingkan dengan label sebenarnya.

*Accuracy* mengukur sejauh mana sistem melakukan proses klasifikasi dengan benar, yaitu persentase prediksi yang tepat dari total prediksi yang dibuat. *Precision* adalah rasio jumlah dokumen yang relevan dan ditemukan oleh sistem klasifikasi terhadap total dokumen yang ditemukan. Ini menunjukkan ketepatan klasifikasi dalam menemukan dokumen yang relevan. *Recall*, di sisi lain, adalah rasio jumlah dokumen relevan yang ditemukan kembali oleh sistem klasifikasi terhadap total dokumen relevan yang ada dalam dataset. Ini mengukur kemampuan sistem dalam menemukan semua dokumen yang relevan.

*F1 Score* kemudian digunakan sebagai metrik yang mengatasi kelemahan dari *precision* dan *recall* ketika digunakan secara terpisah. *F1 Score* dihitung sebagai rata-rata harmonis dari *precision* dan *recall*, memberikan keseimbangan antara keduanya. Namun, karena *precision* dan *recall* hanya fokus pada kelas positif, *F1 Score* pun hanya memberikan gambaran kinerja untuk kelas

tersebut. Ini berarti bahwa performa terhadap kelas negatif mungkin tidak tergambarkan dengan baik.

Untuk mengatasi kekurangan ini, dapat menggunakan versi *weighted* dari *F1 Score*. *Weighted F1 Score* mempertimbangkan seluruh kelas yang ada dalam dataset serta distribusinya. Dengan cara ini, dapat diperoleh gambaran performa yang lebih menyeluruh, yang mencakup baik kelas positif maupun negatif, serta memberikan penilaian yang lebih adil terhadap performa keseluruhan dari sistem klasifikasi. Pendekatan ini memastikan bahwa hasil evaluasi lebih representatif dan dapat memberikan wawasan yang lebih mendalam tentang kemampuan sistem dalam mengklasifikasikan berbagai kelas dengan baik. Perhitungan nilai *accuracy*, *recall*, dan *precision* yang diklasifikasikan menggunakan persamaan yang terdapat pada Persamaan (2),(3), (4), dan (5)[5][17].

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad \dots (2)$$

$$Precision = \frac{TP}{TP+FP} \quad \dots (3)$$

$$Recall = \frac{TP}{TP+FN} \quad \dots (4)$$

$$F1\ Score = 2 * \frac{precision*recall}{precision+recall} \quad \dots (5)$$

Keterangan :

TP	= True Positive
TN	= True Negative
FP	= False Positive
FN	= False Negative

### I. Visualisasi

Pada langkah ini, *confusion matrix* akan divisualisasikan TP (*True Positive*), TN (*True Negative*), FP (*False Positive*), dan FN (*False Negative*). *True Positives* (TP) merupakan jumlah kasus di mana model prediksi secara akurat mengidentifikasi data positif. Ini berarti bahwa data yang seharusnya diklasifikasikan sebagai positif memang benar diperkirakan sebagai positif dari hasil perkiraan model. Di sisi lain, *True Negatives* (TN) mengacu pada jumlah kasus di mana model prediksi secara akurat mengidentifikasi data negatif. Ini berarti bahwa data yang seharusnya diklasifikasikan sebagai negatif memang benar-benar diprediksi sebagai negatif oleh model. Selanjutnya, *False Positives* (FP) merujuk pada jumlah kasus di mana model prediksi salah mengidentifikasi data negatif sebagai positif. Dengan kata lain, ini adalah situasi di mana data yang sebenarnya tidak relevan atau tidak termasuk dalam kategori positif justru diklasifikasikan sebagai positif oleh model. Hal ini dapat menyebabkan kesalahan dalam hasil

prediksi, yang sering disebut sebagai tipe kesalahan tipe I.

Sebaliknya, *False Negatives* (FN) mengacu pada jumlah kasus di mana model prediksi salah mengidentifikasi data positif sebagai negatif. Ini berarti bahwa data yang seharusnya diklasifikasikan sebagai positif tidak terdeteksi dan malah diklasifikasikan sebagai negatif oleh model. Kesalahan ini sering disebut sebagai tipe kesalahan tipe II, yang juga dapat mempengaruhi keandalan dan akurasi keseluruhan dari model prediksi. Dengan demikian, memahami dan menghitung metrik ini sangat penting untuk mengevaluasi kinerja sebuah model pembelajaran mesin, karena mereka memberikan gambaran yang jelas tentang seberapa baik model tersebut dalam mengklasifikasikan data dengan benar dan menghindari kesalahan prediksi. Untuk cara penentuan TP, TN, FP, dan FN nya dapat dilihat pada **Gambar 4**[18].

		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

Gambar 4. Tabel confusion matrix

### III. HASIL DAN PEMBAHASAN

Untuk melakukan analisa sentimen dengan menggunakan SMOTE menggunakan metode k-NN, maka perlu ada tahapan yang dilakukan. Tahapan pertama yang dilakukan adalah pengumpulan data, *cleansing*, dan *casefolding* dari komentar *postingan* instagram terkait komentar masyarakat indonesia mengenai timnas indonesia di aplikasi instagram. Untuk pengumpulan data digunakan *apify* dan diperoleh hasil sebanyak 1280 komentar yang dapat dilihat pada **Tabel I**.

Tabel I. Data awal	
No	Text
0	@edubisa.id simple aja,, terlalu OP dia buat partai final nanti
1	Menyala terusss bangg 
2	Indonesia mendunia    
...	...
1279	Cie?in aku sama raffael pliss

Hasil *dataset* yang didapatkan pada **Tabel I**, dihasilkan melalui pencarian pada instagram terkait 'Timnas Indonesia', 'Timnas Indonesia U23', 'AFC U23'. Setelah proses *scraping* berhasil, data akan disimpan ke dalam format csv. Lalu, akan dilakukan proses *cleansing*. Setelah proses *cleansing* akan dilanjutkan dengan *casefolding* agar setiap kalimat menjadi huruf kecil. Berikut merupakan hasil setelah *cleansing* dan *casefolding* yang berada pada **Tabel II**. Kemudian, untuk tahap kedua adalah tahap *preprocessing* yang terdiri dari normalisasi, *stopword*, *tokenize* dan *stemming*. Tahap-tahap tersebut berada pada **Tabel III**. Setelah dilakukan proses *preprocessing* tersisa 1122 komentar pada **Tabel IV**.

Tabel II. Data Cleansing + CaseFolding	
No	Text
0	simple aja terlalu op dia buat partai final nanti
1	menyala terusss bangg
2	indonesia mendunia
...	...
1279	ciein aku sama raffael pliss

Tabel III. Tahapan Preprocessing	
Preprocessing	Hasil
Normalisasi	simple aja terlalu overpower dia buat partai final nanti
<i>Stopword</i>	['simple', 'aja', 'terlalu', 'overpower', 'dia', 'buat', 'partai', 'final', 'nanti']
<i>Tokenize</i>	[simple, aja, terlalu, overpower, dia, buat, partai, final, nanti]
<i>Stemming</i>	'simple', 'aja', 'terlalu', 'overpower', 'dia', 'buat', 'partai', 'final', 'nanti'
	['simple', 'aja', 'terlalu', 'overpower', 'dia', 'buat', 'partai', 'final', 'nanti']

Tabel IV. Data Hasil Preprocessing	
No	Text
0	simple aja terlalu overpower dia buat partai final nanti
1	nyala terusss bangg
2	indonesia dunia
...	...
1121	ciein aku sama raffael pliss

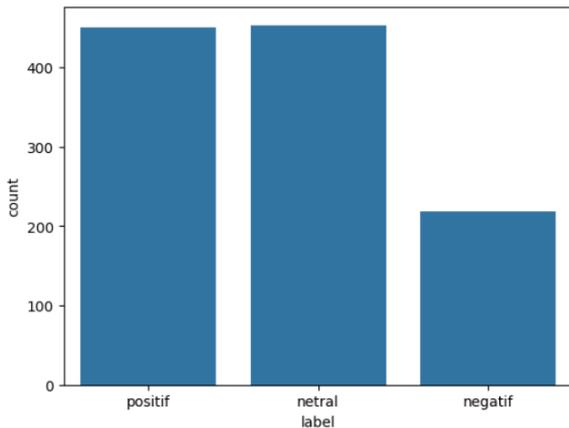
Setelah *preprocessing*, akan dilakukan labelisasi dengan tiga kategori penilaian yakni, kategori positif, negatif dan netral. Yang dapat dilihat pada **Tabel V**.

**Tabel V.** Labelisasi Data

No	Text	Label
0	simple aja terlalu overpower dia buat partai final	positif
1	nyala teruss bangg	positif
2	indonesia dunia	positif
...	...	...
1121	ciein aku sama raffael pliss	netral

Berdasarkan labelisasi yang dilakukan, berhasil dikumpulkan data yang terdiri dari 1122 komentar dengan jumlah komentar positif sebanyak 450 positif, 219 negatif dan 453 netral yang terlihat pada **Gambar 5**.

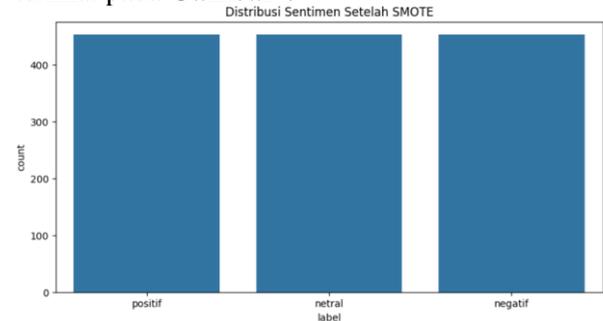
<Axes: xlabel='label', ylabel='count'>



**Gambar 5.** Perbandingan Sentimen tanpa SMOTE

Setelah itu dilanjutkan dengan pembobotan data, lalu diimplementasikan k-NN. Dalam pengimplementasian k-NN, akan dilakukan *split data* dengan data pelatihan sebanyak 80% dan data uji sebanyak 20%. Pengujian dilakukan dengan 2 skenario. Skenario pertama yaitu menggunakan K-fold cross validation dengan iterasi 10 tanpa menggunakan SMOTE dimulai dari nilai k=3, k=5, k=7, dan k=10. Lalu, skenario kedua menggunakan K-fold cross validation dengan iterasi 10 dengan SMOTE dimulai dari nilai k=3, k=5, k=7, dan k=10. Memilih nilai k hingga 10 dalam analisis ini bertujuan untuk mendapatkan pandangan yang lengkap tentang bagaimana model *K-Nearest Neighbors* (k-NN) bekerja dengan berbagai jumlah tetangga. Menggunakan nilai k yang berbeda memungkinkan saya melihat performa model saat dipengaruhi oleh

jumlah tetangga yang dipertimbangkan. Dengan menjalankan *K-fold cross validation* sebanyak 10 kali, dapat diuji konsisten dan stabilitas model. Memilih nilai k yang beragam ini juga membantu membandingkan hasil dengan studi lain yang menggunakan pendekatan serupa, sehingga hasil menjadi lebih valid dan kontekstual. Ketika menggunakan SMOTE hal ini akan berpengaruh terhadap jumlah data yang telah dilabelisasi dan terlihat pada **Gambar 6**.



**Gambar 6.** Perbandingan Sentimen dengan SMOTE

Pada **Tabel VI** akan menjadi skenario pertama dan **Tabel VII** akan menjadi skenario kedua.

**Tabel VI.** Skenario Pertama

K-Fold	Akurasi	Presisi	Recall	F1-Score
3	47%	45%	47%	39%
5	40%	52%	40%	29%
7	42%	59%	42%	33%
10	45%	46%	45%	32%

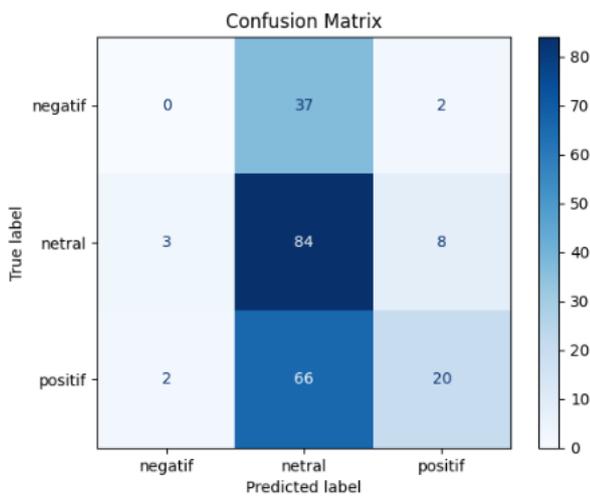
**Tabel VII.** Skenario Kedua

K-Fold	Akurasi	Presisi	Recall	F1-Score
3	55%	68%	55%	52%
5	63%	72%	63%	59%
7	51%	63%	51%	49%
10	50%	73%	50%	45%

Setelah mendapatkan hasil pada **Tabel VI** dan **Tabel VII**, Tahap berikutnya merupakan hasil visualisasi *confusion matrix* berdasarkan kedua skenario tersebut untuk setiap nilai k. Untuk k =3 dapat dilihat pada **Gambar 7** dan **Gambar 8**.

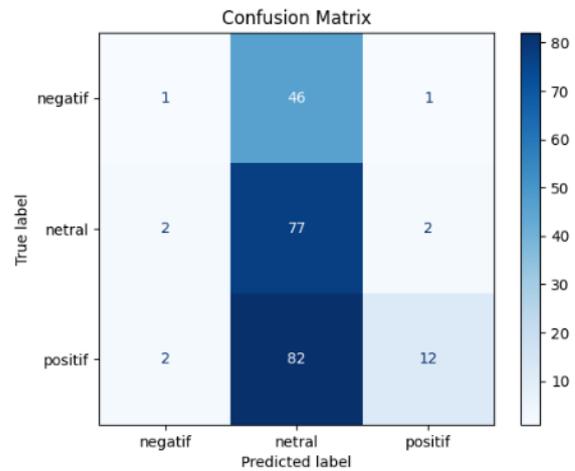
Untuk *confusion matrix* pada **Gambar 7** ini mengungkapkan bahwa model memiliki kesulitan signifikan dalam mengklasifikasikan komentar negatif dengan benar, karena tidak ada komentar negatif yang diklasifikasikan dengan benar (0). Sebanyak 37 komentar negatif salah dikira netral, dan 2 diklasifikasikan sebagai positif. Untuk komentar netral, model cukup akurat dengan 84 klasifikasi benar, namun ada 3 komentar netral

yang salah dikira sebagai negatif dan 8 sebagai positif. Pada komentar positif, model mengklasifikasikan 20 dengan benar, namun 66 salah dikira netral dan 2 sebagai negatif. Secara keseluruhan, confusion matrix ini menunjukkan bahwa meskipun ada beberapa peningkatan dalam klasifikasi komentar positif, model masih cenderung mengklasifikasikan banyak komentar ekstrem (positif dan negatif) sebagai netral, menunjukkan bahwa model memerlukan penyesuaian lebih lanjut untuk meningkatkan akurasi dalam mengidentifikasi sentimen.

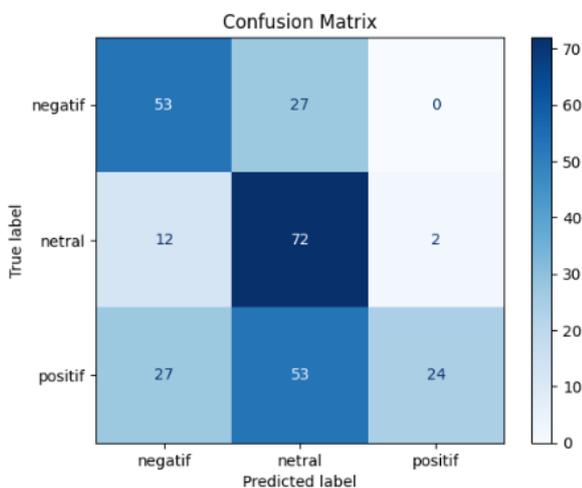


Gambar 7. Visualisasi Skenario Pertama untuk k = 3

positif. Pada kategori komentar positif, model benar mengklasifikasikan 24 komentar, namun 53 komentar positif salah dikategorikan sebagai netral dan 27 sebagai negatif. Dari data ini, dapat disimpulkan bahwa meskipun model dengan SMOTE lebih baik dalam menangani komentar negatif dan positif dibandingkan dengan model tanpa SMOTE, namun masih ada tantangan dalam mengurangi kesalahan klasifikasi terutama dalam membedakan antara komentar positif dan netral. Untuk k = 5 dapat dilihat pada Gambar 9 dan Gambar 10.



Gambar 9. Visualisasi Skenario Pertama untuk k = 5



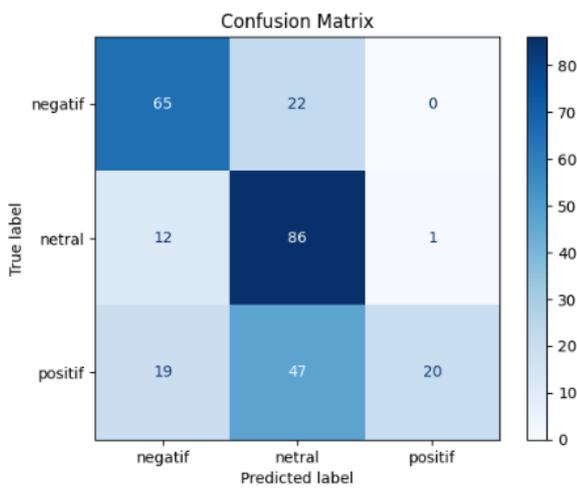
Gambar 8. Visualisasi Skenario Kedua untuk k = 3

Dari confusion matrix pada Gambar 8, dapat dilihat bahwa model berhasil mengklasifikasikan 53 komentar negatif dengan benar, namun masih terdapat 27 komentar negatif yang salah diklasifikasikan sebagai netral, dan tidak ada yang diklasifikasikan sebagai positif. Untuk komentar netral, model mengidentifikasi 72 komentar dengan tepat, tetapi 12 komentar netral salah diklasifikasikan sebagai negatif dan 2 sebagai

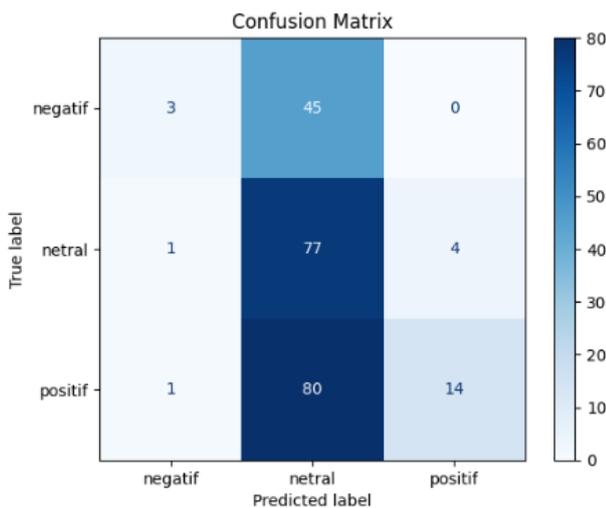
Untuk confusion matrix pada Gambar 9 memperlihatkan bagaimana model klasifikasi sentimen bekerja dalam mengkategorikan komentar Instagram tentang Timnas Indonesia U-23 selama Piala Asia AFC 2024 ketika melakukan skenario pertama. Dari visualisasi tersebut, terlihat bahwa model ini mampu mengidentifikasi 1 komentar negatif dengan benar, tetapi salah mengira 46 komentar negatif sebagai netral. Untuk komentar netral, model berhasil mengenali 77 komentar dengan tepat, namun 2 komentar netral salah dikira sebagai negatif dan 2 sebagai positif. Sedangkan pada komentar positif, model hanya benar pada 12 komentar, sementara 82 komentar positif salah diklasifikasikan sebagai netral dan 2 sebagai negatif. Ini menunjukkan bahwa model ini cenderung mengklasifikasikan banyak komentar negatif dan positif sebagai netral, menunjukkan kesulitan dalam membedakan sentimen yang lebih ekstrem, sehingga perlu peningkatan untuk meningkatkan akurasi klasifikasinya.

Untuk confusion matrix pada Gambar 10 memperlihatkan bagaimana model klasifikasi sentimen bekerja dalam mengkategorikan komentar Instagram tentang Timnas Indonesia U-23 selama Piala Asia AFC 2024 ketika melakukan

skenario kedua. Dari visualisasi tersebut, terlihat bahwa model ini mampu mengidentifikasi 65 komentar negatif dengan benar, tetapi salah mengira 22 komentar negatif sebagai netral. Untuk komentar netral, model berhasil mengenali 86 komentar dengan tepat, namun 12 komentar netral salah dikira sebagai negatif dan 1 sebagai positif. Sedangkan pada komentar positif, model hanya benar pada 20 komentar, sementara 47 komentar positif salah diklasifikasikan sebagai netral dan 19 sebagai negatif. Ini menunjukkan bahwa model mengalami kesulitan dalam membedakan komentar positif dari netral, yang berarti perlu adanya peningkatan untuk lebih akurat dalam klasifikasi sentimen ini.



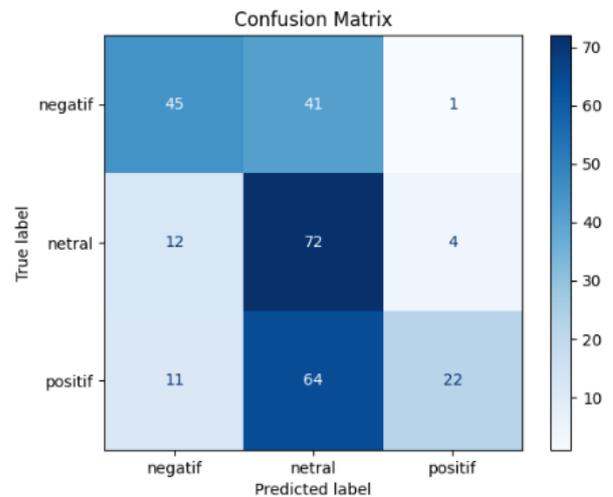
Gambar 10. Visualisasi Skenario Kedua untuk k = 5



Gambar 11. Visualisasi Skenario Pertama untuk k = 7

Confusion matrix pada Gambar 11 ini menunjukkan performa dari sebuah model klasifikasi dengan k=7 pada tiga kelas: negatif, netral, dan positif. Dari 49 sampel sebenarnya negatif, hanya 3 yang terklasifikasi dengan benar sementara 45 salah diklasifikasikan sebagai netral,

dan tidak ada yang salah diklasifikasikan sebagai positif. Untuk 82 sampel netral, 77 diklasifikasi dengan benar, 1 salah diklasifikasikan sebagai negatif, dan 4 salah diklasifikasikan sebagai positif. Dari 95 sampel positif, 14 diklasifikasi dengan benar, namun 80 salah diklasifikasikan sebagai netral, dan 1 salah diklasifikasikan sebagai negatif. Hasil ini menunjukkan bahwa model ini cenderung mengklasifikasikan banyak sampel sebagai netral, terutama yang sebenarnya negatif dan positif, yang menandakan bias yang kuat terhadap kelas netral.

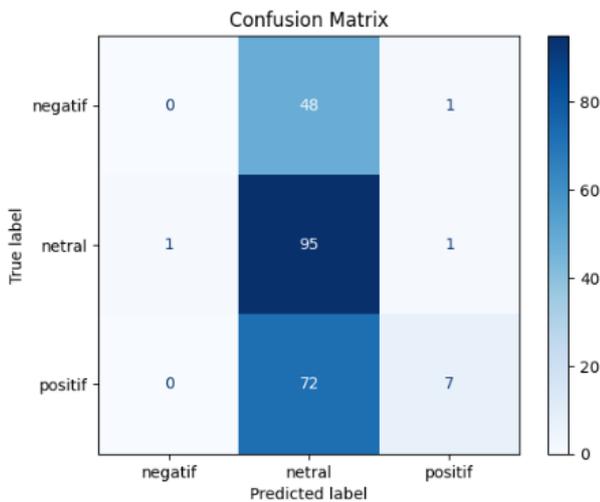


Gambar 12. Visualisasi Skenario Kedua untuk k = 7

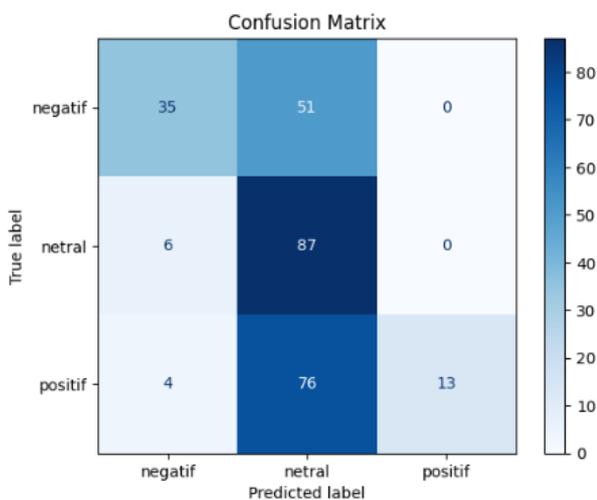
Untuk Gambar 12 confusion matrix ini menunjukkan performa dari model klasifikasi dengan k=7 setelah menerapkan SMOTE untuk menangani ketidakseimbangan data pada tiga kelas: negatif, netral, dan positif. Dari 87 sampel sebenarnya negatif, 45 diklasifikasi dengan benar, 41 salah diklasifikasikan sebagai netral, dan 1 salah diklasifikasikan sebagai positif. Untuk 88 sampel netral, 72 diklasifikasi dengan benar, 12 salah diklasifikasikan sebagai negatif, dan 4 salah diklasifikasikan sebagai positif. Dari 97 sampel positif, 22 diklasifikasi dengan benar, namun 64 salah diklasifikasikan sebagai netral, dan 11 salah diklasifikasikan sebagai negatif. Meskipun model ini menunjukkan peningkatan dalam mendeteksi kelas negatif dan positif dibandingkan dengan sebelumnya tanpa SMOTE, namun untuk nilai k ini masih terdapat bias yang signifikan terhadap kelas netral, dengan banyak sampel positif yang salah diklasifikasikan sebagai netral.

Dari Gambar 13 dapat dilihat dari 49 sampel negatif, tidak ada yang diklasifikasi dengan benar. Sebagian besar, 48 sampel, malah diklasifikasi sebagai netral, dan 1 sampel sebagai positif. Untuk 97 sampel netral, 95 diklasifikasi dengan benar, hanya 1 yang salah diklasifikasi sebagai negatif,

dan 1 lagi sebagai positif. Dari 79 sampel positif, hanya 7 yang benar diklasifikasi, sementara 72 salah diklasifikasi sebagai netral. Ini menunjukkan bahwa model ini cenderung menganggap banyak sampel sebagai netral, mengakibatkan kesalahan klasifikasi signifikan, terutama untuk kelas negatif dan positif. Terlihat jelas bahwa model ini belum mampu mengatasi ketidakseimbangan antar kelas, sehingga hasil klasifikasinya masih belum optimal.



Gambar 13. Visualisasi Skenario Kedua untuk k = 10



Gambar 14. Visualisasi Skenario Kedua untuk k = 10

Untuk Gambar 14 terlihat bahwa model mampu memprediksi kelas "negatif" dengan benar sebanyak 35 kali, tetapi sering salah memprediksi sebagai "netral" sebanyak 51 kali. Untuk kelas "netral", prediksi benar terjadi 87 kali, dengan kesalahan prediksi sebagai "negatif" sebanyak 6 kali. Pada kelas "positif", model memprediksi dengan benar 13 kali, tetapi sering salah memprediksi sebagai "netral" sebanyak 76 kali dan sebagai "negatif" sebanyak 4 kali. Hal ini menunjukkan bahwa model memiliki kesulitan

terbesar dalam membedakan antara kelas "netral" dan "positif".

Melalui tabel pada Tabel VI dan Tabel VII diperoleh pada pengujian pertama tanpa SMOTE hanya mendapatkan tingkat akurasi sebesar 50% untuk k=3, ketika k = 5 akurasi 40%, k = 7 akurasi 42%, dan k = 10 akurasi 45%. Sedangkan, Pada pengujian kedua dengan menggunakan SMOTE dihasilkan akurasi sebesar 55% untuk k = 3, lalu k = 5 akurasi 63%, k = 7 akurasi 51% dan k = 10 akurasi 50%.

Performa terbaik dari algoritma k-NN diperoleh saat nilai k=3, dengan nilai akurasi, presisi, recall, dan F1 Score tertinggi dibandingkan nilai k lainnya, yaitu 50% untuk akurasi, 0.51 untuk presisi, 0.5 untuk recall, dan 0.44 untuk F1 Score. Sedangkan, performa terbaik dari kombinasi k-NN dengan SMOTE dicapai saat nilai k=5, dengan nilai akurasi, recall, dan F1 Score tertinggi dibandingkan nilai k lainnya, yaitu 63% untuk akurasi, 0.72 untuk presisi, 0.63 untuk recall, dan 0.59 untuk F1 Score. Hal ini dapat terjadi dikarenakan sensitivitas terhadap variasi dalam data. Jika k terlalu kecil (misalnya k = 3), model dapat menjadi terlalu sensitif terhadap noise atau variabilitas kecil dalam data latih, yang dapat menghasilkan overfitting. Di sisi lain, jika k terlalu besar (misalnya k = 7 atau k = 10), model dapat menjadi terlalu umum atau cenderung underfitting, di mana informasi dari tetangga terlalu banyak atau terlalu sedikit digunakan.

Hasil yang diperoleh dalam penelitian ini memiliki kesamaan dengan beberapa penelitian terdahulu, dimana hasil ini menunjukkan pentingnya teknik balancing data seperti SMOTE dalam meningkatkan performa model klasifikasi. Meskipun hasil penelitian ini menunjukkan akurasi yang lebih rendah dibandingkan penelitian terdahulu lainnya. Dimana penelitian terdahulu yang membandingkan k-NN dengan dan tanpa SMOTE melaporkan hasil yang lebih tinggi, dengan akurasi 82% tanpa SMOTE dan 90% dengan SMOTE[9]. Perbedaan hasil ini mungkin disebabkan oleh variasi dalam dataset, penyesuaian parameter, atau metode evaluasi yang digunakan. Selain itu, ada juga penelitian yang menunjukkan bahwa metode k-NN dengan SMOTE memiliki akurasi lebih unggul dibandingkan metode Naive Bayes dengan SMOTE, dengan akurasi 88.12% untuk k-NN dan 82.27% untuk Naive Bayes[5].

Berikutnya akan dilakukan visualisasi data menggunakan word cloud untuk memudahkan ilustrasi kata dengan memberikan visualisasi teks yang menarik dan informatif[19]. Word Cloud

untuk data Timnas Indonesia dapat berada pada Gambar 9.



Gambar 9. Word Cloud untuk Sentimen Analisis Timnas Indonesia U-23

Pada Gambar 9 menunjukkan bahwa komentar positif masyarakat instagram sangat mendukung timnas Indonesia dalam piala AFC 2024 yang isinya adalah “otomatis piala dunia”; “lolos olimpiade”; “luar biasa”, “indonesia menang”, bahkan ketika timnas kalah ada komentar “tetap semangat”; “wasitnya” (komentar wasit ini menggambarkan masyarakat Indonesia yang melihat bahwa wasitnya seperti curang dan merugikan timnas Indonesia).

#### IV. KESIMPULAN

Berdasarkan hasil pengujian dan penelitian yang dilakukan terkait analisis sentimen komentar terhadap Timnas Indonesia U23 selama periode piala AFC yang dilakukan melalui dua skenario yaitu menggunakan SMOTE dan tidak menggunakan SMOTE. Melalui kedua skenario tersebut didapatkan sebuah kesimpulan yakni jika data yang tidak seimbang dapat mempengaruhi akurasi yang didapatkan. Hal ini dapat dibuktikan melalui serangkaian pengujian dengan menggunakan *K-fold cross validation* agar mendapatkan *best validation* yang lebih baik. Hal ini membuktikan bahwa klasifikasi menggunakan *K-Nearest Neighbor* (k-NN) dengan SMOTE memiliki akurasi lebih baik dibandingkan tanpa menggunakan SMOTE.

Berdasarkan penelitian ini, diharapkan pada penelitian terkait timnas berikutnya melalui komentar posting instagram dapat menggunakan metode *machine learning* lainnya agar dapat mencapai nilai akurasi yang lebih optimal. Analisis sentimen ini juga dapat menyediakan informasi mengenai sentimen masyarakat terhadap Timnas Indonesia, yang dapat digunakan sebagai umpan

balik untuk terus memberikan masukan terbaik bagi timnas.

#### DAFTAR PUSTAKA

- [1] R. Prajamukti, Jayanta, dan Mayanda Mega Santoni, “Klasifikasi dan Analisis Sentimen pada Data Twitter menggunakan Algoritma Naïve Bayes (Studi Kasus: Timnas Indonesia Senior, U-23, dan U-19),” *Seminar Nasional Informatika, Sistem Informasi, dan Keamanan Siber*, vol. 4, no. 1, hlm. 102–109, 2021.
- [2] Atradinal dan Rika Sepriani, “Pemulihan Kekuatan Otot pada Atlet Sepakbola,” *Jurnal Menssana*, vol. 2, no. 2, hlm. 99–105, 2017.
- [3] M. F. Alfajri, Viranda Adhiazni, dan Qurrotul Aini, “Pemanfaatan Social Media Analytics pada Instagram dalam Peningkatan Efektivitas Pemasaran,” *Interaksi: Jurnal Ilmu Komunikasi*, vol. 8, no. 2, hlm. 1–11, 2019.
- [4] F. S. Jumeilah, “Penerapan Support Vector Machine (SVM) untuk Pengkategorian Penelitian,” *JURNAL RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 1, no. 1, hlm. 19–25, 2017.
- [5] S. D. Prasetyo, Shofa Shofiah Hilabi, dan Fitri Nurapriani, “Analisis Sentimen Relokasi Ibukota Nusantara Menggunakan Algoritma Naïve Bayes dan KNN,” *Jurnal KomtekInfo*, vol. 10, no. 1, hlm. 1–7, 2023, doi: 10.35134/komtekinfo.v10i1.330.
- [6] A. P. Monika, Felisia Elvira Paska Risti, Iwan Binanto, dan Nesti F. Sianipar, “Perbandingan Algoritma Klasifikasi Random Forest, Gaussian Naive Bayes, dan KNearest Neighbor untuk Data Tidak Seimbang dan Data yang diseimbangkan dengan Metode Adaptive Synthetic pada Dataset LCMS Tanaman Keladi Tikus,” *Prosiding Seminar Nasional Teknik Elektro, Informatika & Sistem Informasi (SINTAKS)*, vol. 2, no. 1, hlm. 1–5, 2023.
- [7] A. Tangkelayuk dan Evangs Mailoa, “Klasifikasi Kualitas Air Menggunakan Metode KNN, Naive Bayes Dan Decision Tree,” *JATISI (Jurnal Teknik Informatika dan Sistem Informasi)*, vol. 9, no. 2, hlm. 1109–1119, 2022.
- [8] N. R. Wardani, Sudin Saepudin, dan Cecep Warman, “Sentimen Analisis Kegiatan Trading Pada Aplikasi Twitter dengan Algoritma SVM, KNN Dan Random Forrest,” *J-SAKTI (Jurnal Sains Komputer & Informatika)*, vol. 6, no. 2, hlm. 863–870, 2022.
- [9] A. S. Firmansyah, Abdul Aziz, dan Moh. Ahsan, “Optimasi K-Nearest Neighbor menggunakan Algoritma Smote untuk Mengatasi Imbalance Class pada Klasifikasi Analisis Sentimen,” *Jurnal Mahasiswa Teknik Informatika*, vol. 7, no. 6, hlm. 3341–3347, 2023, doi: 10.36040/jati.v7i6.7257.
- [10] F. D. Astuti dan Febri Nova Lenti, “Implementasi SMOTE untuk mengatasi Imbalance Class pada Klasifikasi Car Evolution menggunakan K-NN,” *JUPITER*, vol. 13, no. 1, hlm. 89–98, 2021.
- [11] A. N. Kasanah, Muladi, dan Utomo Pujianto, “Penerapan Teknik SMOTE untuk Mengatasi Imbalance Class dalam Klasifikasi Objektivitas Berita Online Menggunakan Algoritma KNN,” *JURNAL RESTI(Rekayasa Sistem dan Teknologi Informasi)*, vol. 3, no. 2, hlm. 196–201, 2018.
- [12] J. Muliawan dan Erick Dazki, “Sentiment Analysis of Indonesia’s Capital City Relocation using Three Algorithms: Naïve Bayes, KNN, and Random Forest,” *Jurnal Teknik Informatika (JUTIF)*, vol. 4, no. 5, hlm. 1227–1236, 2023.
- [13] D. Pramana, M Afdal, Mustakim, dan Inggih Permana, “Analisis Sentimen Terhadap Pemindahan Ibu Kota Negara Menggunakan Algoritma Naive Bayes Classifier dan K-Nearest Neighbors,” *JURNAL MEDIA INFORMATIKA BUDIDARMA*, vol. 7, no. 3, hlm. 1306–1314, 2023.
- [14] R. Ferdiana, Fahim Jatmiko, Desi Dwi Purwanti, Artmita Sekar Tri Ayu, dan Wiliam Fajar Dicka, “Dataset Indonesia untuk Analisis Sentimen,” *Jurnal Nasional Teknik Elektro Dan Teknologi Informasi*, vol. 8, no. 4, hlm. 334–339, 2019.
- [15] O. S. Y. Prakasa dan Kemas Muslim Lhaksamana, “Klasifikasi Teks dengan menggunakan Algoritma K-Nearest Neighbor pada Kasus Kinerja Pemerintah di Twitter,” *eProceedings of Engineering*, vol. 5, no. 3, hlm. 8237–8248, 2018.

- [16] V. Saini, "SMOTE." Diakses: 23 Mei 2024. [Daring]. Tersedia pada: <https://varshasaini.in/glossary/smote/>
- [17] H. Nuraliza, Oktariani Nurul Pratiwi, dan Faqih Hamami, "Analisis Sentimen IMBd Film Review Dataset Menggunakan Support Vector Machine (SVM) dan Seleksi Feature Importance," *Jurnal Mirai Manajemen*, vol. 7, no. 1, hlm. 1–17, 2022.
- [18] D. Atmajaya, Annisa Febrianti, dan Herdianti Darwis, "Metode SVM dan Naive Bayes untuk Analisis Sentimen ChatGPT di Twitter," *Indonesian Journal of Computer Science*, vol. 12, no. 4, hlm. 2173–2181, 2023.
- [19] M. R. Pribadi, Hindriyanto Dwi Purnomo, Hendry, Kristoko Dwi Hartomo, Irwan Sembiring, dan Ade Iriani, "Improving the Accuracy of Text Classification Using the Over Sampling Technique in the Case of Sinovac Vaccine," *2022 9th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI2022)*, hlm. 106–110, 2022.