

Eksplorasi *Random Forest* Berbasis Ekstraksi Fitur dan Metode Sampling untuk Analisis Data Multi-label Sentimen dan Emosi

Rebecca Jasmine Surya¹, Yuni Yamasari²

^{1,2}Teknik Informatika/Teknik Informatika, Universitas Negeri Surabaya

rebecca.22052@mhs.unesa.ac.id

yuniyamasari@unesa.ac.id

Abstrak— Penggunaan aplikasi mobile dalam bidang teknologi telah memberikan wadah bagi penggunaannya dalam mengekspresikan emosi dan opini melalui ulasan pengguna. Klasifikasi emosi pada ulasan pengguna dapat membantu dalam memahami tingkat kepuasan pengguna. Namun, proses klasifikasi emosi masih mengalami permasalahan khususnya dalam menghadapi ketidakseimbangan data. Penelitian ini mengusulkan model klasifikasi emosi menggunakan *Random Forest* pada dataset ulasan di *Google Play Store* yang melalui tahapan *pre-processing*, ekstraksi fitur menggunakan *TF-IDF*, penerapan metode sampling berupa *SMOTE*, *Tomek-Links*, dan *SMOTE-Tomek* dalam permasalahan ketidakseimbangan data, serta pemanfaatan *hyperparameter tuning* menggunakan *Particle Swarm Optimization (PSO)*. Model klasifikasi tersebut menunjukkan bahwa penggunaan algoritma *Random Forest* dengan metode sampling *SMOTE* pada pembagian data dengan rasio 95:5 mampu menghasilkan kombinasi dengan kinerja paling optimal dengan nilai *accuracy* dan *recall* masing-masing sebesar 85,24%, *precision* 85,32% dan nilai *F1-score* sebesar 85,20%. Secara keseluruhan, penggunaan *Random Forest* melalui ekstraksi fitur dan metode sampling serta optimasi *hyperparameter* mampu memberikan hasil model klasifikasi emosi yang lebih efektif dan efisien dalam menangani data yang tidak seimbang.

Kata Kunci— *Random Forest*, *TF-IDF*, Metode Sampling, *Particle Swarm Optimization*, Klasifikasi Emosi.

I. PENDAHULUAN

Perkembangan yang pesat dalam bidang teknologi telah mendorong cara manusia dalam berinteraksi di dunia modern dengan pemanfaatan aplikasi *mobile*. Penggunaan aplikasi *mobile* telah memberikan wadah bagi masyarakat dalam mengekspresikan pengalaman mereka yang berisi kritik atau saran melalui *review* atau ulasan pengguna [1]. Ulasan pengguna dapat membantu perusahaan untuk meningkatkan kualitas layanan dan kinerja dari aplikasi. Data ulasan pengguna aplikasi memiliki nilai lebih dalam melakukan klasifikasi emosi untuk dapat memberikan pemahaman terkait tingkat kepuasan dan kebutuhan pengguna terhadap aplikasi yang mereka gunakan [2]. Klasifikasi emosi dapat memberikan informasi terkait perasaan pengguna seperti bahagia, kecewa, puas, atau marah saat menggunakan aplikasi.

Permasalahan yang muncul dalam proses klasifikasi emosi yaitu adanya ketidakseimbangan data. Data dikatakan tidak seimbang jika masing-masing kelas tidak dapat terdistribusi secara merata [3], sehingga model cenderung bias dan

cenderung mengutamakan kelas mayoritas daripada kelas minoritas [4]. Keberagaman data juga menjadi permasalahan yang muncul karena setiap teks pada ulasan memiliki gaya Bahasa dan struktur kalimat yang berbeda sehingga model perlu memahami berbagai variasi makna agar menghasilkan analisis yang akurat.

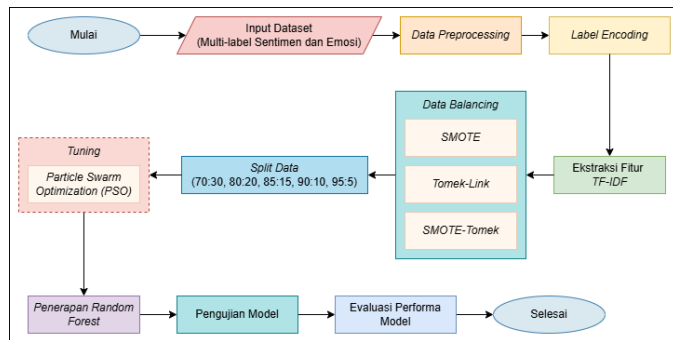
Adanya permasalahan dalam ketidakseimbangan data telah mendorong banyak penelitian dalam melakukan eksplorasi lebih seperti pada penelitian [5] yang melakukan klasifikasi pada data *multi-label* dari ulasan aplikasi *mobile* menggunakan algoritma *Random Forest* dan *Support Vector Machine (SVM)* dengan menerapkan metode *SMOTE*, *Tomek Links* dan *SMOTE-Tomek* dalam mengatasi ketidakseimbangan data yang menunjukkan bahwa kombinasi dari *Random Forest* lebih unggul dibandingkan dengan *Support Vector Machine (SVM)*. Selain itu, penelitian [6] telah menggunakan metode *Particle Swarm Optimization* dengan memberikan nilai akurasi optimal dalam analisis sentiment teks.

Berdasarkan hasil percobaan yang telah dilakukan oleh penelitian sebelumnya, masih belum banyak studi yang melakukan eksplorasi terhadap algoritma *Random Forest* dengan penerapan ekstraksi fitur dan metode sampling pada data *multi-label*. Selain itu, hasil dari beberapa percobaan terdahulu telah menunjukkan bahwa *SMOTE* dan *SMOTE-Tomek* dapat berperan secara aktif dalam mengatasi ketidakseimbangan data. Oleh karena itu, penulis mengusulkan pendekatan model klasifikasi emosi berbasis *Random Forest* dengan menggabungkan ekstraksi fitur melalui representasi vektor *TF-IDF* dan metode sampling seperti *SMOTE*, *Tomek-Links* dan *SMOTE-Tomek* dengan menggunakan *hyperparameter tuning* berupa *PSO*. Metode tersebut harapannya mampu untuk memberikan peningkatan terhadap performa klasifikasi emosi khususnya pada data yang tidak seimbang.

II. METODOLOGI PENELITIAN

Pembangunan model klasifikasi emosi pada penelitian ini diawali dengan tahap persiapan dataset ulasan aplikasi, tahap *pre-processing* teks untuk memperbaiki dan meningkatkan kualitas data sebelum digunakan untuk inputan model. Selanjutnya, tahap ekstraksi fitur menggunakan *TF-IDF*, tahap penyeimbangan data untuk mengatasi ketidakseimbangan pada distribusi data, pembagian data menjadi data *training* dan data *testing*, tahap *hyperparameter tuning* untuk

menemukan kombinasi terbaik, penerapan algoritma *Random Forest*, serta tahap pengujian dan evaluasi performa model klasifikasi. Serangkaian tahapan pada penelitian ini ditunjukkan pada Gambar 1.



Gbr. 1 Tahapan Penelitian.

A. Konfigurasi Penelitian

Lingkungan komputasi berbasis *Google Colaboratory* digunakan dalam penelitian ini dengan implementasi bahasa pemrograman *Python* untuk mendukung seluruh proses pembangunan, pelatihan dan evaluasi model. *Library Scikit-learn* digunakan sebagai *framework* utama pada penelitian ini sebagai implementasi model machine learning. Terdapat beberapa *library* pendukung yang digunakan seperti *Pandas* dan *NumPy* untuk mengolah data, *NLTK* dan *Sastrawi* untuk *preprocessing* teks, *Imbalanced-learn* untuk menangani data tidak seimbang, optimasi *hyperparameter* dengan *PySwarms*, serta *Matplotlib* untuk memvisualisasikan hasil dari model yang dibangun pada penelitian ini.

B. Deskripsi Dataset

Dataset penelitian merupakan data sekunder yang didapatkan melalui platform *Kaggle* yang terdiri atas 21.697 ulasan dari 10 aplikasi *mobile* yang tersedia di *Google Play Store* dalam Bahasa Indonesia [1]. Dataset tersebut bersifat *Multi-label* dengan tiga kolom utama yaitu *content*, *sentiment* dan *emotion*. Kolom '*emotion*' pada dataset terbagi menjadi *anger*, *fear*, *sad*, *happy*, *love*, dan *neutral* sedangkan kolom '*sentiment*' memuat tiga kategori berupa *positive*, *negative*, dan *neutral*. Penelitian ini berfokus pada klasifikasi emosi karena label sentiment pada dataset tersebut didapatkan melalui pengelompokan kategori emosi ke dalam kategori sentiment. Hal ini dapat terlihat pada emosi *anger*, *fear*, dan *sad* yang merupakan sentiment *negative*, emosi *happy* dan *love* tergolong *sentiment positive*, dan *sentiment neutral* berasal dari emosi *neutral*.

TABEL I
SAMPel DATASET PENELITIAN

Content	Sentiment	Emotion
Aplikasi nggak bisa membantu malah nyusahin orang, harus bayar dulu	Negative	Anger
Aplikasinya bagus bgt, mudah, cepat dan aman transaksinya. Bukan aplikasi yg abal abal deh. Thanks !	Positive	Happy

Tiga bintang dulu, semoga beritanya independen n update baru sy kasih lima bintang..so lets see dulu..	Neutral	Neutral
--	---------	---------

C. Pre-processing Data

Pre-processing data merupakan tahapan penting sebelum dilakukan proses klasifikasi dengan memperbaiki kualitas data mentah sebelum digunakan sebagai input model klasifikasi [7]. Tahap *pre-processing* hanya dilakukan pada atribut '*content*' dan bertujuan untuk menghilangkan *noise*, meningkatkan keakuratan, memperjelas fitur, serta mengubah data asli sesuai dengan kebutuhan penelitian [8]. *Pre-processing* data pada penelitian ini akan melalui beberapa tahapan berikut.

1) Penghapusan Data Duplikat

Penghapusan data duplikat dilakukan dengan menghapus data yang memiliki kesamaan pada atribut *content*, *sentiment*, dan *emotion*. Tahapan ini dilakukan untuk memastikan setiap data bersifat unik dan mengurangi redundansi data serta meningkatkan kualitas data sebelum proses klasifikasi [9].

2) Case Folding

Tahap *case folding* dilakukan guna menyamakan format teks dengan mengubah semua huruf pada teks menjadi huruf kecil (*lowercase*) sehingga dapat mengurangi kesalahan penulisan akibat teks yang tidak konsisten [10].

3) Penghapusan Simbol/Tanda Baca

Tahap penghapusan simbol/tanda baca berguna untuk menghilangkan elemen yang tidak diperlukan khususnya berupa simbol dan/ tanda baca yang tidak sesuai dengan konteks penelitian. Tahapan ini dapat membantu dalam mengurangi *noise* pada saat proses pembangunan model [11].

4) Penghapusan Angka

Tahap penghapusan angka dilakukan untuk membersihkan data dari elemen angka yang tidak relevan. Tahapan ini bertujuan untuk menghilangkan *noise* dan meningkatkan akurasi model [12].

5) Penghapusan Spasi Berlebih

Tahap penghapusan spasi berlebih dilakukan untuk mengurangi penggunaan spasi berlebih pada teks sehingga dapat memperbaiki proses tokenisasi, mengurangi *noise*, serta meningkatkan efisiensi proses *pre-processing* [13].

6) Normalisasi Karakter Berulang

Normalisasi karakter berulang dilakukan untuk menangani kata yang memiliki pengulangan huruf secara berlebihan dan tidak sesuai dengan bentuk kata umum. Tahap ini bertujuan untuk menghasilkan representasi kata yang lebih konsisten dan sesuai dengan bentuk aslinya [14].

7) Tokenisasi

Tokenisasi dilakukan untuk memisahkan urutan dari suatu karakter menjadi kalimat yang kemudian akan dipecah menjadi bentuk kata tunggal (token) [15].

8) Stemming

Stemming dilakukan untuk mengubah kata yang memiliki imbuhan menjadi kata dasar. Tahapan ini sangat berpengaruh terhadap keberhasilan teks mining dan bertujuan untuk mengurangi variasi kata yang memiliki makna serupa [5].

D. Ekstraksi Fitur

Tahap ekstraksi fitur dilakukan untuk merepresentasikan teks hasil pre-processing, khususnya pada atribut 'content', ke dalam bentuk vektor numerik. Pada penelitian ini, ekstraksi fitur yang digunakan yaitu *TF-IDF* (*Term Frequency-Inverse Document Frequency*). Teknik *TF-IDF* akan dimulai dengan menghitung frekuensi kata (*Term Frequency*) [16] yang mengacu pada persamaan (1). Kemudian dilanjutkan dengan mengukur pentingnya suatu kata pada seluruh dokumen (*Inverse Document Frequency*) [17] mengacu pada persamaan (2) dan menentukan bobot akhir dari setiap kata (*TF-IDF*) [18] yang mengacu pada persamaan (3).

$$TF_{(t,d)} = \frac{f_{t,d}}{\sum_{t \in d} f_{t,d}} \quad (1)$$

$$IDF_{(t,d)} = \log\left(\frac{N}{df_t}\right) \quad (2)$$

$$TF - IDF_{(t,d)} = TF_{(t,d)} \times IDF_{(t,d)} \quad (3)$$

Pada persamaan (1), (2), dan (3), t ialah kata/*term*, d ialah dokumen, f menunjukkan frekuensi kata, N menunjukkan jumlah total dokumen, dan df merupakan jumlah dokumen yang mengantung kata t .

E. Teknik Penyeimbangan Data

Distribusi data yang tidak proposional pada setiap kelas merupakan contoh dari data yang tidak seimbang sehingga model akan cenderung bias pada kelas mayoritas dan akan mengabaikan kelas minoritas [3] [4]. Ketidakseimbangan tersebut dapat diatasi dengan penerapan metode *sampling* untuk menambahkan data sintesis baru pada kelas minoritas (*oversampling*), mengurangi data yang saling tumpang tindih pada kelas mayoritas (*undersampling*), maupun penggabungan dari keduanya [19]. Terdapat beberapa contoh metode *sampling* yang sering digunakan dalam menangani ketidakseimbangan data seperti *SMOTE* sebagai metode *oversampling*, *Tomek-Links* sebagai metode *undersampling*, dan *SMOTE-Tomek* sebagai metode kombinasi dari *oversampling* dan *undersampling*.

1) SMOTE

Synthetic Minority Oversampling Technique (SMOTE) merupakan teknik *oversampling* untuk mengatasi ketidakseimbangan data. Teknik ini akan dilakukan dengan menghasilkan data sintesis baru pada kelas minoritas [3]. Proses pembentukan data sintesis akan dilakukan berdasarkan kedekatan antar kelas menggunakan *k-nearest neighbors* untuk menentukan tetangga terdekat [20]. Penanganan ketidakseimbangan data melalui metode *sampling SMOTE* akan menghasilkan distribusi data yang lebih seimbang.

2) Tomek-Links

Tomek-Links merupakan teknik *undersampling* untuk mengetahui pasangan sampel dari kelas berbeda yang saling tumpang tindih dan menjadi tetangga terdekat [21]. Metode *Tomek-Links* akan dilakukan dengan mengurangi data dari pasangan sampel kelas berbeda yang memiliki jarak terdekat, khususnya pada kelas mayoritas, sehingga mengurangi tumpang tindih antar kelas [22]. Penerapan metode *Tomek-Links* dapat memberikan peningkatan dalam kualitas data yang dihasilkan dengan memperjelas batas pemisahan antar kelas.

3) SMOTE-Tomek

Permasalahan ketidakseimbangan data dapat diatasi menggunakan metode *sampling* berupa *SMOTE-Tomek*. *SMOTE-Tomek* merupakan salah satu metode gabungan dari metode *SMOTE* untuk menghasilkan sampel sintesis baru pada kelas minoritas dan *Tomek-Links* untuk menghapus sampel pasangan dari kelas yang berbeda dan saling tumpang tindih [23]. Gabungan antara *SMOTE* dan *Tomek-Links* bertujuan untuk meningkatkan proporsi antar kelas sehingga mampu menghasilkan distribusi data yang lebih seimbang dan meningkatkan performa klasifikasi model.

F. Pembagian Data

Dataset yang telah melalui proses penyeimbangan data akan dibagi menjadi data *training* untuk melatih model dan data *testing* untuk mengukur performa model yang telah dilatih. Proporsi pembagian data yang diterapkan oleh beberapa peneliti sebelumnya [24][15], umumnya menggunakan rasio 80%:20%. Namun, penelitian ini akan menggunakan berbagai rasio pembagian data berupa 95%:5%, 90%:10%, 85%:15%, 80%:20%, dan 70%:30% untuk dapat menemukan kombinasi yang optimal [2].

G. Hyperparameter Tuning

Hyperparameter tuning merupakan suatu proses dalam menentukan kombinasi parameter yang optimal selama proses pembelajaran dari suatu algoritma. Pada penerapannya, penggunaan *Particle Swarm Optimization (PSO)* akan digunakan sebagai *hyperparameter tuning* untuk algoritma *Random Forest*. Metode *PSO* yang merupakan metode optimasi metaheuristik berbasis populasi akan direpresentasikan menjadi partikel yang ditentukan oleh ' n particles' yang akan bergerak menuju posisi terbaik [25]. Parameter '*iters*' akan menentukan jumlah siklus yang dilakukan untuk dapat memberikan solusi yang optimal. Persamaan (4) dan (5) menunjukkan proses pembaruan *velocity* dan *position* yang dilakukan [26].

$$v_i^{t+1} = \omega v_i^t + c_1 r_1 (p_{best_i}^t - x_i^t) + c_2 r_2 (g_{best}^t - x_i^t) \quad (4)$$

$$x_i^{t+1} = x_i^t + v_i^{t+1} \quad (5)$$

Pada persamaan (4) dan (5), v ialah *velocity*, x ialah *position*, p menunjukkan posisi partikel terbaik, dan g menunjukkan posisi global terbaik.

H. Pembangunan Model

Algoritma *Random Forest* digunakan pada penelitian ini untuk membangun model klasifikasi emosi. *Random Forest* merupakan metode *ensemble learning* yang menggabungkan beberapa pohon keputusan (*decision tree*) untuk meningkatkan performa klasifikasi dan mengurangi risiko *overfitting* [27]. Proses pemilihan fitur pada setiap node *Random Forest* akan menggunakan *Gini Impurity* untuk mengukur ketidakmurnian atribut. Semakin kecil nilai *Gini Impurity*, maka atribut tersebut semakin baik dalam memisahkan kelas. Perhitungan terkait *Gini Impurity* terlihat pada persamaan (6). Pada persamaan (6), c ialah jumlah kelas dalam data dan p ialah jumlah *sample* pada kelas i [28].

$$Gini = 1 - \sum_{i=1}^c (p_i)^2 \quad (6)$$

I. Pengukuran Kinerja Model

Beberapa metrik evaluasi digunakan pada penelitian ini untuk mengukur kinerja model klasifikasi emosi yang diusulkan. Pengukuran tersebut berguna untuk mengukur dan mengevaluasi seberapa optimal model machine learning dalam mengklasifikasikan data berbentuk teks [15]. Metrik pengukuran kinerja model yang digunakan antara lain:

1) Accuracy

Accuracy merupakan alat ukur yang menunjukkan perbandingan antara jumlah prediksi benar dengan total keseluruhan prediksi. Rumus untuk menghitung nilai *accuracy* dapat menggunakan persamaan (7).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (7)$$

2) Precision

Precision adalah metrik evaluasi untuk mengukur seberapa tepat model dalam melakukan prediksi kelas positif. Mengacu pada (8), *precision* dapat dihitung melalui proporsi antara *true positive* dengan total data yang tergolong ke dalam kategori positif (*True Positive* dan *False Positive*).

$$Presisi = \frac{TP}{TP + FP} \quad (8)$$

3) Recall

Recall atau sensitifitas merupakan alat ukur untuk menilai seberapa baik model dapat menemukan semua data positif. Mengacu pada (9), *recall* diperoleh melalui perhitungan rasio antara *true positive* dengan *actual positive*.

$$Recall = \frac{TP}{TP + FN} \quad (9)$$

4) F1-Score

F1-score merupakan alat ukur performa untuk mengetahui keselarasan antara *precision* dan *recall* dalam proses klasifikasi. Nilai *F1-score* dapat dihitung melalui gabungan antara nilai rata-rata *presisi* dan *recall* sebagaimana ditunjukkan pada persamaan (10).

$$F1 - Score = 2 * \frac{Precision \times Recall}{Precision + Recall} \quad (10)$$

5) ROC

Receiver Operating Characteristic (ROC) merupakan alat ukur performa klasifikasi dalam membedakan antara kelas positif dan negatif. Kurva *ROC* dibentuk berdasarkan hubungan antara *True Positive Rate (TPR)* dan *False Positive Rate (FPR)* yang terlihat pada persamaan (11) dan persamaan (12) [29].

$$TPR = \frac{TP}{TP + FN} \quad (11)$$

$$FPR = \frac{FP}{FP + TN} \quad (12)$$

6) AUC

Area Under Curve (AUC) merupakan alat ukur performa model klasifikasi berdasarkan kurva *Receiver Operating Characteristic (ROC)*. Luas area di bawah kurva *ROC* akan menunjukkan nilai *AUC* sebagaimana terlihat pada persamaan (13) [30].

$$AUC = \int_0^1 ROC(t)dt \quad (13)$$

III. HASIL DAN PEMBAHASAN

Setelah membangun model klasifikasi emosi, maka penelitian ini akan dilanjutkan dengan menganalisis hasil pengujian model klasifikasi berdasarkan berbagai skenario eksperimen yang telah dilakukan. Pembahasan tersebut akan mencakup hasil dari *pre-processing* data, hasil penyeimbangan data, proses klasifikasi menggunakan *Random forest*, analisis *hyperparameter tuning*, serta evaluasi terhadap performa model berdasarkan metrik pengujian yang digunakan. Analisis terkait tahapan yang telah dilakukan bertujuan untuk mengetahui pengaruh dari berbagai penerapan metode terhadap performa model klasifikasi.

A. Hasil Pre-processing

Tahap *Pre-processing* dilakukan untuk meningkatkan kualitas dataset yang nantinya digunakan untuk proses pelatihan dan pengujian model. Setiap tahapan *pre-processing* berguna untuk menghasilkan data yang lebih terstruktur dan mengurangi *noise*. Pada penelitian ini, proses *pre-processing* yang dilakukan meliputi penghapusan data duplikat, *case folding*, penghapusan simbol/tanda baca, penghapusan angka, penghapusan spasi berlebih, normalisasi karakter berulang, tokenisasi, dan *stemming*. Hasil dari penerapan tahap *pre-processing* pada dataset penelitian dapat terlihat pada Tabel III.

TABEL III
HASIL *PRE-PROCESSING*

<i>Pre-processing</i>	Hasil
Data Awal	Seru juga.. Tapi perlu ditingkatkan lgi di bagian arena perang.. Di perlebar dan bisa di zoom juga.. Terima kasih..
<i>Case Folding</i>	seru juga.. tapi perlu ditingkatkan lgi di bagian arena perang.. di perlebar dan bisa di zoom juga.. terima kasih..
Penghapusan Simbol/Tanda Baca	seru juga tapi perlu ditingkatkan lgi di bagian arena perang di perlebar dan bisa di zoom juga terima kasih
Penghapusan	seru juga tapi perlu ditingkatkan lgi di

Spasi Berlebih	bagian arena perang di diperlebar dan bisa di zoom juga terima kasih
Tokenisasi	['seru', 'juga', 'tapi', 'perlu', 'ditingkatkan', 'lgi', 'di', 'bagian', 'arena', 'perang', 'di', 'perlebar', 'dan', 'bisa', 'di', 'zoom', 'juga', 'terima', 'kasih']
Stemming	['seru', 'juga', 'tapi', 'perlu', 'tingkat', 'lgi', 'di', 'bagi', 'arena', 'perang', 'di', 'lebar', 'dan', 'bisa', 'di', 'zoom', 'juga', 'terima', 'kasih']
Hasil Akhir	seru juga tapi perlu tingkat lgi di bagi arena perang di lebar dan bisa di zoom juga terima kasih

B. Label Encoding

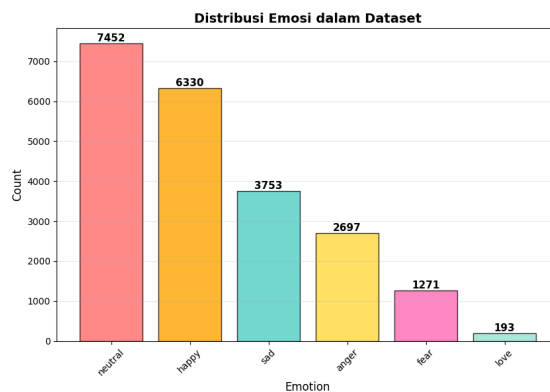
Tahap *label encoding* dilakukan dengan mengubah label 'emotion' dalam bentuk teks menjadi bentuk numerik melalui pemetaan setiap emosi ke dalam nilai tertentu secara berurutan. Tahapan *label encoding* akan menghasilkan kolom baru bernama 'label' yang berisi nilai numerik dari setiap emosi yang dapat digunakan pada tahap pelatihan model. Penerapan *label encoding* pada penelitian ini dapat terlihat pada Tabel III.

TABEL III
LABEL ENCODING

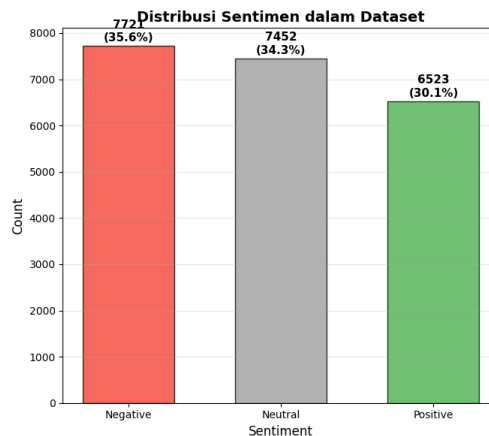
Label	Emotion
0	Anger
1	Fear
2	Sad
3	Happy
4	Love
5	Neutral

C. Hasil Penyeimbangan Data

Dataset pada penelitian ini merupakan dataset yang tidak seimbang pada atribut 'emotion'. Ketidakseimbangan tersebut mengakibatkan model cenderung untuk memprediksi pada kelas mayoritas dan mengurangi kemampuan model dalam melakukan prediksi pada kelas minoritas. Gambar 2 dan Gambar 3 memperlihatkan distribusi data emosi dan sentimen yang digunakan pada dataset penelitian. Berdasarkan distribusi data emosi yang ditunjukkan pada Gambar 2, jumlah data terbesar dimiliki oleh kelas *Neutral* sebanyak 7.452 ulasan sedangkan pada kelas *Fear* hanya sekitar 1.271 ulasan yang menjadikan kelas *Fear* sebagai kelas dengan jumlah data paling rendah. Distribusi tersebut menunjukkan adanya ketimpangan jumlah data yang cukup signifikan antar setiap kelas sehingga perlu penerapan teknik khusus dalam penyeimbangan data seperti *SMOTE*, *Tomek-Links*, dan *SMOTE-Tomek* untuk dapat memberikan hasil klasifikasi yang lebih optimal.



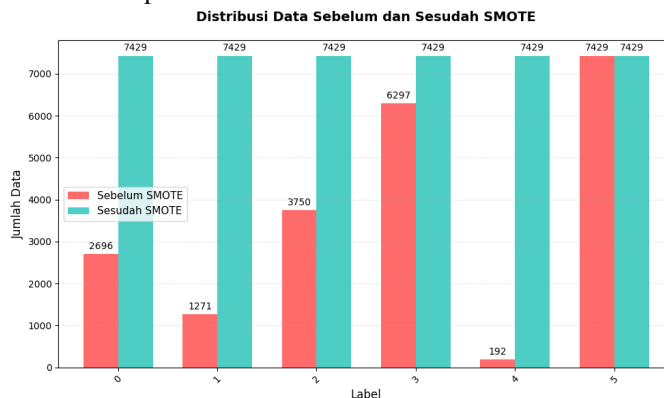
Gbr. 2 Distribusi Data Emosi.



Gbr. 3 Distribusi Data Sentimen

1) SMOTE

Penerapan teknik penyeimbangan data dengan *SMOTE* akan menambah data sintesis baru pada kelas minoritas. Pada Gambar 4, menunjukkan perbandingan distribusi data sebelum dan sesudah penerapan metode *SMOTE* pada setiap label emosi. Berdasarkan perbandingan distribusi tersebut dapat terlihat bahwa seluruh kelas pada atribut 'emotion' menjadi lebih seimbang dengan total 7.429 data pada setiap kelasnya. Distribusi data tersebut menunjukkan bahwa penerapan metode *oversampling* berjalan secara optimal dan mampu mengatasi ketidakseimbangan data dengan menghasilkan data sintesis baru pada kelas minoritas.

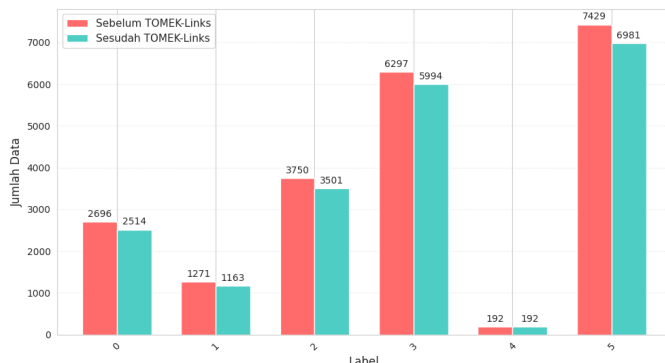


Gbr. 4 Distribusi Data dengan SMOTE

2) Tomek-Links

Tomek-Links pada penerapannya akan mengurangi data dari setiap pasangan sampel kelas berbeda yang saling tumpang tindih untuk memperjelas batas pemisahan antar kelas. Berdasarkan perbandingan distribusi data dengan penerapan undersampling seperti pada Gambar 5, terdapat pengurangan data pada label 5 atau *Neutral* yang awalnya sebesar 7.429 data menjadi 6.981 data. Hal ini berbeda dengan kelas minoritas seperti pada label 4 atau *Love* yang tidak mengalami perubahan data. Distribusi tersebut menunjukkan bahwa penerapan metode *Tomek-Links* akan menjadikan model lebih adil terhadap kelas minoritas tanpa perlu membuat data sintetis baru pada kelas mayoritas.

Distribusi Data Sebelum dan Sesudah TOMEK-Links

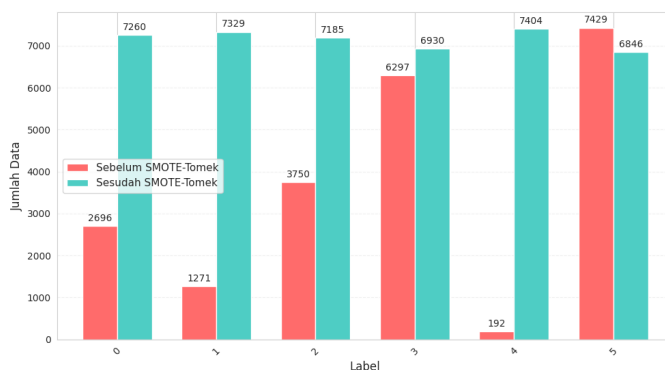


Gbr. 5 Distribusi Data dengan Tomek-Links

3) SMOTE-Tomek

Penerapan *SMOTE-Tomek* sebagai teknik penyeimbangan data akan menggabungkan metode *SMOTE* dan *Tomek-Links*. Penerapan metode *SMOTE-Tomek* terlihat pada Gambar 6 yang menampilkan perbandingan distribusi data sebelum dan sesudah penerapan *SMOTE-Tomek* pada setiap kelas. Berdasarkan Gambar 6, dapat ditunjukkan bahwa distribusi data menjadi lebih seimbang dengan rentang 7.200 hingga 7.400 data pada setiap kelas. Pada label 0 hingga 4 mengalami penambahan data sintesis melalui *SMOTE* sedangkan pada label 5 mengalami pengurangan data melalui *Tomek-Links*. Distribusi tersebut menggambarkan bahwa penerapan *SMOTE-Tomek* mampu menghasikan distribusi yang lebih proporsional sehingga dapat digunakan dalam proses pelatihan model.

Distribusi Data Sebelum dan Sesudah SMOTE-Tomek



Gbr. 6 Distribusi Data dengan SMOTE-Tomek

D. Hasil Hyperparameter Tuning

Tahap *hyperparameter tuning* bertujuan guna mendapatkan konfigurasi parameter terbaik dalam meningkatkan performa klasifikasi model. *Hyperparameter tuning* berupa *Particle Swarm Optimization (PSO)* digunakan pada penelitian ini karena metode tersebut mampu melakukan pencarian solusi berbasis populasi. Konfigurasi parameter terbaik menggunakan metode *Particle Swarm Optimization (PSO)* dalam penelitian ini pada algoritma *Random forest* dapat terlihat pada Tabel IV.

TABEL IVV
HASIL PARAMETER TERBAIK

Parameter	Nilai
'n_estimators'	200
'max_depth'	None
'min_sample_split'	5
'min_sample_leaf'	1
'max_features'	Log2
'iters'	10
'n_particles'	10

Berdasarkan Tabel IV, dapat terlihat bahwa penerapan *hyperparameter tuning* dengan *PSO* mampu meningkatkan performa model klasifikasi melalui kombinasi parameter yang optimal. Hasil kombinasi tersebut menggambarkan bahwa proses pencarian parameter dengan menggunakan metode *PSO* mampu memberikan peningkatan dalam kualitas model dalam proses klasifikasi menggunakan *Random forest*.

E. Evaluasi Model

Tahap evaluasi model pada penelitian ini bertujuan untuk mengetahui kemampuan dari algoritma *Random Forest* dalam melakukan klasifikasi emosi melalui penerapan berbagai teknik penyeimbangan data. Pengujian akan dilakukan menggunakan beberapa rasio pembagian data latih dan data uji seperti 95:5, 90:10, 85:15, 80:20, dan 70:30. Hasil dari evaluasi tersebut akan melihat pengaruh dari penerapan teknik penyeimbangan data (*sampling*) menggunakan *SMOTE*, *Tomek-Links*, dan *SMOTE-Tomek* terhadap performa klasifikasi yang dihasilkan.

1) Random Forest tanpa Sampling

TABEL V
RANDOM FOREST TANPA SAMPLING

Test:Train	Accuracy	Precision	Recall	F1-Score
95:5	61,92%	62,47%	61,92%	59,98%
90:10	62,11%	61,91%	62,11%	60,17%
85:15	62,29%	61,83%	62,29%	60,44%
80:20	61,77%	61,35%	61,77%	60,08%
70:30	61,73%	61,16%	61,73%	59,72%

Hasil evaluasi algoritma *Random Forest* tanpa menggunakan metode *sampling* dapat terlihat pada Tabel V. Hasil tersebut menunjukkan bahwa rasio dari data latih dan data uji berpengaruh terhadap kemampuan model dalam mempelajari pola pada data yang digunakan. Berdasarkan Tabel V, rasio pembagian data 85:15 merupakan model

terbaik dengan nilai *accuracy* sebesar 62,29%, *precision* 61,83%, *recall* 62,29%, dan *F1-score* 60,44% dibandingkan dengan rasio pembagian data lainnya. Namun, performa yang dihasilkan dari model ini masih tergolong rendah karena tidak menerapkan metode *sampling* sehingga model cenderung mengenali kelas mayoritas dibandingkan kelas minoritas.

2) Random Forest dengan SMOTE

TABEL VI
RANDOM FOREST DENGAN SMOTE

Test:Train	Accuracy	Precision	Recall	F1-Score
95:5	85,24%	85,32%	85,24%	85,20%
90:10	84,90%	84,87%	84,90%	84,85%
85:15	83,83%	83,89%	83,83%	83,81%
80:20	83,68%	83,72%	83,68%	83,66%
70:30	82,41%	82,52%	82,41%	82,41%

Tabel VI menampilkan hasil evaluasi *Random Forest* menggunakan metode *sampling* berupa *SMOTE*. Pada Tabel VI dapat terlihat adanya peningkatan terhadap performa model yang dihasilkan pada seluruh rasio pembagian data jika dibandingkan dengan model sebelumnya pada Tabel V. Selain itu, terdapat pola baru yang dihasilkan yaitu semakin besar rasio pada data latih untuk melatih model maka performa model yang dihasilkan cenderung semakin tinggi dan sebaliknya. Menurut Tabel VI, rasio pembagian data berupa 95:5 merupakan model terbaik dengan nilai *accuracy* sebesar 85,24%, *precision* 85,32%, *recall* 85,24% dan *F1-score* 85,20%.

3) Random Forest dengan Tomek-Links

TABEL VII
RANDOM FOREST DENGAN TOMEK-LINKS

Test:Train	Accuracy	Precision	Recall	F1-Score
95:5	63,06%	63,86%	63,06%	60,84%
90:10	63,39%	63,49%	63,39%	60,94%
85:15	62,78%	62,53%	62,78%	60,34%
80:20	62,72%	62,68%	62,72%	60,35%
70:30	62,81%	62,95%	62,81%	60,29%

Hasil evaluasi *Random Forest* menggunakan metode *sampling* berupa *Tomek-Links* dapat terlihat pada Tabel VII. Tabel tersebut menunjukkan bahwa terdapat sedikit peningkatan pada performa klasifikasi dibandingkan dengan model tanpa menggunakan *sampling* tambahan seperti pada Tabel V. Sementara itu, hasil dari setiap rasio pembagian data pada Tabel VII cenderung stabil dan tidak menunjukkan adanya peningkatan yang signifikan atau dapat dikatakan relatif stagnan. Berdasarkan tabel tersebut, model dengan performa terbaik dimiliki oleh rasio pembagian data 90:10 dengan nilai *accuracy* sebesar 63,39%, *precision* 63,49%, *recall* 63,39%, dan *F1-score* 60,94%.

4) Random Forest dengan SMOTE-Tomek

TABEL VIII
RANDOM FOREST DENGAN SMOTE-TOMEK

Test:Train	Accuracy	Precision	Recall	F1-Score
95:5	84,96%	84,98%	84,96%	84,92%
90:10	84,56%	84,59%	84,56%	84,52%
85:15	84,56%	84,59%	84,56%	84,52%
80:20	83,73%	83,77%	83,73%	83,71%
70:30	81,83%	81,83%	81,83%	81,76%

Tabel VIII memperlihatkan hasil evaluasi *Random Forest* menggunakan metode *sampling* berupa *SMOTE-Tomek*. Tabel tersebut menunjukkan adanya peningkatan performa pada model klasifikasi dibandingkan pada model sebelumnya yang tanpa menerapkan metode *sampling*. Di samping itu, terdapat pola yang serupa dengan model pada Tabel VI yang memperlihatkan bahwa performa model akan semakin meningkat seiring dengan besarnya rasio pada data latih untuk melatih model dan performa model cenderung menurun jika data latih yang digunakan semakin sedikit. Menurut Tabel VII, model dengan performa terbaik dimiliki oleh rasio pembagian data 95:5 yang menghasilkan nilai *accuracy* sebesar 84,96%, *precision* 84,98%, *recall* 84,96%, dan *F1-score* 84,92%.

5) Perbandingan Metode Sampling

TABEL IX
PERBANDINGAN METODE SAMPLING

Metode Sampling	Test:Train	Accuracy	Precision	Recall	F1-Score
Tanpa Sampling	85:15	62,29%	61,83%	62,29%	60,44%
SMOTE	95:5	85,24%	85,32%	85,24%	85,20%
Tomek-Links	90:10	63,39%	63,49%	63,39%	60,94%
SMOTE-Tomek	95:5	84,96%	84,98%	84,96%	84,92%

Tabel IX memperlihatkan komparasi hasil evaluasi *Random Forest* dengan penerapan berbagai metode *sampling* yang telah didapatkan sebelumnya. Secara keseluruhan, penerapan metode *sampling* pada penelitian ini mampu memberikan peningkatan terhadap performa klasifikasi model *Random Forest*. Menurut Tabel IX, model dengan performa terbaik dimiliki oleh model *Random Forest* dengan penerapan metode *sampling* berupa *SMOTE* pada rasio pembagian data 95:5. Nilai *accuracy* yang didapatkan dari model terbaik tersebut sebesar 85,24% dengan nilai *precision* dan *recall* masing-masing sebesar 85,32% dan 85,24% serta nilai *F1-score* sebesar 85,20%.

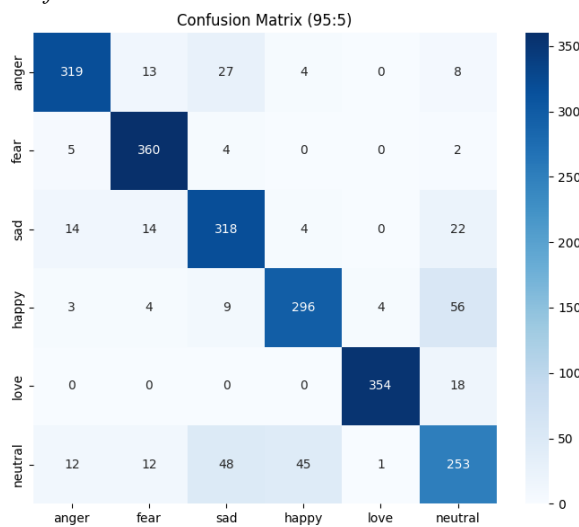
Penerapan *sampling* berupa *SMOTE* didasarkan pada hasil dari Tabel IX, mampu meningkatkan kemampuan model dalam klasifikasi emosi dengan menambah data sintesis baru pada kelas minoritas sehingga distribusi data menjadi lebih seimbang. Selain itu, penerapan *Tomek-Links* nyatanya belum memberikan peningkatan performa yang signifikan terhadap model klasifikasi sedangkan penerapan *SMOTE-Tomek* telah berhasil menggabungkan proses *SMOTE* dan *Tomek-Links* dan memberikan kenaikan pada performa klasifikasi meskipun nilai yang didapatkan masih berada di bawah *SMOTE*. Berdasarkan hasil keseluruhan, dapat dikatakan bahwa peningkatan performa klasifikasi pada penelitian ini

didapatkan melalui penerapan metode sampling tambahan berupa SMOTE.

F. Analisis Model

Setelah didapatkan model terbaik, maka tahap selanjutnya yaitu tahap analisis terhadap kinerja model yang telah dibangun untuk dapat mengukur kemampuan model dalam melakukan prediksi, khususnya prediksi pada kelas emosi secara tepat. Analisis yang dilakukan tidak hanya menitik beratkan nilai *accuracy* secara keseluruhan, tetapi juga analisis terhadap kemampuan model dalam mengenali emosi serta kesalahan klasifikasi yang kemungkinan muncul pada setiap kelasnya. Terdapat beberapa metrik pengukuran pada evaluasi performa model seperti *confusion matrix* dan *ROC-AUC* untuk merepresentasikan kualitas model yang dihasilkan. Metrik pengukuran tersebut diharapkan mampu memperdalam pemahaman terkait performa model yang telah dibangun.

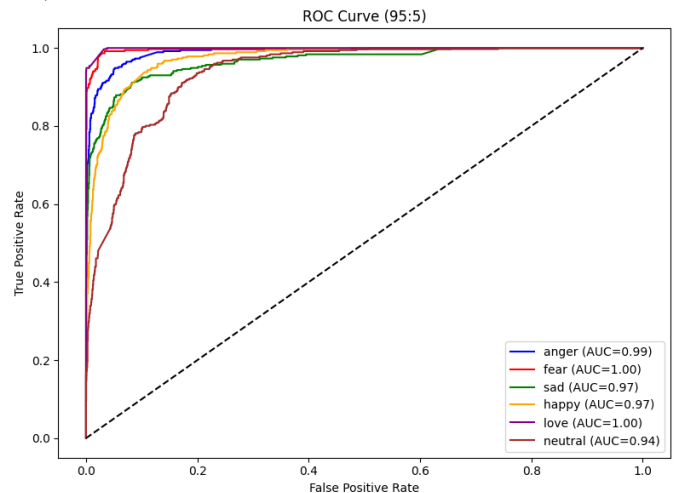
1) Confusion Matrix



Gbr. 7 Confusion Matrix Pada Rasio 95:5

Hasil *confusion matrix* yang ditampilkan pada Gambar 7 menunjukkan bahwa pada rasio pembagian 95:5 dengan metode *sampling SMOTE*, model mampu mengklasifikasikan setiap kelas emosi dengan baik. Hal ini dapat terlihat dari nilai pada sumbu diagonal yang menggambarkan hasil prediksi benar yang dilakukan oleh model klasifikasi pada setiap kelas. Kelas *fear* memiliki performa klasifikasi terbaik dengan total 360 data yang mampu diprediksi secara benar yang menunjukkan karakteristik data pada kelas ini cukup mudah untuk diprediksi oleh model. Kesalahan klasifikasi yang masih sering muncul terdapat pada kelas *neutral* yang hanya mampu diprediksi benar oleh model sebanyak 253 data dengan total kesalahan prediksi terbanyak pada kelas lainnya. Hal ini diduga terjadi karena kurangnya karakteristik data yang unggul pada kelas *neutral* sehingga model cenderung untuk memprediksi ke kelas lain. Secara keseluruhan, model *Random Forest* dengan metode *sampling SMOTE* mampu menghasilkan nilai *accuracy* tertinggi sebesar 85,24% yang menggambarkan kemampuan model yang baik dalam melakukan klasifikasi emosi.

2) ROC-AUC



Gbr. 8 ROC-AUC Pada Rasio 95:5

Pada Gambar 8, dapat terlihat bahwa model dengan rasio pembagian 95:5 dengan metode *sampling SMOTE* mampu menghasilkan performa klasifikasi pada seluruh kelas emosi yang ditandai dengan nilai *Area Under Curve (AUC)* yang tinggi pada setiap kelas emosi. Hasil tersebut ditunjukkan pada kelas *fear* dan *love* yang mendapatkan nilai *AUC* sebesar 100%, kelas *anger* memperoleh nilai *AUC* sebesar 99%, kelas *sad* dan *happy* dengan nilai *AUC* sebesar 97%, dan terakhir pada kelas *neutral* yang memiliki nilai *AUC* sebesar 94%. Nilai *AUC* dari setiap kelas menggambarkan kemampuan model yang sangat baik dalam mengidentifikasi setiap kelas emosi pada data ulasan pengguna. Total nilai rata-rata *macro AUC* yang didapatkan pada model tersebut yaitu sebesar 97,83%. Selain itu, kurva *Receiver Operating Characteristic (ROC)* yang terbentuk menunjukkan seluruh kelas terletak mendekati sudut kiri atas dan berada jauh dari batas garis diagonal. Hasil evaluasi yang didapatkan melalui tingginya nilai *AUC* dan kurva *ROC* menandakan model *Random Forest* dengan penerapan *TF-IDF*, metode *sampling SMOTE*, dan *hyperparameter tuning PSO* mampu melakukan klasifikasi emosi pada data yang tidak seimbang dengan kemampuan diskriminasi kelas yang sangat baik.

IV. KESIMPULAN

Berdasarkan penelitian yang telah dilakukan, dapat diketahui bahwa model *Random Forest* yang dibangun telah berhasil melakukan klasifikasi emosi pada dataset yang tidak seimbang melalui berbagai tahapan seperti *pre-processing* teks, penerapan ekstraksi fitur menggunakan *TF-IDF*, penerapan metode *sampling* seperti *SMOTE*, *Tomek-Links*, dan *SMOTE-Tomek*, serta penerapan *hyperparameter tuning* menggunakan *Particle Swarm Optimization (PSO)*. Mengacu pada eksplorasi yang telah dilakukan, dapat ditarik kesimpulan bahwa model dengan metode *sampling SMOTE* memberikan hasil klasifikasi yang paling optimal. Hal ini terlihat pada

rasio pembagian data 95:5 yang memberikan performa terbaik dengan nilai *accuracy* dan *recall* masing-masing sebesar 85,24%, *precision* 85,32% dan nilai *F1-score* sebesar 85,20%. *Kurva ROC* yang dihasilkan pada model terbaik tersebut juga menunjukkan bahwa model memiliki kemampuan dalam membedakan setiap kelas emosi dengan sangat baik yang terlihat pada nilai rata-rata *macro AUC* sebesar 97,83%. Temuan ini menunjukkan bahwa penggunaan metode *sampling* berupa *SMOTE* dengan penambahan ekstraksi fitur *TF-IDF* dan *hyperparameter tuning PSO* terbukti efektif dalam menangani permasalahan ketidakseimbangan data dan mampu meningkatkan kinerja model secara keseluruhan.

UCAPAN TERIMA KASIH

Penulis panjatkan puji dan syukur kehadirat Tuhan Yang Maha Esa karena atas rahmat dan karunia-Nya penelitian ini dapat terselesaikan dengan baik. Selain itu, penulis sampaikan terima kasih kepada orang tua yang senantiasa memberikan dukungan, doa, dan motivasi selama proses studi dan penelitian ini berlangsung. Tak lupa juga, penulis sampaikan ucapan terima kasih kepada dosen pembimbing dan seluruh staff yang telah memberikan bimbingan, arahan, dan masukan selama proses penyelesaian penelitian ini dari awal hingga akhir. Terakhir, penulis ucapkan terima kasih kepada seluruh teman-teman yang telah memberikan dukungan dan kontribusi baik secara langsung maupun tidak langsung selama pelaksanaan penelitian ini. Dukungan yang telah diberikan dari berbagai pihak sangat membantu penulis untuk dapat menyelesaikan penelitian ini dengan baik.

REFERENSI

- [1] Riccosan and K. E. Saputra, "Multilabel multiclass sentiment and emotion dataset from Indonesian mobile application review," *Data Br.*, vol. 50, p. 109576, Oct. 2023, doi: 10.1016/J.DIB.2023.109576.
- [2] Y. Yamasari, N. Isaena Raharko, A. Qoiriah, E. Irawati Setiawan, W. Yustanti, and R. E. Putra, "Exploring Naive Bayes Variants for Classifying Emotions in e-Commerce Reviews," *Proc. Int. Conf. Comput. Eng. Netw. Intell. Multimedia, CENIM 2024*, 2024, doi: 10.1109/CENIM64038.2024.10882763.
- [3] T. G.S., Y. Hariprasad, S. S. Iyengar, N. R. Sunitha, P. Badrinath, and S. Chennupati, "An extension of Synthetic Minority Oversampling Technique based on Kalman filter for imbalanced datasets," *Mach. Learn. with Appl.*, vol. 8, p. 100267, Jun. 2022, doi: 10.1016/J.MLWA.2022.100267.
- [4] E. F.; Swana *et al.*, "Tomek Link and SMOTE Approaches for Machine Fault Classification with an Imbalanced Dataset," *Sensors 2022, Vol. 22, Page 3246*, vol. 22, no. 9, p. 3246, Apr. 2022, doi: 10.3390/S22093246.
- [5] D. Shabrina Assyifa and A. Luthfiarta, "SMOTE-Tomek Resampling Based on Random Forest Method to Overcome Unbalanced Data for Multi-class Classification," *Inf. J. Ilm. Bid. Teknol. Inf. dan Komun.*, vol. 9, no. 2, pp. 151–160, Jul. 2024, doi: 10.25139/INFORM.V9I2.8410.
- [6] D. Tiwari *et al.*, "SPSO-EFVM: A Particle Swarm Optimization-Based Ensemble Fusion Voting Model for Sentence-Level Sentiment Analysis," *IEEE Access*, vol. 12, pp. 23707–23724, 2024, doi: 10.1109/ACCESS.2024.3363158.
- [7] C. Fan, M. Chen, X. Wang, J. Wang, and B. Huang, "A Review on Data Preprocessing Techniques Toward Efficient and Reliable Knowledge Discovery From Building Operational Data," *Front. Energy Res.*, vol. 9, p. 652801, Mar. 2021, doi: 10.3389/FENRG.2021.652801/BIBTEX.
- [8] S. Suswadi and M. Erkamim, "Sentiment Analysis of Shopee App Reviews Using Random Forest and Support Vector Machine," *Ilk. J. Ilm.*, vol. 15, no. 3, pp. 427–435, Dec. 2023, doi: 10.33096/ilkom.v15i3.1610.427-435.
- [9] A. Petukhova and N. Fachada, "TextCL: A Python package for NLP preprocessing tasks," *SoftwareX*, vol. 19, p. 101122, Jul. 2022, doi: 10.1016/J.SOFTX.2022.101122.
- [10] M. A. Rosid, A. S. Fitriani, I. R. I. Astutik, N. I. Mulloh, and H. A. Gozali, "Improving Text Preprocessing For Student Complaint Document Classification Using Sastrawi," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 874, no. 1, p. 012017, Jun. 2020, doi: 10.1088/1757-899X/874/1/012017.
- [11] R. Haque, N. Islam, M. Tasneem, and A. K. Das, "Multi-class sentiment classification on Bengali social media comments using machine learning," *Int. J. Cogn. Comput. Eng.*, vol. 4, pp. 21–35, Jun. 2023, doi: 10.1016/J.IJCC.2023.01.001.
- [12] H. T. Duong and T. A. Nguyen-Thi, "A review: preprocessing techniques and data augmentation for sentiment analysis," *Comput. Soc. Networks*, vol. 8, no. 1, pp. 1–16, Dec. 2021, doi: 10.1186/S40649-020-00080-X/FIGURES/4.
- [13] M. A. Palomino and F. Aider, "Evaluating the Effectiveness of Text Pre-Processing in Sentiment Analysis," *Appl. Sci. 2022, Vol. 12, Page 8765*, vol. 12, no. 17, p. 8765, Aug. 2022, doi: 10.3390/AP12178765.
- [14] Z. Mansur, N. Omar, S. Tiun, and E. M. Alshari, "A normalization model for repeated letters in social media hate speech text based on rules and spelling correction," *PLoS One*, vol. 19, no. 3, Mar. 2024, doi: 10.1371/JOURNAL.PONE.0299652.
- [15] M. R. Kurniawan, O. N. Pratiwi, and F. Hamami, "KLASIFIKASI SOAL MENGGUNAKAN MULTI-LABEL PROBLEM TRANSFORMATION DENGAN METODE RANDOM FOREST DAN K-NEAREST NEIGHBOR," *JUPI (Jurnal Ilm. Penelit. dan Pembelajaran Inform.)*, vol. 10, no. 1, pp. 367–380, Jan. 2025, doi: 10.29100/JUPI.V10I1.5910.
- [16] M. Y. Khan, A. Qayoom, M. S. Nizami, M. S. Siddiqui, S. Wasi, and S. M. K. U. R. Raazi, "Automated Prediction of Good Dictionary EXamples (GDEX): A Comprehensive Experiment with Distant Supervision, Machine Learning, and Word Embedding-Based Deep Learning Techniques," *Complexity*, vol. 2021, 2021, doi: 10.1155/2021/2553199.
- [17] L. D. Cahya, A. Luthfiarta, J. I. T. Krisna, S. Winarno, and A. Nugraha, "Improving Multi-label Classification Performance on Imbalanced Datasets Through SMOTE Technique and Data Augmentation Using IndoBERT Model," *J. Nas. Teknol. dan Sist. Inf.*, vol. 9, no. 3, pp. 290–298, Jan. 2024, doi: 10.25077/TEKNOSI.V9I3.2023.290-298.
- [18] E. Deniz, H. Erbay, and M. Coşar, "Multi-Label Classification of E-Commerce Customer Reviews via Machine Learning," *Axioms 2022, Vol. 11, Page 436*, vol. 11, no. 9, p. 436, Aug. 2022, doi: 10.3390/AXIOMS11090436.
- [19] K. M. Hasib *et al.*, "A Survey of Methods for Managing the Classification and Solution of Data Imbalance Problem," *J. Comput. Sci.*, vol. 16, no. 11, pp. 1546–1557, Nov. 2020, doi: 10.3844/JCSSP.2020.1546.1557.
- [20] C. M. Yeşilkanat and S. Akkoyun, "SMOTE-based data augmentation for accurate classification of neutron halo nuclei," *Knowledge-Based Syst.*, vol. 318, Jun. 2025, doi: 10.1016/J.KNOSYS.2025.113580.
- [21] N. Katsikis *et al.*, "Comparative Analysis of Resampling Techniques for Class Imbalance in Financial Distress Prediction Using XGBoost," *Math. 2025, Vol. 13, Page 2186*, vol. 13, no. 13, p. 2186, Jul. 2025, doi: 10.3390/MATH13132186.
- [22] W. Chandra, B. Suprihatin, Y. Resti, J. A. Jenderal Yani No, and S. Selatan, "Median-KNN Regressor-SMOTE-Tomek Links for Handling Missing and Imbalanced Data in Air Quality Prediction," *Symmetry 2023, Vol. 15, Page 887*, vol. 15, no. 4, p. 887, Apr. 2023, doi: 10.3390/SYM15040887.
- [23] N. Katsikis *et al.*, "Comparative Analysis of Resampling Techniques for Class Imbalance in Financial Distress Prediction Using XGBoost," *Math. 2025, Vol. 13, Page 2186*, vol. 13, no. 13, p. 2186, Jul. 2025, doi: 10.3390/MATH13132186.
- [24] J. Tao and X. Fang, "Toward multi-label sentiment analysis: a transfer learning based approach," *J. Big Data*, vol. 7, no. 1, pp. 1–26, Dec. 2020, doi: 10.1186/S40537-019-0278-0/TABLES/13.

- [25] A. Sethi and D. Kataria, "Analyzing Emergent Complexity in Particle Swarm Optimization using a Rolling Technique for Updating Hyperparameter Coefficients," *Procedia Comput. Sci.*, vol. 193, pp. 513–523, Jan. 2021, doi: 10.1016/J.PROCS.2021.10.053.
- [26] A. G. Gad, "Particle Swarm Optimization Algorithm and Its Applications: A Systematic Review," *Arch. Comput. Methods Eng.* 2022 295, vol. 29, no. 5, pp. 2531–2561, Apr. 2022, doi: 10.1007/S11831-021-09694-4.
- [27] K. Guo, X. Wan, L. Liu, Z. Gao, and M. Yang, "Fault Diagnosis of Intelligent Production Line Based on Digital Twin and Improved Random Forest," *Appl. Sci.* 2021, Vol. 11, Page 7733, vol. 11, no. 16, p. 7733, Aug. 2021, doi: 10.3390/APP11167733.
- [28] T. A. U. Azmi, L. Hakim, D. C. R. Novitasari, and W. D. U. D. Utami, "Application Random Forest Method for Sentiment Analysis in Jamsostek Mobile Review," *Telematika*, vol. 20, no. 1, p. 117, Mar. 2023, doi: 10.31315/TELEMATIKA.V20I1.8868.
- [29] K. Namdar, M. A. Haider, and F. Khalvati, "A Modified AUC for Training Convolutional Neural Networks: Taking Confidence Into Account," *Front. Artif. Intell.*, vol. 4, p. 582928, Nov. 2021, doi: 10.3389/FRAI.2021.582928/TEXT.
- [30] J. Sun, C. Tang, W. Xie, and X. H. Zhou, "Nonparametric receiver operating characteristic curve analysis with an imperfect gold standard," *Biometrics*, vol. 80, no. 3, Jul. 2024, doi: 10.1093/BIOMTC/UJAE063.