ZONAsi Jurnal Sistem Informasi
ZONAsi VOL. 7 NO. 1 JANUARI 2025
Page: 49 - 60
ISSN: 2656-7407 (Online)
2656-7393 (Print)

EKSPLORASI FITUR FASTTEXT, TF-IDF DAN INDOBERT PADA METODE K-NEAREST NEIGHBOR UNTUK KLASIFIKASI SENTIMEN

Atika Putri[1], Surya Agustian[2*], Jasril[3], Iis Afrianty[4]
[1,2,3,4]Fakultas Sains dan Teknologi, Teknik Informatika, Universitas Islam Negeri Sultan Syarif Kasim Riau, Pekanbaru, Indonesia
12050124460@students.uin-suska.ac.id[1], Surya.agustian@uin-suska.ac.id[2*],
jasril@uin-suska.ac.id[3], iis.afrianty@uin-suska.ac.id[4]

Abstrak

Klasifikasi sentimen penting untuk menganalisis opini publik, terutama pada isu-isu di media sosial. Salah satu tantangan utama dalam klasifikasi sentimen adalah terbatasnya jumlah data training yang sering kali memengaruhi performa model dalam menghasilkan prediksi yang akurat. Penelitian ini mengkaji pengangkatan Kaesang Pengarep sebagai ketua PSI dengan metode ekstraksi fitur seperti FastText, TF-IDF, dan IndoBERT, serta algoritma K-Nearest Neighbor (KNN). Langkah optimasi meliputi penambahan data eksternal, konfigurasi preprocessing teks, scaling data, dan pencarian parameter terbaik. Model baseline mencapai akurasi 44% dan F1-score 39% dengan FastText. Setelah optimasi dan mengganti ke IndoBERT, model optimal mencapai akurasi 57% dan F1-score 49%, meningkat 10%. Hasil ini menunjukkan bahwa optimasi, seperti ekstraksi fitur canggih dan tuning parameter, memiliki dampak signifikan pada klasifikasi sentimen. Penelitian selanjutnya dapat fokus pada teknik optimasi lanjutan untuk mengatasi keterbatasan data dan meningkatkan performa analisis sentimen.

Kata kunci: *Klasifikasi Sentimen, Optimasi Model, K-Nearest Neighbor, FastText dan TF-IDF, IndoBERT*

Abstract

Sentiment classification is important for analysing public opinion, especially on issues on social media. One of the main challenges in sentiment classification is the limited amount of training data which often affects the performance of the model in producing accurate predictions. This research examines the appointment of Kaesang Pengarep as PSI chairman with feature extraction methods such as FastText, TF-IDF, and IndoBERT, and the K-Nearest Neighbor (KNN) algorithm. Optimisation steps include adding external data, configuring text preprocessing, scaling data, and finding the best parameters. The baseline model achieved 44% accuracy and 39% F1-score with FastText. After optimisation and changing to IndoBERT, the optimal model achieved 57% accuracy and 49% F1-score, a 10% improvement. These results show that optimisation, such as advanced feature extraction and parameter tuning, has a significant impact on sentiment classification. Future research can focus on advanced optimisation techniques to overcome data limitations and improve sentiment analysis performance.

Keywords: *Sentiment Classification, Model Optimisation, K-Nearest Neighbor, FastText, and TF-IDF, IndoBERT.*

1. PENDAHULUAN

Isu politik saat ini menjadi hal yang sangat menarik perhatian. Opini publik terhadap isu politik adalah sesuatu yang sangat penting bagi pihak yang terkait, mengetahui pandangan masyarakat terhadap isu politik memengaruhi strategi komunikasi pihak terkait dalam membangun citra mereka dalam politik[1]. Opini publik yang disampaikan masyarakat berupa respons dalam bentuk teks terhadap suatu isu atau peristiwa tertentu yang banyak didapatkan melalui salah satu platform media sosial, yaitu Twitter. Untuk memastikan pendapat atau opini tersebut bisa bermanfaat, beberapa proses perlu dilakukan agar informasi penting bisa didapatkan melalui klasifikasi sentimen[2].

Klasifikasi sentimen dilakukan untuk mengklasifikasikan teks dari media sosial berdasarkan sentimen yang terkandung di dalamnya, serta mengidentifikasi apakah opini tersebut cenderung positif, netral, atau negatif [3]. Saat ini telah banyak diperkenalkan metode Machine learning untuk mengukur sentimen terhadap suatu isu dari media sosial, salah satunya adalah metode *K-Nearest Neighbor* (KNN). KNN memiliki keunggulan sebagai algoritma yang sederhana dan fleksibel [4] Namun, KNN juga memiliki kelemahan dalam kebutuhan penentuan parameter. Meskipun demikian, KNN tetap menarik untuk diteliti, karena kemudahannya dalam implementasi dan kemampuannya untuk menghasilkan hasil yang cukup baik dalam analisis sentimen dari data sosial media [5]. Masih banyak ruang yang dapat dikembangkan dari metode KNN agar kompetitif setidaknya dengan metode ML konvensional lainnya. Dalam penelitian sentimen terhadap program vaksinasi COVID-19, KNN hanya menempati urutan di level menengah ke bawah dari beberapa penelitian [6].

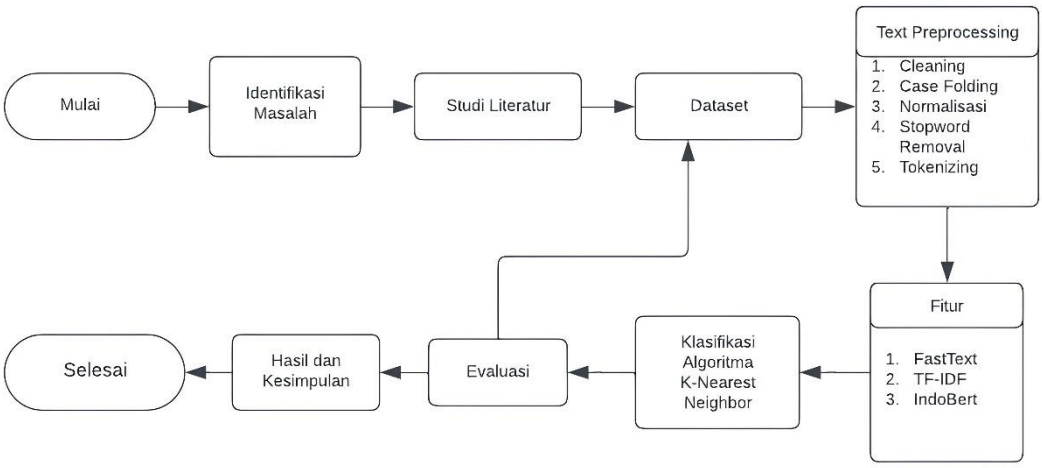
Penelitian ini merujuk pada tantangan klasifikasi sentimen yang diberikan [7], berupa tantangan klasifikasi dengan terbatasnya jumlah data training, dikarenakan memberi label yang cukup pada data *training* akan membutuhkan waktu dan sumber daya yang cukup banyak. Namun, hal ini sering kali menyebabkan model menjadi tidak optimal dan menghasilkan prediksi yang tidak akurat pada data uji. Beberapa penelitian sebelumnya mengenai Kaesang dengan jumlah data Training terbatas dan beberapa ekstraksi fitur yang digunakan, yaitu penelitian [8] menggunakan metode *Support Vector Machine* (SVM) dengan ekstraksi fitur *FastText*, memberikan hasil terbaik pada data uji dengan F1-score sebesar 54,23%, akurasi 63,81%. Selanjutnya, penelitian [9] menggunakan metode *Support Vector Machine* (SVM) dan pendekatan *Term Frequency-Inverse Document Frequency* (TF-IDF). Model SVM diuji dan menunjukkan hasil F1 Score 0.51. Penambahan data eksternal pada penelitian ini terbukti meningkatkan kinerja model.

Penelitian ini akan berfokus pada penggunaan jumlah data training yang terbatas dengan mengeksplorasi beberapa metode ekstraksi fitur. Fitur yang akan dieksplorasi adalah model bahasa *bag of words* TF-IDF, *word embeddings* FastText, dan BERT *embeddings*. Sedangkan, studi kasus yang digunakan pada *shared task* ini adalah Pengangkatan Kaesang Pengarep, putra bungsu dari Presiden Joko Widodo sebagai Ketua Umum Partai Solidaritas Indonesia (PSI), yang banyak menimbulkan sikap pro dan kontra sehingga banyak memicu opini publik [10]. *FastText* digunakan karena memiliki keunggulan dalam mengatasi kata-kata yang tidak pernah ditemui sebelumnya [11], TF-IDF memberikan bobot yang lebih tinggi pada kata-kata yang lebih spesifik dan relevan untuk dokumen tertentu, sementara mengurangi bobot pada kata-kata yang umum dan kurang informatif [12], selanjutnya IndoBERT menunjukkan kinerja yang lebih baik dalam tugas-tugas pemahaman bahasa alami, dikarenakan model dilatih terlebih dahulu pada korpus teks yang sangat besar dan kemudian disesuaikan dengan tugas spesifik [13]

Selanjutnya, dilakukan proses optimasi seperti penambahan data dari topik yang berbeda (data eksternal), pencarian konfigurasi *text preprocessing* yang tepat, penerapan scaling data yang merujuk pada penelitian [14] bahwa penggunaan scaling data dapat memberikan kinerja yang baik, serta melakukan proses optimasi pada algoritma yang akan digunakan dengan mencari parameter terbaik, seperti nilai *K* dan metrik jarak (metrics). Proses tersebut dilakukan untuk mengetahui pengaruhnya terhadap peningkatan kinerja model pada algoritma machine learning yang akan digunakan.

Kontribusi penelitian ini adalah menunjukkan bagaimana langkah-langkah optimasi yang dilakukan untuk menangani masalah keterbatasan data *training*, dapat meningkatkan performa yang cukup signifikan bila dibandingkan dengan metode baseline (dasar) tanpa optimasi. Hal itu meliputi langkah optimasi untuk tahapan text preprocessing, langkah pemilihan fitur dengan BERT *embeddings* yang dipadukan dengan metode konvensional machine learning KNN, dan penambahan data training dengan data eksternal dengan topik berbeda.

2. METODE PENELITIAN



Gambar 1. Tahapan Penelitian

Gambar 1 menunjukkan beberapa tahapan utama pada penelitian. Tahap pertama adalah identifikasi masalah dan studi literatur untuk memahami konsep dasar serta metode yang relevan dalam klasifikasi sentimen. Selanjutnya, dilakukan analisis dataset untuk mengevaluasi struktur, kualitas, dan relevansi data dengan topik penelitian. Data kemudian diproses melalui tahapan *preprocessing*, termasuk *cleaning*, normalisasi, *stopword removal*, *case folding*, *dan tokenizing*. Setelah itu, dilakukan ekstraksi fitur menggunakan metode *FastText*, TF-IDF, dan IndoBERT, diikuti dengan klasifikasi menggunakan algoritma *K-Nearest Neighbor* (KNN) sebagai model awal. Setelah proses klasifikasi, dilakukan evaluasi model menggunakan metrik akurasi dan F1-score. Jika performa model belum optimal, penelitian akan kembali ke tahap analisis dataset untuk mencari strategi optimasi. Tahapan terakhir adalah penyajian hasil dan penarikan kesimpulan untuk merumuskan temuan penelitian.

2.1. Dataset

Dataset yang digunakan pada penelitian bersumber pada penelitian Agustian, dkk [7], berupa tantangan klasifikasi dengan data training terbatas. Data tersebut berupa Tweet yang diambil dari Twitter, yaitu data tentang pengangkatan Kaesang menjadi ketua umum PSI, yang pengumpulan datanya telah dilakukan dari tanggal September 2022 sampai dengan tanggal 03 Oktober 2022, dengan menggunakan kata kunci "Kaesang PSI". Data terbagi menjadi dua set, yaitu data *training* yang terbagi menjadi dua versi: Kaesang V1 dan Kaesang V2, lalu terdapat juga data uji dengan topik yang sama. Selanjutnya, data tentang sentimen vaksin Covid-19 yang akan digunakan sebagai data *training* tambahan, data tersebut didapatkan dari penelitian [6]. Rincian dataset yang digunakan dapat dilihat pada Tabel 1 berikut:

Tabel 1. Dataset

Dataset	Jumlah Tweet
Data Train Kaesang V1	300
Data Train Kaesang V2	300
Data Test Kaesang	924
Data Train Covid	8.000

Data kaesang V2 yang akan digunakan akan dibagi menjadi 2 subset yaitu data *training* dan data validasi. Data lainnya seperti data Kaesang V1 yang jumlahnya sama dengan data Kaesang V2 yaitu berjumlah 300 tweet, data Covid yang berjumlah 8000 tweet yang juga telah diberi label positif, netral, dan negatif akan dijadikan sebagai data tambahan untuk data *training*. Lalu, juga terdapat data Uji sebanyak 924 tweet yang mana akan dijadikan data testing.

Data Kaesang V2 akan dibagi menjadi 80% data *Training*, dan 20% data Validasi. Pembagian ini untuk mengevaluasi kinerja model pada data yang tidak digunakan selama pelatihan. Ini membantu dalam mengukur seberapa baik model akan bekerja pada data baru yaitu data validasi nantinya. Berikut merupakan contoh dataset yang telah diberi label positif, netral, dan negatif.

Tabel 2. Contoh Dataset

No	Tweet	Label
1	@gibran_gen @kaesangp Psi pasti semakin maju di tangan mas Kaesang	Positif
2	@HusinShihab @kaesangp @psi_id Habib Alwi itu prestasi Kaesang sudah bentuk pengkaderan berapa ratus ribu kader PSI bisa langsung jadi Ketum? apa karena anak Presiden atau apa yah? hanya nanya saja nih.â˜ºï¸_x008f_	Netral
3	@KompasTV Jokowi ketua umum PDI Kaesang ket umum PSI Gibran cawapres Jokowi gilanya sudah terlalu Rakyat tidak akan menerima Demokrasi total lumpuh Yg sengsara rakyat	Negatif

2.2. Text Preprocessing

Text preprocessing mengacu pada proses pembersihan dan penyiapan data teks sebelum digunakan untuk tugas analisis atau pembelajaran mesin [15]. Berikut beberapa tahapan yang dilakukan dalam proses text preprocessing:

a. *Cleaning*: Proses menghapus elemen-elemen yang tidak relevan dalam analisis teks, seperti username, URL, angka, tanda baca, dan emoji.
b. *Case Folding*: Proses mengubah semua huruf dalam teks menjadi huruf kecil.
c. Normalisasi: Proses mengubah berbagai bentuk kata menjadi bentuk dasar atau standar, menggunakan file CSV yang berisi pasangan kata asli dan kata yang telah dinormalisasi.
d. *Stopword Removal*: Proses penyaringan kata-kata umum dalam Bahasa Indonesia menggunakan library Natural Language Toolkit (NLTK). Contoh kata yang sering dihapus termasuk "dan", "untuk", dan "dari".
e. Tokenizing: Proses memecah teks menjadi unit-unit lebih kecil menggunakan metode Regex Tokenization, untuk memecah teks menjadi token berdasarkan pola tertentu.

Pada proses *text preprocessing* yang dilakukan pada penelitian mengharuskan data melewati serangkaian tahapan untuk memastikan kesiapan data untuk proses selanjutnya. Hasil dari beberapa tahapan yang telah dilakukan dapat dilihat pada Tabel 3 berikut:

Tabel 3. Hasil *Text Preprocessing*

No	Tahap	Sebelum	Sesudah
1	Cleaning	@GarethBJ4w4 @Uki23 @kaesangp @psi_id Mas Kaesang jd KETUM PSI , koq kamu yg frustasi ya ?	Mas Kaesang jd KETUM PSI koq kamu yg frustasi ya
2	Case Folding	Mas Kaesang jd KETUM PSI koq kamu yg frustasi ya	mas kaesang jd ketum psi koq kamu yg frustasi ya
3	Normalisasi	mas kaesang jd ketum psi koq kamu yg frustasi ya	mas kaesang jadi ketum pssi koq kamu yang frustasi ya
4	Stopword Removal	mas kaesang jadi ketum pssi koq kamu yang frustasi ya	mas kaesang ketum pssi koq frustasi ya
5	Tokenizing	mas kaesang ketum pssi koq frustasi ya	[‘mas’, ‘kaesang’, ‘ketum’, ‘pssi’, ‘koq’, ‘frustasi’, ‘ya’]

2.3 Ekstraksi Fitur

Ekstraksi fitur adalah proses penting dalam klasifikasi sentimen, yang bertujuan untuk mengubah data teks menjadi representasi numerik yang dapat diproses oleh algoritma pembelajaran mesin. Beberapa metode ekstraksi fitur yang digunakan dalam penelitian ini adalah *FastText*, TF-IDF, dan IndoBERT.

2.3.1. FastText

Model *FastText* [11] memecah kata menjadi n-gram karakter. *FastText* tidak hanya mempertimbangkan kata utuh, tetapi juga memecah kata menjadi n-gram karakter. Ini memungkinkan model untuk menghasilkan representasi kata bahkan untuk kata-kata yang tidak ada dalam kosakata pelatihan (*out-of-vocabulary*) dengan menjumlahkan vektor n-gram yang ada.

2.3.2. Term Frequency-Inverse Document Frequency (TF-IDF)

Term Frequency-Inverse Document Frequency (TF-IDF) [12] adalah sebuah metode untuk menilai relevansi sebuah kata dalam dokumen dengan memberikan skor numerik pada setiap kata yang menunjukkan seberapa penting kata tersebut dalam dokumen tertentu berdasarkan frekuensi kemunculannya. Berdasarkan [16] metode ini terdiri dari dua komponen utama, yaitu *Term Frequency* (TF), untuk menghitung seberapa sering sebuah kata muncul dalam dokumen. *Inverse Document Frequency* (IDF), untuk mengukur seberapa penting sebuah kata di seluruh kumpulan dokumen. Berikut persamaan-persamaan dalam menghitung TF-IDF:

$$W_{d,t} = tf_{d,t} \; x \; IDF_{d,t} \qquad (1)$$

2.3.3. Bidirectional Encoder Representations from Transformers (BERT)

Bidirectional Encoder Representations from Transformers (BERT) [17] menggunakan arsitektur Transformer untuk pre-training representasi bahasa dari teks yang tidak berlabel. BERT dilatih dengan dua tugas utama, yaitu *Masked Language Model* (MLM), di mana beberapa token dalam kalimat diacak dan model memprediksi token yang hilang, dan *Next Sentence Prediction* (NSP), yang memprediksi apakah satu kalimat mengikuti kalimat lainnya. Model representasi teks yang digunakan adalah IndoBERT. IndoBERT [18] adalah model bahasa BERT yang dilatih secara monolingual dengan sumber corpus dokumen bahasa Indonesia.

2.4. Scaling Data

Scaling data [14] merupakan proses yang digunakan untuk mengubah rentang nilai dari fitur-fitur dalam dataset. Proses ini penting karena algoritma machine learning seringkali sensitif terhadap skala data. Penelitian ini menggunakan dua metode scaling data, yaitu *Standar Scaler* dan *Robust Scaler* untuk meningkatkan kinerja algoritma.

a. Standar Scaler

Metode ini mengubah data sehingga memiliki rata-rata 0 dan deviasi standar 1. Ini berguna ketika data memiliki distribusi normal.

$$z = \frac{x - Q2}{Q3 - Q1} \qquad (2)$$

b. Robust Scaler

Metode ini menggunakan median dan rentang antar quartile range untuk mengurangi pengaruh outlier.

$$z = \frac{x - \mu}{\sigma} \qquad (3)$$

2.5. Klasifikasi K-Nearest Neighbor

Proses *K-Nearest Neighbor* akan menentukan label data berdasarkan mayoritas tetangga terdekat [4]. Metode yang digunakan dalam perhitungan jarak ialah *Euclidean* dan *Manhattan*. Jarak *Euclidean*, menghitung jarak antara dua titik dalam ruang n-dimensi. Sedangkan jarak *Manhattan*, menghitung jarak antara dua titik dengan menjumlahkan nilai absolut dari perbedaan koordinat [19]. Rumus berikut digunakan untuk menentukan jarak:

a. Jarak *Euclidean*

$$d = \sqrt{\sum_{i=1}^{p}(x_{2i} - x_{1i})^2} \quad (4)$$

b. Jarak *Manhattan*

$$d = \sum_{i=1}^{p}|x_{2i} - x_{1i}| \quad (5)$$

Keterangan:

x_1 = Data sampel
x_2 = Data uji atau data pengujian
i = Variabel data
d = Nilai jarak
p = Dimensi data

Performa *K-Nearest Neighbor* (KNN) sangat bergantung pada pemilihan parameter yang tepat. Parameter tuning menjadi langkah penting dalam mengoptimalkan akurasi model, terutama dalam menentukan jumlah tetangga terdekat (K) dan metode perhitungan jarak yang sesuai.

2.5.1. Parameter Tuning

Parameter tuning adalah proses penyesuaian parameter dalam model *machine learning* untuk mencapai kinerja yang optimal [20]. Sebagai salah satu langkah optimasi model, proses penyesuaian parameter akan dilakukan menggunakan metode *GridSearchCV* yang tersedia dalam Python untuk menentukan parameter terbaik. Pendekatan ini memungkinkan pencarian kombinasi parameter secara lebih sistematis dibandingkan dengan metode coba-coba secara manual [21].

Tabel 4. Parameter yang akan diuji

Parameter	Nilai
K	[1 – 30]
Metric	[euclidean, manhattan]

Tabel 4 menunjukkan parameter yang akan diuji menggunakan metode *GridSearchCV* yaitu pencarian parameter K dengan nilai 1 sampai dengan 30, dan pencarian Metric *euclidean* atau *manhattan*. Tahap ini bagian dari proses penyesuaian parameter untuk model.

2.6. Evaluasi

Confusion matrix digunakan untuk mengevaluasi kinerja model klasifikasi dalam machine learning. *Confusion matrix* memberikan gambaran yang jelas tentang jumlah prediksi yang benar dan salah [16]. F1-score digunakan sebagai skor resmi dalam penelitian. F1-score mengukur akurasi prediksi pada tiap kelas, F1-score yang tinggi menunjukkan keseimbangan antara precision dan recall, yang penting dalam evaluasi performa model [22].

$$\text{F1} = 2\,x\left(\frac{Precision\ x\ Recall}{Precision + Recall}\right) \quad (5)$$

3. HASIL DAN PEMBAHASAN

Bagian hasil dan pembahasan dalam penelitian ini menyajikan temuan dari eksperimen yang telah dilakukan.

3.1. Eksperimen Set Up

Eksperimen Set Up dilakukan untuk mencari langkah optimasi terbaik dalam meningkatkan performa klasifikasi sentimen. Proses ini melibatkan beberapa tahap eksperimen dengan skenario sebagai berikut:

1. Ekplorasi Fitur
   Penelitian ini dilakukan dengan tiga skenario utama, di mana setiap skenario menerapkan model *K-Nearest Neighbor* (KNN) dengan metode ekstraksi fitur yang berbeda. Ekstraksi fitur yang digunakan meliputi *FastText*, TF-IDF, dan IndoBERT.
2. Penerapan Model Baseline (Dasar)
   Model baseline digunakan sebagai acuan awal untuk mengevaluasi performa model klasifikasi sentimen sebelum diterapkannya langkah optimasi lebih lanjut. Pada tahap ini, model dijalankan menggunakan parameter default tanpa penyesuaian atau optimasi tertentu. Proses ini dimulai dengan penerapan konfigurasi text preprocessing, guna mengevaluasi pengaruh setiap tahapan terhadap performa model. Dataset yang digunakan pada tahap ini hanya data Kaesang V2 berjumlah 240 Tweet (telah malalui proses pembagian data *Training* dan data Validasi).
3. Penambahan/ Penggabungan Data *Training*
   Pada tahap ini, dilakukan upaya penambahan atau penggabungan data training untuk mengatasi keterbatasan jumlah data dan meningkatkan kinerja model klasifikasi sentimen. Proses ini dilakukan secara coba-coba dengan menggabungkan dataset utama yaitu data Kaesang V2 dengan data eksternal dari topik serupa maupun berbeda yaitu data Kaesang V1 dan data Covid. Penambahan data dilakukan dalam jumlah tertentu, yaitu dengan kelipatan 100 tweet untuk setiap label (positif, netral, dan negatif), hingga mencapai jumlah maksimal data yang tersedia. Hanya beberapa eksperimen yang dilakukan untuk mengevaluasi pengaruh penambahan data terhadap performa model,
4. Penerapan Scaling Data
   Pada penelitian ini, dua metode scaling data digunakan, yaitu *Standar Scaling* dan *Robust Scaling*. Proses scaling dilakukan untuk setiap ekstraksi fitur, seperti *FastText*, TF-IDF, dan IndoBERT, dengan tujuan mengevaluasi pengaruh scaling terhadap performa model KNN. Eksperimen dilakukan untuk melihat perbedaan hasil antara data tanpa scaling dan data dengan scaling.
5. Parameter Tuning
   Parameter tuning dilakukan untuk mengoptimalkan performa model KNN dengan menyesuaikan nilai parameter yang digunakan (dapat dilihat pada Tabel 4).

3.2. Baseline Model

Baseline ialah model dasar yang belum melalui proses optimasi, pada *baseline* semua parameter hanya menggunakan nilai *default* yang disediakan oleh *library*. Pada tahap ini, dilakukan berbagai konfigurasi pada proses text preprocessing untuk mengevaluasi pengaruh setiap tahapan dan menemukan komposisi yang paling tepat sehingga dapat menjadi acuan untuk tahap berikutnya. Dikarenakan karakteristik data tweet yang cenderung terdiri atas kalimat-kalimat tidak baku dan menggunakan ragam bahasa informal menjadi salah satu penyebab utama. Akibatnya, tidak semua tahapan *text preprocessing* mampu menghasilkan output yang optimal sesuai dengan tujuan yang diharapkan. Model baseline pada tahap ini akan diuji menggunakan data validasi.

Tabel 5. Pengujian Baseline pada Data Validasi

Normal-isasi	Stop-word Remo-val	Case Folding	FASTTEXT		TF-IDF		IndoBERT	
			Accuracy	F1 (%)	Accuracy (%)	F1 (%)	Accuracy (%)	F1 (%)
✓	✓	✓	52	52	50	49	48	48
✓	✓	-	52	52	50	49	50	50
✓	-	✓	**63**	**64**	**60**	**60**	42	42
-	✓	✓	55	55	55	54	52	52
-	-	✓	48	49	58	58	**53**	**54**

Tabel 5 menunjukkan hasil evaluasi kinerja model pada proses klasifikasi teks dengan beberapa konfigurasi *Text Preprocessing* dan ekstraksi fitur, yaitu *FastText*, TF-IDF, dan IndoBERT. *FastText* dan TF-IDF mencapai hasil tertinggi saat menggunakan konfigurasi ketika hanya menerapkan Normalisasi dan *Case Folding*, mengindikasikan bahwa keberadaan kata-kata umum dapat memperkaya informasi fitur bagi model. Sedangkan, IndoBERT memperlihatkan hasil yang lebih rendah secara keseluruhan, dengan performa tertinggi pada konfigurasi yang hanya menerapkan *Case Folding.*

3.3. Optimasi Model

Setelah memperoleh Konfigurasi *text preprocessing,* pada eksperimen model optimal hanya menggunakan Konfigurasi *Text Preprocessing* yang memiliki nilai F1 Score yang cukup tinggi. Langkah selanjutnya adalah melakukan proses penambahan data, scaling data, dan parameter tuning. Tahapan ini bertujuan untuk meningkatkan kinerja model secara keseluruhan dengan menyesuaikan parameter agar sesuai dengan karakteristik data, serta memperluas variasi data untuk memperkuat generalisasi model. Pada tahap ini penambahan data eksternal, penerapan scaling data serta penggunaan parameter tuning memiliki dampak yang berbeda pada performa masing-masing metode ekstraksi fitur, seperti terlihat pada Tabel 6-8 di bawah ini.

Tabel 6. Pencarian Model Optimal dengan Ekstraksi Fitur *FastText*

Komposisi Dataset	**Scaling Data**		**Parameter**		**Accuracy (%)**	**F1 (%)**
	Standar	**Robust**	**K**	**Metrics**		
Kaesang V2 + Covid	-	-	18	*euclidean*	42	42
	✓	-	29	*euclidean*	58	58
	-	✓	29	*euclidean*	58	59
Kaesang V2 + Kaesang V1	-	-	9	*euclidean*	**62**	**60**
	✓	-	17	*manhattan*	60	57
	-	✓	12	*manhattan*	60	56
Kaesang V2 + Kaesang V1 + Covid	-	-	18	*manhattan*	57	54
	✓	-	16	*manhattan*	60	57
	-	✓	28	*euclidean*	62	59

Tabel 6 menunjukkan hasil pencarian model optimal dengan menggunakan ekstraksi fitur *FastText*, yang mencakup optimasi melalui penambahan data, scaling data, dan parameter tuning. Pada dataset Kaesang V2 + Kaesang V1, tanpa scaling data, model mencapai performa terbaik dengan akurasi sebesar 62% dan F1-score sebesar 60%, menggunakan metrik *euclidean* dan K = 9. Meskipun penambahan data eksternal serta penerapan scaling data seperti Standar Scaling dan Robust Scaling dilakukan, performa model secara keseluruhan tetap tidak menunjukkan peningkatan yang signifikan dibandingkan eksperimen awal atau *baseline*, yang bahkan mencapai akurasi 63% dan F1-score 64%. Hal ini menunjukkan bahwa *FastText* relatif stabil terhadap variasi data eksternal dan penerapan scaling, tetapi juga menunjukkan bahwa langkah optimasi tersebut tidak selalu memberikan dampak

pada kinerja model ini. FastText tampaknya lebih efektif dengan konfigurasi baseline tanpa intervensi kompleks.

Tabel 7. Pencarian Model Optimal dengan Ekstraksi Fitur TF-IDF

Komposisi Dataset	Scaling Data		Parameter		Accuracy (%)	F1 (%)
	Standar	Robust	K	Metrics		
Kaesang V2 + Covid	-	-	29	*euclidean*	62	61
	✓	-	2	*euclidean*	37	26
	-	✓	5	*euclidean*	38	38
Kaesang V2 + Kaesang V1	-	-	11	*euclidean*	**63**	**63**
	✓	-	23	*euclidean*	32	16
	-	✓	1	*euclidean*	40	40
Kaesang V2 + Kaesang V1 + Covid	-	-	11	*euclidean*	62	61
	✓	-	24	*euclidean*	43	33
	-	✓	11	*euclidean*	53	52

Tabel 7 menunjukkan hasil pencarian model optimal dengan menggunakan ekstraksi fitur TF-IDF. Penerapan langkah optimasi, termasuk penambahan data, scaling data, dan parameter tuning, menghasilkan hasil yang bervariasi tergantung pada kombinasi komposisi dataset, jenis scaling, dan pemilihan parameter. Pada dataset Kaesang V2 + Kaesang V1, tanpa scaling data, model mencapai performa terbaik dengan akurasi 63% dan F1-score 63%, menggunakan metrik *euclidean* dan K = 11. Hasil ini mencerminkan potensi TF-IDF untuk mencapai kinerja optimal pada konfigurasi tertentu tanpa intervensi scaling. Namun, beberapa konfigurasi optimasi justru menyebabkan penurunan performa yang signifikan. Misalnya, penerapan *Standar Scaling* pada dataset Kaesang V2 + Kaesang V1 menghasilkan akurasi 32% dan F1-score 16%, jauh lebih rendah dibandingkan tanpa scaling. Penurunan serupa juga terjadi pada beberapa kombinasi lain yang menggunakan *Robust Scaling*. Hasil ini menunjukkan bahwa TF-IDF dapat menghasilkan performa yang kompetitif pada konfigurasi tertentu, tetapi langkah optimasi tertentu, seperti penerapan *Standar Scaling* atau *Robust Scaling*, dapat memberikan dampak negatif yang signifikan pada performa model, tergantung pada kombinasi dataset dan parameter yang digunakan. Oleh karena itu, pemilihan strategi optimasi perlu dilakukan dengan hati-hati untuk menghindari penurunan kinerja.

Tabel 8. Pencarian Model Optimal dengan Ekstraksi Fitur IndoBert

Komposisi Dataset	Scaling Data		Parameter		Accuracy (%)	F1 (%)
	Standar	Robust	K	Metrics		
Kaesang V2 + Covid	-	-	22	*euclidean*	55	56
	✓	-	27	*euclidean*	**63**	**63**
	-	✓	9	*euclidean*	52	52
Kaesang V2 + Kaesang V1	-	-	17	*manhattan*	53	52
	✓	-	15	*manhattan*	53	51
	-	✓	18	*manhattan*	52	47
Kaesang V2 + Kaesang V1 + Covid	-	-	9	*euclidean*	62	58
	✓	-	15	*euclidean*	57	55
	-	✓	21	*euclidean*	55	51

Tabel 8 menunjukkan hasil pencarian model optimal dengan menggunakan ekstraksi fitur IndoBERT. Pada dataset Kaesang V2 + Covid, penerapan Standar Scaling memberikan hasil terbaik, dengan akurasi 63% dan F1-score 63%, dengan menggunakan K= 27 metrik *euclidean*. Hal ini menunjukkan bahwa scaling data menggunakan metode *Standar Scaling* dapat meningkatkan kinerja model secara signifikan, dibandingkan tanpa scaling, yang hanya mencapai akurasi 55% dan F1-score 56%, dan model baseline yang hanya mencapai akurasi 53% dan F1-score 54%. Hal ini menunjukkan

bahwa penerapan scaling data memiliki dampak yang beragam pada kinerja model IndoBERT, tergantung pada metode scaling dan dataset yang digunakan. Sementara *Standar Scaling* terbukti efektif dalam meningkatkan performa pada beberapa konfigurasi. Selanjutnya, akan digunakan model optimal dengan ekstrasi fitur IndoBert untuk diuji menggunakan data uji.

3.4. Pengujian dan Perbandingan Data Uji

Setelah mendapatkan hasil terbaik dari eksperimen model optimal, selanjutnya adalah pengujian menggunakan data test yang tidak pernah dilihat saat pelatihan, guna memastikan bahwa model mampu melakukan generalisasi dengan baik terhadap data yang baru dan tidak dikenal. Berikut hasil pengujian menggunakan data test:

Tabel 9. Tabel Pengujian Data Test

Run	Metode	Accuracy (%)	F1 (%)
Run 1	FastText + KNN (Baseline)	44	39
Run 2	FastText + KNN	45	40
Run 3	IndoBert + KNN	**57**	**49**

Tabel 9 memperlihatkan hasil eksperimen menggunakan model *K-Nearest Neighbors* (KNN) dengan berbagai metode ekstraksi fitur pada tiga run berbeda. Pada Run 1, KNN dengan ekstraksi fitur *FastText* dan menggunakan model *baseline* menghasilkan akurasi sebesar 44% serta F1-score 39%. Pada Run 2, konfigurasi yang sama yaitu *FastText* dengan model yang telah melalui tahap optimasi pada langkah sebelumnya, menunjukkan sedikit peningkatan performa, menghasilkan akurasi 45% dan F1-score 40%. Run 3 menunjukkan peningkatan yang signifikan setelah mengganti ekstraksi fitur menjadi IndoBert dengan model yang telah dioptimasi. Model ini mencapai akurasi tertinggi sebesar 57% dan F1-score 49%, menjadikannya model optimal dibandingkan konfigurasi sebelumnya. Hasil ini menunjukkan bahwa penggunaan IndoBert sebagai metode ekstraksi fitur secara signifikan meningkatkan performa dibandingkan dengan *FastText.*

Tabel 10. Perbandingan Hasil *Leaderboard*

Tim	Metode	Accuracy (%)	F1 (%)
Rank 1	BERT	**70**	**60**
Admin	Baseline	45	40
Penelitian Ini	IndoBert + KNN	57	49

Tabel 10 menunjukkan hasil perbandingan akurasi dan F1-score antara beberapa metode dalam pengujian *leaderboard*, di mana metode yang digunakan oleh penelitian ini dibandingkan dengan metode dari tim lain. Tim Rank 1 menggunakan metode BERT dan mencapai akurasi tertinggi sebesar 70% dan F1-score 60%, menunjukkan performa terbaik di antara semua tim. Selanjutnya, penelitian ini yang menggunakan KNN dengan ekstraksi fitur IndoBert, berada di posisi berikutnya dengan akurasi 57% dan F1-score 49%. Meskipun metode sederhana seperti K-Nearest Neighbor (KNN) belum mampu melampaui performa metode berbasis transformer seperti BERT, penggunaan KNN dengan ekstraksi fitur IndoBert menunjukkan peningkatan yang signifikan dari baseline awal. Sementara itu, admin yang hanya menggunakan model *Baseline* menunjukkan performa yang paling rendah, dengan akurasi 45% dan F1-score 40%.

4. KESIMPULAN

Penelitian ini menunjukkan bahwa metode *K-Nearest Neighbor* (KNN) dapat digunakan untuk klasifikasi sentimen dengan hasil yang signifikan, meskipun kinerjanya sangat bergantung pada teknik ekstraksi fitur dan langkah-langkah optimasi yang diterapkan. Dari tiga metode ekstraksi fitur yang diuji, *FastText* awalnya memberikan performa terbaik, tetapi optimasi lebih lanjut seperti penambahan data dan scaling tidak memberikan peningkatan yang berarti. Sebaliknya, TF-IDF menunjukkan hasil

yang bervariasi, dengan beberapa konfigurasi memberikan peningkatan performa, meskipun teknik scaling tertentu justru menurunkan akurasinya. Di sisi lain, IndoBERT berhasil menunjukkan peningkatan kinerja setelah optimasi, meskipun tetap sensitif terhadap teknik scaling dan parameter yang digunakan.

Pada pengujian dengan data uji, model *baseline* menggunakan *FastText* menunjukkan performa awal yang cukup rendah dengan akurasi sebesar 44% dan F1-score 39%, tetapi melalui optimasi yang mencakup penggantian metode ekstraksi fitur menjadi IndoBERT, terjadi peningkatan yang signifikan mencapai akurasi tertinggi sebesar 57% dan F1-score 49%, mengalami peningkatan performa sebesar 10%. Hal ini membuktikan pentingnya memilih metode ekstraksi fitur dan langkah optimasi yang tepat dalam meningkatkan kinerja model klasifikasi sentimen.

Penelitian ini merekomendasikan pengembangan lebih lanjut dengan menerapkan teknik optimasi lanjutan, seperti eksplorasi hyperparameter lebih mendalam dan pemanfaatan data eksternal yang lebih beragam untuk meningkatkan hasil klasifikasi sentimen secara lebih signifikan. Upaya ini diharapkan dapat memberikan model yang lebih akurat dan andal dalam menangani tugas klasifikasi sentimen di masa depan.

Daftar Pustaka

[1] D. A. Vonega, A. Fadila, and D. E. Kurniawan, "Analisis Sentimen Twitter Terhadap Opini Publik Atas Isu Pencalonan Puan Maharani dalam PILPRES 2024," *Journal of Applied Informatics and Computing*, vol. 6, no. 2, pp. 129–135, 2022.

[2] R. Harun, "Analisis Sentimen Opini Publik Pengguna Twitter Terhadap Kenaikan Harga BBM Menggunakan Algoritma Naïve Bayes," *Jurnal Ilmiah Ilmu Komputer Banthayo Lo Komputer*, vol. 2, no. 1, pp. 26–33, 2023.

[3] E. Budianita, E. P. Cynthia, A. Pranata, and D. Abimanyu, "Pendekatan berbasis Machine Learning dan Leksikal Pada Analisis Sentimen," *Seminar Nasional Teknologi Informasi, Komunikasi dan Industri (SNTIKI)*, pp. 99–104, 2022.

[4] O. Kherif, Y. Benmahamed, M. Teguar, A. Boubakeur, and S. S. M. Ghoneim, "Accuracy Improvement of Power Transformer Faults Diagnostic Using KNN Classifier With Decision Tree Principle," *IEEE Access*, vol. 9, pp. 81693–81701, 2021, doi: 10.1109/ACCESS.2021.3086135.

[5] N. Sepriadi, E. Budianita, M. Fikry, and Pizaini, "Analisis Sentimen Review Aplikasi Mypertamina Menggunakan Word Embedding Fasttext Dan Algoritma K-Nearest Neighbor," *INFORMASI (Jurnal Informatika dan Sistem Informasi)*, vol. 15, no. 1, pp. 91–109, May 2023, doi: 10.37424/informasi.v15i1.222.

[6] A. Naldi and S. Agustian, "KLASIFIKASI SENTIMEN VAKSIN COVID-19 MENGGUNAKAN K-NEAREST NEIGHBOR BERDASARKAN WORD EMBEDDINGS FASTTEXT PADA TWITTER," *ZONAsi: Jurnal Sistem Informasi*, vol. 5, no. 2, pp. 323–333, Jun. 2023, doi: 10.31849/zn.v5i2.12548.

[7] 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," 2024. [Online]. Available: https://arxiv.org/abs/2407.05627

[8] 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.

[9] 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.

[10] A. N. Yahya, "Pro dan Kontra Kaesang Pangarep Jadi Ketum PSI ," *KOMPAS*, Sep. 26, 2023.

[11] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *Trans Assoc Comput Linguist*, vol. 5, pp. 135–146, 2017.

[12] A. Addiga and S. Bagui, "Sentiment Analysis on Twitter Data Using Term Frequency-Inverse Document Frequency," *Journal of Computer and Communications*, vol. 10, no. 08, pp. 117–128, 2022, doi: 10.4236/jcc.2022.108008.

[13] K. S. Nugroho, A. Y. Sukmadewa, H. Wuswilahaken DW, F. A. Bachtiar, and N. Yudistira, "Bert fine-tuning for sentiment analysis on indonesian mobile apps reviews," in *Proceedings of the 6th International Conference on Sustainable Information Engineering and Technology*, 2021, pp. 258–264.

[14] M. Ahsan, M. Mahmud, P. Saha, K. Gupta, and Z. Siddique, "Effect of Data Scaling Methods on Machine Learning Algorithms and Model Performance," *Technologies (Basel)*, vol. 9, no. 3, p. 52, Jul. 2021, doi: 10.3390/technologies9030052.

[15] S. Alam and N. Yao, "The impact of preprocessing steps on the accuracy of machine learning algorithms in sentiment analysis," *Comput Math Organ Theory*, vol. 25, no. 3, pp. 319–335, Sep. 2019, doi: 10.1007/s10588-018-9266-8.

[16] H. Zhou, "Research of Text Classification Based on TF-IDF and CNN-LSTM," *J Phys Conf Ser*, vol. 2171, no. 1, p. 012021, Jan. 2022, doi: 10.1088/1742-6596/2171/1/012021.

[17] J. D. M.-W. C. Kenton and L. K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of naacL-HLT*, Minneapolis, Minnesota, 2019, p. 2.

[18] F. Koto, A. Rahimi, J. H. Lau, and T. Baldwin, "IndoLEM and IndoBERT: A benchmark dataset and pre-trained language model for Indonesian NLP," *arXiv preprint arXiv:2011.00677*, 2020.

[19] J. U. U. Nysa, A. Mahmudi, and K. Auliasari, "PERBANDINGAN JARAK EUCLIDEAN, MANHATTAN, CHEBYSHEV PADA KLASIFIKASI STATUS GIZI BALITA MENGGUNAKAN METODE K-NEAREST NEIGHBORS (KNN)," *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 7, no. 4, pp. 2443–2450, 2023.

[20] Muhammad Luthfi Al-Ghifari and Ken Ditha Tania, "Sentiment Analysis Performance Value Optimization Using Hyperparamater Tunning With Grid Search On Shopee App Reviews," *Indonesian Journal of Computer Science*, vol. 12, no. 5, Oct. 2023, doi: 10.33022/ijcs.v12i5.3384.

[21] I. G. T. Permana and I. B. G. Dwidasmara, "Evaluasi Performance dengan Grid Search Terhadap K Nearest Neighbor (KNN) untuk Klasifikasi Penderita Diabetes Melitus," *Jurnal Elektronik Ilmu Komputer Udayana p-ISSN*, vol. 2301, p. 5373.

[22] P. Yohana, S. Agustian, and S. K. Gusti, "Klasifikasi Sentimen Masyarakat terhadap Kebijakan Vaksin Covid-19 pada Twitter dengan Imbalance Classes Menggunakan Naive Bayes," in *Seminar Nasional Teknologi Informasi Komunikasi dan Industri*, pp. 69–80.