JIKOMTI:
Jurnal Ilmiah Ilmu Komputer dan Teknologi Informasi
e- ISSN: 3089-2996
Vol. 1 No. 2, Mei 2025

# Optimasi Klasifikasi Data Teks Menggunakan Algoritma Logistic Regression dengan TF-IDF dan SMOTE

Angge Firizkiansah[1], Ali Muhammad[2], Imron Rizki Maulana[3]
[1,2,3]Universitas Sains Indonesia, Kabupaten Bekasi

E-mail:
angge.firizkiansah@lecturer.sains.ac.id[1](korespondensi), ali.muhammad@lecturer.sains.ac.id[2],
imron.rizki@sains.ac.id[3]

***Abstract***

*Text classification using machine learning is a key area within Natural Language Processing (NLP), aimed at automatically categorizing textual data into predefined classes. This technique is widely applied in fields such as sentiment analysis, topic detection, and spam filtering. Effective classification relies heavily on thorough preprocessing, as text data is inherently unstructured and must be transformed into a format suitable for machine learning algorithms. This study focuses on optimizing automatic text classification by employing logistic regression combined with TF-IDF feature extraction, and compares its performance against the use of SMOTE to handle class imbalance. Evaluation results indicate that logistic regression with TF-IDF alone achieves a higher accuracy of 76.9%, outperforming the model that incorporates SMOTE. The findings suggest that SMOTE does not enhance and may even diminish the accuracy of logistic regression in this text classification context.*

***Keywords****: logistic regression; text classification; text data; smote; tf-idf*

**Abstrak**

Klasifikasi teks menggunakan algoritma *machine learning* merupakan bagian dari cabang ilmu *Natural Language Processing* (NLP). Klasifikasi ini dilakukan untuk mengkategorisasikan data tekstual secara otomatis dalam sekumpulan kategori yang telah ditetapkan. Klasifikasi teks ini menjadi salah satu alat yang berguna di berbagai bidang, diantaranya analisis sentimen, deteksi topik, dan penyaringan spam. Pemodelan klasifikasi teks sangat dipengaruhi *preprocessing data* yang teliti. Data teks merupakan jenis data tidak terstruktur yang perlu diolah dengan mengubah dan mentransformasikan data teks melalui metode yang relevan, sehingga data teks berubah menjadi bentuk yang dapat dikenali oleh algoritma *machine learning* untuk dianalisis. Berdasarkan hal tersebut, penelitian ini bertujuan untuk mengoptimasi klasifikasi otomatis teks menggunakan algoritma *machine learning,* yaitu *logistic regression* dengan ekstraksi fitur TF-IDF dan dibandingkan dengan metode SMOTE untuk penanganan *imbalance class.* Berdasarkan hasil evaluasi model, diperoleh bahwa model *machine learning* algoritma *logistic regression* dengan ektraksi fitur TF-IDF menghasilkan tingkat akurasi yang lebih baik, yaitu sebesar 76,9% dibandingkan dengan model yang dilengkapi dengan SMOTE. Hal tersebut dapat disimpulkan bahwa metode SMOTE tidak mempengaruhi, bahkan menurunkan tingkat akurasi model algoritma *logistic regression* pada data teks yang menjadi domain dalam penelitian ini.

**Kata kunci**: data teks; klasifikasi teks; *logistic regression*; smote; tf-idf

## 1. PENDAHULUAN

Klasifikasi teks yang menjadi landasan dalam *Natural Language Processing* (NLP), merupakan kategorisasi otomatis data tekstual ke dalam serangkaian kelas yang telah ditentukan sebelumnya, dimana memainkan peran penting dalam mengatur dan mengekstrak informasi dari *volume* teks yang terus berkembang dan dihasilkan setiap hari [1]. Klasifikasi teks didefinisikan sebagai penetapan sekumpulan dokumen teks dalam satu atau lebih kategori/klasifikasi yang didasarkan pada konten dan semantiknya [2]. Klasifikasi teks merupakan alat serbaguna yang dapat diterapkan di berbagai domain, termasuk analisis sentimen, deteksi topik, dan penyaringan spam [3]. Meningkatnya relevansi klasifikasi teks berasal dari proliferasi data tekstual di berbagai sumber, termasuk konten web, umpan media sosial, ulasan pelanggan, maupun dokumen perusahaan [4], [5]. Sistem klasifikasi teks pada umumnya melibatkan tahapan, yaitu: *preprocessing*, *feature extraction*, *feature selection*, dan *classification* [6]. Pendekatan *Machine Learning* (ML) untuk klasifikasi teks secara luas dikategorikan dalam *supervised learning* dan *unsupervised learning* [7].

Dasar dari klasifikasi teks yang efektif terletak pada *preprocessing data* yang teliti, yaitu serangkaian transformasi yang diterapkan pada teks mentah untuk mengubahnya menjadi format yang terstruktur dan dapat dianalisis menggunakan algoritma ML Langkah-langkah *preprocessing data* ini, diantaranya seperti *tokenization*, *stop word removal*, dan *stemming*. Hal ini bertujuan untuk mengurangi *noise* dan meningkatkan nilai yang ada didalam teks [8]. *Tokenization* merupakan proses memecah teks menjadi kata atau *token* individu, yang berfungsi sebagai unit dasar untuk proses analisis [9]. *Stop word removal*, merupakan tahap untuk menghapus istilah yang sangat umum seperti 'the,' 'a,' dan 'is,' karena tidak mengandung makna untuk mendukung klasifikasi teks, serta frekuensi kemunculannya pada teks cenderung tinggi [9]. *Stemming* digunakan untuk mengurangi/menghapus kata berimbuhan ke bentuk dasarnya, menormalkan variasi, dan mengelompokkan istilah yang terkait bersama-sama. Ekstraksi fitur melibatkan transformasi teks yang telah melewati tahap *preprocessing data* menjadi representasi numerik yang menangkap esensi semantik dari teks tersebut, sehingga memungkinkan algoritma ML untuk mengenali pola dan hubungan dalam data [10]. Algoritma ML tradisional memiliki kecenderungan bergantung pada rekayasa fitur untuk mengidentifikasi karakteristik yang relevan dalam klasifikasi [11]. *Term Frequency-Invers Document Frequency* (TF-IDF) memberikan bobot pada kata-kata berdasarkan frekuensinya dalam sebuah dokumen dan frekuensi terbaliknya di seluruh korpus, dengan menekankan istilah yang umum dalam dokumen tertentu namun relatif jarang di seluruh kumpulan data [12].

*Imbalance class* di dalam suatu dataset, merupakan tantangan signifikan bagi algoritma ML, yang sering kali menghasilkan model bias terhadap kelas mayoritas dan menunjukkan kinerja yang buruk pada kelas minoritas. *Synthetic Minority Oversampling Technique* (SMOTE) adalah teknik *oversampling* yang mengatasi ketidakseimbangan kelas dengan mensintesis *instance* baru untuk kelas minoritas berdasarkan sampel yang ada, sehingga meningkatkan representasi kelas yang kurang terwakili dan memperbaiki generalisasi model [9]. SMOTE bekerja dengan memilih *instance* kelas minoritas dan membuat titik data sintetis di sepanjang segmen garis yang menghubungkan setiap *instance* kelas minoritas dengan k tetangga terdekatnya dari kelas yang sama [13]. Dengan menghasilkan sampel sintetis, SMOTE meningkatkan ukuran kelas minoritas dan mengurangi bias terhadap kelas mayoritas.

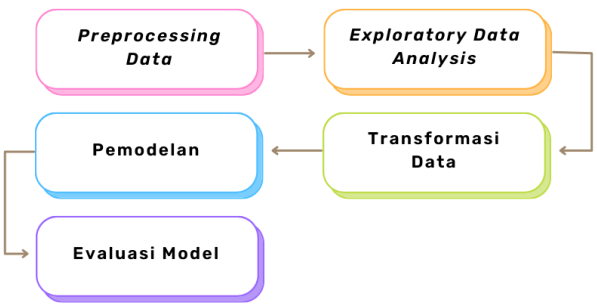
Tujuan penelitian ini adalah untuk menyelidiki optimasi kinerja klasifikasi teks melalui integrasi strategis TF-IDF sebagai metode ekstraksi fitur dan SMOTE untuk mengatasi kelas pada dataset yang tidak seimbang. Sedangkan *Logistic Regression* akan digunakan sebagai algoritma ML untuk melalukan klasifikasi terhadap dataset. Studi ini berupaya mengidentifikasi efek sinergis dari teknik-teknik tersebut dengan tujuan

membangun kerangka kerja yang teroptimasi untuk meningkatkan akurasi klasifikasi, khususnya dalam skenario yang ditandai oleh distribusi data yang tidak seimbang dan ruang fitur berdimensi tinggi, dengan tujuan akhir menyediakan metode yang kuat dan dapat diandalkan.

Beberapa hasil penelitian yang relevan untuk mendukung penelitian ini, diantaranya adalah implementasi algoritma *logistic regression* dengan ekstraksi fitur TF-IDF dalam mengklasifikasikan data teks berita diperoleh akurasi sebesar 97% [10]. Selanjutnya, klasifikasi data teks medis menggunakan *logistic regression* dan TF-IDF diperoleh akurasi sebesar 77,8% [14]. Kemudian, terdapat penelitian tentang penanganan *imbalance class* menggunakan TF-IDF dan SMOTE terhadap data teks komentar *toxic* media sosial menggunakan algoritma *logistic regression* diperoleh tingkat akurasi sebesar 94% [15]. Selain itu, penggunaan TF-IDF untuk ekstraksi fitur maupun SMOTE dalam menangani *imbalance class* pada penerapan berbagai algoritma *machine learning* menghasilkan tingkat akurasi yang baik [1][10][13][16].

## 2. METODE

Penelitian ini diawali *preprocessing* data, *exploratory data analysis*, transformasi data dengan TF-IDF dan SMOTE, serta selanjutnya dilakukan pemodelan klasifikasi menggunakan algoritma *logistic regression*. Kemudian hasil klasifikasi dievaluasi menggunakan *confusion matrix* untuk diketahui nilai *accuracy*, *precision*, *recall*, dan *F1-score*. Tahapan penelitian tersebut dapat digambarkan seperti gambar 1.




**Gambar 1.** Tahap Penelitian

### 2.1. *Preprocessing* Data

Penelitian ini menggunakan data dari repositori publik kaggle.com, yaitu data teks kesehatan mental. Dataset tersebut terdiri dari 53042 baris dan dua kolom (kolom *statement* dan status). Kolom *statement* berisikan data teks pernyataan kondisi personal, dan kolom status menggambarkan kategori kesehatan mental yang terbagi menjadi 7 (tujuh), yaitu normal, *depression*, *suicidal*, *anxiety*, *stress*, *bipolar*, dan *personality disorder*.

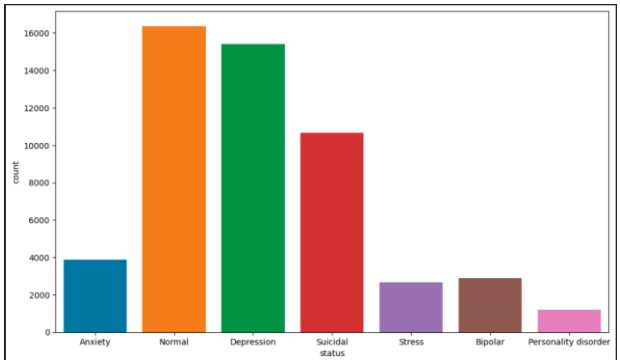
Selanjutnya, dilakukan *preprocessing*, sehingga dapat dianalisis menggunakan algoritma ML. *Preprocessing* data ini diawali dengan pembersihan data (*data cleaning*) dengan cara pengecekan spasi kosong (*white space*); duplikasi data; pembersihan tanda baca, karakter spesial, dan url yang tidak diperlukan; serta pengubahan setiap baris teks dengan huruf kecil (*lower case*) [17][18]. Hasil dari pembersihan data diperoleh dataset yang siap dianalisis seperti digambarkan sebagai berikut.

| | statement | status | cleaned_statement |
|---|---|---|---|
| 0 | oh my gosh | Anxiety | oh my gosh |
| 1 | trouble sleeping, confused mind, restless hear... | Anxiety | trouble sleeping confused mind restless heart ... |
| 2 | All wrong, back off dear, forward doubt. Stay ... | Anxiety | all wrong back off dear forward doubt stay in ... |
| 3 | I've shifted my focus to something else but I'... | Anxiety | ive shifted my focus to something else but im ... |
| 4 | I'm restless and restless, it's been a month n... | Anxiety | im restless and restless its been a month now ... |
| ... | ... | ... | ... |
| 53038 | Nobody takes me seriously I've (24M) dealt wit... | Anxiety | nobody takes me seriously i've dealt with depr... |
| 53039 | selfishness "I don't feel very good, it's lik... | Anxiety | selfishness i dont feel very good its like i d... |
| 53040 | Is there any way to sleep better? I can't slee... | Anxiety | is there any way to sleep better i cant sleep ... |

**Gambar 2.** Dataset Bersih

### 2.2. *Exploratory Data Analysis*

Pada tahap ini diawali dengan melihat sebaran kategori data yang menjadi target klasifikasi. Hal tersebut dapat dilihat sebagai berikut.




**Gambar 3.** Sebaran Kategori Data

JIKOMTI:
Jurnal Ilmiah Ilmu Komputer dan Teknologi Informasi
e- ISSN: 3089-2996
Vol. 1 No. 2, Mei 2025

Berdasarkan gambar tersebut, dapat diketahui bahwa jumlah data pada setiap kelas beragam, yaitu Normal = 16351 baris, *Depression* = 15404 baris, *Suicidal* = 10653 baris, *Anxiety* = 3888 baris, *Bipolar* = 2877 baris, *Stress* = 2669 baris, dan *Personality disorder* = 1201 baris.

Selanjutnya, data teks tersebut ditransformasikan menggunakan *stop word removal* dengan tujuan untuk menghapus kata-kata yang bersifat umum dan tidak memiliki makna dalam mendukung klasifikasi yang akan dilakukan oleh algoritma ML [28]. Berdasarkan hasil *exploratory data analysis*, dapat diketahui kata paling sering muncul dalam data teks yang dianalisis seperti data *wordcloud* yang dapat dilihat pada gambar 4.




**Gambar 4.** WordCloud

## 2.3. Transformasi Data

Berdasarkan hasil *preprocessing data* dan *exploratory data analysis*, selanjutnya data teks tersebut dilakukan tranformasi teks menjadi *array* atau vektor-vektor dalam bentuk numerik menggunakan TF-IDF. TF-IDF diterapkan untuk mengubah teks menjadi representasi numerik yang memiliki makna tertentu. *Term frequency* (TF) digunakan untuk menormalkan kemunculan setiap kata dengan ukuran dataset, sedangkan *inverse document frequency* (IDF) digunakan untuk menghilangkan kata-kata yang tidak banyak berkontribusi dalam menentukan makna kalimat. Oleh karena itu, jika suatu istilah muncul dalam teks secara umum, nilainya akan menjadi nol. Teknik TF-IDF mengeliminasi kata-kata umum sekaligus mengekstraksi fitur-fitur relevan dari korpus [10]. Fokus utama algoritma TF-IDF adalah pada kata yang memiliki frekuensi tinggi dalam teks, namun muncul dalam korpus dalam rentang yang lebih kecil. Kata tersebut kemudian memiliki kemampuan paling kuat untuk membedakan kelas-kelas yang berbeda dalam teks [10].

Secara matematis, persamaan untuk mengetahui nilai TF-IDF dapat dituliskan sebagai berikut [14].

$$TF - IDF = TF_{(t,doc)} \times IDF_{(t)} \quad (1)$$

Di mana TF(t, doc) merepresentasikan frekuensi kemunculan term t dalam setiap dokumen dan IDF(t) dapat dihitung berdasarkan frekuensi kata di seluruh korpus, seperti yang digambarkan pada persamaan matematika berikut [14].

$$IDF = log\left(\frac{|D|}{DF_{(t)}}\right) \quad (2)$$

|D| menggambarkan jumlah total dokumen dalam dataset, sedangkan DF(t) menjelaskan jumlah dokumen di mana term t muncul setidaknya satu kali [14].

Selain menggunakan TF-IDF, mengingat hasil *exploratory data analysis* diketahui bahwa jumlah data pada setiap kelas beragam. Berdasarkan hal tersebut, dapat diketahui bahwa jumlah data kelas terendah berjumlah 1201 baris data pada kelas *Personality Disorder* dan terbanyak pada kelas Normal, yaitu 16351 baris data. Hal tersebut dapat menyebabkan *imbalance class* dan dapata menimbulkan bias pada saat algoritma ML memprediksi klasifikasi data teks.

Oleh karena itu, dapat dilakukan transformasi data menggunakan SMOTE. Metode ini akan mengubah jumlah data pada setiap kelas, mengingat SMOTE akan membentuk data sintetis berdasarkan data yang sudah ada untuk menambahkan jumlah data pada kelas minoritas [9].

## 2.4. Pemodelan

*Logistic Regression* termasuk dalam algoritma *supervised learning*. Penggunaan algoritma ini semakin meningkat dalam beberapa waktu terakhir. Algoritma ini digunakan untuk mengklasifikasikan individu ke dalam kategori berdasarkan fungsi logistik [10]. Diketahui

beberapa kasus dimana grafik yang sempurna dan sesuai dengan semua titik data tidak dapat diperoleh. Untuk mengatasi hal tersebut, diperlukan pendekatan algoritma *logistic regression*. Hal tersebut dikarenakan *logistic regression* bekerja menggunakan fungsi sigmoid. Fungsi sigmoid ini diperoleh dari pengembangan persamaan matematika yang berawal dari persamaan linear regresi sederhana seperti berikut [10].

$$y = b_0 + b_1 \times x \quad (3)$$

Selanjutnya, fungsi sigmoid diimplementasikan ke dalamnya dengan persamaan matematika sebagai berikut [10].

$$p = \frac{1}{1-e^{-y}} \quad (4)$$

Kemudian, kedua persamaan tersebut dipadukan dan diperoleh persamaan berikut [10].

$$In\left(\frac{p}{1-p}\right) = b_0 + b_1 \times x \quad (5)$$

Atau

$$logit(S) = b_0 + b_1 M_1 + b_2 M_2 + \cdots b_k M_k \quad (6)$$

Dimana S merupakan probabilitas dari fitur yang menarik. $M_1$, $M_2$, $M_3$ ... $M_k$ merupakan nilai prediktor dan $b_0$, $b_1$, $b_2$, ... $b_k$ merupakan intersepsi model [10].

Beberapa hal yang merupakan asumsi dalam *logistic regression*, yaitu [10]:

1. Tidak ada hubungan linier antara variabel dependen dan independen dalam regresi logistik.
2. Variabel dependen tidak dapat dibagi menjadi dua bagian.
3. Variabel dependen tidak boleh berdistribusi normal, melainkan harus berhubungan secara linier.

### 2.5. Evaluasi Model

Pemodelan yang telah dilakukan menggunakan algoritma *logistic regression* perlu dievaluasi guna mengetahui seberapa baik algoritma tersebut dapat mengklasifikasikan data teks yang menjadi data latih dan data testing. Evaluasi ini dilakukan menggunakan *confusion matrix*. *Confusion matrix* menggambarkan nilai *accuracy, precision, recall,* dan *F1-score. Confusion matrix* digambarkan seperti pada tabel 1 berikut.

**Tabel 1.** *Confusion Matrix*

| | Predicted | |
|---|---|---|
| Actual | Positive | Negative |
| Positive | TP | FN |
| Negative | FP | TN |

Secara umum, *accuracy* menggambarkan nilai untuk mengetahui jumlah prediksi klasifikasi benar terhadap keseluruhan prediksi yang telah diketahui. Secara matematis, nilai *accuracy* dapat dihitung menggunakan persamaan sebagai berikut [14][17].

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

Sedangkan nilai *precision* menggambarkan seberapa banyak nilai prediksi pada *true positive* terhadap total prediksi *positive*. Nilai *precision* dapat dihitung menggunakan persamaan berikut [14][17].

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

Selanjutnya, metriks evaluasi yang tidak kalah penting, yaitu *recall*. Nilai *recall* menggambarkan seberapa banyak rasio prediksi pada *true positive* terhadap kelas aktual. *Recall* dapat dihitung menggunakan persamaan sebagai berkut [14][17].

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

Berikutnya nilai *F1-score* menggambarkan nilai rata-rata dari *precision* dan *recall*, serta berfungsi sebagai metrik komprehensif untuk menilai kinerja model ML. Persamaan matematika dari metrik *F1-score* dapat dilihat sebagai berikut [10].

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

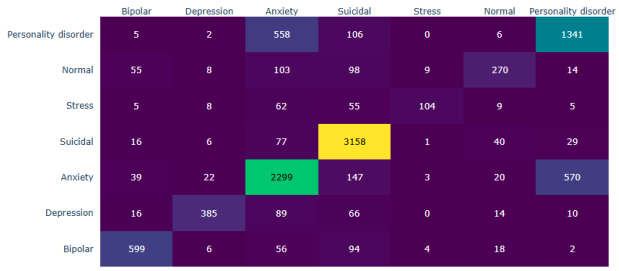
## 3. HASIL DAN PEMBAHASAN

Berdasarkan hasil penelitian yang telah dilakukan, diperoleh hasil evaluasi penerapan

algoritma *logistic regression* dalam mengklasifikasikan data teks dengan ekstraksi fitur TF-IDF dan penanganan *imbalance class* menggunakan SMOTE yang dijelaskan sebagai berikut.

### 3.1. Evaluasi Prediksi *Logistic Regression* dan TF-IDF

Berdasarkan hasil pemodelan yang telah dilakukan menggunakan algoritma *logistic regression* dengan ekstraksi fitur TF-IDF, diperoleh hasil evaluasi menggunakan diagram *confusion matrix* seperti dapat dilihat pada Gambar 5 berikut.




**Gambar 5.** *Confusion Matrix Logistic Regression*+TF-IDF

Berdasarkan diagram *confusion matrix* tersebut, dapat diketahui nilai evaluasi model menggunakan metrik *accuracy*, *precision*, *recall*, dan *F1-score* seperti dapat dilihat pada tabel 2 berikut.

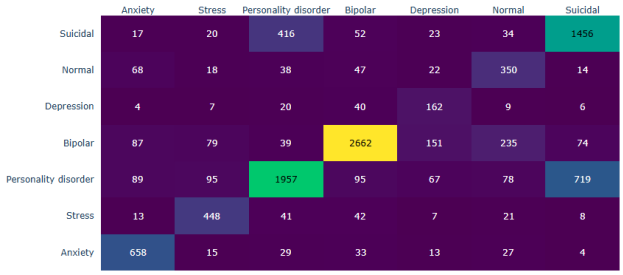
**Tabel 2.** Evaluasi Model *Logistic Regression*+TF-IDF

| Kelas | *Precision* | *Recall* | F1-score |
|---|---|---|---|
| Anxiety | 81% | 77% | 79% |
| Bipolar | 88% | 66% | 76% |
| Depression | 71% | 74% | 72% |
| Normal | 85% | 95% | 90% |
| Personality disorder | 86% | 42% | 56% |
| Stress | 72% | 48% | 58% |
| Suicidal | 68% | 66% | 67% |

Berdasarkan Tabel 2, dapat diketahui bahwa hasil evaluasi tertinggi dapat dilihat pada kelas Normal, mengingat jumlah data pada kelas tersebut adalah tertinggi. Secara keseluruhan nilai *accuracy* terhadap hasil klasifikasi menggunakan algoritma *logistic regression* dengan ekstraksi fitur TF-IDF diperoleh sebesar 76,9%.

### 3.2. Evaluasi Prediksi *Logistic Regression*, TF-IDF, dan SMOTE

Selanjutnya, penelitian ini membandingkan hasil pemodelan algoritma *logistic regression* dengan ekstraksi fitur TF-IDF yang didukung dengan penanganan *imbalance class* menggunakan metode SMOTE. Berdasarkan hasil pemodelan yang telah dilakukan, diperoleh hasil evaluasi menggunakan diagram *confusion matrix* seperti dapat dilihat pada Gambar 5 berikut.




**Gambar 6.** Confusion Matrix *Logistic Regression*+TF-IDF+SMOTE

Berdasarkan diagram *confusion matrix* tersebut, dapat diketahui nilai evaluasi model menggunakan metrik *accuracy*, *precision*, *recall*, dan *F1-score* seperti dapat dilihat pada tabel 3 berikut.

**Tabel 3.** Evaluasi Model *Logistic Regression*+TF-IDF+SMOTE

| Kelas | *Precision* | *Recall* | F1-score |
|---|---|---|---|
| Anxiety | 70% | 84% | 77% |
| Bipolar | 66% | 77% | 71% |
| Depression | 77% | 63% | 69% |
| Normal | 90% | 80% | 85% |
| Personality disorder | 36% | 65% | 47% |
| Stress | 46% | 63% | 53% |
| Suicidal | 64% | 72% | 68% |

Berdasarkan Tabel 3, dapat diketahui bahwa hasil evaluasi secara umum terjadi penurunan nilai baik pada metrik *precision*, *recall*, dan *F1-score*. Sedangkan nilai *accuracy* model diperoleh sebesar 72,5%. Nilai tersebut juga terlihat menurun dibandingkan dengan hasil pemodelan tanpa menggunakan SMOTE [19].

Hasil pemodelan klasifikasi data teks pernyataan personal terkait kesehatan mental

menggunakan *logistic regression* dengan ekstraksi fitur TF-IDF dan penanganan *imbalance class* menggunakan SMOTE dapat dijelaskan bahwa secara nilai evaluasi diperoleh model ML yang lebih baik adalah model tanpa menggunakan SMOTE, mengingat secara umum nilai setiap metrik evaluasi lebih tinggi dibandingkan dengan model ML menggunakan SMOTE. Namun, perlu diketahui bahwa algoritma *logistic regression* merupakan algoritma klasifikasi yang lebih baik dalam menangani kelas biner (hanya terdiri dari dua kelas saja) [14]. Sedangkan, data yang digunakan dalam penelitian ini memiliki *multi-class* dengan jumlah sebanyak 7 kelas.

Selanjutnya, implementasi metode SMOTE dalam menangani *imbalance class* juga diperoleh hasil yang terkadang tidak memberikan dampak bahkan menurunkan hasil evaluasi model yang telah dilakukan [20]. Hal ini mungkin dapat dijelaskan karena SMOTE lebih baik dalam menangani *imbalance class* pada data berdimensi rendah [20]. Pada data berdimensi tinggi, SMOTE malah dapat menghasilkan bias dan noise baru pada data kelas minoritas [20].

## 4. KESIMPULAN

Berdasarkan hasil penelitian yang telah dilakukan dapat disimpulkan bahwa pemodelan algoritma *logistic regression* dalam menangani permasalahan klasifikasi data teks menghasilkan tingkat akurasi yang cukup baik, yaitu lebih dari 70%. Namun dalam rangka optimasi model ML menggunakan metode SMOTE untuk menangani permasalahan *imbalance class* pada dataset, hal tersebut menyebabkan sedikit penurunan tingkat akurasi prediksi model, sehingga dapat dikatakan penggunaan SMOTE dalam penelitian ini tidak mengoptimasi model awal. Selain hal itu, mengingat kondisi jumlah data pada setiap kelas masih dapat dikatakan *imbalance class*, maka dapat dikatakan juga bahwa hasil tersebut masih perlu ditingkatkan karena dapat diasumsikan masih terdapat bias dan noise data. Penggunaan algoritma ML yang lebih tepat juga dapat menjadi salah satu faktor yang dapat meningkatkan model yang lebih baik untuk menangani data teks.

Melihat hal tersebut, tentunya penelitian ini masih dapat dikembangkan lebih lanjut untuk mengoptimasi tingkat akurasi model ML dalam penanganan data teks, seperti penerapan *preprocessing data* yang lebih komprehensif, (penggunaan tokenisasi dan stemming), pemilihan algoritma ML yang lebih tepat untuk menangani jenis data teks dan *multi-class*, serta metode-metode lain yang dapat digunakan untuk ekstraksi fitur berdimensi tinggi dan mengatasi *imbalance class*, diantaranya metode BERT dan ADASYN.

## 5. DAFTAR PUSTAKA

[1] L. Xiang, “Application of an Improved TF-IDF Method in Literary Text Classification,” *Adv. Multimed.*, vol. 2022, pp. 1–10, May 2022, doi: 10.1155/2022/9285324.

[2] V. Dogra *et al.*, “A Complete Process of Text Classification System Using State-of-the-Art NLP Models,” *Comput. Intell. Neurosci.*, vol. 2022, pp. 1–26, Jun. 2022, doi: 10.1155/2022/1883698.

[3] A. Mahabal, J. Baldridge, B. Karagol Ayan, V. Perot, and D. Roth, “Text Classification with Few Examples using Controlled Generalization,” in *Proceedings of the 2019 Conference of the North*, Stroudsburg, PA, USA: Association for Computational Linguistics, 2019, pp. 3158–3167. doi: 10.18653/v1/N19-1319.

[4] P. Gao, J. Zhao, Y. Ma, T. Ahmad, and B. Jin, “HFT-ONLSTM: Hierarchical and Fine-Tuning Multi-label Text Classification,” 2022, Accessed: May 02, 2025. [Online]. Available: https://www.loc.gov/aba/cataloging/classification/

[5] S. Minaee, N. Kalchbrenner, E. Cambria, N. Nikzad, M. Chenaghlu, and J. Gao, “Deep Learning Based Text Classification: A Comprehensive Review,” vol. 1, no. 1, pp. 1–43, 2020, [Online]. Available: http://arxiv.org/abs/2004.03705

[6] M. A. Shah, M. J. Iqbal, N. Noreen, and I. Ahmed, “An Automated Text Document

Classification Framework using BERT," *Int. J. Adv. Comput. Sci. Appl.*, vol. 14, no. 3, pp. 279–285, 2023, doi: 10.14569/IJACSA.2023.0140332.

[7] W. Medhat, A. Hassan, and H. Korashy, "Sentiment analysis algorithms and applications: A survey," *Ain Shams Eng. J.*, vol. 5, no. 4, pp. 1093–1113, Dec. 2014, doi: 10.1016/j.asej.2014.04.011.

[8] A. Ternikov and E. Aleksandrova, "Demand for skills on the labor market in the IT sector," *Bus. Informatics*, vol. 14, no. 2, pp. 64–83, Jun. 2020, doi: 10.17323/2587-814X.2020.2.64.83.

[9] A. Thöni, A. Taudes, and A. M. Tjoa, "An information system for assessing the likelihood of child labor in supplier locations leveraging Bayesian networks and text mining," *Inf. Syst. E-bus. Manag.*, vol. 16, no. 2, pp. 443–476, May 2018, doi: 10.1007/s10257-018-0368-0.

[10] K. Shah, H. Patel, D. Sanghvi, and M. Shah, "A Comparative Analysis of Logistic Regression, Random Forest and KNN Models for the Text Classification," *Augment. Hum. Res.*, vol. 5, no. 1, 2020, doi: 10.1007/s41133-020-00032-0.

[11] Q. Li *et al.*, "A Survey on Text Classification: From Shallow to Deep Learning," *ACM Trans. Intell. Syst. Technol.*, vol. 37, no. 4, 2020, [Online]. Available: http://arxiv.org/abs/2008.00364

[12] A. Occhipinti, L. Rogers, and C. Angione, "A pipeline and comparative study of 12 machine learning models for text classification," *Expert Syst. Appl.*, vol. 201, p. 117193, Sep. 2022, doi: 10.1016/j.eswa.2022.117193.

[13] J. A. Sáez, J. Luengo, J. Stefanowski, and F. Herrera, "SMOTE–IPF: Addressing the noisy and borderline examples problem in imbalanced classification by a re-sampling method with filtering," *Inf. Sci. (Ny).*, vol. 291, no. C, pp. 184–203, Jan. 2015, doi: 10.1016/j.ins.2014.08.051.

[14] G. Ben Abdennour, K. Gasmi, and R. Ejbali, "An Optimal Model for Medical Text Classification Based on Adaptive Genetic Algorithm," *Data Sci. Eng.*, vol. 9, no. 4, pp. 378–392, Dec. 2024, doi: 10.1007/s41019-024-00257-8.

[15] V. Rupapara, F. Rustam, H. F. Shahzad, A. Mehmood, I. Ashraf, and G. S. Choi, "Impact of SMOTE on Imbalanced Text Features for Toxic Comments Classification Using RVVC Model," *IEEE Access*, vol. 9, pp. 78621–78634, 2021, doi: 10.1109/ACCESS.2021.3083638.

[16] V. García, J. S. Sánchez, R. Martín-Félez, and R. A. Mollineda, "Surrounding neighborhood-based SMOTE for learning from imbalanced data sets," *Prog. Artif. Intell.*, vol. 1, no. 4, pp. 347–362, Dec. 2012, doi: 10.1007/s13748-012-0027-5.

[17] A. Firizkiansah, A. Muhammad, and D. Setiawan, "Implementasi Algoritma k-Nearest Neighbor (k-NN) pada Data Ulasan Pelaksanaan Pembelajaran Daring," *JIKOMTI J. Ilm. Ilmu Komput. dan Teknol. Inf.*, vol. 1, no. 1, pp. 16–23, Dec. 2024, Accessed: Mar. 20, 2025. [Online]. Available: https://ojs.sains.ac.id/index.php/Jikomti/article/view/35/35

[18] D. Setiawan, A. Muhammad, A. Firizkiansah, U. S. Indonesia, and K. Bekasi, "Pengklasifikasian Dokumen Teks Bahasa Indonesia berbasis Vector Space Model dengan menggunakan Metode k- Nearest Neighbor (k-NN) dan Euclidean Distance," *JIKOMTI J. Ilm. Ilmu Komput. dan Teknol. Inf.*, vol. 1, no. 1, pp. 30–37, 2024.

[19] A. A. P. T. Mara, E. Sediyono, and H. Purnomo, "Penerapan Algoritma K-Nearest Neighbors Pada Analisis Sentimen Metode Pembelajaran Dalam Jaringan (DARING) Di Universitas Kristen Wira Wacana Sumba."

[20] B. Y. Siow, "A Practical Approach to using Supervised Machine Learning Models to Classify Aviation Safety Occurrences," pp. 1–9, Apr. 2025, [Online]. Available: http://arxiv.org/abs/2504.09063