Institut Riset dan Publikasi Indonesia (IRPI)
MALCOM: Indonesian Journal of Machine Learning and Computer Science
Journal Homepage: https://journal.irpi.or.id/index.php/malcom
Vol. 5 Iss. 3 July 2025, pp: 908-919
ISSN(P): 2797-2313 | ISSN(E): 2775-8575

Sentiment Classification Using Multilayer Perceptron Algorithm with TF-IDF Features

Klasifikasi Sentimen Menggunakan Metode Multilayer Perceptron dengan Fitur TF-IDF

Abdurrahman Arasy[1*], Surya Agustian[2], Lestari Handayani[3] , Iwan Iskandar[4]

[1,2,3,4,5]Program Studi Teknik Informatika, Fakultas Sains dan Teknologi,
Universitas Islam Negeri Sultan Syarif Kasim Riau, Indonesia

E-Mail: [1]11950114943@students.uin-suska.ac.id, [2]surya.agustian@uin-suska.ac.id,
[3]lestari.handayani@uin-suska.ac.id, [4]iwan.iskandar@uin-suska.ac.id

Received Apr 16th 2025; Revised Jun 16th 2025; Accepted Jun 20th 2025; Available Online Jun 24th 2025; Published Jun 24th 2025
Corresponding Author: Abdurrahman Arasy

Abstract

Social media, particularly Twitter (X), has become the main platform for discussions on politics and government policy. The term used for messages sent on Twitter is a "tweet", which consists of a message with a maximum of 280 characters. Although Tweets are often just text, they can also include hyperlinks, videos, and other types of media that can be used to gauge public opinion. This study aims to classify public sentiment regarding the appointment of Kaesang Pangarep as Chairman of the Indonesian Solidarity Party (PSI) using the Multi-Layer Perceptron (MLP) Classifier method with the Term Frequency-Inverse Document Frequency (TF-IDF) approach, utilizing the Python programming language. The data used consists of 300 Tweets, with 100 Tweets per class or option for optimal results. The three categories are positive, neutral, and negative. Based on the research conducted, the best method achieved an F1-score of 0.6767 and an accuracy of 0.6667. These results indicate that the combination of the MLP Classifier and TF-IDF can overcome the limitations of the dataset to a certain extent compared to the baseline method. This study also provides insights into sentiment classification optimization under limited data conditions, which can be applied to other topics with similar issues.

Keyword: Classification, Multi-Layer Perceptron, Sentimen Analysis, Term Frequency-Inverse Document Frequency

Abstrak

Media sosial, khususnya Twitter (X), telah menjadi platform utama dalam diskusi politik dan kebijakan pemerintah. Istilah dalam pengiriman pesan pada Twitter dikenal sebagai Tweet yang terdiri dari pesan dengan maksimal 280 karakter. Meskipun Tweet seringkali hanya berupa teks, juga dapat menyertakan *hyperlink*, video, dan jenis media lainnya yang dapat digunakan untuk mengukur opini publik. penelitian ini bertujuan mengklasifikasikan sentimen masyarakat terkait pengangkatan Kaesang Pangarep sebagai Ketua Umum Partai Solidaritas Indonesia (PSI) dengan metode *Multi-Layer Perceptron* (MLP) *Classifier* dengan pendekatan *Term Frequency-Inverse Document Frequency* (TF-IDF) menggunakan bahasa pemograman *python*. Data yang digunakan terdiri dari 300 tweet, dengan 100 tweet perkelas atau opsi untuk hasil yang optimal. Tiga kategori tersebut adalah positif, netral, dan negatif. Berdasarkan penelitian yang telah dilakukan metode terbaik mencapai *F1-score* sebesar 0,6767 dan akurasi 0,6667. Hasil ini menunjukkan bahwa kombinasi MLP Classifier dan TF-IDF dapat mengatasi keterbatasan dataset hingga tingkat tertentu dibandingkan metode *baseline*. Penelitian ini juga memberikan wawasan tentang optimasi klasifikasi sentimen dalam kondisi data terbatas, yang dapat diterapkan pada topik lain dengan permasalahan serupa.

Kata Kunci: Analisis Sentimen, Klasifikasi, Multi-Layer Perceptron, Term Frequency-Inverse Document Frequency

1. PENDAHULUAN

Media sosial menjadi semakin penting dalam diskusi politik dan kebijakan, salah satu contohnya yaitu diskusi rakyat pada salahsatu akun twitter Presiden [1]. Twitter adalah situs web microblog dimana pengguna dapat mengirim pesan yang dikenal sebagai Tweet. Tweet ini dapat terdiri dari pesan dengan maksimal 280 karakter. Tweet itu sendiri, meskipun seringkali hanya berupateks, dapat juga dapat menyertakan *hyperlink,*

video, dan jenis media lainnya [2]. Data ini dapat digunakan untuk mengukur opini publik terkait penunjukan Kaesang Pangarep sebagai ketua umum PSI. Karenanya, analisis sentiment membutuhkan kategori sasi untuk memperkirakan kelas dengan label yang tidak pasti.

Analisis sentiment adalah klasifikasi Bahasa yang dikaitkan dengan kemajuan dibidang *Natural Language Processing* (NLP). Hal ini sangat membantu untuk mengklasifikasikan teknik berdasarkan kemajuan yang signifikan dalam *NLP*, karena terobosan tersebut mewakili pergeseran paradigma yang signifikan dalam pemrosesan dan penilaian teks [3][4]. Salah satu algoritma yang digunakan untuk analisis sentiment adalah MLP *Classifier (Multi-layer Perceptron). MLP* memberikan banyak fleksibilitas dan telah terbukti berguna dan dapat diandalkan dalam berbagai masalah klasifikasi dan regresi [5]. MLP adalah sebuah *Artificial Neural Network* (ANN) yang memiliki lapisan tersembunyi. *MLP* memiliki tiga lapisan yaitu input, tersembunyi, dan keluaran. Data pelatihan digunakan untuk mempelajari bobot tersembunyi antara atribut dan label kelas [6].

Penelitian mengenai analisis sentiment untuk ahli anestesiologi brazil menggunakan pengklasifikasi *Multi-Layer Perceptron* setelah dilabeli menggunakan *TextBlob* menghasilkan akurasi 94,44%, presisi 94,44%, *recall* 92%, dan *f1_score* 93%. Disisi lain, *Random Forest* menghasilkan akurasi 96,42%, presisi 94,44%, *recall* 96,66%, dan *f1_score* 95,56%. Kasus pelecehan seksual mendapatkan sentimen negatif yang lebih tinggi di media sosial Twitter. Secara umum, *Random Forest* berkerja lebih baik daripada *MLP*, mungkin karena dimensi tinggi dari tweet dan jumlah *teks* yang sedikit dalam tweet serta ukuran sampel [7]. Pada penelitian tentang analisis *sentiment* hasil *rivew* mahasiswa dengan berbagai jenis skema *Machine Learning* menunjukkan bahwa MLP *Classifier* lebih unggul dibandingkan pengklasifikasi lainnya dengan *F1-Score* mencapai 67% dan lebih unggul dari metode lainnya [8]. Penelitian menggunakan metode *K-Nearest Neighbor* (K-NN) untuk klasifikasi sentimen pada data terbatas, membandingkan tiga teknik ekstraksi fitur: *FastText*, *Term Frequency–Inverse Document Frequency* (TF-IDF), dan IndoBERT. Hasil pengujian menunjukkan bahwa setelah optimasi dan penggunaan IndoBERT, performa model meningkat signifikan dari akurasi 44% dan F1-score 39% menjadi akurasi 57% dan F1-score 49% [11]. Dari berbagai penelitian menunjukkan bahwa klasifikasi sentimen pada data latih kecil tetap dapat menghasilkan performa yang baik melalui optimasi seperti penambahan data eksternal, preprocessing, tuning parameter, dan penggunaan representasi fitur canggih seperti TF-IDF, *FastText,* dan IndoBERT. Metode seperti K-NN, *Bidirectional Long Short-Term Memory* (Bi-LSTM), *Support Vector Machines* (SVM), dan *Passive Aggressive* terbukti mengalami peningkatan performa signifikan setelah optimasi, dengan IndoBERT dan teknik transfer *learning* memberikan kontribusi besar terhadap peningkatan *F1-score* dan akurasi.

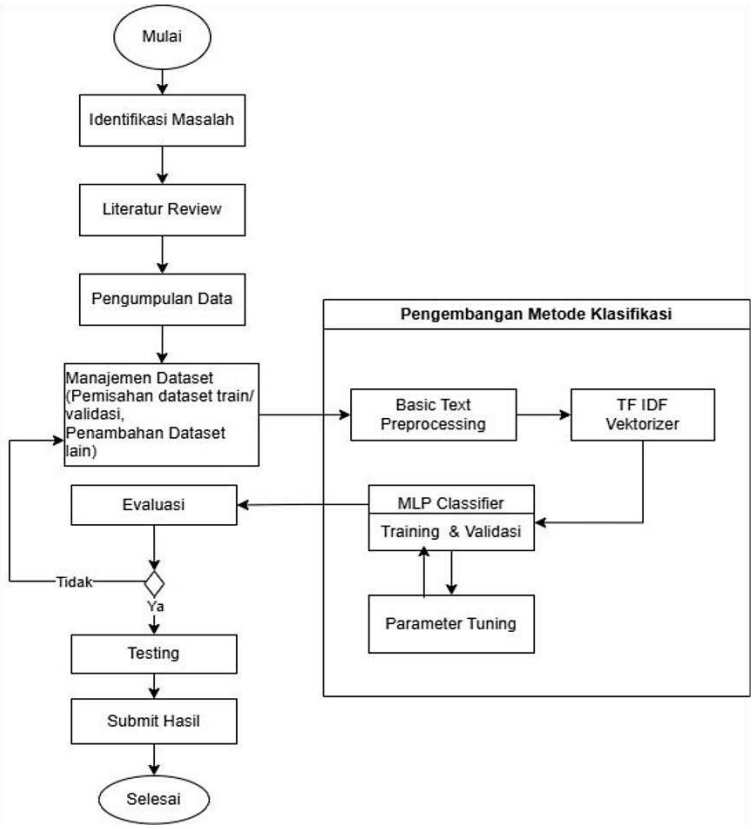
Penggunaan data pada penelitian ini relatif sedikit, khususnya untuk melatih model *machine learning*. Sejalan dengan *shared task* terkait klasifikasi sentiment seperti yang dijabarkan pada studi. Dataset terdiri dari 300 tweet sebagai data training, yang dinilai kurang memenuhi syarat untuk mendapatkan hasil yang optimal. Penelitian ini akan berfokus pada klasifikasi sentiment dari tiga kategori utama: positif, netral, dan negatif pada dataset kecil atau terbatas. Dengan studi kasus sentimen masyarakat di Twitter terkait pengangkatan Kaesang Pangarep sebagai Ketua Umum Partai Solidaritas Indonesia (PSI). Metode yang digunakan adalah *Multilayer Perceptron* dengan Fitur TF-IDF metode ini sangat populer karena efektif dalam merepresentasikan data teks dan memiliki keunggulan dalam berbagai penggunaannya [10]. Penelitian ini bertujuan memberikan kontribusi melalui penerapan algoritma *machine learning*, khususnya menggunakan metode MLP *Classifier*. Hasil yang diharapkan adalah peningkatan performa *system* klasifikasi yang ditunjukkan dengan tingkat *F1-score* dan akurasi yang lebih baik dari metode baseline. Proses klasifikasi sentiment akan dilakukan menggunakan bahasa pemograman *Python*.

2. METODE PENELITIAN

Studi ini menggunakan metodologi yang terdiri dari langkah yang terstruktur untuk menjamin keakuratan dan keandalan temuan. Langkah tersebut mencakup proses penelitian, pengelolaan dataset, *preprocessing* teks, penerapan metode *vektorisasi*, penggunaan algoritma MLP, serta pengujian model. Setiap langkah dirancang secara cermat untuk mengatasi tantangan dalam analisis sentiment pada data set yang terbatas, sekaligus menjamin model yang dihasilkan mampu memberi kinerja yang optimal. Tahapan penelitian yang dilakukan mengikuti diagram yang digambarkan pada Gambar 1.

2.1. Identifikasi Masalah

Langkah pertama dalam Teknik penelitian adalah perumusan masalah,yang merumuskan, menganalisis masalah, dan menentukan latar belakang masalah dengan studi yang dilakukan. Masalah yang diangkat dalam penelitian ini adalah keterbatasan data berlabel yangdigunakan untuk pelatihan metode *Machine Learning* MLP. Hal ini disebabkan karena dalam dunia nyata, kebutuhan analisis sentimen harus dapat dilakukan dengan cepat, sehingga tidak mungkin menghabiskan waktu memberi label pada data pelatihan dari suatu kasus sentiment yang ingin dianalisis.



Gambar 1. Tahapan Penelitian

2.2. Literatur Review

Pada tahap ini, dilakukan penelusuran terhadap informasi, teori, dan konsep dasar yang berkaitan dengan materi terkait penelitian. Beberapa penelitian yang dapat digunakan sebagai bahan referensi atau perbandingan karena telah menerapkan metode atau topik penelitian yang sama dengan penelitian ini.

Penelitian oleh Atika Putri [11] menggunakan metode K-NN untuk klasifikasi sentimen pada data terbatas, sama dengan yang digunakan dalam penelitian ini. Mereka membandingkan tiga teknik ekstraksi fitur: *FastText,* TF-IDF*,* dan IndoBERT. Optimasi dilakukan melalui penambahan data eksternal, *preprocessing teks*, scaling, dan tuning parameter. Hasil pengujian menunjukkan bahwa setelah optimasi dan penggunaan IndoBERT, performa model meningkat signifikan dari akurasi 44% dan F1-score 39% menjadi akurasi 57% dan F1-score 49%. Penelitian Ridho Illahi mengembangkan [12] model klasifikasi sentimen menggunakan Bi-LSTM yang dikombinasikan dengan representasi teks IndoBERT untuk menganalisis opini publik di media sosial dengan data training yang terbatas. Optimasi dilakukan melalui preprocessing, vektorisasi, dan tuning *hyperparameter*. Hasilnya, model mencapai *F1-score* 71% pada data validasi dan 59% pada data uji, menunjukkan efektivitas pendekatan ini dalam menangani isu politik seperti pengangkatan Kaesang sebagai ketua PSI.

Penelitian yang dilakukan Joni Pranata [13] menunjukkan bahwa optimasi model klasifikasi sentimen dengan data terbatas dapat meningkatkan performa secara signifikan. Dengan menggabungkan embedding IndoBERT, optimasi praproses, penambahan data eksternal, dan tuning parameter pada algoritma Random Forest, model mampu mencapai peningkatan *F1-score* sebesar 6%. Hasil ini membuktikan bahwa pendekatan tersebut efektif untuk meningkatkan akurasi dan generalisasi model pada bahasa Indonesia. Selain itu, penggunaan IndoBERT memberikan kontribusi besar dalam kualitas representasi kata.

Penelitian Yazid Abdullah Subhi [14] menunjukkan bahwa penggunaan model BERT sebagai fitur input dalam metode Passive Aggressive dapat meningkatkan performa klasifikasi sentimen, khususnya pada dataset kecil. Penggunaan BERT menghasilkan *F1-score* 0.52, lebih baik dibandingkan dengan TF-IDF yang hanya memperoleh *F1-score* 0.42. Penambahan data eksternal dan eksplorasi teknik ekstraksi fitur lain juga berkontribusi signifikan terhadap peningkatan performa. Pendekatan transfer learning dengan BERT memperlihatkan kemampuan yang lebih baik dalam menangani konteks teks yang kompleks. Saran penelitian selanjutnya adalah memperluas dataset dan mengeksplorasi *fine-tuning* serta metode ensembel untuk peningkatan lebih lanjut.

Pada penelitian yang telah dilakukan oleh Yoga El Saputra [15] menunjukkan bahwa metode *SVM* dengan pendekatan TF-IDF efektif untuk mengklasifikasikan sentimen terhadap pengangkatan Kaesang Pangarep sebagai Ketua Umum PSI. Model ini menghasilkan F1 Score sebesar 0.53, akurasi 0.62, presisi 0.52,

dan *recall* 0.57. Penambahan data eksternal terkait Covid-19 meningkatkan kinerja model. Meskipun demikian, ukuran dataset yang kecil dan potensi bias dalam anotasi data menjadi keterbatasan utama. Penelitian selanjutnya disarankan untuk menggunakan dataset lebih besar dan menggabungkan teknik pemrosesan teks lanjutan seperti *word embeddings* atau *deep learning*.

Sedangkan pada peneliatian yang dilakukan oleh Safrizal [16] bahwa metode SVM dengan representasi fitur *FastText* dan kernel *RBF* mampu mengklasifikasikan sentimen publik terhadap Kaesang Pangarep secara efektif, meskipun data *training* awal terbatas. Penambahan dataset eksternal dari isu Covid-19 berhasil meningkatkan performa model. Eksperimen terbaik (ID C2) menghasilkan *F1-Score* sebesar 53.59% dan akurasi 62.73% pada data uji. Proses *grid search* dan validasi silang turut berperan dalam menghasilkan model optimal. Hasil ini menegaskan pentingnya representasi fitur berbasis *word embedding* dan perluasan data untuk meningkatkan performa klasifikasi sentimen.

2.3. Pengumpulan Dataset

Data Kaesang0 dan Kaesang1 dikumpulkan menggunakan teknik *crawling* untuk mendapatkan data twitter yang merepresentasikan sentiment *public* terhadap Kaesang Pangarep, yang ditunjuk sebagai ketua umum PSI, seperti yang dirinci dalam penelitian [17]. Proses *crawling* dilakukan secaraotomatis dengan menggunakan Bahasa pemrograman *Python*. Katakunci yang digunakan antara lain "Kaesang Pangarep","Kaesang Pangarep ketua umum PSI", "Ketua Umum PSI", "Partai Solidaritas Idonesia", dan kata kunci terkait lainnya seputar Kaesang Pangarep yang menjadi Ketua umum PSI. Penelitian ini memanfaatkan data *eksternal* yaitu datacovid dan data open *topic* untuk menambah dan memenuhi kebutuhan jumlah data *t raining*. Data covid yang tersedia berjumlah 8000 data dan data open topic berjumlah 7569 data. Detail mengenai dataset yang digunakan untuk data latih dan uji, beserta distribusinya, disajikan di Tabel 1[18].

Tabel 1. Benchmark Dataset

No	Dataset	Penggunaan	Jumlah Sampel Data	Distribusi kelas		
				Positif	Negatif	Netral
1	*Dataset* Kaesang0	*Training*	300	100	100	100
2	*Dataset* Kaesang1	*Training*	300	100	100	100
3	*Dataset* Covid	*Training*	8000	463	6664	873
4	*Dataset Open Topic*	*Training*	7569	1505	3408	2626
5	*Dataset* Kaesang	*Testing*	924			

2.4. Manajemen Data Set

Sebagai acuan penilaian, dataset untuk melatih metode pembelajaran mesin disajikan peneliti beserta sejumlah data uji. Dengan adanya data ini, performa setiap metode dapat dievaluasi dan dibandingkan. Pada tahap ini, data yang akan digunakan telah dikompilasi dan diperiksa apakah ada informasi yang hilang atau tidak akurat.Selanjutnya, analisis data dilakukan untuk menemukan *variable* yang dapat mempengaruhi akurasi penelitian. Dataset akhir dari data pelatihan penelitian ini terdiri dari 300 tweet, dengan 100 tweet perkelas atau opsi untuk mencampurnya untuk hasil yang optimal. Tiga kategori tersebut adalah positif, netral, dan negatif. Data pelatihan, validasi, dan data uji dipisahkan sehingga data yang berbeda dapat digunakan untuk menguji model selama pengujian

2.5. Text Processing

Preprocessing teks, yang meliputi *case folding, tokenizing, filtering,* dan *stemming,* merupakan langkah dalam proses pemilihan data analisis sentimen. Keakuratan klasifikasi analisis sentimen dapat dipengaruhi oleh hasil dari langkah prapemrosesan. Dalam teknik dan aplikasi *text mining*, langkah *preprocessing* sangat penting Tahapan *preprocessing* yang dilakukan:

1. *Cleaning*
   Cleaning merupakan proses menghilangkan karak terkhusus,tanda baca,atau symbol seperti menghapus karakter *noise* seperti tautan (*hyperlink*), tanda "@",tanda pagar (tagar,#) dan lainnya yang tidak relevan atau mengganggu dalam analisis [19].

2. *Case Folding*
   Case folding adalah salah satu teknik dalam pra-pemrosesan teks yang bertujuan untuk menyamakan huruf kapital dan kecil dalam teks dengan mengonvers iseluruh huruf menjadi huruf kecil *(lowercase)* [20].

3. *Tokenizing*
   Tokenizing yaitu membagi teks menjadi bagian-bagian kecil,seperti kalimat atau kata-kata, disebut tokenisasi. Ini adalah salah satu tugas dasar dalam pemrosesan Bahasa alami.Token bisa berupa

karakter, kata, atau bagian dari kata. Karena itu, tokenisasi bisa dibagi menjadi tiga jenis: level karakter, level kata, dan level sub-kata [21].

4. *Normalization*
   Normalization adalah bentuk normalisasi yang memetakan data kedalam interval [0,1]. Buat objek yang dinormalisasi menggunakan varians rata-rata, normalkan menggunakan set pelatihan, catat parameter yang diperlukan, lalu gunakan atribut set pelatihan untuk menstandarkan set pengujian. Jelaskan rentang distribusi data, termasuk standar deviasi, varians, dan sebagainya [22].

5. *Stopword removal*
   Proses menghilangkan kata umum seperti "dan", "atau", "yang", dan kata lain yang tidak secara signifikan memajukan pemahaman teks dikenal sebagai stopword. Kata-kata tanpa komponen sentimental ditampilkan dalam *stopword* [23].

6. *Stemming*
   Ditahap *stemming,* seluruh kata ditransformasikan keformat dasarnya. Algoritma *stemming* yang diterapkan ialah *Enhanced Confix Stripping (ECS).*

2.6. TF-IDF

TF atau Term Frequency mengacu pada frekuensi suatu kata timbul pada suatu teks, sementara *IDF* adalah *indeks* frekuensi dokumen terbalik. Ide dasar dibalik TF-IDF adalah bahwa kata yang timbul lebih sering pada suatu dokumen dan minim muncul di dokumen lainnya seharusnya dipandang lebih krusial karena mereka lebih berguna untuk klasifikasi [24]. TF-IDF terbukti menjadi metode yang sangat andal dalam analisis teks,membantu model MLP mencapai tingkat akurasi cukup tinggi di bandingkan dengan model lain seperti yang di jelaskan pada penelitian [25]. Pada dasarnya untuk perhitungan atau rumus *TF-IDF* terbagi menjadi dua yaitu *Term Frequency* (TF) dan *Inverse Document Frequency* (IDF) dengan rumus dan cara kerja yang berbeda akan digabungkan diakhir perhitungan antara *TF* dan *IDF* sebagai berikut[10].

1. TF: Menilai frekuensi suatu kata (t) timbul pada suatu dokumen (d). Semakin banyak frekuensi kata tersebut timbul dalam dokumen, akan kian besar skor TF-nya.

$$tf_{t,d} = \frac{n_{t,d}}{(\text{Total number o0 term 1n document})} \tag{1}$$

2. IDF: Menilai pentingnya pada semua koleksi dokumen (corpus). Kata yang sering timbul di banyak dokumen bisa mempunyai IDF yang rendah, sementara kata yang jarang timbul di banyak dokumen bisa mempunyai *IDF* yang tinggi.

$$idf_d = log \frac{Number\ o0\ document}{(Total\ number\ o0\ term\ 1n\ document)} \tag{2}$$

3. TF-IDF: Ialah capaian perkalian antara TF dan IDF pada suatu kata (t) yang ada di dokumen (d). Tujuannya adalah untuk menyediakan bobot yang lebih besar pada kata yang sering timbul di dalam dokumen namun jarang timbul di seluruh dokumen lainnya.

$$tfidf_{t,d} = tf_{t,d} \times idf_d \tag{3}$$

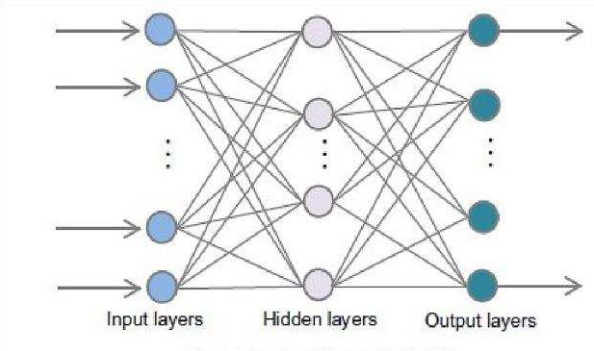
Penerapan Metode TF-IDF Menggunakan *Library Scikit-learn*: Untuk melakukan *ekstraksi* fitur dari data teks, penelitian ini menggunakan metode TF-IDF yang diimplementasikan melalui *library Scikit-learn (sklearn)* pada *Python*. Metode ini bertujuan untuk memberikan bobot pada setiap kata dalam dokumen berdasarkan tingkat kepentingannya, dengan mempertimbangkan *frekuensi* kemunculan kata dalam dokumen tertentu dibandingkan dengan seluruh korpus dokumen.

2.7. Parameter Tuning

Sebagai alternatif, kita bisa menggunakan strategi optimasi parameter *tuning* untuk meningkatkan performa model *machine learning* dari *dataset* yang sedikit [27]. Strategi ini adalah cara menyesuaikan parameter model berdasarkan data agar model bisa meminimalkan kesalahan dan bekerja lebih baik. Dalam proses ini, berbagai kombinasi pengaturan parameter diuji untuk menemukan konfigurasi yang memberikan hasil terbaik. Biasanya, pertama-tama model dianalisis menggunakan pengaturan parameter *default,* lalu dibandingkan dengan hasil yang didapat setelah dilakukan *hyperparameter tuning* [28].

2.8. Multi-layer Perceptron classifer

Multi-layer Perceptron Classifier (MLP *Classifier*) bergantung pada *neural network* yang mendasarinya untuk melakukan klasifikasi. Pengklasifikasi MLP mengimplementasikan algoritma MLP dan melatih Jaringan Syaraf menggunakan *back propagation* [29]. Pada dasarnya, MLP adalah model *feed-forward* meliputi satu lapisan input, satua tau lebih *hidden layer* dan satu lapisan *out put* seperti pada Gambar 2.



Gambar 2. *MLPClassifier*

Umumnya, jumlah *node* lapisan input tergantung pada faktor yang dipilih dalam sumber data, dan jumlah *neuron* tersembunyi dikuantifikasi berdasarkan dataset pelatihan tertentu. *Hidden layer* digunakan untuk komputasi, dan lapisan keluaran mewakili tujuan pemodelan [30]. Terdapat lebih dari satu *hidden layer,* setiap *node* dari *hidden layer* harus terhubung dengan semua *node* dari lapisan *input,* dan kemudian setiap *node* dari lapisan *output* harus terhubung ke semua *node* di lapisan yang tersembunyi [31].

2.9. Implemntasi

Implementasi merupakan tahap krusial dalam penelitian, di mana penerapan model dilangsungkan menurut temuan analisis serta penyusun yang sudah disusun. Tahapan ini melibatkan pemanfaatan berbagai alat dan teknologi, salah satunya adalah Google *Colab*, yang digunakan sebagai *platform* pendukung untuk memfasilitasi proses pengolahan data, pelatihan model, dan evaluasi hasil secara efektif dan efisien.

2.10. Evaluasi

Pada tahap evluasi, dilakukan pengukuran kinerja model yang telah dibangun dengan menghitung tingkat akurasi menggunakan MLP *Classifier Report.* MLP *Classifier Report* adalah keluaran evaluasi kinerja model MLP *Classifier* dalam menyelesaikan tugas klasifikasi. Laporan ini biasanya melibatkan metrik seperti *Precision, Recall, F1-Score,* dan Akurasi yang membantu dalam memahami seberapa baik model melakukan prediksi pada data uji.

1. *Precision* (Presisi): menilai proporsi prediksi yang relevan diantara seluruh prediksi positif. *Inter*pretasi dari *Precision* adalah tinggi jika model menghasilkan sedikit prediksi positif yang salah. Ditunjukkan pada persamaan 4.

$$Precision = \frac{True\ Positif\ (TP)}{True\ Positif\ TP+False\ Positif\ (FP)} \tag{4}$$

2. *Recall* (Sensitivitas atau *True Positive Rate*) : menilai proporsi prediksi positif yang relevan diantara seluruh kejadian positif sebenarnya. Interpretasi dari *recall* adalah tinggi jika model mampu menemukan sebagian besar kasus positif. Ditunjukkan pada persamaan 5.

$$Recall = \frac{True\ Positif\ (TP)}{True\ Positif\ TP+False\ Negatif\ (FN)} \tag{5}$$

3. *F1-Score*: rata-rata harmonis antara *Precision* dan *Recall*, memberikan keseimbangan antara keduanya. *Interpretasi* dari *F1-Score* adalah berguna ketika ada ketidak seimbangan antara kelas positif dan negatif. Ditunjukkan pada persamaan 6.

$$F1\ Score = \frac{Precision \times Recall}{Precision+Recall} \tag{6}$$

4. Akurasi: proporsi total prediksi yang benar. *Interpretasi* dari akurasi adalah tinggi jika model berhasil memprediksi sebagian besar data dengan benar. Ditunjukkan pada persamaan 7.

$$Akurasi = \frac{Jumlah\ Prediksi\ Benar}{Jumlah\ Total\ Data} \quad (7)$$

3. HASIL DAN PEMBAHASAN

3.1. Data Set

Tahap berikutnya adalah pembagian data ke dalam set data pelatihan dan pengujian. *Crawling* di Twitter digunakan untuk mengumpulkan data Tweet tentang Kaesang, yang secara keseluruhan berjumlah 1.524 Tweet setelah melalui tahap seleksi data. Setelah itu, data dipisahkan menjadi dua bagian: Data Latih sebanyak 300 Tweet (dengan label kaesang0 dan kaesang1) dan Data Uji sebanyak 924 Tweet.Tujuan dari pembagian data ini adalah untuk digunakan sebagai data latih. Selain itu, data train kaesang0 yang tersusun atas 300 Tweet dapat dipisahkan lagi menjadi 80% data pelatihan dan 20% data validasi untuk menentukan model terbaik. Meskipun jumlah data pelatihan yang digunakan sangat sedikit, pilihan ini mungkin dibuat karena beberapa alasan, termasuk keterbatasan sumber daya atau untuk mencegah model menjadi overfit. Penelitian ini juga menggunakan data tambahan dari luar, termasuk *Open Topic* Dataset dan Covid Dataset.

Tabel 2. Contoh hasil *crawling dataset*

No	Tweet	Label
1	@psi_id @kaesangp Asli ini re-marketing @psi_id ke ibu-ibu dan wanita bagus banget... Lgsg salfok sama baju imutnya kaesang. Ini kena banget dan politik jd adeeemmmmm banget... Ngga ada kalimat kasar, vulgar, caci maki, dsb... Mantap...	Positif
2	@bangherwin Banyak x cakap @adearmando61 basiii. Yang jelas selama ente di @psi_id ga bisa derek parpol itu naik. Harus ada Kaesang.	Netral
3	@abdulmukti691 Kaesang itu hanya boneka PSI untuk mendongkrak suara saja.	Negatif
4	@awe_adhi @Uki23 Ya nggak apa toh, apalagi Kaesang juga punya pengalaman berorganisasi dan bisnis. PSI kan bukan partai besar bisa jadi tempat tumbuh belajar.	Positif
5	@FrankMi61711987 @psi_id @uki Jilid 1 PDIP dibesarkan oleh Jokowi Jilid 2 PSI dibesarkan oleh Kaesang.....kita lihat nanti apa yang terjadi	Netral

Analisis klasifikasi sentimen dilakukan menggunakan data Tweet yang telah diklasifikasikan berdasarkan sentimen. Opini yang mendukung Kaesang menjadi Ketua Umum PSI tercermin dalam emosi positif.Sementara itu, sentimen netral mewakili opini yang tidak cenderung ke arah pandangan positif maupun negatif. Adapun sentimen negatif menggambarkan pemikiran yang kritis atau tidak setuju terhadap pencalonan atau kepemimpinan Kaesang.

3.2. Text Processing

Tabel 3 di bawah ini menunjukkan tahapan *preprocessing* yang diterapkan pada data teks untuk analisis lebih lanjut. *Preprocessing* bertujuan untuk membersihkan, menyederhanakan, dan menormalkan teks agar lebih mudah diproses oleh algoritma analisis data atau *machine learning*. Setiap tahap memiliki fungsi spesifik untuk mengubah teks mentah menjadi format yang lebih terstruktur dan relevan. Berikut adalah penjelasan masing-masing tahap.

Tabel 3. *Text Preprocessing*

No	Preprocessing	Sebelum	Sesudah
1	Cleaning	"Pengamat politik Rocky Gerung memberikan komentarnya terkait Kaesang Pangarep yang menjadi Ketua Umum Partai Solidaritas Indonesia (PSI). https://t.co/3wDBLz00CU https://t.co/f7YlAWkuNO"	"Pengamat politik Rocky Gerung memberikan komentarnya terkait Kaesang Pangarep yang menjadi Ketua Umum Partai Solidaritas Indonesia PSI"
2	Case folding	"Pengamat politik Rocky Gerung memberikan komentarnya terkait Kaesang Pangarep yang menjadi Ketua Umum Partai Solidaritas Indonesia PSI"	"pengamat politik rocky gerung memberikan komentarnya terkait kaesang pangarep yang menjadi ketua umum partai solidaritas indonesia psi"
3	Tokenizing	"pengamat politik rocky gerung memberikan komentarnya terkait kaesang pangarep yang menjadi ketua umum partai solidaritas indonesia psi"	["pengamat", "politik", "rocky", "gerung", "memberikan", "komentarnya", "terkait", "kaesang",

No	Preprocessing	Sebelum	Sesudah
			"pangarep", "yang", "menjadi", "ketua", "umum", "partai", "solidaritas", "indonesia", "psi"]
4	Normalisasi	"komentarnya, terkait"	"komentar, terkait"
5	Stopword Removal	["pengamat", "politik", "rocky", "gerung", "memberikan", "komentarnya", "terkait", "kaesang", "pangarep", "yang", "menjadi", "ketua", "umum", "partai", "solidaritas", "indonesia", "psi"]	["pengamat", "politik", "rocky", "gerung", "komentar", "terkait", "kaesang", "pangarep", "menjadi", "ketua", "umum", "partai", "solidaritas", "indonesia", "psi"]
6	Stemming	["pengamat", "politik", "rocky", "gerung", "komentar", "terkait", "kaesang", "pangarep", "menjadi", "ketua", "umum", "partai", "solidaritas", "indonesia", "psi"]	["pengamat", "politik", "rocky", "gerung", "komentar", "terkait", "kaesang", "pangarep", "ketua", "partai", "solidaritas", "indonesia", "psi"]

Tabel 3 menunjukkan proses transformasi *teks* melalui beberapa tahap *text preprocessing*, yaitu *cleaning, case folding, takenizing, normalization, hanling, stopword removal* dan *stemming*. Langkah-langkah ini sangat penting untuk meningkatkan akurasi dan kinerja model klasifikasi sentimen dalam penelitian ini, karena memungkinkan teks menjadi lebih bersih dan konsisten sehingga model dapat mengenali pola-pola dengan lebih efektif.

3.3. Set Up Eksperimen

Tujuan dari *eksperimen* ini adalah untuk menemukan model MLP terbaik yang mampu menghasilkan performa maksimal. Dalam prosesnya, akan dilakukan beberapa percobaan untuk mengevaluasi efek dari implementasi berbagai teknik *preprocessing teks*, seperti *stemming*, penghapusan *stopword*, dan normalisasi. Kombinasi tahapan preprocessing tersebut dijelaskan secara lengkap pada Tabel 4 di bagian Pengaturan *Eksperimen*.

Tabel 4. Kombinasi *Teks Prosesing*

Nomor Eksperimen	Normalisasi	Stopword Removal	Steming
1	No	No	No
2	No	No	Yes
3	No	Yes	No
4	No	Yes	Yes
5	Yes	No	No
6	Yes	No	Yes
7	Yes	Yes	No
8	Yes	Yes	Yes

Tabel 4 menunjukkan berbagai kombinasi langkah *preprocessing teks* dengan status *"Yes"* atau *"No"* untuk setiap teknik, termasuk *Cleaning, Case Folding, Tokenizing, Normalisasi, Stopword Removal,* dan *Stemming*. Setiap baris dalam tabel menggambarkan kombinasi yang berbeda, mulai dari tanpa *preprocessing* sama sekali (semua bernilai *"No"*) hingga penerapan semua langkah *preprocessing* (semua bernilai *"Yes"*). Tujuan dari variasi ini adalah untuk mengevaluasi dampak masing-masing teknik *preprocessing* terhadap kinerja model MLP.

3.4. Parameter Tuning

Sebagai bagian dari pengembangan model yang optimal, dilakukan proses parameter tuning menggunakan kombinasi parameter tertentu. Parameter tuning ini bertujuan untuk menemukan kombinasi terbaik dari arsitektur model, fungsi aktivasi, algoritma optimasi,tingkat regulasi, dan jumlah iterasi maksimal, berikut menunjukkan konfigurasi parameter yang digunakan untuk proses tuning pada model yang dikembangkan.

Hiddenlayersize :[(10,20),(10,8),(5,3)]
Activation :[‘relu’,‘tanh’]
Solver :[‘adam’,‘sgd’]
Alpha :[0,01,0,001]
MaxIter :[20,50]

Kombinasi *hyper* parameter yang diuji meliputi arsitektur model dengan ukuran *hidden layer* (10,20), (10,8), dan (5,3), *activation functionrelu* dan *tanh*, algoritma *optimizationa*, dan dan *sgd*, nilai *regularization alpha* sebesar 0.01 dan 0.001, serta jumlah *maximum iterations* sebanyak 20 dan 50. Penyesuaian ini bertujuan untuk *mengeksplorasi* arsitektur model yang optimal, memilih *activation function* yang sesuai, algoritma *optimization* yang efisien, menerapkan *regularization* untuk mencegah *overfitting*, serta memastikan proses pelatihan dapat *konvergen* dengan jumlah iterasi yang memadai. Hasil dari proses tuning ini akan menjadi dasar untuk evaluasi kinerja model pada data set yang digunakan seperti table berikut.

Tabel 5. Kombinasi Parameter *Tuning Result*

Nomor Experimen	Best Parameters Found				
	Acivation	Alpha	HidenLayer	MaxIter	Solver
1	Tanh	0.01	(10,20)	20	Adam
2	Tanh	0.001	(10,20)	20	Adam
3	Tanh	0.01	(10,20)	50	Adam
4	Tanh	0.001	(5,3)	50	Adam
5	ReLU	0.001	(10,20)	20	Adam
6	Tanh	0.01	(10,8)	20	Adam
7	Tanh	0.01	(10,20)	50	Adam
8	Tanh	0.01	(10,8)	50	Adam
9	ReLU	0.001	(10,20)	20	Adam

3.5. Optimasi Parameter MLP Classifier berdasarkan Penelusuran Model Optimal

Pengelompokan dengan teknik MLP *Classifier* adalah langkah selanjutnya. Untuk mendapatkan nilai akurasi dan nilai f1score, data akan terlebih dahulu menjalani training dan testing sebelum dimodelkan dengan menggunakan pendekatan *Multi-layer Perceptron Classifier* (MLP *Classifier*).

3.6. Penelusuran Model Optimal

Dengan mengikuti *setup eksperimen* di atas, kami melakukan berbagai *eksperimen* dengan berbagai kombinasi tahapan optimasi sebagai bagian dari proses pengujian model. *Eksperimen* ini dirancang untuk mengevaluasi kinerja model terhadap data validasi menggunakan parameter yang telah ditentukan sebelumnya. Hasil dari setiap kombinasi tahapan optimasi disajikan dalam bentuk tabel untuk mempermudahan alisis dan *interpretasi*. Tabel 6 menunjukkan hasil pengujian model terhadap data validasi berdasarkan metrik evaluasi yang digunakan.

Tabel 6. Training dan Pengujian Model Data Validasi

Nomor Eksperimen	Penambahan Dataset Covid & Open Topic			Data Validasi	
	Positif	Negatif	Netral	F1Score	Accuracy
1	100	100	100	0.5571	0.5833
2	200	200	200	0.6220	0.6333
3	300	300	300	0.5356	0.5500
4	100	100	200	0.6220	0.6333
5	200	200	400	0.5772	0.6000
6	300	300	500	0.5144	0.5500
7	60	60	120	0.6247	0.6333
8	80	80	80	0.6070	0.6167
9 (Kaesang)	-	-	-	0.6119	0.6500
10 (K+Op)	60	60	120	0.6767	0.6667

Tabel 6 menunjukkan hasil pengujian model terhadap data validasi dengan berbagai *scenario* penambahan data pada masing-masing kelas (Positif, Negatif, Netral). Setiap *eksperimen* dilakukan dengan variasi jumlah data untuk melihat pengaruhnya terhadap performa model, yang diukur menggunakan metrik *F1-Score* dan *Accuracy*. Temuan menunjukkan bahwa kombinasi data dengan jumlah tertentu dapat meningkatkan kinerja model, seperti pada *eksperimen* 10 di mana model mencapai nilai *F1-Score* sebesar 0,6767 dan *Accuracy* 0,6667, yang merupakan performa terbaik diantara semua *eksperimen*. Hal ini menunjukkan pentingnya pemilihan jumlah data yang seimbang untuk setiap kelas dalam meningkatkan akurasi dan kemampuan model untuk mengklasifikasikan data secara efektif.

Tabel 7. Hasil Experimen Pada Perbandingan data Set

Eksperimen	Metode	F1Score	Accuracy	Precision	Recall
1.1	MLPTF-IDF(K0+K1)	0.6119	0.6500	0.6998	0.6409
7	MLPTF-IDF(K0+K1+Op+Cv)	0.6247	0.6333	0.6517	0.6277
7.2	MLPTF-IDF(K0+K1+Op)	0.6767	0.6667	0.7188	0.6670

Berdasarkan *F1Score, Accuracy, Precision,* dan *Recall*, hasil evaluasi dari beberapa uji coba model pada data validasi ditampilkan pada Tabel 7. Pada *eksperimen* (7.2) memberikan hasil yang paling unggul secara keseluruhan kinerja dengan *F1Score* 0.6767 *Accuracy* 0.6667 *Precision* 0.7188 *Recall* 0.6670, Sedangkan (1.1) memberikan hasil *F1Score* 0.6119 terendah *Accuracy* 0.6500 *Precision* 0.6998 *Recall* 0.6409 hanya memberikan kontribusi signifikan terhadap peningkatan kerja model. Pada *eskperimen* (7) memberikan hasil performa yang seimbang *F1Score* 0.6247 *Accurasy* 0.6333 *Precision* 0.6516 *Recall* 0.6266. Dengan demikian, pada *eksperimen* (7.2) adalah yang paling optimal dalam hal keseluruh kinerja.

Tabel 8. Hasil Konstitusi *Leaderboard*

Nama	Metode	F1Score	Accurasy	Precision	Recall
Abdurrahman Arasy	Mlp TF-IDF Vectorizer	0.5027	0.5915	0.5327	0.5851
Dina Deswara	Random Forest (KS1+KS)	0.4989	0.5829	0.4916	0.5647
Atika Putri	KNN+IndoBert	0.4938	0.5796	0.5079	0.5764
Admin	BaseLane	0.4038	0.4545	0.4953	0.488

Tabel 8 memperlihatkan capaian leaderboard dari empat partisipan melalui penerapan metode yang berbeda dalam memeproses data. Abdurrahman Arasy menggunakan metode *TFIDFVectorizer* dan berhasil memperoleh *F1Score* sebesar 0.5027 *Accurasy* 0.5915 *Precision* 0.5327 *Recall* 0.5851. Mengunakan metode *Random Forest* mencatat (K1+K2) *F1Score* sebesar 0.4989 *Accurasy* 0.5829 *Precision* 0.4916 *Recall* 0.5647, sedangkan menggunakan *KNN*+Indo*Bert* mencatat *F1Score* sebesar 0.4938 *Accurasy* 0.5796 *Precision* 0.5079 *Recall* 0.5764 dan menggunakan *BaseLane* berhasil mencatat *F1Score* sebesar 0.4038 *Accurasy* 0.4545 *Precision* 0.4953 *Recall* 0.488.

4. KESIMPULAN

Penelitian ini secara efektif mengklasifikasikan opini mengenai penunjukan Kaesang Pangarep sebagai Ketua Umum PSI dengan menggunakan metode MLP *Classifier* yang dikombinasikan dengan metodologi TF-IDF. Dari skenario *eksperimen* yang dilakukan, hasil terbaik dicapai pada *Eksperimen* 7 (7.2), dengan konfigurasi *hyperparameter* menggunakan *hidden layer* (10, 20), fungsi aktivasi *relu*, *solver Adam, alpha* sebesar 0,001, dan iterasi maksimum 20. Model ini menghasilkan nilai *F1Score* sebesar 0,5027 dan akurasi 0,5915. Meskipun masih ada peluang untuk peningkatan kinerja,terutama dibidang akurasi dan *F1 Score*, angka ini membuktikan pendekatan MLP *Classifier* menggunakan *TF-IDF* dapat mengklasifikasikan sentimen secara sangat efektif pada konteks dataset yang diterapkan. Optimasi lebih lanjut terhadap parameter model dan teknik *text preprocessing* dapat dipertimbangkan untuk penelitian dimasa mendatang.

Untuk meningkatkan akurasi dan kapasitas model untuk generalisasi, set data yang lebih banyak dan lebih bervariasi harus digunakan dalam penelitian selanjutnya.Penggunaan metode pemrosesan teks yang canggih, seperti *deep learning* atau penyematan kata, juga bisa dipertimbangkan dalam mengembangkan kinerja model. Meskipun studi ini menawarkan landasan yang kuat dalam analisis sentimen dengan MLP *Classifier* dan TF-IDF, masih tersedia ruang bagiperbaikan untuk meningkatkan luasnya analisis dan *output*.

REFERENSI

[1] S. Kanchan and A. Gaidhane, "Social Media Role and Its Impact on Public Health: A Narrative Review," *Cureus*, Jan. 2023, doi: 10.7759/cureus.33737.

[2] H. W. A. Hanley and Z. Durumeric, "Twits, Toxic Tweets, and Tribal Tendencies: Trends in Politically Polarized Posts on Twitter," Jul. 2023, [Online]. Available: http://arxiv.org/abs/2307.10349

[3] M. Rodríguez-Ibánez, A. Casánez-Ventura, F. Castejón-Mateos, and P.-M. Cuenca-Jiménez, "A review on sentiment analysis from social media platforms," *Expert Syst Appl*, vol. 223, p. 119862, 2023, doi: https://doi.org/10.1016/j.eswa.2023.119862.

[4] İ. Bucak, "Bayesian Inference-Recent Trends: Recent Trends," 2024.

[5] A. Al Bataineh, D. Kaur, and S. M. J. Jalali, "Multi-Layer Perceptron Training Optimization Using Nature Inspired Computing," *IEEE Access*, vol. 10, pp. 36963–36977, 2022, doi: 10.1109/ACCESS.2022.3164669.

[6] A. C. Cinar, "Training Feed-Forward Multi-Layer Perceptron Artificial Neural Networks with a Tree-Seed Algorithm," *Arab J Sci Eng*, vol. 45, no. 12, pp. 10915–10938, Dec. 2020, doi: 10.1007/s13369-020-04872-1.

[7] J. Asian, M. Dholah Rosita, and T. Mantoro, "Sentiment Analysis for the Brazilian Anesthesiologist Using Multi-Layer Perceptron Classifier and Random Forest Methods," *Jurnal Online Informatika*, vol. 7, no. 1, p. 132, Sep. 2022, doi: 10.15575/join.v7i1.900.

[8] I. Ali Kandhro, M. Ameen Chhajro, K. Kumar, H. N. Lashari, and U. Khan, "Student Feedback Sentiment Analysis Model Using Various Machine Learning Schemes A Review," *Indian J Sci Technol*, vol. 14, no. 12, pp. 1–9, Apr. 2019, doi: 10.17485/ijst/2019/v12i14/143243.

[9] A. K. Singh, S. Kumar, S. Bhushan, P. Kumar, and A. Vashishtha, "A Proportional Sentiment Analysis of MOOCs Course Reviews Using Supervised Learning Algorithms," *Ingénierie des systèmes d information*, vol. 26, no. 5, pp. 501–506, Oct. 2021, doi: 10.18280/isi.260510.

[10] E. R. N. Mustaqim, U. Pagalay, and C. Crysdian, "Prediksi Tingkat Kepercayaan Masyarakat Terhadap Pilpres 2024 Menggunakan TF-IDF Dan Bow Menggunakan Metode Svm," *Jurnal Cahaya Mandalika ISSN 2721-4796 (online)*, vol. 5, no. 1, pp. 515–530, Jun. 2024, [Online]. Available: https://www.ojs.cahayamandalika.com/index.php/jcm/article/view/3114

[11] A. Putri, "Eksplorasi Fitur Fasttext, Tf-Idf Dan Indobert Pada Metode K-Nearest Neighbor Untuk Klasifikasi Sentimen," *Jurnal Sistem Informasi*, vol. 7, no. 1, pp. 49–60, 2025, doi: 10.31849/zn.v7i1.24779

[12] R. Illahi, "Klasifikasi Sentimen Menggunakan Bidirectional Lstm Dan Indobert Dengan Dataset Terbatas," *Jurnal Sistem Informasi*, vol. 7, no. 1, pp. 74–84, 2025, doi: 10.31849/zn.v7i1.25091

[13] J. PRANATA, "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, pp. 1668–1676, 2025, doi: 10.47065/bits.v6i3.6335

[14] Y. A. Subhi, S. Agustian, M. Irsyad, and F. Insani, "Klasifikasi Sentimen Menggunakan Metode Passive Aggressive dengan Menggunakan Model Bahasa BERT pada Dataset Kecil," *Building of Informatics, Technology and Science (BITS)*, vol. 6, no. 3, pp. 1838–1847, Dec. 2024, doi: 10.47065/bits.v6i3.6389.

[15] Y. EL SAPUTRA, "Klasifikasi Sentimen SVM Dengan Dataset yang Kecil Pada Kasus Kaesang Sebagai Ketua Umum PSI," *KLIK: Kajian Ilmiah Informatika dan Komputer*, vol. 4, no. 6, pp. 2902–2908, 2024, doi: 10.30865/klik.v4i6.1944

[16] S. .Safrizal, S. Agustian, A. Nazir, and Y. Yusra, "Klasifikasi Sentimen Terhadap Pengangkatan Kaesang Sebagai Ketua Umum Partai PSI Menggunakan Metode Support Vector Machine," *Building of Informatics, Technology and Science (BITS)*, vol. 6, no. 1, Jun. 2024, doi: 10.47065/bits.v6i1.5340.

[17] D. A. Nurdeni, I. Budi, and A. B. Santoso, "Sentiment Analysis on Covid19 Vaccines in Indonesia: From The Perspective of Sinovac and Pfizer," in *2021 3rd East Indonesia Conference on Computer and Information Technology (EIConCIT)*, IEEE, Apr. 2021, pp. 122–127. doi: 10.1109/EIConCIT50028.2021.9431852.

[18] S. SAKTHI VEL, "Pre-Processing techniques of Text Mining using Computational Linguistics and Python Libraries," in *2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS)*, IEEE, Mar. 2021, pp. 879–884. doi: 10.1109/ICAIS50930.2021.9395924.

[19] 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," Mar. 29, 2021, *Frontiers Media S.A.* doi: 10.3389/fenrg.2021.652801.

[20] S. Khomsah and Agus Sasmito Aribowo, "Text-Preprocessing Model Youtube Comments in Indonesian," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 4, no. 4, pp. 648–654, Aug. 2020, doi: 10.29207/resti.v4i4.2035.

[21] D. Darwis, E. S. Pratiwi, and A. F. O. Pasaribu, "Penerapan Algoritma Svm Untuk Analisis Sentimen Pada Data Twitter Komisi Pemberantasan Korupsi Republik Indonesia," *Edutic - Scientific Journal of Informatics Education*, vol. 7, no. 1, Nov. 2020, doi: 10.21107/edutic.v7i1.8779.

[22] S. Sunny, S. Pinky, S. Jalal, M. Kayser, M. Wadud, and N. Mansoor, "Bangla E-Commerce Sentiment Analysis Optimization Using Tokenization and TF-IDF," in *2024 International Conference on Advances in Computing, Communication, Electrical, and Smart Systems (iCACCESS)*, IEEE, 2024, pp. 1–6. doi: 10.1109/iCACCESS61735.2024.10499476

[23] S. Akuma, T. Lubem, and I. T. Adom, "Comparing Bag of Words and TF-IDF with different models for hate speech detection from live tweets," *International Journal of Information Technology*, vol. 14, no. 7, pp. 3629–3635, 2022, doi: 10.1007/s41870-022-01096-4

[24] E. Elgeldawi, A. Sayed, A. R. Galal, and A. M. Zaki, "Hyperparameter tuning for machine learning algorithms used for arabic sentiment analysis," in *Informatics*, MDPI, 2021, p. 79. doi: 10.3390/informatics8040079

[25] A. M. Goh and X. L. Yann, "A Novel Sentiments Analysis Model Using Perceptron Classifier," *International Journal of Electronics Engineering and Applications*, vol. 10, no. 2, pp. 01–10, Sep. 2021, doi: 10.30696/IJEEA.IX.IV.2021.01-10.

[26] J. Naskath, G. Sivakamasundari, and A. A. S. Begum, "A study on different deep learning algorithms used in deep neural nets: MLP SOM and DBN," *Wirel Pers Commun*, vol. 128, no. 4, pp. 2913–2936, 2023.

[27] B. T. Pham, M. D. Nguyen, K.-T. T. Bui, I. Prakash, K. Chapi, and D. T. Bui, "A novel artificial intelligence approach based on Multi-layer Perceptron Neural Network and Biogeography-based Optimization for predicting coefficient of consolidation of soil," *Catena (Amst)*, vol. 173, pp. 302–311, Feb. 2019, doi: 10.1016/j.catena.2018.10.004.