PENINGKATAN KINERJA SVM PADA KLASIFIKASI SENTIMEN ULASAN IPUSNAS BERBASIS IMBALANCE HANDLING

Akmal Mustafa[1], Rudi Kurniawan[2], Bani Nurhakim[3], Puji Pramudya Marta[4], Khaerul Anam[5]

[1,2,5] Program Studi Teknik Informatika,STMIK IKMI Cirebon, Jawa Barat, Indonesia
[3]Program Studi Manajemen Informatika, STMIK IKMI Cirebon , Jawa Barat, Indonesia
[4]Program Studi Sistem Informasi, STMIK IKMI Cirebon , Jawa Barat, Indonesia

e-mail: [1]akmalmustafa.sch@gmail.com, [2]rudi226ikmi@gmail.com,
[3]baninurhakim@gmail.com, [4]prammarta88@gmail.com, [5]jodiust9@gmail.com

ABSTRAK

Ulasan pengguna aplikasi iPusnas merupakan sumber informasi penting untuk memahami persepsi pengguna terhadap layanan perpustakaan digital. Namun, distribusi sentimen pada data ulasan cenderung tidak seimbang sehingga model klasifikasi berisiko bias terhadap kelas mayoritas. Penelitian ini bertujuan mengoptimalkan klasifikasi sentimen ulasan aplikasi iPusnas menggunakan algoritma Linear Support Vector Machine dengan menerapkan teknik class imbalance handling. Data ulasan diberi label sentimen ke dalam tiga kelas, yaitu negatif, netral, dan positif. Tahapan penelitian meliputi preprocessing teks, ekstraksi fitur menggunakan TF-IDF, penerapan SMOTE, RandomUnderSampler, dan SMOTE-Tomek, pemodelan Linear SVM, serta evaluasi menggunakan accuracy, precision macro, recall macro, F1-score macro, cross validation, dan uji signifikansi statistik. Hasil penelitian menunjukkan bahwa model baseline memperoleh accuracy sebesar 0,851 dengan F1-score macro 0,593. Penerapan SMOTE dan SMOTE-Tomek menghasilkan performa terbaik dengan accuracy dan F1-score macro masing-masing sebesar 0,970, serta rata-rata cross validation 0,9730. Uji statistik menunjukkan seluruh teknik penanganan class imbalance berbeda signifikan dibandingkan baseline. Dengan demikian, SMOTE dan SMOTE-Tomek terbukti paling efektif dalam meningkatkan performa dan stabilitas klasifikasi sentimen ulasan iPusnas.

Kata kunci: klasifikasi sentimen; *class imbalance*; support vector machine; iPusnas; aplikasi literasi digital.

ABSTRACT

This study investigates sentiment classification of user reviews on the iPusnas digital library application to provide an objective overview of service quality and user experience. Numerous complaints related to login failures, application crashes, and issues in accessing digital books indicate the need for a computational approach capable of processing large volumes of user feedback. The proposed method integrates Natural Language Processing (NLP) techniques with the Support Vector Machine (SVM) algorithm. The workflow consists of collecting 2,000 reviews, applying text cleaning and normalization, tokenization, stopword removal, stemming, rating-based sentiment annotation, and feature extraction using TF-IDF. The dataset was divided using a train–test split for model training and evaluation. Experimental results show that the SVM model achieves 90.1% accuracy, demonstrating strong performance in detecting negative sentiments and moderate performance for positive sentiments due to class imbalance. These

findings highlight the effectiveness of NLP and SVM for extracting user perceptions and indicate the potential of this model as a decision-support tool for improving iPusnas application services. Overall, the study contributes to the advancement of digital service innovation in Indonesia.

Keywords*: Sentiment Analysis, Natural Language Processing, Support Vector Machine, iPusnas, Digital Literacy*

1. PENDAHULUAN

Transformasi layanan perpustakaan digital mendorong peningkatan penggunaan aplikasi berbasis mobile sebagai sarana akses literatur, salah satunya iPusnas sebagai aplikasi Perpustakaan Digital Nasional. Ulasan pengguna pada platform aplikasi menjadi sumber data penting karena memuat pengalaman, keluhan, kepuasan, serta persepsi pengguna terhadap kualitas layanan. Informasi tersebut dapat dianalisis menggunakan pendekatan analisis sentimen untuk mengelompokkan opini ke dalam kelas positif, netral, dan negatif. Dalam konteks pengembangan aplikasi, klasifikasi sentimen dapat membantu pengelola memahami pola respons pengguna secara lebih terukur sehingga perbaikan layanan tidak hanya didasarkan pada penilaian umum, tetapi juga pada bukti tekstual yang terstruktur.

Analisis sentimen berbasis teks ulasan memiliki tantangan karena data yang dihasilkan pengguna umumnya bersifat tidak baku, mengandung variasi bahasa, simbol, angka, kesalahan penulisan, serta kata-kata informal. Oleh karena itu, tahap *preprocessing* seperti *cleaning*, *case folding*, normalisasi, tokenisasi, *stopword removal*, dan *stemming* diperlukan untuk meningkatkan kualitas fitur sebelum proses pemodelan [1], [2]. Representasi teks menggunakan TF-IDF juga banyak digunakan karena mampu mengubah teks menjadi bobot numerik berdasarkan kepentingan *term*, sedangkan penggunaan *n-gram* dapat menangkap konteks kata yang lebih informatif dibandingkan *unigram* tunggal [3]. Pada data berdimensi tinggi seperti teks, Support Vector Machine (SVM) relevan digunakan karena memiliki kemampuan membentuk *hyperplane* pemisah yang efektif pada ruang fitur *sparse* dan linear.

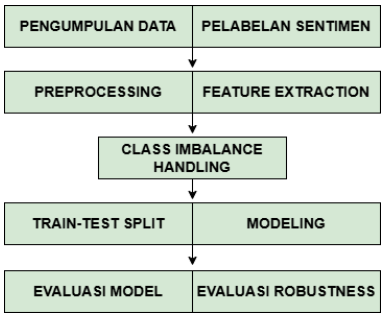
Meskipun demikian, permasalahan utama dalam klasifikasi sentimen ulasan aplikasi adalah ketidakseimbangan distribusi kelas. Kondisi *class imbalance* menyebabkan model cenderung mempelajari kelas mayoritas secara dominan, sehingga performa pada kelas minoritas dapat menurun meskipun nilai akurasi terlihat tinggi. Penelitian sebelumnya menunjukkan bahwa teknik *rebalancing* seperti *oversampling*, *undersampling*, dan pendekatan *hybrid* dapat memengaruhi kinerja klasifikasi pada data tidak seimbang, tetapi efektivitasnya bergantung pada karakteristik dataset, distribusi kelas, dan algoritma yang digunakan [4], [5], [6], [7], [8]. Pada konteks sentimen berbahasa Indonesia, penerapan *preprocessing* dan SMOTE juga dilaporkan berpengaruh terhadap performa klasifikasi sentimen [9].

Kesenjangan penelitian yang masih perlu diperhatikan adalah kecenderungan sebagian studi untuk menilai performa model terutama berdasarkan akurasi, padahal akurasi kurang representatif pada data multikelas yang tidak seimbang. Evaluasi yang lebih tepat perlu melibatkan *F1-macro* karena metrik ini memberikan bobot yang seimbang terhadap setiap kelas, *term*asuk kelas minoritas [10], [11], [12]. Selain itu, konsistensi performa model perlu diuji melalui *cross-validation* dan uji signifikansi statistik agar peningkatan performa tidak hanya bersifat numerik pada satu skenario pembagian data[13], [14].

Berdasarkan permasalahan tersebut, penelitian ini bertujuan mengoptimalkan klasifikasi sentimen ulasan aplikasi iPusnas menggunakan algoritma Linear SVM dengan membandingkan model *baseline* terhadap model yang menerapkan teknik penanganan *class imbalance*, yaitu SMOTE, RandomUnderSampler, dan SMOTE-Tomek. Evaluasi dilakukan menggunakan *accuracy*, *precision macro*, *recall macro*, *F1-macro*, *cross-validation*, dan uji signifikansi statistik untuk memperoleh penilaian performa yang lebih komprehensif dan andal.

2. METODE PENELITIAN

Penelitian ini menggunakan pendekatan eksperimen kuantitatif untuk mengoptimalkan klasifikasi sentimen ulasan aplikasi iPusnas dengan algoritma Support Vector Machine (SVM) melalui penanganan *class imbalance*. Alur penelitian terdiri atas lima tahap utama, yaitu pengumpulan dan pelabelan data, *preprocessing* dan ekstraksi fitur, penanganan ketidakseimbangan kelas, pembagian data dan pemodelan, serta evaluasi model dan evaluasi *robustness*. Tahapan tersebut disusun agar proses eksperimen dapat direplikasi secara sistematis pada dataset ulasan pengguna aplikasi digital.



Gambar 1 Diagram Alur Metode Penelitian

Pengumpulan dan Pelabelan Data

Data penelitian berupa ulasan pengguna aplikasi iPusnas yang diperoleh dari Google Play Store dengan bantuan pustaka google_play_scraper. Pengambilan data dilakukan pada aplikasi dengan identitas paket mam.reader.ipusnas, jumlah maksimum 2000 ulasan, dan pengurutan berdasarkan Sort.MOST_RELEVANT. Data yang dikumpulkan kemudian diperiksa melalui tahap data understanding, meliputi identifikasi ukuran data, struktur kolom, tipe data, informasi nilai kosong, data duplikat, distribusi label, serta panjang ulasan. Pemeriksaan awal ini bertujuan memastikan bahwa data yang digunakan sesuai untuk eksperimen klasifikasi teks.

Pelabelan sentimen dilakukan secara manual melalui *human annotation* ke dalam tiga kelas, yaitu positif, netral, dan negatif. Kelas positif diberikan pada ulasan yang secara dominan menunjukkan kepuasan, apresiasi, atau pengalaman baik terhadap aplikasi. Kelas negatif diberikan pada ulasan yang menunjukkan keluhan, kekecewaan, atau kritik terhadap aplikasi. Kelas netral diberikan pada ulasan yang bersifat informatif atau memuat kombinasi sentimen positif dan negatif secara bersamaan. Setelah proses pembersihan data duplikat dan data kosong, jumlah data yang digunakan dalam pemodelan adalah 1942 ulasan, terdiri atas 1576 ulasan negatif, 274 ulasan netral, dan 92

ulasan positif. Distribusi ini menunjukkan adanya ketidakseimbangan kelas yang menjadi fokus utama penelitian.

Preprocessing dan Feature Extraction

Tahap *preprocessing* dilakukan untuk mengubah teks ulasan yang tidak terstruktur menjadi representasi yang lebih bersih dan konsisten. Proses ini mencakup *text cleaning*, *case folding*, normalisasi kata, tokenisasi, penghapusan *stopword*, *stemming*, penghapusan data duplikat, dan penghapusan data kosong. *Text cleaning* dilakukan dengan menghapus URL, tag HTML, emoji, simbol, dan angka. *Case folding* digunakan untuk mengubah seluruh huruf menjadi huruf kecil. Normalisasi dilakukan dengan kamus kata tidak baku dan kamus khusus agar variasi kata informal dapat dikonversi ke bentuk baku. Tokenisasi dilakukan dengan pemisahan kata berbasis spasi, sedangkan penghapusan *stopword* menggunakan daftar *stopword* bahasa Indonesia dari NLTK yang dikombinasikan dengan custom *stopword*. Proses *stemming* dilakukan menggunakan Sastrawi dengan beberapa pengecualian kata agar makna penting tidak hilang.

Untuk mengubah teks hasil *stemming* menjadi fitur numerik yang dapat diproses oleh algoritma SVM, penelitian ini menggunakan pembobotan Term Frequency-Inverse Document Frequency (TF-IDF). TF-IDF menghitung bobot suatu term berdasarkan frekuensi kemunculannya dalam dokumen dan tingkat keunikannya terhadap keseluruhan korpus. Semakin sering suatu term muncul dalam dokumen tertentu dan semakin jarang term tersebut muncul pada dokumen lain, maka semakin tinggi bobot informatifnya. Perhitungan TF-IDF dinyatakan pada rumus Persamaan (1) sampai Persamaan (3).

$$TF(t,d) = \log\left(f_{t,d}\right) \tag{1}$$

$$IDF(t) = \log\left(\frac{N}{df(t)}\right) \tag{2}$$

$$TF - IDF(t,d) = TF(t,d) \times IDF(t) \tag{3}$$

Parameter yang digunakan adalah ngram_range=(1,2), min_df=2, max_df=0.9, max_features=6000, dan sublinear_tf=True. Penggunaan *unigram* dan *bigram* bertujuan menangkap informasi kata tunggal serta pasangan kata yang relevan, sedangkan batas min_df dan max_df digunakan untuk mengurangi fitur yang terlalu jarang atau terlalu umum. Fitur hasil TF-IDF menjadi masukan numerik bagi model klasifikasi SVM.

Class Imbalance Handling

Penanganan *class imbalance* dilakukan dengan membandingkan empat skenario data. Skenario pertama adalah *baseline*, yaitu data asli tanpa penanganan ketidakseimbangan kelas. Skenario kedua menggunakan Synthetic Minority Oversampling Technique (SMOTE) untuk menambah sampel sintetis pada kelas minoritas. Skenario ketiga menggunakan RandomUnderSampler (RUS) untuk mengurangi jumlah sampel pada kelas mayoritas. Skenario keempat menggunakan SMOTE-Tomek sebagai teknik *hybrid* yang menggabungkan *oversampling* SMOTE dan pembersihan data Tomek Links untuk memperbaiki batas keputusan antarkelas. Seluruh teknik sampling menggunakan random_state=42 agar hasil eksperimen dapat direplikasi.

Train-Test Split dan Modeling

Setiap skenario data dibagi menjadi data latih dan data uji menggunakan rasio 70:30 dengan parameter test_size=0.3, random_state=42, dan stratify=y. Penggunaan stratifikasi bertujuan menjaga proporsi kelas pada data latih dan data uji sesuai distribusi masing-masing skenario. Model klasifikasi yang digunakan adalah Linear Support Vector Machine melalui implementasi LinearSVC. Linear SVM dipilih karena sesuai untuk data teks berdimensi tinggi dan jarang (*sparse*) yang dihasilkan dari representasi TF-IDF. Model dibangun pada empat skenario, yaitu SVM *baseline*, SVM dengan SMOTE, SVM dengan RUS, dan SVM dengan SMOTE-Tomek. Fungsi keputusan Linear SVM dinyatakan pada rumus Persamaan (4).

$$f(x) = w^T x + b \tag{4}$$

Evaluasi Model dan Evaluasi Robustness

Evaluasi model dilakukan menggunakan *accuracy*, *precision macro*, *recall macro,* dan *F1-score macro*. *F1-score macro* digunakan sebagai metrik utama karena dapat menilai performa tiap kelas secara seimbang tanpa didominasi oleh kelas mayoritas. Dengan demikian, evaluasi model menjadi lebih adil, terutama jika distribusi data antar kelas tidak seimbang. Rumus evaluasi model dinyatakan pada rumus Persamaan (5) sampai Persamaan (9).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{5}$$

$$Precision_i = \frac{TP_i}{TP_i + FP_i} \tag{6}$$

$$Recall_i = \frac{TP_i}{TP_i + FN_i} \tag{7}$$

$$F1_i = \frac{2 \times Precision_i \times Recall_i}{Precision_i + Recall_i} \tag{8}$$

$$F1\_Macro = \frac{1}{C}\sum_{i=1}^{C} F1_i \tag{9}$$

Evaluasi *robustness* dilakukan melalui 5-*fold* Stratified Cross Validation dengan metrik *F1-macro*. Rata-rata skor *cross validation* digunakan untuk melihat performa umum model pada seluruh *fold*, sebagaimana dinyatakan pada rumus Persamaan (10).

$$CV = \frac{1}{k}\sum_{i=1}^{k} S_i \tag{10}$$

Selanjutnya, uji signifikansi statistik dilakukan menggunakan *paired t-test* untuk membandingkan performa *baseline* dengan setiap skenario penanganan *class imbalance*. Nilai *p-value* di bawah 0,05 digunakan sebagai dasar bahwa perbedaan performa antarmodel bersifat signifikan secara statistik. Rumus *paired t-test* dinyatakan pada rumus Persamaan (11).

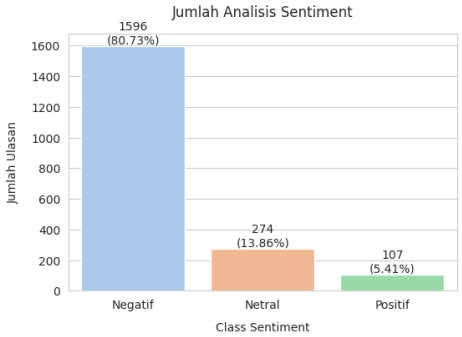
$$t = \frac{\bar{d}}{s_d/\sqrt{n}} \tag{11}$$

Dengan rancangan evaluasi ini, penelitian tidak hanya menilai peningkatan performa model pada satu pembagian data, tetapi juga menguji konsistensi dan keandalan model pada beberapa lipatan validasi.

3. HASIL DAN PEMBAHASAN

Pengumpulan dan Pelabelan Sentimen

Hasil pengumpulan dan pelabelan sentimen menunjukkan bahwa ulasan pengguna aplikasi iPusnas memiliki distribusi kelas yang tidak seimbang. Sebagaimana ditampilkan pada Gambar 2, kelas negatif menjadi kelas mayoritas dengan jumlah 1596 ulasan atau 80,73% dari keseluruhan data. Kelas netral berada pada posisi kedua dengan 274 ulasan atau 13,86%, sedangkan kelas positif hanya berjumlah 107 ulasan atau 5,41%. Komposisi tersebut memperlihatkan bahwa sebagian besar pengguna dalam dataset lebih banyak menyampaikan keluhan, kritik, atau pengalaman yang kurang memuaskan dibandingkan ulasan yang bersifat apresiatif.



Gambar 2 Distribusi Sentimen

Dominasi sentimen negatif pada Gambar 2 mengindikasikan bahwa ulasan aplikasi iPusnas banyak berkaitan dengan permasalahan pengalaman pengguna. Dalam konteks aplikasi perpustakaan digital, ulasan negatif umumnya dapat mencerminkan hambatan penggunaan, seperti kendala akses aplikasi, kesulitan membaca atau meminjam buku, gangguan sistem, pembaruan aplikasi, maupun kebutuhan perbaikan fitur. Oleh karena itu, distribusi sentimen ini tidak hanya menunjukkan karakteristik data untuk pemodelan, tetapi juga memberikan gambaran awal mengenai persepsi pengguna terhadap kualitas layanan aplikasi.

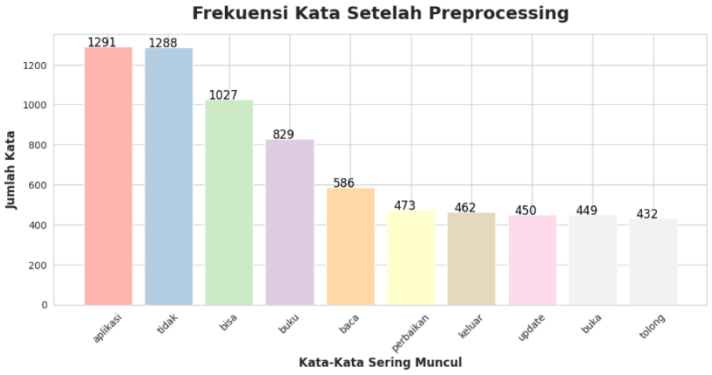
Dari sudut pandang klasifikasi, ketimpangan antara kelas negatif, netral, dan positif menunjukkan adanya permasalahan *class imbalance*. Perbedaan jumlah antara kelas negatif dan positif sangat besar, yaitu 1489 ulasan. Kondisi ini berpotensi menyebabkan model klasifikasi lebih mudah mempelajari pola kelas negatif karena jumlah representasinya jauh lebih dominan dibandingkan kelas lain. Sebaliknya, kelas positif berisiko kurang terwakili dalam proses pembelajaran model karena jumlah datanya sangat kecil. Pada dataset tidak seimbang, model dapat menghasilkan nilai akurasi yang tampak tinggi, tetapi gagal mengenali kelas minoritas secara optimal [1], [2].

Distribusi pada Gambar 2 juga menjadi dasar penting dalam menentukan strategi evaluasi penelitian. Penggunaan akurasi saja tidak memadai karena kelas mayoritas dapat mendominasi hasil pengukuran. Jika model lebih sering memprediksi kelas negatif, akurasi dapat tetap tinggi meskipun performa pada kelas netral dan positif rendah. Oleh sebab itu, penelitian ini lebih menekankan *F1-score macro* sebagai metrik utama karena metrik tersebut menghitung rata-rata performa setiap kelas secara seimbang tanpa memberikan dominasi pada kelas mayoritas [3].

Hasil pelabelan ini memperkuat relevansi penerapan teknik penanganan *class imbalance* pada tahapan eksperimen berikutnya. Ketidakseimbangan distribusi kelas yang terlihat pada Gambar 2 menjadi alasan empiris bahwa model *baseline* perlu dibandingkan dengan model yang menerapkan teknik penyeimbangan data. Dengan demikian, tahap pengumpulan dan pelabelan sentimen tidak hanya menghasilkan data berlabel, tetapi juga memperlihatkan masalah utama penelitian, yaitu dominasi kelas negatif yang dapat memengaruhi objektivitas dan kestabilan performa klasifikasi sentimen.

Preprocessing dan Feature Extraction

Hasil *preprocessing* menunjukkan bahwa teks ulasan aplikasi iPusnas telah berubah menjadi representasi kata yang lebih bersih dan terstruktur. Sebagaimana ditampilkan pada Gambar 3, kata yang paling sering muncul setelah *preprocessing* adalah “aplikasi” sebanyak 1291 kali, “baik” sebanyak 1288 kali, “bisa” sebanyak 1027 kali, “buku” sebanyak 829 kali, dan “baca” sebanyak 586 kali. Kata lain yang juga muncul dengan frekuensi tinggi adalah “perbaikan” sebanyak 473 kali, “kasih” sebanyak 462 kali, “update” sebanyak 450 kali, “buka” sebanyak 449 kali, dan “tolong” sebanyak 432 kali.



Gambar 3 Frekuensi Kata Setelah Preprocessing

Distribusi kata pada Gambar 3 memperlihatkan bahwa hasil *preprocessing* masih mempertahankan istilah yang relevan dengan konteks aplikasi perpustakaan digital. Kata “aplikasi”, “buku”, dan “baca” merepresentasikan objek utama yang dibahas pengguna, yaitu penggunaan aplikasi iPusnas sebagai media membaca buku digital. Sementara itu, kemunculan kata “perbaikan”, “*update*”, “buka”, dan “tolong” menunjukkan bahwa sebagian ulasan berisi permintaan perbaikan layanan, kendala akses, atau masalah teknis yang dialami pengguna. Dengan demikian, kata-kata dominan yang tersisa setelah *preprocessing* masih memiliki hubungan langsung dengan isu pengalaman pengguna. Hasil ini menunjukkan bahwa tahapan *preprocessing* tidak hanya berfungsi menghapus

unsur yang tidak relevan, tetapi juga memperjelas pola informasi yang terkandung dalam ulasan. Teks ulasan pengguna umumnya mengandung variasi penulisan, kata tidak baku, simbol, angka, dan ekspresi informal, sehingga pembersihan dan normalisasi diperlukan agar fitur yang dihasilkan lebih konsisten [3], [4]. Dalam konteks penelitian ini, keberhasilan *preprocessing* terlihat dari munculnya kata-kata yang merepresentasikan domain penelitian, bukan kata acak atau simbol yang tidak memiliki kontribusi semantik.

Tahap *feature extraction* menghasilkan matriks fitur berukuran (1942, 2810). Nilai 1942 menunjukkan jumlah dokumen atau ulasan yang digunakan setelah data melalui proses pembersihan, sedangkan 2810 menunjukkan jumlah fitur teks yang terbentuk dari representasi TF-IDF. Jumlah fitur tersebut menunjukkan bahwa setiap ulasan telah dikonversi menjadi vektor numerik berbasis bobot kata dan pasangan kata. Representasi ini penting karena algoritma Linear SVM tidak memproses teks mentah secara langsung, melainkan membutuhkan bentuk numerik yang dapat digunakan untuk membentuk batas keputusan antarkelas. Jumlah fitur sebanyak 2810 juga menunjukkan bahwa ruang fitur yang terbentuk cukup kaya untuk merepresentasikan variasi kata dalam ulasan iPusnas, tetapi tetap terkendali untuk proses pemodelan. Meskipun parameter maksimum fitur ditetapkan sampai 6000, jumlah fitur aktual yang terbentuk adalah 2810 karena hanya *term* yang memenuhi kriteria frekuensi dan relevansi yang dipertahankan. Hal ini menunjukkan bahwa proses ekstraksi fitur tidak mengambil seluruh kata secara bebas, tetapi menyaring fitur agar lebih representatif dan tidak terlalu bising. Pendekatan seperti ini penting pada klasifikasi teks karena fitur yang terlalu jarang atau terlalu umum dapat menurunkan kualitas model [5].

Secara analitis, hasil *preprocessing* dan *feature extraction* berkontribusi langsung terhadap performa klasifikasi pada tahap berikutnya. Kata-kata dominan pada Gambar 3 memperlihatkan bahwa data mengandung pola topik yang kuat terkait penggunaan aplikasi, akses buku, dan kebutuhan perbaikan. Ketika pola tersebut dikonversi ke dalam fitur TF-IDF, model memperoleh dasar numerik untuk membedakan kecenderungan sentimen negatif, netral, dan positif. Oleh karena itu, kualitas hasil pada tahap ini menjadi fondasi penting sebelum penerapan teknik *class imbalance handling* dan pemodelan menggunakan Linear SVM.

Class Imbalance Handling

Hasil identifikasi distribusi kelas setelah data melalui tahap *preprocessing* menunjukkan bahwa dataset masih memiliki ketimpangan kelas yang cukup besar. Sebagaimana ditampilkan pada Tabel 1, skenario *baseline* terdiri atas 1576 data negatif, 274 data netral, dan 92 data positif dengan total 1942 data. Komposisi ini menunjukkan bahwa kelas negatif menjadi kelas mayoritas, sedangkan kelas positif menjadi kelas minoritas paling rendah. Perbedaan jumlah yang sangat besar antara kelas negatif dan positif mengindikasikan bahwa model berpotensi lebih banyak mempelajari pola kelas negatif dibandingkan kelas lain.

Tabel 1 Distribusi Class Imbalance Handling

	Negatif	Netral	Positif	Total
Baseline	1576	274	92	1942
SMOTE	1576	1576	1576	4728
RUS	92	92	92	276
SMOTE-Tomek	1576	1576	1576	4728

Berdasarkan Tabel 1, rasio antara kelas negatif dan positif pada data *baseline* mencapai sekitar 17:1. Kondisi tersebut menunjukkan bahwa data positif memiliki representasi yang jauh lebih kecil dalam proses pembelajaran. Pada klasifikasi sentimen, ketimpangan seperti ini dapat membuat model lebih cenderung mengoptimalkan prediksi pada kelas mayoritas, sementara kelas minoritas menjadi lebih sulit dikenali. Oleh karena itu, penanganan *class imbalance* menjadi tahap penting agar model tidak hanya menghasilkan performa tinggi pada kelas dominan, tetapi juga mampu mengenali kelas netral dan positif secara lebih proporsional [6], [7].

Penerapan SMOTE menghasilkan perubahan distribusi yang paling besar karena jumlah data pada kelas netral dan positif meningkat hingga setara dengan kelas negatif, yaitu masing-masing 1576 data. Total data pada skenario SMOTE menjadi 4728 data. Hasil ini menunjukkan bahwa teknik *oversampling* mampu memperkuat representasi kelas minoritas tanpa mengurangi jumlah data kelas mayoritas. Secara analitis, kondisi ini menguntungkan bagi model karena seluruh kelas memiliki jumlah sampel yang seimbang, sehingga peluang model untuk mempelajari pola setiap kelas menjadi lebih merata.

Skenario RandomUnderSampler menghasilkan distribusi yang seimbang dengan masing-masing kelas berjumlah 92 data. Namun, keseimbangan tersebut diperoleh dengan mengurangi jumlah data kelas negatif dari 1576 menjadi 92 dan kelas netral dari 274 menjadi 92. Akibatnya, total data hanya tersisa 276 data. Meskipun RUS berhasil menghilangkan dominasi kelas mayoritas, pengurangan data yang terlalu besar berpotensi menghilangkan variasi informasi penting dari kelas negatif dan netral. Hal ini menjadi kelemahan utama *undersampling*, terutama pada data teks yang sangat bergantung pada keragaman kata dan konteks ulasan.

Skenario SMOTE-Tomek menghasilkan distribusi yang sama dengan SMOTE, yaitu 1576 data pada setiap kelas dengan total 4728 data. Kesamaan ini menunjukkan bahwa proses pembersihan menggunakan Tomek Links tidak mengurangi jumlah data secara signifikan setelah proses *oversampling*. Dengan kata lain, pada dataset ini tidak banyak sampel hasil SMOTE yang teridentifikasi sebagai pasangan data tumpang tindih atau berada pada batas kelas yang perlu dihapus. Kondisi tersebut mengindikasikan bahwa distribusi hasil *oversampling* relatif stabil terhadap mekanisme pembersihan Tomek Links.

Perbandingan pada Tabel 1 memperlihatkan bahwa setiap teknik penanganan *class imbalance* memiliki karakteristik yang berbeda. SMOTE dan SMOTE-Tomek mempertahankan seluruh data mayoritas sekaligus memperkuat kelas minoritas melalui penambahan data sintetis. Sebaliknya, RUS menyeimbangkan data dengan mengurangi

kelas mayoritas sehingga ukuran dataset menjadi jauh lebih kecil. Perbedaan ini penting karena performa model tidak hanya dipengaruhi oleh keseimbangan kelas, tetapi juga oleh jumlah dan variasi data yang tersedia untuk pelatihan. Dengan demikian, hasil pada tahap *class imbalance handling* menunjukkan bahwa SMOTE dan SMOTE-Tomek lebih potensial untuk mendukung pembelajaran model dibandingkan RUS. Kedua teknik tersebut menghasilkan distribusi kelas yang seimbang tanpa menghilangkan informasi pada kelas mayoritas. Sementara itu, RUS tetap relevan sebagai pembanding karena mampu menunjukkan dampak penyeimbangan melalui pengurangan data, tetapi risikonya lebih besar terhadap hilangnya informasi pelatihan. Temuan ini menjadi dasar untuk menafsirkan hasil evaluasi model pada tahap berikutnya, khususnya perbedaan performa antara model *baseline*, SMOTE, RUS, dan SMOTE-Tomek.

Train-Test Split dan Modeling

Hasil pembagian data latih dan data uji pada setiap skenario ditampilkan pada Tabel 2. Skenario *baseline* memiliki total 1942 data, yang terbagi menjadi 1359 data latih dan 583 data uji. Skenario SMOTE dan SMOTE-Tomek memiliki jumlah data yang lebih besar, yaitu 4728 data, dengan komposisi 3309 data latih dan 1419 data uji. Sementara itu, skenario RUS memiliki jumlah data paling kecil, yaitu 276 data, yang terbagi menjadi 193 data latih dan 83 data uji.

Tabel 2 Distribusi Pembagian Data

	Training	**Testing**	**Total**
Baseline	1359	583	1942
SMOTE	3309	1419	4728
RUS	193	83	276
SMOTE-Tomek	3309	1419	4728

Berdasarkan Tabel 2, perbedaan jumlah data pada setiap skenario menunjukkan dampak langsung dari teknik penanganan *class imbalance* terhadap volume data yang digunakan dalam pemodelan. Pada *baseline*, model dibangun menggunakan data asli yang masih merepresentasikan ketidakseimbangan kelas. Kondisi ini memungkinkan model mempelajari pola dari data aktual tanpa modifikasi distribusi, tetapi tetap membawa risiko bias terhadap kelas mayoritas. Dengan jumlah data latih 1359, *baseline* memiliki informasi pelatihan yang cukup besar, namun distribusi kelas yang tidak seimbang dapat membatasi kemampuan model dalam mengenali kelas minoritas secara optimal [6], [7].

Skenario SMOTE dan SMOTE-Tomek menghasilkan jumlah data latih yang jauh lebih besar dibandingkan *baseline*. Jumlah 3309 data latih pada kedua skenario menunjukkan bahwa kelas minoritas telah diperkuat melalui pembentukan data sintetis. Dari sisi pemodelan, kondisi ini memberikan peluang yang lebih baik bagi Linear SVM untuk membentuk batas keputusan yang tidak hanya didominasi oleh kelas negatif, tetapi juga mempertimbangkan pola kelas netral dan positif. Hal ini penting karena algoritma SVM bekerja dengan mencari *hyperplane* terbaik untuk memisahkan kelas, sehingga

representasi kelas yang lebih seimbang dapat membantu pembentukan batas klasifikasi yang lebih proporsional [8].

Sebaliknya, skenario RUS hanya menyisakan 193 data latih dan 83 data uji. Walaupun distribusi kelas pada skenario ini menjadi seimbang, jumlah data yang tersedia untuk pembelajaran menjadi sangat terbatas. Pada data teks, pengurangan data secara besar dapat menghilangkan banyak variasi kata, frasa, dan konteks ulasan. Akibatnya, model berisiko kehilangan informasi penting yang seharusnya dapat digunakan untuk membedakan sentimen. Kondisi ini menjelaskan mengapa *undersampling* sering menghasilkan model yang lebih seimbang dibandingkan *baseline*, tetapi belum tentu memberikan performa terbaik apabila data yang tersisa terlalu sedikit [9].

Pemodelan pada seluruh skenario dilakukan dengan algoritma Linear Support Vector Machine. Model *baseline* digunakan sebagai acuan untuk menilai performa SVM pada data asli yang tidak seimbang. Model SMOTE, RUS, dan SMOTE-Tomek digunakan untuk menilai sejauh mana perubahan distribusi kelas memengaruhi kemampuan klasifikasi. Dengan penggunaan algoritma yang sama pada seluruh skenario, perbedaan performa yang muncul dapat dikaitkan secara lebih jelas dengan strategi penanganan *class imbalance*, bukan karena perbedaan arsitektur atau karakteristik algoritma. Hasil pembagian data pada Tabel 2 juga menunjukkan bahwa strategi *oversampling* lebih menguntungkan dari sisi ketersediaan informasi pelatihan dibandingkan *undersampling*. SMOTE dan SMOTE-Tomek tidak menghilangkan data kelas mayoritas, tetapi menambah representasi kelas minoritas sehingga model tetap memperoleh variasi data yang luas. Sebaliknya, RUS menyeimbangkan kelas dengan mengurangi data mayoritas, sehingga informasi yang digunakan dalam pelatihan menjadi jauh lebih sedikit. Perbedaan ini menjadi faktor penting dalam interpretasi hasil evaluasi, terutama ketika membandingkan performa *F1-macro* dan stabilitas model pada tahap pengujian.

Dengan demikian, tahap train-test split dan modeling memperlihatkan bahwa kualitas pemodelan tidak hanya dipengaruhi oleh algoritma yang digunakan, tetapi juga oleh distribusi dan ukuran data latih. Linear SVM berpotensi bekerja lebih optimal ketika data latih memiliki keseimbangan kelas yang baik dan tetap mempertahankan keragaman informasi. Oleh karena itu, skenario SMOTE dan SMOTE-Tomek secara awal memiliki kondisi yang lebih mendukung untuk menghasilkan performa klasifikasi yang stabil dibandingkan *baseline* dan RUS.

Evaluasi Model dan Evaluasi Robustness

Hasil evaluasi model pada Tabel 3 menunjukkan bahwa terdapat perbedaan performa yang jelas antara model *baseline* dan model yang menerapkan penanganan *class imbalance*. Model *baseline* memperoleh nilai *accuracy* sebesar 0,851, *precision macro* sebesar 0,673, *recall macro* sebesar 0,551, dan *F1-score macro* sebesar 0,593. Nilai *accuracy baseline* terlihat cukup tinggi, tetapi nilai *recall macro* dan *F1-score macro* masih rendah. Kondisi ini menunjukkan bahwa model *baseline* belum mampu mengenali seluruh kelas sentimen secara seimbang, terutama pada kelas minoritas.

Tabel 3 Evaluasi Model

	Accuracy	**Precision (Macro)**	**Recall (Macro)**	**F1-score (Macro)**
Baseline	0.851	0.673	0.551	0.593
SMOTE	0.970	0.970	0.970	0.970
RUS	0.663	0.674	0.664	0.656
SMOTE-Tomek	0.970	0.970	0.970	0.970

Perbedaan antara *accuracy* dan *F1-score macro* pada *baseline* memperlihatkan bahwa akurasi tidak cukup representatif untuk menilai model pada dataset yang tidak seimbang. Nilai *accuracy* sebesar 0,851 dapat dipengaruhi oleh dominasi kelas negatif sebagai kelas mayoritas. Namun, *F1-score macro* sebesar 0,593 menunjukkan bahwa performa rata-rata model pada semua kelas masih terbatas. Dalam klasifikasi multikelas yang mengalami *class imbalance*, metrik makro lebih sesuai karena memberikan bobot yang sama pada setiap kelas, sehingga performa kelas minoritas tetap diperhitungkan secara adil [6], [7].

Penerapan SMOTE menghasilkan peningkatan performa yang signifikan dibandingkan *baseline*. Berdasarkan Tabel 3, model SVM dengan SMOTE memperoleh nilai *accuracy*, *precision macro*, *recall macro*, dan *F1-score macro* masing-masing sebesar 0,970. Dibandingkan *baseline*, terjadi peningkatan *F1-score macro* sebesar 0,377 poin, dari 0,593 menjadi 0,970. Peningkatan ini menunjukkan bahwa penambahan sampel sintetis pada kelas minoritas mampu memperbaiki kemampuan model dalam mengenali kelas negatif, netral, dan positif secara lebih seimbang.

Hasil yang sama diperoleh pada skenario SMOTE-Tomek. Model ini juga menghasilkan *accuracy*, *precision macro*, *recall macro*, dan *F1-score macro* sebesar 0,970. Kesamaan nilai antara SMOTE dan SMOTE-Tomek menunjukkan bahwa pada dataset ini proses pembersihan Tomek Links tidak memberikan perubahan performa yang berbeda dari SMOTE. Hal tersebut selaras dengan hasil distribusi sebelumnya, ketika jumlah data SMOTE dan SMOTE-Tomek sama-sama berjumlah 4728 data. Dengan demikian, kontribusi utama peningkatan performa pada skenario ini lebih kuat berasal dari proses *oversampling* dibandingkan proses pembersihan Tomek Links [15].

Skenario RUS menghasilkan *accuracy* sebesar 0,663, *precision macro* sebesar 0,674, *recall macro* sebesar 0,664, dan *F1-score macro* sebesar 0,656. Jika dibandingkan *baseline*, nilai *F1-score macro* RUS meningkat sebesar 0,063 poin. Peningkatan ini menunjukkan bahwa pengurangan dominasi kelas mayoritas dapat membantu model menjadi lebih seimbang dalam mengenali kelas. Namun, penurunan *accuracy* dari 0,851 menjadi 0,663 menunjukkan adanya konsekuensi dari pengurangan data yang terlalu besar. Pada data teks, penghapusan banyak sampel mayoritas dapat menghilangkan variasi kata dan konteks ulasan yang penting bagi pembelajaran model [8].

Evaluasi *robustness* melalui *cross validation* memperkuat hasil evaluasi utama. Sebagaimana ditampilkan pada Tabel 4, *baseline* memperoleh rata-rata *F1-macro* sebesar 0,5590 dengan standar deviasi 0,0417. Nilai ini menunjukkan bahwa performa *baseline*

relatif rendah dan masih mengalami variasi antar*fold*. Hal ini mengindikasikan bahwa model *baseline* belum stabil ketika diuji pada beberapa variasi pembagian data.

Tabel 4 Cross Validation Score

	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Mean	Std
Baseline	0.5478	0.6107	0.6047	0.5091	0.5228	0.5590	0.0417
SMOTE	0.9743	0.9727	0.9727	0.9711	0.9742	0.9730	0.0012
RUS	0.5753	0.7629	0.6663	0.5442	0.6510	0.6399	0.0765
SMOTE-Tomek	0.9743	0.9727	0.9727	0.9711	0.9742	0.9730	0.0012

Skenario SMOTE dan SMOTE-Tomek memperoleh hasil paling stabil dengan rata-rata *F1-macro* sebesar 0,9730 dan standar deviasi 0,0012. Nilai standar deviasi yang sangat kecil menunjukkan bahwa performa kedua skenario tersebut konsisten pada seluruh *fold*. Dengan kata lain, peningkatan performa yang dihasilkan oleh SMOTE dan SMOTE-Tomek tidak hanya terjadi pada satu pembagian data, tetapi tetap stabil pada beberapa variasi data latih dan validasi. Temuan ini memperkuat bahwa *oversampling* memberikan representasi kelas yang lebih proporsional bagi model Linear SVM.

Sementara itu, RUS memperoleh rata-rata *F1-macro* sebesar 0,6399 dengan standar deviasi 0,0765. Nilai rata-rata tersebut lebih tinggi dibandingkan *baseline*, tetapi standar deviasinya paling besar di antara seluruh skenario. Hal ini menunjukkan bahwa performa RUS lebih fluktuatif. Ketidakstabilan tersebut dapat dikaitkan dengan jumlah data yang jauh lebih kecil setelah proses *undersampling*. Ketika jumlah data latih terbatas, perubahan komposisi data pada setiap *fold* dapat berdampak lebih besar terhadap hasil klasifikasi.

Hasil uji signifikansi statistik pada Tabel 5 menunjukkan bahwa seluruh teknik penanganan *class imbalance* memberikan perbedaan performa yang signifikan dibandingkan *baseline*. Perbandingan *baseline* dengan SMOTE menghasilkan nilai t-statistic sebesar -19,868 dan p-value sebesar 0,000. Perbandingan *baseline* dengan SMOTE-Tomek juga menghasilkan nilai yang sama, yaitu t-statistic -19,868 dan p-value 0,000. Nilai p-value tersebut berada di bawah batas signifikansi 0,05, sehingga peningkatan performa pada kedua skenario dapat dinyatakan signifikan secara statistik.

Tabel 5 Statistical Significance Test

	T-statistic	P-value	Significant
Baseline vs SMOTE	-19.868	0.000	Yes
Baseline vs RUS	-3.222	0.032	Yes
Baseline vs SMOTE-Tomek	-19.868	0.000	Yes

Perbandingan *baseline* dengan RUS juga menunjukkan hasil signifikan dengan t-statistic sebesar -3,222 dan p-value sebesar 0,032. Namun, nilai signifikansi RUS tidak diikuti oleh peningkatan performa sebesar SMOTE dan SMOTE-Tomek. Hal ini menunjukkan bahwa RUS memang memberikan perubahan performa yang bermakna dibandingkan *baseline*, tetapi efektivitasnya lebih terbatas. Dengan demikian, hasil uji statistik tidak hanya menunjukkan ada atau tidaknya perbedaan, tetapi juga perlu dibaca bersama nilai rata-rata performa dan standar deviasi pada Tabel 4.

Secara keseluruhan, hasil pada Tabel 3, Tabel 4, dan Tabel 5 menunjukkan pola yang konsisten. Model *baseline* memiliki *accuracy* yang cukup tinggi, tetapi lemah pada *F1-score macro* karena distribusi kelas tidak seimbang. RUS mampu meningkatkan *F1-score macro* dibandingkan *baseline*, tetapi performanya kurang stabil karena banyak informasi data mayoritas yang dihilangkan. SMOTE dan SMOTE-Tomek menjadi skenario terbaik karena menghasilkan performa tertinggi, stabilitas terbaik, serta peningkatan yang signifikan secara statistik.

Temuan ini menegaskan bahwa penanganan *class imbalance* berpengaruh langsung terhadap kualitas klasifikasi sentimen ulasan aplikasi iPusnas. Pada dataset ini, strategi *oversampling* lebih efektif dibandingkan *undersampling* karena mampu memperkuat kelas minoritas tanpa menghilangkan informasi dari kelas mayoritas. Oleh karena itu, Linear SVM bekerja lebih optimal ketika distribusi kelas dibuat seimbang melalui SMOTE atau SMOTE-Tomek. Hasil ini juga memperkuat pentingnya penggunaan *F1-score macro*, *cross validation*, dan uji signifikansi statistik dalam evaluasi model klasifikasi sentimen pada data multikelas yang tidak seimbang [9], [10].

4. KESIMPULAN

Berdasarkan hasil penelitian dan pengujian yang telah dilakukan, diperoleh beberapa kesimpulan sebagai berikut.

1. Permasalahan utama dalam penelitian ini adalah *class imbalance* pada dataset ulasan aplikasi iPusnas. Dataset awal didominasi kelas negatif sebesar 80,73%, sedangkan kelas positif hanya 5,41%. Setelah pembersihan data, dataset pemodelan berjumlah 1942 ulasan dengan distribusi 1576 negatif, 274 netral, dan 92 positif.
2. Model *baseline* Linear SVM tanpa penanganan *class imbalance* menghasilkan *accuracy* sebesar 0,851, tetapi *F1-score* macro hanya 0,593. Hasil ini menunjukkan bahwa model cenderung bias terhadap kelas mayoritas dan belum optimal mengenali kelas minoritas.
3. Penerapan teknik penanganan *class imbalance* mampu meningkatkan performa klasifikasi. SMOTE dan SMOTE-Tomek memberikan hasil terbaik dengan *accuracy*, *precision macro*, *recall macro*, dan *F1-score macro* sebesar 0,970. RUS meningkatkan *F1-score macro* menjadi 0,656, tetapi menurunkan *accuracy* menjadi 0,663.

4. Evaluasi *cross validation* dan uji signifikansi menunjukkan bahwa peningkatan performa setelah penanganan *class imbalance* bersifat konsisten dan signifikan dibandingkan *baseline*.
5. Kelebihan penelitian ini adalah membandingkan beberapa teknik *imbalance handling* dan menggunakan *F1-score macro* sebagai metrik utama. Kekurangannya, dataset masih terbatas pada iPusnas dan data sintetis perlu validasi lanjutan. Penelitian berikutnya dapat memperluas dataset, menguji algoritma lain, dan menambahkan optimasi *hyperparameter*.

DAFTAR PUSTAKA

[1] N. A. K. M. Haris, S. Mutalib, A. M. A. Malik, S. Abdul-Rahman, and S. N. K. Kamarudin, "Sentiment Classification From Reviews for Tourism Analytics," *Int. J. Adv. Intell. Informatics*, vol. 9, no. 1, p. 108, 2023, doi: 10.26555/ijain.v9i1.1077.

[2] T. Kolajo, O. Daramola, A. A. Adebiyi, and A. Seth, "A Framework for Pre-Processing of Social Media Feeds Based on Integrated Local Knowledge Base," *Inf. Process. Manag.*, vol. 57, no. 6, p. 102348, 2020, doi: 10.1016/j.ipm.2020.102348.

[3] E. H. Muktafin, P. Pramono, and K. Kusrini, "Sentiments Analysis of Customer Satisfaction in Public Services Using K-Nearest Neighbors Algorithm and Natural Language Processing Approach," *Telkomnika (Telecommunication Comput. Electron. Control.*, vol. 19, no. 1, p. 146, 2021, doi: 10.12928/telkomnika.v19i1.17417.

[4] A. B. P. Negara, "The Influence of Applying Stopword Removal and Smote on Indonesian Sentiment Classification," *Lontar Komput. J. Ilm. Teknol. Inf.*, vol. 14, no. 03, pp. 172–185, 2025, doi: 10.24843/lkjiti.2023.v14.i03.p05.

[5] N. N. Alabid and Z. Naseer, "Summarizing Twitter Posts Regarding COVID-19 Based on N-Grams," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 31, no. 2, p. 1008, 2023, doi: 10.11591/ijeecs.v31.i2.pp1008-1015.

[6] I. A. Farha and W. Magdy, "A Comparative Study of Effective Approaches for Arabic Sentiment Analysis," *Inf. Process. Manag.*, vol. 58, no. 2, p. 102438, 2021, doi: 10.1016/j.ipm.2020.102438.

[7] R. Guido, M. C. Groccia, and D. Conforti, "A Hyper-Parameter Tuning Approach for Cost-Sensitive Support Vector Machine Classifiers," *Soft Comput.*, vol. 27, no. 18, pp. 12863–12881, 2022, doi: 10.1007/s00500-022-06768-8.

[8] M. Khairy, T. M. Mahmoud, and T. A. El-Hafeez, "The Effect of Rebalancing Techniques on the Classification Performance in Cyberbullying Datasets," *Neural Comput. Appl.*, vol. 36, no. 3, pp. 1049–1065, 2023, doi: 10.1007/s00521-023-09084-w.

[9] S. F. TAŞKIRAN, B. Türkoğlu, E. Kaya, and T. Aşuroğlu, "A Comprehensive Evaluation of Oversampling Techniques for Enhancing Text Classification Performance," *Sci. Rep.*, vol. 15, no. 1, 2025, doi: 10.1038/s41598-025-05791-7.

[10] W. Rhmann, "An Empirical Study on the Class Imbalance Handling Techniques for Different Diseases," *Soft Comput.*, vol. 28, no. 19, pp. 11439–11456, 2024, doi: 10.1007/s00500-024-09881-y.

[11] S. Riyanto, I. S. Sitanggang, T. Djatna, and T. D. Atikah, "Comparative Analysis Using Various Performance Metrics in Imbalanced Data for Multi-Class Text Classification," *Int. J. Adv. Comput. Sci. Appl.*, vol. 14, no. 6, 2023, doi: 10.14569/ijacsa.2023.01406116.

[12] M. Reusens *et al.*, "Evaluating Text Classification: A Benchmark Study," *Expert Syst. Appl.*, vol. 254, p. 124302, 2024, doi: 10.1016/j.eswa.2024.124302.

[13] E. Elyan, C. F. Moreno-García, and C. Jayne, "CDSMOTE: Class Decomposition and Synthetic Minority Class Oversampling Technique for Imbalanced-Data Classification," *Neural Comput. Appl.*, vol. 33, no. 7, pp. 2839–2851, 2020, doi: 10.1007/s00521-020-05130-z.

[14] W. Saidi, A. E. Abderrahmani, and K. Satori, "Effective Comparative Evaluation of Sentiment Analysis Using Paired T-Test: A Performance Study of Supervised Methods," *J. Southwest Jiaotong Univ.*, vol. 58, no. 5, 2023, doi: 10.35741/issn.0258-2724.58.5.28.

[15] P. Vuttipittayamongkol, E. Elyan, and A. Petrovski, "On the Class Overlap Problem in Imbalanced Data Classification," *Knowledge-Based Syst.*, vol. 212, p. 106631, 2021, doi: 10.1016/j.knosys.2020.106631.