ZONAsi Jurnal Sistem Informasi
ZONAsi VOL. 7 NO. 1 JANUARI 2025
Page: 95 - 107
ISSN: 2656-7407 (Online)
2656-7393 (Print)

PENERAPAN METODE *LOGISTIC REGRESSION* UNTUK KLASIFIKASI SENTIMEN PADA *DATASET TWITTER* TERBATAS

Adilah Atikah Putri[1], Surya Agustian[2*], Rahmad Abdillah[3], Pizaini[4]
[1,2,3,4]Program Studi Teknik Informatika Fakultas Sains dan Teknologi
Universitas Islam Negeri Sultan Syarif Kasim Riau
Jl. H.R. Soebrantas KM. 15, Simpang Baru, Panam, Pekanbaru, Riau, telp (0761) 562051
e-mail: [1]12050120340@students.uin-suska.ac.id, [2*]surya.agustian@uin-suska.ac.id,
[3]rahmad.abdillah@uin-suska.ac.id, [4]pizaini@uin-suska.ac.id

Abstrak

Kecepatan dan akurasi menjadi semakin penting dalam analisis sentimen pada media sosial, yang sering digunakan oleh tokoh publik maupun partai politik akhir-akhir ini. Penelitian ini mengaplikasikan metode Logistic Regression (LR) untuk klasifikasi sentimen pada dataset terbatas yang hanya terdiri dari 300-600 sampel, dengan kategori label positif, negatif, dan netral. Studi kasus yang diuji adalah sentimen masyarakat di Twitter terhadap pengangkatan Kaesang Pangarep sebagai Ketua Umum Partai Solidaritas Indonesia (PSI) tanpa melalui mekanisme yang seharusnya. Data eksternal dapat digunakan untuk menambah ukuran data training, dari dataset sentimen program vaksinasi COVID-19 dan data setdengan topik yang tidak spesifik (open topic) untuk meningkatkan akurasi hasil klasifikasi. Metode TF-IDF digunakan sebagai fitur representasi teks. Grid search diterapkan untuk mencari parameter model LR optimal. Hasil kinerja klasifikasi dievaluasi menggunakan F1-score sebagai metrik ofisial. Metode baseline tanpa optimasi menghasilkan performa F1-score sebesar 40.83% pada data uji. Model optimal yang dipilih menghasilkan F1-score mencapai 52,68% dengan akurasi 61,76%. Penelitian ini menunjukkan bahwa proses klasifikasi pada dataset terbatas dengan metode Logistic Regression yang dioptimalkan dapat meningkatkan performa model dalam analisis sentimen secara signifikan.

Kata kunci: *Logistic Regression, klasifikasi sentimen, Twitter, TF-IDF*

Abstract

Speed and accuracy have become increasingly important in sentiment analysis on social media, which is often used by public figures and political parties lately. This study applies the Logistic Regression (LR) method for sentiment classification on a limited dataset consisting of only 300-600 samples, with label categories of positive, negative, and neutral. The case study examined is public sentiment on Twitter regarding the appointment of Kaesang Pangarep as the Chairman of the Indonesian Solidarity Party (PSI) without following the expected mechanisms. External data can be used to expand the training dataset, incorporating datasets on COVID-19 vaccination sentiment and open-topic datasets to improve classification accuracy. The TF-IDF method was used for text representation features, and grid search was applied to find the optimal parameters for the LR model. Classification performance was evaluated using the F1-score as the official metric. The baseline method without optimization achieved an F1-score of 40.83% on test data. The selected optimal model achieved an F1-score of 52.68% with an accuracy of 61.76%. This study demonstrates that sentiment classification on a limited dataset using an optimized Logistic Regression method can significantly improve model performance in sentiment analysis.

Keywords*: Logistic Regression, sentiment classification, Twitter, TF-IDF*

1. PENDAHULUAN

Pesatnya perkembangan internet dan penggunaan media sosial telah memungkinkan masyarakat untuk menyampaikan pandangan mereka dengan lebih mudah. Melalui komentar-komentar

tersebut, mereka memberikan penilaian terhadap berbagai isu yang berkaitan dengan pengalaman pribadi, umpan balik atau kritik terhadap berbagai objek, layanan, atau kebijakan pemerintah [1]. Media sosial telah menjadi sarana penting untuk interaksi publik, salah satunya adalah Twitter yang kini dikenal dengan X. Twitter merupakan platform yang sering dipilih oleh masyarakat untuk mengekspresikan tanggapan dan pandangan mereka dengan kata kunci yang relevan [2].

Peristiwa yang menarik perhatian publik, terutama di media sosial, menentukan dinamika politik Indonesia pada tahun 2023. *Trending* topik di Twitter adalah pengangkatan Kaesang Pangarep, putra bungsu mantan Presiden Joko Widodo, sebagai Ketua Umum Partai Solidaritas Indonesia (PSI) [3]. Keputusan ini memunculkan beragam tanggapan: sebagian melihatnya sebagai strategi menarik perhatian generasi muda, sementara lainnya mempertanyakan kapasitas politiknya. Pengamat politik menawarkan perspektif lebih kritis, menafsirkannya sebagai upaya mempertahankan "dinasti politik" keluarga Jokowi melalui PSI. Hal ini menunjukkan bagaimana media sosial tidak hanya menjadi suara masyarakat, tetapi juga wadah perdebatan yang intens.

Untuk melihat bagaimana tanggapan masyarakat terhadap isu, data yang ada di media sosial, seperti keluhan, informasi, atau saran, dapat digunakan sebagai sumber data [4]. Proses klasifikasi terhadap opini masyarakat diperlukan untuk mendapatkan hasil *feedback* yang jelas. Data opini ini dapat diolah menggunakan algoritma klasifikasi berbasis *machine learning* untuk mendapatkan pemahaman yang lebih mendalam tentang bagaimana masyarakat menerima keputusan ini. Metode yang digunakan adalah *Logistic Regression*. Pemilihan algoritma *Logistic Regression* didasarkan pada di mana algoritma ini terbukti memiliki kelebihan dalam menangani masalah klasifikasi pada data dengan dimensi tinggi, terutama saat jumlah sampel data yang tersedia terbatas [5]. Penelitian ini menghasilkan model yang dapat dengan baik dan tepat menganalisis sentimen masyarakat terkait isi yang diteliti. Diharapkan model ini akan memberikan wawasan yang akurat tentang persepsi masyarakat dan reaksi mereka terhadap isu tersebut.

Teknik pemrosesan bahasa alami atau dikenal dengan *Natural Language Processing* dapat digunakan untuk berbagai tugas, seperti *clustering, named entity recognition, automatic summarization, information classification* (baik *supervised* maupun *unsupervised*), dan *information retrieval* [6]. Penelitian ini berfokus pada penggunaan TF-IDF untuk mengurangi ketidakcocokan kosakata dalam pemrosesan bahasa alami. Penggunaan TF-IDF dapat meningkatkan efisiensi pengambilan data penting. Penulis menggunakan TF-IDF untuk klasifikasi *tweet* untuk memperluas fitur representasi teks. Teknik pembobotan TF-IDF (*Term Frequency–Inverse Document Frequency*) adalah teknik yang umum digunakan dalam *information retrieval* dan *data mining*. Salah satu prinsip utama TF-IDF adalah bahwa kata-kata yang muncul lebih sering dalam satu dokumen daripada yang muncul lebih jarang seharusnya lebih penting karena lebih bermanfaat untuk klasifikasi [7].

Beberapa penelitian sebelumnya telah mengkaji analisis sentimen [8] melakukan *Feature Expansion for Sentiment Analysis in Twitter*, hasil pengujian diperoleh bahwa fitur perluasan dapat digunakan dan terbukti meningkatkan akurasi analisis sentimen. Dari dua perluasan fitur, kami melihat peningkatan signifikan dalam perluasan fitur dengan fitur berbasis *tweet*, di mana akurasi tertinggi 98,81% dicapai menggunakan *Logistic Regression Classifier* [9] menganalisis *Sentiment Analysis using Logistic Regression* prediksi *tweet* krisis kesehatan mental yang potensial mendapatkan akurasi tertinggi 81% dan direkomendasikan untuk digunakan dalam analisis sentimen. Kemudian, beberapa penelitian menggunakan TF-IDF. [10] melakukan Twitter *sentiment analysis of* COVID-19 *using term weighting TF-IDF and logistic regresion* menyatakan bahwa penggunaan TF-IDF selama proses ekstraksi memiliki dampak signifikan terhadap kinerja klasifikasi analisis sentimen pada tweet terkait COVID-19, dengan akurasi 94,71%. [11] melakukan *Detecting Hate Speech and Offensive Language on Twitter using Machine Learning : An N-gram and TFIDF based Approach* bahwa dimungkinkan untuk mendeteksi ujaran kebencian dan bahasa ofensif di Twitter dengan menggunakan fitur n-gram yang diberi bobot dengan nilai TF-IDF dengan akurasi 95,6%, *Logistic Regression* memberikan hasil terbaik. [12] melakukan *Sentiment Analysis on Twitter Using Machine Learning Techniques and TF-IDF Feature Extraction: A Comparative Study* menunjukkan bahwa menggunakan TF-IDF untuk ekstraksi fitur dan menerapkan model *Logistic Regression* memberikan hasil terbaik dalam analisis sentimen pada data Twitter, dengan akurasi 64,86% untuk *dataset* maskapai penerbangan AS dan 71,84% untuk *dataset* umum.

Penelitian ini dilakukan dalam konteks *shared task*, dengan tantangan yang dihadapi adalah terbatasnya jumlah data pelatihan, yang disebabkan oleh kebutuhan mendesak pelanggan untuk mengetahui sentimen terhadap isu secara cepat. Dalam penelitian ini, *shared task* digunakan sebagai pendekatan untuk mendorong minat beberapa peneliti berpartisipasi dalam pengembangan metode yang mereka kembangkan untuk mengatasi problem penelitian yang diberikan. *Dataset* yang digunakan untuk mengembangkan model *machine learning* terdiri dari 300 *tweet* yang dikategorikan menjadi tiga kelas: positif, negatif, dan netral. *Shared task* di dalam penelitian klasifikasi teks banyak diperkenalkan di seluruh dunia, misalnya klasifikasi *hate speech* pada *shared task* HASOC 2023 [13].

Kasus yang diteliti dalam penelitian ini adalah pengangkatan Kaesang Pangarep sebagai Ketua Umum PSI. Isu ini menjadi perhatian publik dan memicu perdebatan, karena proses pengangkatan tersebut tidak mengikuti prosedur standar dalam partai politik. Penggunaan data eksternal dari vaksinasi COVID-19 dan topik umum bertujuan untuk meningkatkan kinerja model, mengingat keterbatasan sampel data Kaesang. Proses analisis ini mencakup penggunaan kata kunci yang relevan, dan data tersebut diproses menggunakan metode *Logistic Regression Classifier* yang dioptimalkan dengan TF-IDF. Metode ini mengubah teks menjadi representasi numerik, sehingga algoritma dapat menganalisis dan mengklasifikasikan sentimen dengan lebih efektif.

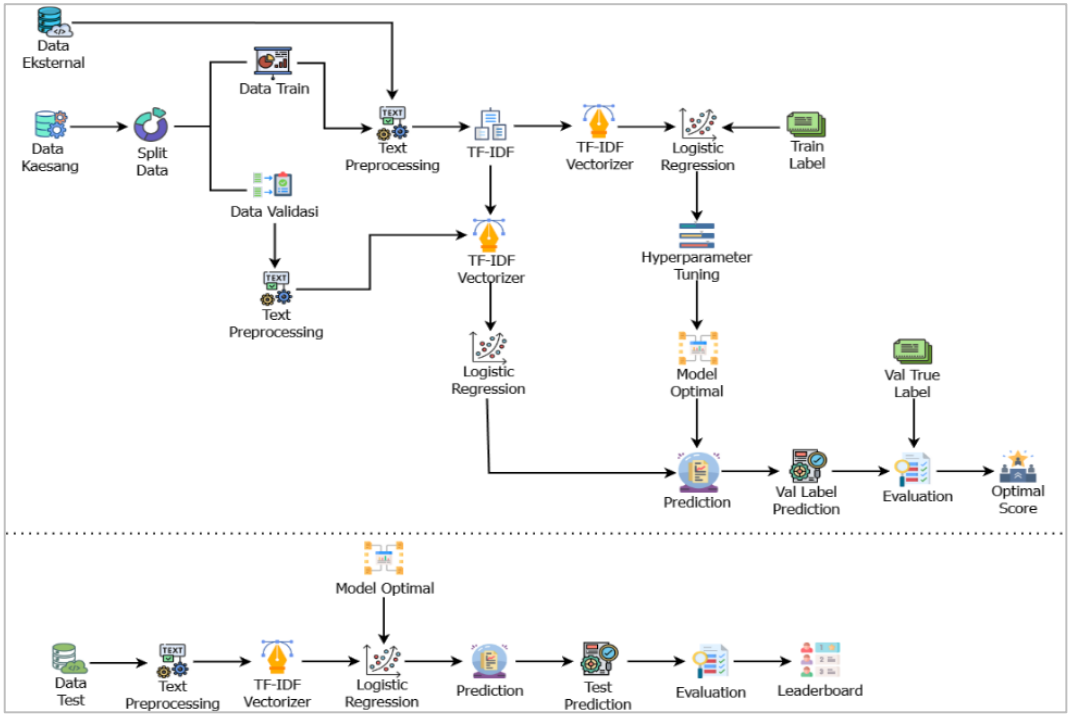
2. METODE PENELITIAN

Penelitian ini terdiri dari beberapa tahap, sebagaimana terlihat pada Gambar 1. *Dataset* yang tersedia dibuka, kemudian untuk data Kaesang dilakukan *splitting* menjadi 80:20, yaitu 80% untuk *training* (*data train*) dan 20% untuk data validasi. Data eksternal lainnya dipersiapkan untuk dipergunakan sebagai penambahan data *train*. Porsi penambahan data eksternal diatur sedemikian rupa sehingga perbandingan data eksternal dengan data topik utama tetap relatif seimbang, baik dari komposisi topiknya maupun komposisi label kelasnya.

Setelah itu, data *tweet* mengalami *text preprocessing* dilakukan untuk membersihkan dan menyiapkan teks yang dikumpulkan. Ekstraksi fitur *bag of words* berdasarkan *data train* dilakukan menggunakan metode TF-IDF untuk menghasilkan representasi numerik dari teks. Data validasi divektoriasi berdasarkan model *bag of words* TF-IDF tersebut. Data *tweet* yang telah ditransformasi ke dalam bentuk vektor TF-IDF tersebut, di inputkan kepada *Logistic Regression classifier* untuk di *training* sehingga menghasilkan model klasifikasi yang optimal. *Hyperparameter tuning* dilakukan dengan teknik *Grid Search* untuk mengoptimalkan kinerja model berdasarkan F1-score dari hasil prediksi terhadap data validasi. Model yang menghasilkan F1-score tertinggi dianggap model optimal yang dapat dicapai, kemudian digunakan untuk memprediksi label pada data *test*. Hasil prediksi data *test* di *submit* ke suatu sistem *Leaderboard* untuk mengetahui F1-score, akurasi, *precision* dan *recall*, dan posisi peringkatnya di antara peneliti lain yang mengerjakan *shared task* ini.

2.1. *Dataset*

Dataset untuk penelitian ini berasal dari tiga sumber yang diperoleh melalui metode *crawling Twitter*. *Dataset* untuk penelitian ini berasal dari tiga sumber yang diperoleh melalui metode *crawling Twitter*. *Dataset* sentimen terkait pengangkatan Kaesang sebagai ketua umum PSI, COVID-19, dan *Open Topic*. Fokus utama penelitian ini adalah *dataset* Kaesang, yang dikumpulkan secara *shared task*, dengan tujuan untuk mempelajari bagaimana penggunaan data terbatas memengaruhi pelatihan *machine learning*.



Gambar 1. Tahapan Penelitian

Dataset Kaesang diperoleh melalui *crawling Twitter* dari 25 September 2023 hingga 3 Oktober 2023 dengan kata kunci "Kaesang PSI". Proses *crowdsourcing* dengan beberapa orang anotator untuk setiap *tweet* digunakan untuk memberikan label sentimen pada setiap *tweet*. *Majority voting* menentukan label positif, netral, atau negatif. *Tweet* yang tidak memiliki label dominan dihapus dan dianggap tidak sah.

Data *training* Kaesang terdiri dari dua *dataset*, Kaesang v1 dan Kaesang v2, masing-masing dengan 300 *tweet* yang diberi label positif, negatif, dan netral. Untuk pengujian, 924 *tweet* dengan label *gold standard* tersedia di *server leaderboard*. Model terbaik dari penelitian ini akan diterapkan pada data uji. Hasil prediksi dikirim ke sistem *leaderboard* untuk mendapatkan skor evaluasi.

Data COVID-19 juga digunakan sebagai *training* untuk data Kaesang. Data COVID-19 yang dikumpulkan dari penelitian sebelumnya [14], [15], dan [16] terdiri dari 8.000 *tweet* dengan label positif, negatif, dan netral.

Data *Open Topic* digunakan untuk proses penambahan sampel data. Data *Open Topic* adalah data *tweet* yang tidak berfokus pada topik atau masalah tertentu. Data ini diambil dari *Twitter* dari Januari 2021 hingga Februari 2021 tanpa menggunakan kata kunci tertentu. Proses *scrapping* mengumpulkan lebih dari 30 ribu *tweet*. Proses memberikan label (positif, negatif dan netral) untuk kelas klasifikasi dilakukan oleh 42 orang anotator. Dari sampel *tweet* yang dikumpulkan, banyak yang tidak relevan, seperti yang berkaitan dengan pornografi dan bahasa asing. Untuk mengatasi masalah ini, kelas baru ditambahkan: "dewasa" untuk *tweet* dengan konten dewasa dan "asing" untuk *tweet* dengan bahasa asing. *Tweet* yang memiliki kelas ini hanya dianotasi oleh satu orang dan dihapus dari *dataset*. Label diberikan melalui sistem *majority voting*, dan *detailing* juga dilakukan dengan menghapus *tweet* dengan konten dewasa (pornografi), *tweet* dalam bahasa asing, dan duplikasi (*tweet* yang sama atau *retweet*). Hasilnya, 7596 *tweet* dengan label positif, negatif, dan netral telah dikumpulkan dan dapat dimanfaatkan sebagai data pelatihan tambahan. Tabel 1 berikut menunjukkan pembagian masing-masing *dataset* [17].

Tabel 1. *Dataset* penelitian

No.	*Dataset*	Penggunaan	Jumlah Sampel *Tweet*	Distribusi Kelas		
				Positif	Netral	Negatif
1.	Data Kaesang v1	*Training*	300	100	100	100
2.	Data Kaesang v2	*Training*	300	100	100	100
3.	Data Kaesang	*Testing*	924	-	-	-
4.	Data Open Topic	*Training*	7569	1505	3408	2656
5.	Data COVID-19	*Training*	8000	463	6664	873

2.2. *Text Preprocessing*

Text Preprocessing merupakan proses sistematis untuk mengubah data menjadi bentuk yang lebih terstruktur sehingga dapat diproses dengan lebih efisien. *Text Preprocessing* dilakukan dengan tujuan memperkecil ukuran dan mengubah data agar lebih mudah dalam pengolahan kata [18]. Optimasi *Text Preprocessing* pada penelitian ini, menerapkan serangkaian tahap yaitu konversi *emoji*, *cleaning*, *stopword removal*, dan *stemming* [19]. Langkah ini untuk mengubah teks mentah menjadi bentuk yang lebih terstruktur dan homogen sehingga dapat lebih mudah diproses oleh algoritma *machine learning*. Pemilihan teknik yang tepat untuk setiap langkah *preprocessing* dapat secara signifikan meningkatkan kualitas data dan, pada akhirnya, performa model.

Konversi *emoji* merupakan sebutan proses yang mengubah simbol-simbol *emoji* dan *emotikon* yang terdapat dalam teks menjadi representasi teks yang lebih mudah dipahami. *Cleaning* merupakan proses menghilangkan informasi yang dianggap tidak diperlukan, seperti *mention*, *emoticon*, dan *Uniform Resource Locator* (URL), *case folding*, dan lain-lain. *Stopword Removal* merupakan proses untuk menghilangkan atau membersihkan kata-kata yang tidak diperlukan seperti kata sambung, kata ganti, kata depan serta kata-kata yang bisa dikecualikan. *Stemming* merupakan proses mengubah kata-kata menjadi bentuk dasar dilakukan dengan cara menghilangkan akhiran atau awalan tertentu yang melekat pada kata tersebut. Tabel 2 menunjukkan simulasi langkah *text preprocessing* yang dilakukan. Namun dalam eksperimen pencarian model optimal, bisa saja beberapa tahap *text preprocessing* tidak dilakukan untuk mendapatkan hasil klasifikasi yang baik.

Tabel 2. Hasil proses *text preprocessing*

No.	Langkah *Text Preprocessing*	Hasil Setelah Proses*
1.	*Original tweet*	@abdulmukti691 Contoh aja bro ada rektor muda yang memimpin profesor dll, apakah itu salah? Kan tidak Kenapa kaesang di permasalahkan? Toh semua di PSI juga setuju 😊
2.	Konversi *emoji*	@abdulmukti691 Contoh aja bro ada rektor muda yang memimpin profesor dll, apakah itu salah? Kan tidak Kenapa kaesang di permasalahkan? Toh semua di PSI juga setuju wajah tersenyum dengan mata bahagia
3.	*Cleaning*	USER contoh aja bro ada rektor muda yang memimpin profesor dll apakah itu salah kan tidak kenapa kaesang di permasalahkan toh semua di psi juga setuju wajah tersenyum dengan mata bahagia
4.	*Stopword removal*	USER contoh bro rektor muda memimpin profesor salah kaesang permasalahkan psi setuju wajah tersenyum mata bahagia
5.	*Stemming*	USER contoh aja bro rektor muda pimpin profesor salah kaesang masalah psi tuju wajah senyum mata bahagia

*biru: kata-kata yang akan berubah di tahap selanjutnya (proses *cleaning*)
*garis-bawah: kata-kata yang akan berubah di tahap selanjutnya (proses *stopword removal*)
*merah: hasil perubahan akibat langkah *text preprocessing* sebelumnya

2.3. *Word2Vec*

Word2Vec adalah metode *embedding* kata yang diperkenalkan oleh Mikolov et al. pada tahun 2013. Teknik ini dirancang untuk menggambarkan kata-kata beserta makna dan konteksnya dalam suatu dokumen. Terdapat dua pendekatan utama yang digunakan, yaitu algoritma *Continuous Bag-of-Words*

(CBOW) dan *Skip-Gram*. Meskipun keduanya berbagi struktur jaringan yang serupa, pendekatan ini memiliki perbedaan dalam hal bagaimana data *input* dan *output* diproses [20].

Penelitian ini menggunakan model *Skip-gram*. Cara kerjanya adalah dengan mempertimbangkan kedekatan kata-kata di sekitar (disebut sebagai kata target) terhadap kata utama (disebut sebagai kata konteks). Untuk menentukan kata target, diperlukan *window size*, yaitu jumlah kata yang akan diperiksa sebelum dan sesudah kata konteks [21].

2.4. TF-IDF

Metode *Term Frequency-Inverse Document Frequency* (TF-IDF), yang menghitung bobot kata yang diekstrak dari kumpulan tweet, digunakan untuk membagi dokumen teks. Banyak orang menggunakan teknik ini untuk mengumpulkan informasi dan dianggap akurat [22]. Memberikan nilai bobot pada setiap kata adalah tujuan pembobotan kata. Frekuensi kata dalam dokumen menunjukkan jumlah kata yang ada di dalamnya, dan frekuensi balik dokumen menunjukkan jumlah kata yang muncul di seluruh dokumen. TF-IDF digunakan untuk mencari kata-kata penting dari *tweet* untuk memprediksi sentimen [23]. Untuk menghitung bobot TF-IDF digunakan persamaan (1)-(2) dari korpus *data tweet training* [24].

$$IDF_t = log \frac{d}{df_t} \quad (1)$$

$$W_{dt} = tf_t \times IDF_t \quad (2)$$

Keterangan:
IDF_t = bobot *IDF* ke token *t*
d = jumlah dokumen di dalam korpus
df_t = jumlah dokumen yang mengandung term *t*.
W_{dt} = Bobot term *t* dalam dokumen *d*
tf_t = jumlah kata *t* muncul dalam dokumen *d*, dibagi panjang dokumen (jumlah kata)

Penelitian ini menggunakan *library* **TfidfVectorizer** dari *scikit-learn* untuk membuat vektor dari teks yang telah melalui tahap *preprocessing* sebelumnya. Gambar 2 di bawah menunjukkan beberapa hasil vektorisasi kata, dari korpus *tweet* gabungan data *train* Kaesang v1 dan v2 (80% atau 480 *tweet*), data Covid (8000 *tweet*) dan data Open Topic (7569 *tweet*). Terlihat di dalam Gambar 2, bahwa jumlah kata (*feature*) di dalam *bag of words*-nya adalah 26,523 kata.

```
  feature     tfidf
0    gara  0.633529
1    baba  0.413827
2   habis  0.617493
3   ribut  0.604608
4  malang  0.404339

Ukuran matrix TF-IDF: (16049, 26532)
```

Gambar 2. Hasil dari Vektorisasi Data

2.5. *Logistic Regression*

Logistic Regression merupakan jenis regresi analisis yang menjelaskan hubungan antara variabel dependen dan variabel independen. Ini menjelaskan hubungan antara satu atau lebih variabel bebas dan variabel terikat dari jenis kategori tertentu, yang dapat berkisar antara 0 dan 1 dan benar atau salah, besar atau kecil [25]. Ini adalah alasan mengapa *Logistic Regression* berbeda dari *multiple regression* atau *linear regression*. Persamaan (3) dan (4) ini menunjukkan persamaan *Logistic Regression* yang menggunakan skala logaritmik.

$$Ln\left(\frac{p}{1-p}\right) = B_0 + B_1 X \quad (3)$$

$$p = \left(\frac{e^{(B_0+B_1X)}}{1+e^{(B_0+B_1X)}}\right) \quad (4)$$

Keterangan:
B_0 = konstanta
B_1 = koefisien dari masing-masing variabel
p = peluang ($Y = 1$)

2.6. *Hyperparameter Tuning*

Hyperparameter Tuning merupakan proses penyesuaian nilai-nilai parameter dalam model *Logistic Regression* untuk meningkatkan performanya. Salah satu metode untuk *parameter tuning* yang optimal adalah menggunakan algoritma *grid search*. Seperti namanya, *Grid Search* akan mencari parameter "*Grid*" dan *Cross Validation* (CV) untuk mengevaluasi kinerja. *Grid Search Cross Validation* adalah metode yang secara otomatis menguji dan memvalidasi kombinasi parameter dan *hyperparameter* untuk menemukan model dengan performa terbaik untuk prediksi [26].

Proses pencarian *grid*, dilakukan pada sekumpulan nilai untuk setiap *hyperparameter* yang didefinisikan dan semua kombinasi yang mungkin diuji [27], serta validasi silang *(Cross Validation),* yang memastikan model tidak hanya baik pada data pelatihan tetapi juga dapat digeneralisasi dengan baik pada data baru [7]. *Grid search* memberikan cara yang sistematis untuk mengevaluasi berbagai kombinasi *hyperparameter* dan memilih yang terbaik berdasarkan metrik kinerja yang diinginkan, seperti akurasi atau *F1-score*. Hal ini sangat penting untuk menciptakan model pembelajaran mesin yang sukses karena pilihan *hyperparameter* memiliki dampak besar pada hasilnya.

2.7. Evaluasi

Setelah mengumpulkan kombinasi parameter dan fitur yang ideal, model *Logistic Regression* akan digunakan untuk evaluasi. *Confusion Matrix* digunakan untuk menunjukkan hasil klasifikasi dengan jumlah data yang diklasifikasikan secara tepat atau tidak tepat. Biasanya digunakan untuk mengukur *accuracy, recall, precision* dan *f1-score*. Nilai *recall* dan *precision* dihitung untuk mengetahui seberapa baik teknik yang dikembangkan dapat menemukan kasus sentimen di setiap kelasnya [28]. Nilai *F1-score,* yang digunakan sebagai skor resmi (*official*) dalam *shared task* ini, akan digunakan untuk menilai hasil penelitian ini. Nilai *F1-score* akan dihitung untuk masing-masing kelas, yaitu positif, netral, dan negatif, menggunakan rumus F1 berikut ini. *Score* final yang dipakai untuk mengukur kinerja masing-masing metode yang diusulkan adalah rata-rata makro (*macro average*) dari *F1-score* dari masing-masing kelas.

$$F1 = 2\ x\ \frac{precision\ X\ recall}{precision\ +\ recall} \quad (5)$$

Selain itu, data uji akan dianalisis untuk mengevaluasi kinerja model. Untuk memastikan bahwa model dapat digeneralisasi dengan baik dan membuat ramalan yang akurat pada data baru daripada data latih. Penelitian ini dapat mengetahui kelebihan dan kekurangan model klasifikasi sentimen melalui evaluasi yang menyeluruh.

3. HASIL DAN PEMBAHASAN

3.1. Eksperimen *Setup*

Penelitian ini melibatkan 4 langkah optimasi untuk mengoptimalkan performa model, sebagaimana pada Tabel 3. Pertama, tahap teks *preprocessing*, dimulai dengan konversi *emoji* menjadi teks agar mudah diproses, pembersihan karakter yang tidak perlu, penghapusan *stopword* atau kata-kata umum yang kurang informatif, dan *stemming* untuk menyeragamkan kata ke bentuk dasarnya. Langkah-langkah pada *text preprocessing* dapat diterapkan semua maupun sebagian saja.

Selanjutnya, mengeksplorasi beberapa komposisi agregasi *dataset*, yaitu menggabungkan data *train* KSv1, KSv2, *dataset* Covid-19, dan *dataset Open Topic*, untuk melihat pengaruh penambahan

data *train* terhadap hasil klasifikasi. Penambahan data dilakukan secara empiris, dengan menambahkan secara berimbang sejumlah *tweet* dari topik-topik eksternal. Penambahan dilakukan per kelipatan 100 untuk setiap kelas positif, negatif dan netral.

Langkah optimasi ketiga adalah penerapan fitur-fitur representasi teks. Fitur dasar adalah TF-IDF, dan fitur pembanding adalah *word embeddings* dengan Word2Vec. Pengaruh jumlah fitur *bag of words* pada TF-IDF juga diselidiki, yaitu menerapkan seluruh fitur (*all*) yang diekstrak sesuai dengan ukuran gabungan data *train*, atau membatasi jumlah fitur ke dalam {1000, 2000, 3000, 4000, 5000} fitur kosa kata saja.

Terakhir, *hyperparameter tuning* dilakukan untuk menyempurnakan model, dengan fokus pada penentuan nilai C yang optimal dan pemilihan *solver logistic regression* yang tepat. Semua langkah ini dilakukan secara sistematis untuk menemukan konfigurasi terbaik yang menghasilkan performa model yang paling baik. Tabel 4 menunjukkan eksperimen setup yang digunakan.

Tabel 3. *Setup* Eksperimen

No.	Langkah Optimasi	Optimasi/Variasi
1.	Tahap *text preprocessing*	1. Konversi *emoji* 2. *Cleaning* 3. *Stopword removal* 4. *Stemming*
2.	Komposisi *dataset*	1. KSv1 2. KSv2 3. Covid-19 4. *Open topic*
3.	Fitur	1. *Word2Vec* 2. TF-IDF
4.	*Hyperparameter tuning*	1. Nilai C 2. *Solver*

3.2. Optimasi Model

Berdasarkan tabel yang disajikan, tujuan dari penelitian ini adalah untuk menemukan model *Logistic Regression* terbaik dengan mengeksplorasi berbagai langkah dari *text preprocessing*, kombinasi *dataset*, percobaan fitur *Word2Vec* dan penggunaan TF-IDF dengan *max_features* yang berbeda. Tabel 4 di bawah menyediakan rangkuman mengenai variasi eksperimen yang telah dilakukan.

3.3. Eksperimen Mencari Model Optimal

Pencarian *text preprocessing* optimal untuk *Logistic Regression* dilakukan pada eksperimen. Pendekatan penyetelan *hyperparameter* telah digunakan dalam banyak penelitian. Ini terutama berlaku untuk *solver* dan parameter C dalam *Logistic Regression*. Parameter C berfungsi untuk mengontrol kompleksitas model; Dalam penelitian ini, nilai C yang diuji adalah [0.1, 1, 10], yang mencakup rentang yang cukup untuk mengeksplorasi efek regularisasi terhadap kinerja model.

Tabel 4. Optimasi Model

ID Eksperimen		*Text Preprocessing*	*Dataset*				*Fitur*	
			KSv1	**KSv2**	***COVID-19***	***Open Topic***	***Word2Vec***	**TF-IDF**
C1		*cleaning*	300	-	-	-	✓	-
C2		*cleaning*	300	-	300	-	✓	-
C3	C3	konversi *emoji, cleaning, stemming*	600	300	-	900	-	*all*
	C3v1							1000
	C3v2							2000
	C4v3							3000
	C3v4							4000
	C3v5							5000
C4	C4		600	300	900	-	-	*all*

	C4v1	konversi *emoji*, *cleaning*, *stopword removal*, *stemming*						1000
	C4v2							2000
	C4v3							3000
	C4v4							4000
	C4v5							5000
C5	C5	konversi *emoji*, *cleaning*	600	300	300	900	-	*all*
	C5v1							1000
	C5v2							2000
	C5v3							3000
	C5v4							4000
	C5v5							5000
C6	C6	konversi *emoji*, *cleaning*	600	300	900	900	-	*all*
	C6v1							1000
	C6v2							2000
	C6v3							3000
	C6v4							4000
	C6v5							5000
C7	C7	konversi *emoji*, *cleaning*	600	300	600	600	-	*all*
	C7v1							1000
	C7v2							2000
	C7v3							3000
	C7v4							4000
	C7v5							5000

Parameter selanjutnya yang akan diselidiki adalah *solver*. *Solver* adalah algoritma yang digunakan untuk mengoptimalkan fungsi biaya dalam model. *Solver* yang digunakan adalah ['*newton-cg*', '*lbfgs*', '*liblinear*']. Penggunaan *5-fold Cross Validation* dalam penelitian ini bertujuan untuk mengevaluasi kinerja model secara lebih akurat.

Tabel 5. *Hyperparameter tuning dengan Grid Search 5-fold Cross Validation*

ID Eksperimen	***Best Parameter***		**Data *Training***		**Data Validasi**	
	C	***Solver***	**F1 (%)**	**Akurasi (%)**	**F1 (%)**	**Akurasi (%)**
C1	1	liblinear	55.26	55.83	56.60	57.50
C2	1	liblinear	62.74	63.33	60.34	60.83
C3	0.1	newton-cg	68.98	69.17	65.23	65.00
C3v1			66.53	66.67	63.33	63.33
C3v2			63.97	64.17	64.17	64.17
C4v3			63.02	63.33	63.41	63.33
C3v4			65.67	65.83	65.22	65.00
C3v5			68.16	68.33	65.14	65.00
C4	0.1	newton-cg	**69.96**	**70.00**	66.19	65.83
C4v1			64.01	64.17	63.42	63.33
C4v2			66.44	66.67	64.37	64.17
C4v3			69.15	69.17	64.50	64.17
C4v4			69.96	70.00	66.19	65.83
C4v5			69.96	70.00	66.19	65.83

C5	C5	0.1	newton-cg	66.36	66.67	64.02	64.17
	C5v1			66.23	66.67	63.53	63.33
	C5v2			63.73	64.17	63.51	63.33
	C5v3			67.17	67.50	65.00	65.00
	C5v4			65.38	65.83	65.93	65.83
	C5v5			64.60	65.00	64.89	65.00
C6	C6	0.1	newton-cg	65.50	65.83	65.90	65.83
	C6v1			62.03	62.50	61.78	61.67
	C6v2			62.90	63.00	63.44	63.33
	C6v3			62.83	63.33	61.95	61.67
	C6v4			64.72	65.00	63.49	63.33
	C6v5			64.75	65.00	65.75	65.83
C7	C7	0.1	newton-cg	67.31	67.50	**66.58**	**66.67**
	C7v1			61.51	61.67	65.30	65.00
	C7v2			65.69	65.83	64.19	64.17
	C7v3			64.90	65.00	66.56	66.50
	C7v4			66.45	66.67	65.00	65.00
	C7v5			67.33	67.50	65.00	65.00

Pada Tabel 5 dapat dilihat bahwa nilai parameter terbaik untuk nilai C adalah [0.1] dan *solver* adalah [newton-cg]. Kemudian, data *training* tertinggi pada eksperimen C4 dengan fitur TF-IDF. Sedangkan, data validasi tertinggi pada eksperimen C7 dengan fitur TF-IDF. Hasil dari *hyperparameter tuning* menunjukkan bahwa kinerja model pada data *training* yang tinggi tidak menjamin akan menghasilkan performa tertinggi pada data validasi. Hal ini tergantung pada kombinasi *dataset* yang digunakan dan nilai *max_features*, di mana terkadang pengaturan ini dapat menghasilkan kinerja yang baik pada kedua data *training* dan validasi.

Model terbaik dipilih berdasarkan kombinasi parameter yang menghasilkan skor *F1* tinggi pada data validasi, seperti dilihat pada Tabel 5, yang menampilkan nilai C dan *solver* yang terbaik di setiap eksperimen.

3.4. Pengujian terhadap Data Uji

Untuk pengujian final, model optimal yang telah dilatih diterapkan pada data *test* yang belum pernah digunakan sebelumnya untuk *training.* Hasil implementasi model *Logistic Regression* pada data *testing*, bisa dilihat pada tabel 6 berikut.

Tabel 6. Pengujian Data Uji

ID Eksperimen	Run	*Data Val*		*Data Test*	
		***F1* (%)**	**Akurasi (%)**	***F1* (%)**	**Akurasi (%)**
C1	Run 1	56.60	57.50	45.86	55.04
C2	Run 2	60.34	60.83	46.26	55.90
C7	Run 3	**66.58**	**66.67**	**52.68**	**61.76**

Berdasarkan hasil eksperimen yang ditunjukkan dalam Tabel 6 mengungkapkan dinamika performa yang menarik di antara ketiga percobaan. Run-1 dan Run-2 adalah model dari eksperimen awal C1 dan C2, yang menggunakan *Word2Vec* sebagai metode *word embedding.* Ekspektasinya, penggunaan vektor *word embeddings* akan menunjukkan hasil yang bagus. Kenyataannya, hasil yang diperoleh pada Run-1 dan Run-2 kurang menjanjikan dan butuh peningkatan.

Transformasi signifikan terjadi ketika eksperimen beralih menggunakan TF-IDF sebagai metode representasi teks pada C3 hingga C7 pada data validasi. Perubahan ini terbukti dimana C7 dengan performa yang mengesankan. Keunggulan TF-IDF dibandingkan *Word2Vec* dalam konteks *Logistic Regression* ini menunjukkan bahwa representasi berbasis frekuensi kata lebih efektif dalam menangkap data teks untuk tugas klasifikasi ini. Peningkatan performa yang konsisten dari C3 hingga C7 dengan penggunaan TF-IDF memvalidasi keputusan untuk beralih dari *Word2Vec*, membuktikan

bahwa pendekatan yang lebih sederhana namun tepat dapat menghasilkan hasil yang lebih optimal dalam pembelajaran mesin.

3.5. Perbandingan Pengujian

Tabel 7 menunjukkan perbandingan kinerja skor F1, akurasi, presisi, dan *recall* antara hasil penelitian ini dan penelitian sebelumnya. Perbandingan ini menunjukkan bahwa metode BERT *classifier* (Kaesangv1+Final) dari Pranata dkk. Rank 1 mengungguli metode lainnya. Di posisi ketujuh, penelitian ini menggunakan metode LR+TF-IDF+emoji yang menunjukkan hasil cukup kompetitif. Sementara metode SVM (Kaesangv1+Covid) dari tim *Organizer* berada di peringkat 12 dengan performa yang tidak jauh berbeda. *Baseline* yang digunakan sebagai pembanding dasar menempati peringkat terendah (18) dengan performa yang cukup jauh di bawah metode-metode lainnya. Hasil evaluasi metrik seperti skor F1 menunjukkan bahwa penerapan *Logistic Regression* dengan fitur TF-IDF menunjukkan pendekatan yang efektif meskipun belum mencapai yang paling unggul.

Hasil pengujian menunjukkan bahwa metode TF-IDF yang dikombinasikan dengan *Logistic Regression* memberikan performa yang lebih unggul dibandingkan pendekatan berbasis *Word2Vec* seperti yang telah dijelaskan pada 3.7. Pengujian Data Uji. Peningkatan *F1-score* dari eksperimen C3 hingga C7 juga mengindikasikan pentingnya langkah optimasi, baik dalam pemilihan fitur, *tuning hyperparameter*, maupun kombinasi *dataset* eksternal. Dengan menambahkan variasi data, model mampu belajar dari konteks yang lebih beragam, yang kemudian meningkatkan akurasi pada data uji.

Tabel 7. Perbandingan Pengujian

Rank	Tim	Metode	*F1*	Akurasi	Presisi	*Recall*
1	Pranata, J., dkk [27]	BERT *Classifier*	**60.63**	**70.53**	**59.36**	**67.75**
7	Penelitian ini	*Logistic Regression*	52.68	61.76	53.54	60.97
12	*Organizer* [17]	SVM	51.28	61.21	52.89	57.22
18	Admin [17]	*Baseline*	40.38	45.45	49.53	48.80

3.6. Pembahasan

Hasil penelitian menunjukkan bahwa penerapan metode *Logistic Regression* untuk klasifikasi sentimen pada *dataset* terbatas dapat ditingkatkan dengan menambah data eksternal dan teknik optimasi. Dengan menggabungkan *dataset* dari sentimen terkait COVID-19 dan topik umum, model yang dioptimalkan mencapai *F1-score* tertinggi sebesar 52,68% dengan akurasi *61,76%*. Hal ini menunjukkan bahwa variasi dalam *dataset* pelatihan dapat membantu model memahami konteks sentimen yang lebih kompleks. Dalam perbandingan dengan penelitian sebelumnya, meskipun hasil penelitian ini menunjukkan performa yang lebih baik dari *baseline* (40,83%), masih terdapat celah signifikan dibandingkan dengan metode yang lebih canggih seperti BERT. Penelitian ini menekankan perlunya eksplorasi lebih lanjut terhadap metode baru dalam pemrosesan bahasa alami dan *deep learning* untuk meningkatkan akurasi klasifikasi sentimen. Secara keseluruhan, penelitian ini memberikan wawasan lebih mendalam tentang optimasi metode *Logistic Regression* untuk tugas klasifikasi sentimen. Penelitian ini juga membuktikan bahwa, dengan pendekatan yang sesuai, model ini mampu bersaing dengan metode yang lebih canggih dalam analisis sentimen di *platform* media sosial.

4. KESIMPULAN

Hasil penelitian menunjukkan bahwa optimasi performa model klasifikasi *Logistic Regression* melalui penambahan data dari topik yang berbeda (*dataset* eksternal) secara signifikan meningkat. Hal ini terlihat dari hasil eksperimen pada tiga skenario terbaik, yaitu C1, C2, dan C7, dengan F1-Score masing-masing 56,60%, 60,34%, dan 66,58% pada data validasi, serta 45,86%, 46,26%, dan 52,68% pada data *testing*. Berdasarkan temuan ini, penggunaan *dataset* eksternal terbukti memberikan kontribusi yang signifikan dalam meningkatkan F1-Score, menunjukkan bahwa metode *Logistic Regression* dengan teknik TF-IDF efektif untuk klasifikasi sentimen *tweet* terkait Kaesang Pangarep sebagai Ketua Umum PSI di Twitter. Penelitian ini menekankan pentingnya integrasi data dan strategi optimasi dalam menciptakan model klasifikasi yang lebih efektif, sekaligus memberikan pemahaman untuk penelitian lanjutan dalam klasifikasi sentimen.

Daftar Pustaka

[1] R. A. Husen, R. Astuti, L. Marlia, and L. Efrizoni, "Sentiment Analysis of Public Opinion on Twitter Toward BSI Bank Using Machine Learning Algorithms Analisis Sentimen Opini Publik pada Twitter Terhadap Bank BSI Menggunakan Algoritma Machine Learning," vol. 3, no. October, pp. 211–218, 2023.

[2] S. Suryono, E. Utami, and E. T. Luthfi, "Klasifikasi Sentimen Pada Twitter Dengan Naive Bayes Classifier," *Angkasa J. Ilm. Bid. Teknol.*, vol. 10, no. 1, p. 89, 2018, doi: 10.28989/angkasa.v10i1.218.

[3] B. N. Indonesia, "Kaesang resmi menjadi Ketum PSI, apa artinya bagi pertarungan Pilpres 2024?," vol., no., p., Sep. 25, 2023. [Online]. Available: https://www.bbc.com/indonesia/articles/crg8mpexwxgo

[4] Y. Pratama, D. T. Murdiansyah, and K. M. Lhaksmana, "Analisis Sentimen Kendaraan Listrik Pada Media Sosial Twitter Menggunakan Algoritma Logistic Regression dan Principal Component Analysis," vol. 7, pp. 529–535, 2023, doi: 10.30865/mib.v7i1.5575.

[5] N. L. P. C. Savitri, R. A. Rahman, R. Venyutzky, and N. A. Rakhmawati, "Analisis Klasifikasi Sentimen Terhadap Sekolah Daring pada Twitter Menggunakan Supervised Machine Learning," *J. Tek. Inform. dan Sist. Inf.*, vol. 7, no. 1, pp. 47–58, 2021, doi: 10.28932/jutisi.v7i1.3216.

[6] A. Khatua, A. Khatua, and E. Cambria, "A tale of two epidemics: Contextual Word2Vec for classifying twitter streams during outbreaks," *Inf. Process. Manag.*, vol. 56, no. 1, pp. 247–257, 2019, doi: 10.1016/j.ipm.2018.10.010.

[7] H. A. O. Liu, X. I. Chen, and X. Liu, "A Study of the Application of Weight Distributing Method Combining Sentiment Dictionary and TF-IDF for Text Sentiment Analysis," *IEEE Access*, vol. 10, pp. 32280–32289, 2022, doi: 10.1109/ACCESS.2022.3160172.

[8] E. B. Setiawan, D. H. Widyantoro, and K. Surendro, "Feature expansion for sentiment analysis in twitter," *Int. Conf. Electr. Eng. Comput. Sci. Informatics*, vol. 2018-Octob, pp. 509–513, 2018, doi: 10.1109/EECSI.2018.8752851.

[9] S. Kumar, N. Kaur, Kavita, and A. Joshi, "Tweet Sentiment Analysis using Logistic Regression," *IET Conf. Proc.*, vol. 2023, no. 11, pp. 332–336, 2023, doi: 10.1049/icp.2023.1801.

[10] Imamah and F. H. Rachman, "Twitter sentiment analysis of Covid-19 using term weighting TF-IDF and logistic regresion," *Proceeding - 6th Inf. Technol. Int. Semin. ITIS 2020*, pp. 238–242, 2020, doi: 10.1109/ITIS50118.2020.9320958.

[11] A. Gaydhani, V. Doma, S. Kendre, and L. Bhagwat, "Detecting Hate Speech and Offensive Language on Twitter using Machine Learning : An N-gram and TFIDF based Approach".

[12] W. Ahmed, "Sentiment Analysis on Twitter Using Machine Learning Techniques and TF-IDF Feature Extraction : A Comparative Study," vol. 10, pp. 2–7, 2023.

[13] O. E. Ojo, O. O. Adebanji, H. Calvo, A. Gelbukh, A. Feldman, and G. Sidorov, "Hate and Offensive Content Identification in Indo-Aryan Languages using Transformer-based Models," *CEUR Workshop Proc.*, vol. 3681, no. November, pp. 383–392, 2023, doi: 10.13140/RG.2.2.31562.54721.

[14] M. Ihsan, Benny Sukma Negara, and Surya Agustian, "LSTM (Long Short Term Memory) for Sentiment COVID-19 Vaccine Classification on Twitter," *Digit. Zo. J. Teknol. Inf. dan Komun.*, vol. 13, no. 1, pp. 79–89, 2022, doi: 10.31849/digitalzone.v13i1.9950.

[15] M. K. Kusairi and S. Agustian, "Metode SVM dengan Fitur Representasi FastText untuk Klasifikasi Sentimen Twitter Mengenai Program Vaksinasi Covid-19," *J. Teknol. Inf. dan Komun.*, vol. 13, no. 2, pp. 140–150, 2022.

[16] Ash Shiddicky and Surya Agustian, "Analisis Sentimen Masyarakat Terhadap Kebijakan Vaksinasi Covid-19 pada Media Sosial Twitter menggunakan Metode Logistic Regression," *J. CoSciTech (Computer Sci. Inf. Technol.*, vol. 3, no. 2, pp. 99–106, 2022, doi: 10.37859/coscitech.v3i2.3836.

[17] S. Agustian, M. I. Syah, N. Fatiara, and R. Abdillah, "New Directions in Text Classification Research : Maximizing The Performance of Sentiment Classification from Limited Data Arah Baru Penelitian Klasifikasi Teks : Memaksimalkan Kinerja Klasifikasi Sentimen dari Data Terbatas," pp. 1–10, 2024, [Online]. Available: https://arxiv.org/abs/2407.05627

[18] A. Poornima and K. S. Priya, "A Comparative Sentiment Analysis of Sentence Embedding

Using Machine Learning Techniques," *2020 6th Int. Conf. Adv. Comput. Commun. Syst. ICACCS 2020*, pp. 493–496, 2020, doi: 10.1109/ICACCS48705.2020.9074312.

[19] M. F. Karaca, "Effects of preprocessing on text classification in balanced and imbalanced datasets," *KSII Trans. Internet Inf. Syst.*, vol. 18, no. 3, pp. 591–609, 2024, doi: 10.3837/tiis.2024.03.004.

[20] F. P. Giovanni Di Gennaro, A. Buonanno, "Considerations about learning Word2Vec," *J. Supercomput.*, vol. 77, no. 11, 2021, doi: 10.1007/S11227-021-03743-2/FIGURES/7.

[21] Siti Khomsah, Rima Dias Ramadhani, and Sena Wijaya, "The Accuracy Comparison Between Word2Vec and FastText On Sentiment Analysis of Hotel Reviews," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 6, no. 3, pp. 352–358, 2022, doi: 10.29207/resti.v6i3.3711.

[22] V. Amrizal, "Penerapan Metode Term Frequency Inverse Document Frequency (Tf-Idf) Dan Cosine Similarity Pada Sistem Temu Kembali Informasi Untuk Mengetahui Syarah Hadits Berbasis Web (Studi Kasus: Hadits Shahih Bukhari-Muslim)," *J. Tek. Inform.*, vol. 11, no. 2, pp. 149–164, 2018, doi: 10.15408/jti.v11i2.8623.

[23] M. R. Hasan, M. Maliha, and M. Arifuzzaman, "2019 International Conference on Computer, Communication, Chemical, Materials and Electronic Engineering (IC4ME2)," *2019 Int. Conf. Comput. Commun. Chem. Mater. Electron. Eng.*, pp. 1–4, 2019.

[24] Mega Kurnia Maulidina, "ANALISIS SENTIMEN KOMENTAR WARGANET TERHADAP POSTINGAN INSTAGRAM MENGGUNAKAN METODE NAIVE BAYES CLASSIFIER DAN TF-IDF (Studi Kasus: Instagram Gubernur Jawa Barat ridwan Kamil)," *Perpustakaan Universitas Teknologi Yogyakarta. Perpustakaan Universitas Teknologi Yogyakarta.*, 2020. http://lib.uty.ac.id/index.php?p=show_detail&id=13386

[25] F. D. Pramakrisna, F. D. Adhinata, and N. A. F. Tanjung, "Aplikasi Klasifikasi SMS Berbasis Web Menggunakan Algoritma Logistic Regression," *Teknika*, vol. 11, no. 2, pp. 90–97, 2022, doi: 10.34148/teknika.v11i2.466.

[26] N. R. Robynson and Y. Sibaroni, "Analisis Tren Sentimen Masyarakat Terhadap Pembatasan Sosial Berskala Besar Kota Jakarta Menggunakan Algoritma Support Vector Machine," *e-Proceeding Eng.*, vol. 8, no. 5, pp. 10166–10178, 2021.

[27] J. (2016). Brownlee, "How to grid search hyperparameters for deep learning models in python with keras," *machinelearningmastery*, 2016. https://machinelearningmastery.com/grid-search-hyperparameters-deep-learning-models-python-keras/

[28] P. Yohana, S. Agustian, and S. K. Gusti, "Klasifikasi Sentimen Masyarakat terhadap Kebijakan Vaksin Covid-19 pada Twitter dengan Imbalance Classes Menggunakan Naive Bayes," *Semin. Nas. Teknol. …*, pp. 69–80, 2022, [Online]. Available: http://ejournal.uin-suska.ac.id/index.php/SNTIKI/article/view/19012%0Ahttp://ejournal.uin-suska.ac.id/index.php/SNTIKI/article/viewFile/19012/8336

[27] J. Pranata, S. Agustian, Jasril, E. Haerani, " Penggunaan Model Bahasa indoBERT pada metode Random Forest untuk Klasifikasi Sentimen dengan Dataset Terbatas", *Building of Informatics, Technology and Science* (BITS), vol 6 No 3 (2024): December 2024, doi:10.47065/bits.v6i3.6335