Building of Informatics, Technology and Science (BITS)
Volume 6, No 3, Desember 2024 Page: 1668−1676
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v6i3.6335

Penggunaan Model Bahasa indoBERT pada Metode Random Forest Untuk Klasifikasi Sentimen Dengan Dataset Terbatas

Joni Pranata, Surya Agustian*, Jasril, Elin Haerani

Sains dan Teknologi, Teknik Informatika, Universitas Islam Negeri Sultan Syarif Kasim Riau, Pekanbaru, Indonesia
Email: [1]11950111700@students.uin-ac.id, [2,*] surya.agustian@uin-suska.ac.id, [3]jasril@uin-suska.ac.id, [4] elin.haerani@uin-suska.ac.id
Email Penulis Korespondensi: surya.agustian@uin-suska.ac.id
Submitted: **22/11/2024**; Accepted: **12/12/2024**; Published**: 18/12/2024**

Abstrak−Masalah keterbatasan data latih menjadi tantangan utama dalam klasifikasi sentimen di berbagai bahasa, termasuk bahasa Indonesia, terutama untuk analisis sentimen terkait topik tertentu. Hal ini disebabkan oleh berbagai faktor, dan umumnya adalah kebutuhan untuk mengetahui dengan segera bagaimana sentimen terhadap suatu isu, sehingga tidak mungkin menghabiskan waktu untuk memberi label yang cukup pada data untuk proses pelatihan. Penelitian ini mengusulkan model klasifikasi sentimen dengan sumber data pelatihan yang sedikit, pada studi kasus pengangkatan Kaesang Pangarep sebagai ketua umum PSI. Model dasar (*baseline*) menggunakan representasi kata dengan FastText dan algoritma Random Forest, Model ini kemudian dioptimasi dengan menggunakan *word embedding* IndoBERT, sebuah model berbasis BERT yang telah dilatih khusus untuk teks bahasa Indonesia, serta optimasi praproses, menambahkan data eksternal (*data aggregation*) dan parameter tuning. Hasil penelitian menunjukkan bahwa metode IndoBERT dengan Random Forest yang dioptimasi memberikan peningkatan performa yang signifikan dibandingkan baseline, sebesar 6%. Hasil klasifikasi model yang paling optimal sebesar 54% unutk F1-score dan 63% akurasi. Temuan ini menegaskan bahwa penambahan data eksternal dan optimasi parameter dapat meningkatkan kemampuan generalisasi model dalam klasifikasi sentimen bahasa Indonesia. Penelitian ini diharapkan dapat menjadi referensi metodologis bagi studi klasifikasi sentimen serupa yang menghadapi kendala ukuran dataset.

Kata Kunci: Analisis Sentimen, IndoBERT, Klasifikasi Sentimen, Optimasi Model, Random Forest.

Abstract−The limitation of training data is a major challenge in sentiment classification across various languages, including Indonesian, especially for sentiment analysis on specific topics. This is caused by various factors, primarily the need to quickly understand the sentiment towards an issue, making it impractical to spend time labeling enough data for the training process. This study proposes a sentiment classification model with a limited training data source, using the case study of Kaesang Pangarep's appointment as the chairman of PSI. The baseline model uses word representation with FastText and the Random Forest algorithm. This model is then optimized using the IndoBERT word embedding, a BERT-based model specifically trained for Indonesian text, along with text preprocessing optimization, the addition of external data (data aggregation), and parameter tuning. The results show that the IndoBERT with optimized Random Forest method provides a significant performance improvement of 6% compared to the baseline. The optimal model classification achieved an F1-score of 54% and an accuracy of 63%. These findings highlight that adding external data and optimizing parameters can improve the generalization ability of models in Indonesian sentiment classification. This study is expected to serve as a methodological reference for similar sentiment classification studies facing dataset size constraints.

Keywords: Sentiment Analysis, IndoBERT, Sentiment Classification, Model Optimization, Random Forest

1. PENDAHULUAN

Pesatnya perkembangan teknologi informasi dan komunikasi telah membuat media sosial menjadi platform utama bagi banyak orang untuk berbagi informasi, pandangan, dan opini. Salah satu *platform* media sosial yang paling populer adalah Twitter, yang memungkinkan pengguna untuk mempublikasikan pesan singkat yang dikenal sebagai tweet. Setiap hari, jutaan tweet dipublikasikan oleh orang-orang dari berbagai belahan dunia, yang membahas beragam topik, seperti isu sosial, politik, ekonomi, dan hiburan [1]. Data yang dihasilkan dari Twitter ini sangat kaya dan memberikan wawasan yang berharga untuk memahami sentimen publik terhadap topik-topik tersebut. Dalam bidang sosial dan politik, analisis sentimen memungkinkan untuk menggali pandangan publik terkait isu-isu krusial, kebijakan pemerintah, dan tindakan politisi secara *real-time*. Dengan metode ini, para peneliti, pembuat kebijakan, dan politisi dapat mengidentifikasi kecenderungan sentimen, baik positif, negatif, dan netral, terhadap peristiwa tertentu, memprediksi dampak kebijakan, serta memantau reaksi masyarakat secara efektif [2].

Untuk mengolah data yang dihasilkan oleh platform seperti Twitter, diperlukan metode yang efektif untuk menganalisis sentimen dalam teks. Proses ini umumnya menggunakan algoritma pembelajaran mesin dan metode pemrosesan bahasa alami (*Natural Language Processing/NLP*), yang memungkinkan pengelompokan sentimen berdasarkan pola ekspresi emosi yang ada dalam teks [3]. Pengklasifikasi teks berbasis *supervised learning* dapat disesuaikan dengan kelas dan teks baru tanpa perlu memodifikasi algoritma, cukup dengan menyediakan himpunan data pelatihan yang telah dianotasi. Namun, dataset pelatihan semacam ini sering kali tidak tersedia untuk kelas atau topik tertentu, sehingga data khusus perlu diberi anotasi secara manual [4], proses ini memakan waktu dan biaya, terutama untuk jumlah data yang besar. Oleh karena itu, penelitian ini membatasi annotasi manual pada 300 hingga 600 data sebagai dasar untuk membangun model. Selanjutnya, digunakan data tambahan yang telah dianotasi untuk memperkaya data pelatihan, sehingga model dapat mencapai performa optimal tanpa membutuhkan sumber daya yang berlebihan.

Building of Informatics, Technology and Science (BITS)
Volume 6, No 3, Desember 2024 Page: 1668−1676
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v6i3.6335

Penelitian sebelumnya telah mengeksplorasi berbagai pendekatan untuk menangani tantangan keterbatasan dataset dalam klasifikasi sentimen. Misalnya, penelitian yang menggunakan algoritma *Naïve Bayes* yang dioptimalkan dengan *Particle Swarm Optimization* (PSO) menunjukkan bahwa penggunaan data tambahan, seperti dataset COVID-19, dapat meningkatkan performa model secara signifikan, menghasilkan *f1-score* tertinggi 50% [5]. Penelitian lain yang menggunakan algoritma *Support Vector Machine* (SVM) dengan fitur *FastText* juga menunjukkan bahwa penambahan data eksternal dapat meningkatkan nilai *F1-score* hingga 53% [6]. Selain itu, algoritma Random Forest digunakan untuk mengklasifikasikan sentimen dengan memanfaatkan fitur TF-IDF dan seleksi fitur berbasis *Chi-Square*. Penelitian ini menunjukkan bahwa penambahan data eksternal dapat meningkatkan performa model hingga *F1-score* sebesar 52% [7]. pendekatan-pendekatan ini berhasil meningkatkan performa model secara signifikan. Namun, masih diperlukan eksplorasi metode yang lebih efektif untuk meningkatkan akurasi model dan kemampuan dalam memahami hubungan semantik secara lebih mendalam.

Sebagai upaya untuk mengatasi keterbatasan yang ditemukan dalam penelitian sebelumnya, penelitian ini menerapkan pendekatan berbasis *word embedding FastText* dan algoritma *Random Forest* sebagai tahap awal (*baseline*). *FastText* dipilih karena pada penelitian sebelumnya telah menunjukkan hasil yang baik dan memiliki kemampuan untuk menangani kata-kata yang tidak ada dalam kamus (*out-of-vocabulary words*) [8]. *Random Forest* digunakan karena keandalannya dalam menangani dataset dengan variabilitas tinggi dan kemampuannya menghasilkan prediksi yang stabil [9]. Model ini kemudian dioptimasi dengan menggunakan *word embedding* IndoBERT, sebuah model berbasis BERT yang telah dilatih khusus untuk teks berbahasa Indonesia, untuk menghasilkan representasi kata yang lebih kaya dan kontekstual. IndoBERT dipilih karena kemampuannya dalam memahami konteks semantik yang lebih mendalam dalam bahasa Indonesia [10]. Untuk mengoptimalkan proses klasifikasi, penelitian ini juga melakukan beberapa langkah penting, yaitu mengoptimalkan proses praproses data (*text preprocessing*) untuk menghasilkan representasi teks yang lebih baik, menambahkan data eksternal (*Data Aggregation*), dan melakukan *parameter tuning*. Langkah-langkah ini bertujuan untuk memperkuat kualitas data pelatihan dan menyesuaikan model agar menghasilkan performa yang lebih akurat dan stabil dalam analisis sentimen.

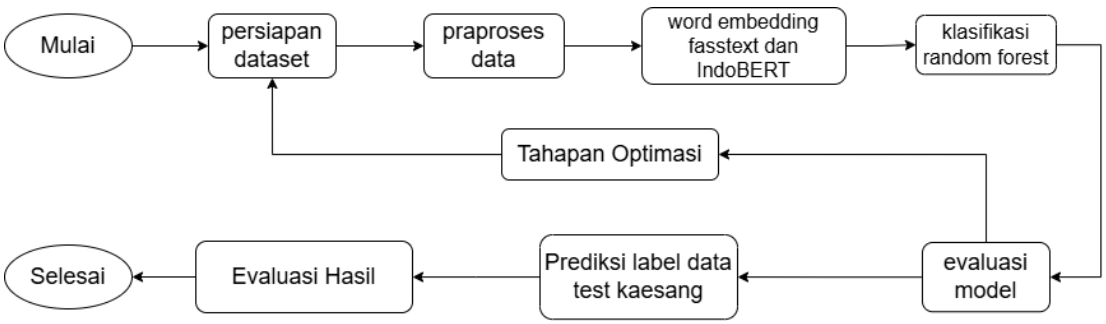
Penelitian ini merupakan kelanjutan dari penelitian bersama yang diselenggarakan oleh Agustian dkk. [11] dalam bentuk *shared task*. Dalam penelitian ini, tantangan utama yang dihadapi adalah keterbatasan data pelatihan, yang disebabkan oleh biaya pengumpulan data yang tinggi, keterbatasan akses, serta kesulitan dalam memperoleh data yang relevan secara cepat. Studi kasus yang digunakan dalam penelitian ini adalah isu pengangkatan Kaesang Pangarep, sebagai ketua umum PSI. Proses pengangkatan ini menuai berbagai sentimen di masyarakat, baik positif maupun negatif, karena dianggap tidak sesuai dengan mekanisme yang lazim dalam partai politik. Ketidakwajaran ini terjadi ketika Kaesang baru bergabung sebagai anggota partai, namun langsung diangkat menjadi ketua umum tanpa melalui proses kaderisasi atau pemilihan yang biasa dilakukan dalam struktur partai politik. Penelitian ini bertujuan untuk menganalisis sentimen masyarakat terhadap isu ini dan mengevaluasi pendekatan klasifikasi pada dataset dengan sumber daya terbatas.

Penelitian ini diharapkan dapat memberikan kontribusi signifikan dalam mengatasi tantangan dataset terbatas dengan memanfaatkan kombinasi *word embedding* IndoBERT dan algoritma Random Forest yang dioptimalkan. Dengan hasil yang diperoleh, penelitian ini juga berupaya menjadi acuan bagi pengembangan model klasifikasi sentimen dalam bahasa Indonesia, khususnya untuk studi-studi serupa di masa depan.

2. METODOLOGI PENELITIAN

2.1 Tahapan Penelitian

Berikut tahapan-tahapan yang dilakukan dalam penelitian ini.



Gambar 1. Tahapan Penelitian

2.2 Persiapan Data

Dalam penelitian ini, pengumpulan *dataset* dilakukan dengan menggunakan beberapa sumber data yang diambil dari penelitian Agustian dkk. [11] mengenai klasifikasi sentimen dengan *dataset* terbatas. Salah satu dataset utama yang digunakan berasal dari media sosial Twitter, dengan jumlah total 2000 sampel tweet yang dikumpulkan melalui proses

Building of Informatics, Technology and Science (BITS)
Volume 6, No 3, Desember 2024 Page: 1668−1676
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v6i3.6335

scraping dari tanggal 25 September hingga 3 Oktober 2023. Tweet-tweet tersebut berkaitan dengan isu pengangkatan Kaesang Pangarep sebagai Ketua Umum PSI, yang kemudian diberi label sentimen (positif, negatif, dan netral) oleh tim anotator melalui *crowdsourcing*. Dari total 2000 tweet yang digunakan, sebagian kecil (300 sampai 600 tweet) dipilih sebagai data training, sementara sisanya digunakan untuk pengujian *(testing)* tanpa label kelas. *Dataset* yang digunakan sebagai data latih dan data uji, beserta distribusinya, dapat dilihat pada Tabel 1 di bawah ini, dan dapat diunduh melalui situs Github [12].

Tabel 1. *Dataset* Penelitian

No	Dataset	Penggunaan	Jumlah sampel data	Distribusi Kelas		
				Positif	Negatif	Netral
1	Dataset Kaesang v1	Training	300	100	100	100
2	Dataset Kaesang v2	Training	300	100	100	100
3	Dataset Program Vaksin Covid	Training	8000	463	6664	873
4	Dataset *Open Topic*	Training	7569	1505	3408	2656
5	Data Test Kaesang	Testing	924			

Pada tahap awal penelitian ini, digunakan dataset Kaesang v1 dan v2 sebagai data pelatihan utama. Kedua dataset tersebut digabungkan menjadi 600 data, kemudian dibagi menjadi dua bagian: 80% digunakan untuk melatih model (*data train*) dan 20% untuk validasi (*data validasi*). Dataset Program vaksin Covid dan Dataset *Open Topic* merupakan data ekternal digunakan sebagai data tambahan (*data aggregation*). Kemudian Data Test Kaesang tanpa label digunakan untuk pengujian model. Berikut adalah contoh distribusi data yang digunakan untuk pelatihan model:

Tabel 2. Distribusi *Dataset* Penelitian

No	Dataset	Penggunaan	Jumlah sampel	Distribusi Kelas		
				Positif	Negatif	Netral
1	Train Kaesang	Train (80%)	420	160	160	160
2	Val Kaesang	Validasi (20%)	120	40	40	40
3	Covid 300	Train	300	100	100	100
4	Open 300	Train	300	100	100	100
5	Train Kaesang + Covid 300	Train	780	260	260	260

Pada tahap optimasi, untuk memperkaya *dataset* dan meningkatkan akurasi model, *dataset* vaksin covid dan *open* topik digunakan sebagai penambahan data *(data aggregation)*. Tahapan optimasi pada penambahan data dilakukan secara bertahap untuk menjaga keseimbangan data. Pada tahap pertama, sebanyak 100 data per label (positif, negatif, dan netral) dari masing-masing dataset vaksin covid dan open topik ditambahkan ke dalam data pelatihan utama, contoh penambahan data dapat dilihat pada nomor 5 pada Tabel 2. Setelah model dievaluasi, penambahan data dilanjutkan dengan menambahkan 200 data per label pada setiap iterasi berikutnya. Proses ini dilakukan secara bertahap hingga diperoleh hasil yang optimal.

2.3 Praproses Data

Tahapan praproses data dalam penelitian ini dilakukan untuk memastikan bahwa data yang digunakan siap untuk dianalisis dan menghasilkan model yang akurat [13]. Berikut adalah langkah-langkah praproses data yang dilakukan:

a. Normalisasi Teks (*Case Folding*): Seluruh teks diubah menjadi huruf kecil untuk menjaga konsistensi dalam analisis kata, sehingga perbedaan penggunaan huruf besar dan kecil tidak memengaruhi hasil klasifikasi.
b. Konversi *Emoticon*: *emoticon* dapat mengandung makna emosional yang signifikan. Proses ini melibatkan penggantian simbol *emoticon* dengan kata atau frasa yang mewakili perasaan yang ditunjukkan oleh *emoticon* tersebut.
c. Pembersihan Teks (*Text Cleaning*): Pada tahap ini, dilakukan penghapusan karakter yang tidak relevan seperti tanda baca, angka, simbol, *hyperlink*, serta *mention* (@username). Langkah ini bertujuan untuk menghilangkan elemen yang tidak diperlukan dalam analisis sentimen.
d. Penghapusan *Stopwords*: Kata-kata umum (*stopwords*) seperti “dan”, “ke”, “di” yang tidak memberikan kontribusi signifikan terhadap sentimen dihilangkan, sehingga model fokus pada kata-kata yang lebih bermakna dalam analisis. Daftar *stop words* bahasa Indonesia diambil dari pustaka *Natural Language Toolkit* (NLTK), yang menyediakan kumpulan kata-kata umum untuk berbagai bahasa, termasuk bahasa Indonesia [14].
e. Tokenisasi: setiap kalimat atau teks yang telah dibersihkan akan dipisahkan menjadi token-token yang lebih kecil, seperti kata atau frasa. Proses ini dilakukan dengan memecah teks berdasarkan pemisah yang telah ditentukan, seperti spasi, tanda baca, atau karakter lainnya.

Setelah tahap praproses data yang mencakup normalisasi teks, konversi emotikon, pembersihan teks, penghapusan *stopwords*, dan tokenisasi, data yang telah diproses selanjutnya akan digunakan untuk tahap *word embedding.*

2.4 Word Embedding dengan FastText

Setelah tahap tokenisasi, penelitan ini menggunakan *FastText* untuk mengonversi kata-kata yang telah dipisahkan menjadi representasi numerik yang dapat diproses oleh model klasifikasi. Untuk itu, dalam penelitian ini, digunakan teknik *word embedding* dengan model *FastText*. *FastText* adalah salah satu teknik *word embedding* yang

Building of Informatics, Technology and Science (BITS)
Volume 6, No 3, Desember 2024 Page: 1668−1676
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v6i3.6335

dikembangkan oleh Facebook, yang bertujuan untuk mengubah kata-kata dalam teks menjadi vektor berdimensi rendah yang menggambarkan makna semantik kata tersebut [15].

FastText berbeda dari teknik *word embedding* lainnya seperti Word2Vec dan GloVe, karena model ini tidak hanya mengonversi kata utuh menjadi vektor, tetapi juga memecah kata menjadi n-grams (sub-kata). Pendekatan ini memungkinkan model untuk menangkap makna kata yang lebih mendalam dan menangani kata yang tidak ada dalam kamus (*out-of-vocabulary words*) dengan lebih efektif. Sebagai contoh, kata ”menghibur” dalam FastText akan dipecah menjadi beberapa n-grams seperti ”me”, ”meng”, ”hibur”, ”ibur”, dan ”r”, yang meningkatkan kemampuan model untuk menangkap makna bahkan dari kata yang jarang ditemukan dalam dataset pelatihan.

Penggunaan *FastText* pada tahap awal bertujuan untuk memperoleh representasi kata yang lebih efektif, terutama dalam menangani kata-kata yang tidak ditemukan dalam dataset pelatihan. Namun, pada tahap optimasi, *FastText* digantikan dengan IndoBERT untuk mendapatkan representasi yang lebih kaya dan kontekstual, yang lebih sesuai untuk teks bahasa Indonesia yang lebih kompleks.

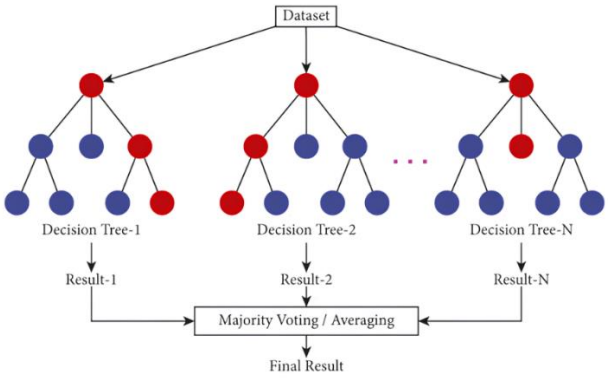
2.5 Word Embedding dengan IndoBERT

Model indoBERT adalah model bahasa berbasis *transformer* yang dirancang khusus untuk menangani teks berbahasa indonesia. Dikembangkan dari arsitektur BERT *(Bidirectional Encoder Representations from Transformers)*, indoBERT dilatih menggunakan korpus teks dalam bahasa indonesia sehingga mampu menangkap makna dan konteks kata-kata dengan lebih akurat sesuai karakteristik bahasa [16]. setelah melalui proses tokenisasi, setiap token dari teks diubah menjadi representasi numerik atau *embedding* yang memungkinkan model untuk memahami konteks dan makna dari kata-kata dalam kalimat.

Pada indoBERT, setiap token diubah menjadi vektor berdimensi tetap biasanya berukuran 768 dimensi yang mewakili hubungan semantik dan konteks kata tersebut dalam kalimat. *Embedding* ini mencakup informasi kontekstual dari setiap kata, memungkinkan algoritma untuk menangkap makna yang lebih mendalam dari teks. Selain itu, indoBERT juga menambahkan token khusus seperti [CLS] di awal dan [SEP] di akhir kalimat, yang masing-masing memberikan informasi tambahan untuk tugas klasifikasi dan pemisahan kalimat [17].

2.6 Metode Klasifikasi dengan Random Forest

Random Forest adalah algoritma pembelajaran mesin berbasis *ensemble* yang dikembangkan oleh Leo Breiman pada 2001, digunakan untuk klasifikasi dan regresi [18]. Algoritma ini mengurangi overfitting dan meningkatkan akurasi dengan membangun beberapa pohon keputusan menggunakan sampling acak dan subset fitur. Prediksi akhir dihasilkan dengan menggabungkan hasil dari setiap pohon melalui *majority vote* untuk klasifikasi dan rata-rata untuk regresi [19]. Berikut cara kerja *random forest:*



Gambar 2. Ilustrasi Random Forest

Cara kerja algoritma *Random Forest* dapat dijabarkan dalam langkah-langkah berikut:

a. Pemilihan Sampel Acak *(Bootstrap Sampling):* Algoritma memilih sejumlah sampel acak dari dataset utama, menciptakan beberapa dataset pelatihan yang berbeda. Proses ini disebut *bagging* atau *bootstrap aggregation*.
b. Pembuatan *Decision Tree:* Setiap sampel acak yang dipilih akan digunakan untuk membangun sebuah *decision tree* menggunakan subset data dan subset fitur yang dipilih secara acak. Ini membantu mengurangi korelasi antar pohon dan meningkatkan ketahanan model terhadap overfitting.
c. Proses Voting: Setelah semua *decision tree* terbentuk, masing-masing pohon memberikan hasil prediksinya. Untuk masalah klasifikasi, voting dilakukan untuk memilih prediksi yang paling sering muncul (*modus*), sedangkan untuk masalah regresi, hasil prediksi dari setiap pohon dihitung rata-ratanya (*mean*).
d. Pemilihan Hasil Akhir: Prediksi akhir diperoleh dari hasil voting dengan memilih prediksi yang paling banyak mendapat suara. Pendekatan ini memberikan hasil yang lebih stabil dan akurat.

Dalam proses pembentukan *decision tree*, Indeks Gini sering digunakan sebagai metrik untuk menentukan split terbaik di setiap node [20].Indeks Gini untuk suatu node dihitung dengan rumus berikut:

Building of Informatics, Technology and Science (BITS)
Volume 6, No 3, Desember 2024 Page: 1668−1676
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v6i3.6335

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

Pada suatu node tertentu, D merujuk pada dataset yang ada, dengan n menunjukkan jumlah kelas yang terdapat dalam dataset tersebut. Proporsi data yang termasuk dalam kelas ke-iii di node tersebut dapat dihitung sebagai p_i, yang menggambarkan seberapa besar kontribusi kelas ke-iii terhadap keseluruhan data pada node tersebut.

2.7 Parameter Tuning pada Algoritma Random Forest

Pada penelitian ini, model parameter *tuning Random forest* yang digunakan adalah *library sklearn*[21] dengan pemrograman python. Optimasi model dilakukan melalui parameter tuning untuk mendapatkan performa terbaik dari algoritma *Random Forest* yang digunakan dalam klasifikasi sentimen. Parameter *tuning* adalah proses menyesuaikan nilai-nilai parameter penting dalam model agar menghasilkan kombinasi optimal yang memberikan kinerja terbaik, terutama dalam hal akurasi, *F1-Score, precision,* dan *recall*[22]. Ada beberapa parameter utama yang disesuaikan dalam algoritma *Random Forest*, di antaranya:

a. n_estimators: Parameter ini mengatur jumlah pohon keputusan yang akan dibangun dalam Random Forest. Pada penelitian ini, nilai yang diuji adalah 100 dan 200 pohon.
b. max_depth: Parameter ini membatasi kedalaman maksimum dari setiap pohon dalam Random Forest. Dengan mengatur kedalaman maksimal, model dapat diatur untuk menghindari *overfitting*. Pada penelitian ini, nilai yang diuji adalah 10, 20, dan None (tak terbatas). Kedalaman pohon yang optimal akan menghasilkan keseimbangan antara kompleksitas dan kemampuan generalisasi model.
c. min_samples_split: Parameter ini menentukan jumlah sampel minimum yang dibutuhkan untuk membagi sebuah node. Nilai yang diuji dalam penelitian ini adalah 2 dan 5.
d. min_samples_leaf: Parameter ini menentukan jumlah sampel minimum yang dibutuhkan untuk sebuah *leaf node* atau node akhir. Dengan menetapkan Batasan ini, model menjadi lebih *robust*, terutama pada data yang memiliki variasi tinggi. Nilai yang diuji adalah 1 dan 2.

Dalam penelitian ini, parameter tuning dilakukan melalui grid search untuk mengevaluasi kombinasi terbaik dari ketiga parameter tersebut. Proses ini diharapkan dapat menghasilkan model Random Forest yang memiliki akurasi tinggi serta mampu menangani variabilitas data secara efektif.

2.8 Optimasi Model

Untuk meningkatkan kinerja model, beberapa tahapan optimasi dilakukan sebagai berikut:

a. Mengoptimalkan Praproses Data: Pada tahap praproses data, optimasi dilakukan dengan cara menguji kombinasi pengaturan untuk empat fitur yang berbeda, yaitu normalisasi teks, konversi emotikon, teks cleaning, dan penghapusan stopwords. Setiap fitur memiliki dua opsi: aktif atau tidak aktif. Dengan konfigurasi ini, terdapat 16 kemungkinan kombinasi yang dihasilkan dari berbagai keadaan aktif dan non-aktif pada fitur-fitur tersebut, setiap kombinasi diuji satu per satu untuk melihat dampaknya terhadap akurasi model.
b. Penambahan *Dataset* Eksternal: *Dataset* tambahan yang relevan, yaitu data vaksin covid dan data open topik, ditambahkan untuk memperkaya data pelatihan dan menangkap variasi sentimen yang lebih luas..
c. Menggunakan *Word Embeding* IndoBERT: *FastText* digantikan dengan IndoBERT untuk word embedding. IndoBERT, sebagai varian BERT yang dilatih untuk bahasa Indonesia, memberikan representasi kata yang lebih kontekstual dan mampu menangkap makna semantik dalam kalimat dengan lebih baik.
d. Parameter *Tuning* pada *Random Forest*: Untuk mengoptimalkan kinerja model, dilakukan pencarian parameter terbaik menggunakan teknik grid search untuk mengatur jumlah pohon (n_estimators), kedalaman pohon (max_depth), dan jumlah fitur yang dipertimbangkan untuk pembagian (max_features).

2.9 Evaluasi

Setelah proses parameter *tuning* selesai dan model *Random Forest* mencapai konfigurasi optimal, dilakukan evaluasi model menggunakan *classification report*, yang memberikan informasi mendetail mengenai performa model dalam klasifikasi, termasuk metrik *F1-score, precision, recall,* dan *support* untuk setiap kelas.

Dimana, *precision* menunjukkan seberapa tepat model dalam memprediksi setiap kelas, *recall* menunjukkan kemampuan model menemukan semua sampel yang benar di setiap kelas, dan *F1-score* mengukur keseimbangan antara ketepatan (*precision*) dan ketercakupan (*recall*) [23]. Pada penelitian ini, *F1-score* dijadikan sebagai metrik utama atau patokan untuk mengevaluasi performa model. *F1-score* dipilih karena metrik ini mampu menyeimbangkan antara ketepatan (*precision*) dan kemampuan deteksi (*recall*), memberikan gambaran lebih akurat tentang kinerja model dalam mengklasifikasikan setiap kelas secara konsisten [24]. Berdasarkan hasil dari classification report, model dapat dianalisis lebih lanjut untuk memastikan apakah model bekerja dengan baik pada semua kelas atau terdapat ketidakseimbangan dalam prediksi.

2.10 Pengujian Menggunakan Data Test Kaesang

Pada tahap akhir penelitian, model baseline dan model optimal dipilih untuk memprediksi data test kaesang, hasil prediksi di *import* ke sistem leaderboard [25]. Proses pengujian dilakukan dengan langkah-langkah berikut:

Building of Informatics, Technology and Science (BITS)
Volume 6, No 3, Desember 2024 Page: 1668−1676
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v6i3.6335

a. Prediksi Label Sentimen: Model yang telah dilatih sebelumnya digunakan untuk memprediksi label sentimen (positif, negatif, atau netral) pada setiap sampel dalam data uji. Model melakukan prediksi berdasarkan pola yang telah dipelajari dari data latih.
b. Evaluasi Skor pada *Leaderboard*: Setelah label prediksi dihasilkan, hasil prediksi ini dievaluasi dengan cara mengunggahnya ke halaman *leaderboard* yang disediakan oleh penyelenggara penelitian. Skor yang diperoleh pada *leaderboard* menunjukkan seberapa baik model dalam memprediksi sentimen secara keseluruhan.
c. Metrik Evaluasi: Performansi model dalam pengujian diukur menggunakan F1-score (*official*), *precision*, dan *recall.*

Dengan pengujian menggunakan data uji, penelitian ini dapat memverifikasi apakah optimasi yang telah dilakukan pada tahap pelatihan memberikan peningkatan performa yang signifikan dalam klasifikasi sentimen.

3. HASIL DAN PEMBAHASAN

Hasil penelitian ini ditampilkan berdasarkan perbandingan antara hasil tahap awal (*baseline*) dan hasil setelah optimasi dilakukan. Pada tahap awal, model menunjukkan performa awal yang menjadi acuan, sementara pada tahap optimasi, peningkatan kinerja dicapai melalui penyesuaian parameter yang optimal.

3.1 Hasil Tahap Awal (*Baseline*)

Metode *baseline* dalam klasifikasi adalah pendekatan awal yang bertujuan untuk mendapatkan gambaran dasar performa model klasifikasi tanpa melakukan optimasi. Metode *baseline* meliputi, *dataset* yang digunakan gabungan data kaesang v1 dan kaesang v2, teks prapocessing menggunakan firur *case folding* dan *cleaning text* dan word embedding FastText. Hasil klasifikasi terhadap data validasi, yang diukur secara macro-average untuk *F1-score*, *accuracy*, *precision* dan *recall* secara berturut-turut adalah 0.5502, 0.5500, 0.5560 dan 0.5466. Hasil baseline ini menunjukkan performa awal model klasifikasi dalam menangani dataset yang digunakan, serta menjadi acuan dalam mengukur peningkatan yang mungkin dicapai melalui optimasi atau penyesuaian parameter model. hasil dari metode baseline ini kemudian diuji menggunakan data *test* pada halaman *leaderboard* untuk melihat kemampuan model dalam mengklasifikasikan data baru. Hasil performa *baseline random forest* dicatat pada tabel 6 pada baris pertama dengan nama metode *Baseline.*

3.2 Hasil Tahap Optimasi

Setelah *baseline* ditetapkan, penelitian ini dapat melakukan berbagai upaya optimasi, seperti pemilihan fitur pada tahap prapocessing, penambahan data ekternal, penggunaan word embedding indoBERT dan parameter *tuning* untuk meningkatkan performa model. Model yang dioptimasi memiliki performa yang lebih baik dibandingkan *baseline.*
Berikut hasil optimasi model penelitian ini :

a. Hasil Pencarian Fitur Optimal Praposes Data

Pada tahap ini, optimasi dilakukan untuk mencari kombinasi fitur terbaik dalam proses praproses data, guna meningkatkan kinerja model klasifikasi sentimen. kemudian optimasi dilakukan dengan cara melakukan ekperimen pemilihan fitur terbaik yaitu Normalisasi Teks *(Case Folding)*, Pembersihan Teks *(Text Cleaning)*, Konversi *Emoticon*, Penghapusan *Stopwords*, di mana masing-masing fitur memiliki dua opsi: aktif atau tidak aktif. Hasil ekperimen dapat dilihat pada Tabel 3.

Table 3. Hasil Pencarian Fitur Optimal Praproses Data

no	Case Folding	Konversi Emoticon	Text Cleanin	Stopwords	F1-Score
1	aktif	aktif	aktif	aktif	0.6683
2	aktif	aktif	aktif	tidak aktif	0.6209
3	aktif	tidak aktif	aktif	aktif	0.6584
4	aktif	tidak aktif	aktif	tidak aktif	0.6584
5	aktif	aktif	tidak aktif	aktif	0.7010
6	aktif	aktif	tidak aktif	tidak aktif	0.6816
7	aktif	tidak aktif	tidak aktif	aktif	0.6750
8	aktif	tidak aktif	tidak aktif	tidak aktif	0.6750
9	tidak aktif	aktif	aktif	aktif	0.6417
10	tidak aktif	aktif	aktif	tidak aktif	0.6667
11	tidak aktif	tidak aktif	aktif	aktif	0.6500
12	tidak aktif	tidak aktif	aktif	tidak aktif	0.6333
13	tidak aktif	aktif	tidak aktif	aktif	0.7000
14	tidak aktif	aktif	tidak aktif	tidak aktif	0.6583
15	tidak aktif	tidak aktif	tidak aktif	aktif	0.6983
16	tidak aktif	tidak aktif	tidak aktif	tidak aktif	0.6750

Dari berbagai kombinasi fitur praproses data yang diuji dengan data validasi, hasil terbaik diperoleh dengan menggunakan *case folding* aktif,, *text cleaning* tidak aktif, konversi *emoticon* aktif, *stopwords* aktif, menghasilkan

Building of Informatics, Technology and Science (BITS)
Volume 6, No 3, Desember 2024 Page: 1668−1676
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v6i3.6335

F1-Score sebesar 0,7010. hasil terbaik ini kemudian digunakan pada tahap berikutnya, yaitu penambahan data eksternal untuk meningkatkan variasi dan performa model dalam klasifikasi sentimen.

b. Hasil Pengujian Dengan Penambahan Data Eksternal
Setelah mendapatkan model yang optimal dari proses optimasi praproses data, model tersebut digunakan untuk eksperimen berikutnya dengan menambahkan data eksternal ke data *train* kaesang . Data *train* Kaesang yang digunakan dalam eksperimen ini adalah data *train* kaesang, data ekternal menggunakan data covid dan open topic yang ditambahkan secara bertahap, dan pengujian menggunakan data validasi kaesang.. Hasil eksperimen ini dapat dilihat pada Tabel 4.

Table 4. Hasil Pengujian Dengan Tambahan Data Eksternal

no	Dataset	Jumlah Data	F1-Score	Akurasi	Precision	Recall
1	Kaesang + Covid 300	900	0.6919	0.6917	0.7055	0.6905
2	Kaesang + Covid 600	1200	0.6909	0.6833	0.6833	0.7090
3	Kaesang + Covid 900	1500	0.6529	0.6500	0.6737	0.6478
4	Kaesang + Covid 1200	1800	0.7145	0.7083	0.7341	0.7075
5	Kaesang + Open 300	900	0.6833	0.6417	0.6879	0.6833
6	Kaesang + Open 600	1200	0.6707	0.6667	0.6799	0.6670
7	Kaesang + Open 900	1500	0.6582	0.6500	0.6789	0.6504
8	Kaesang + Open 1200	1800	0.6685	0.6667	0.6823	0.6659
9	Kaesang + Covid 300 + Open 300	1200	0.7137	0.7083	0.7356	0.7071
10	Kaesang + Covid 600 + Open 600	1800	0.6654	0.6583	0.6858	0.6578
11	Kaesang + Covid 900 + Open 900	2400	0.6825	0.6750	0.7046	0.6745
12	Kaesang + Covid 1200 + Open 1200	3000	0.6825	0.6750	0.7046	0.6745

Penambahan data eksternal dalam bentuk dataset dari isu "covid" terbukti efektif dalam meningkatkan performa model. berdasarkan tabel, kombinasi dataset "kaesang + covid 1200" menghasilkan f1-score tertinggi, yaitu sebesar 0.7145. hasil ini menunjukkan bahwa penambahan data dari isu lain dapat membantu model dalam menangkap pola sentimen yang lebih kaya dan beragam, yang pada akhirnya berkontribusi pada peningkatan kinerja klasifikasi.

c. Parameter *Tuning*
Pada penelitian ini, optimasi model dilakukan melalui parameter tuning untuk mendapatkan performa terbaik dari algoritma *Random Forest* yang digunakan dalam klasifikasi sentimen. Parameter tuning telah diterapkan pada setiap eksperimen yang dilakukan dalam penelitian ini. Proses *tuning* dilakukan untuk setiap kombinasi data yang diujikan, memastikan bahwa model yang dihasilkan berada pada konfigurasi optimal dalam hal akurasi dan stabilitas performa. Hasil dari setiap kombinasi parameter yang diuji dapat dilihat pada Tabel 3. Hasil pencarian model optimal praproses data dan Tabel 4. Hasil pengujian dengan penambahan data eksternal.

3.3 Hasil Pengujian Model pada Data Test Kaesang

Pengujian ini dilakukan untuk kedua model, yaitu model tahap awal (*baseline*) dan model optimal, dengan langkah pertama melakukan prediksi label terhadap data test Kaesang yang tidak memiliki label. Pengujian ini bertujuan untuk menilai kemampuan kedua model dalam mengklasifikasikan sentimen pada data yang belum dilabeli dan mengevaluasi performa model pada data baru. Tabel 6 di bawah ini menampilkan hasil pengujian model pada sistem *leaderboard*, termasuk metrik evaluasi seperti *F1-score*, akurasi, *precision*, dan *recall* untuk setiap kelas sentimen. Hasil ini memberikan gambaran lengkap tentang performa tiap metode pada data uji Kaesang.

Table 6. Hasil Pengujian Model pada Data Test Kaesang

Model	Metode	Dataset	Run	F1-Score	Akurasi	Precision	Recall
Baseline	FastText dan RF	Kaesang	1	0.48	0.52	0.50	0.55
Optimasi	IndoBERT dan RF	Kaesang + Covid 1200	2	0.54	0.63	0.58	0.63

Berdasarkan hasil pengujian pada data uji, dapat disimpulkan bahwa model *baseline random forest* memperoleh skor f1 sebesar 0.48. setelah dilakukan optimasi, model *random forest* menunjukkan peningkatan dengan nilai *f1-score* 0.54. Peningkatan ini mengindikasikan bahwa langkah-langkah optimasi, termasuk penambahan dataset eksternal dan penggantian metode *word embedding*, telah berhasil meningkatkan kinerja model dalam mengklasifikasikan sentimen dengan lebih akurat.

3.4 Hasil Perbandingan dengan Metode Lain pada Sistem *Leaderboard*

Setelah mendapatkan hasil pengujian dari model yang dioptimasi, langkah berikutnya adalah membandingkan performa model ini dengan metode lain yang tercantum di sistem *leaderboard* penelitian. Pada sistem *leaderboard* memuat hasil dari berbagai metode yang diuji dalam tugas klasifikasi sentimen ini. Perbandingan ini bertujuan untuk menilai seberapa baik model yang dikembangkan dalam penelitian ini dibandingkan dengan metode lain dalam hal akurasi dan *F1-score*.

Building of Informatics, Technology and Science (BITS)
Volume 6, No 3, Desember 2024 Page: 1668−1676
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v6i3.6335

Berikut adalah tabel perbandingan performa model yang digunakan dalam penelitian ini dengan metode lain pada *leaderboard*:

Table 5. Hasil Perbandingan Dengan Metode lain

Rank	Tim	Metode	Run	F1-Score	Akurasi	Precision	Recall
1	BERT	BERT Clasification	2	0.60	0.62	0.65	0.63
5	Safrizal dkk [10]	SVM + Fasttext	2	0,53	0,62	0,53	0,59
6	Rasyid dkk [25]	Random forest (tfidf)	1	0,52	0.63	0.53	0,58
11	Saputra dkk [8]	SVM TF-IDF	3	0.51	0,61	0,52	0,59
12	Organizer [5]	SVM + TF IDF	2	0.51	0.50	0.51	0.51
13	Ravil dkk [9]	NB + PSO	3	0,50	0,59	0,52	0,58
24	Penelitian ini	Baseline RF	1	0.48	0.52	0.50	0.55
4	Penelitian ini	Optimasi RF	3	0.54	0.63	0.58	0.63

Berdasarkan hasil perbandingan pada tabel pengujian, dapat dilihat bahwa metode *random forest* yang telah dioptimasi dengan *f1-score* sebesar 0.54 berhasil mengungguli metode dari *organizer* (*f1-score* 0.51), penelitian lain menggunakan random forest dengan fitur input tfidf (*f1-score* 0.52) dan Penelitian laiinya menggunakakan metode SVM (*f1-score* 0.51). Model BERT *Clasification* berhasil menempati peringkat pertama dengan *f1-score* tertinggi yaitu 0.60, menunjukkan performanya yang lebih baik dalam memahami konteks dan memberikan hasil klasifikasi sentimen yang lebih akurat dibandingkan metode lainnya.

3. KESIMPULAN

Berdasarkan penelitian ini, dapat disimpulkan bahwa optimalisasi algoritma *random forest* melalui mengoptimalkan praproses data, penambahan data ekstern1al, penggunaan *word embedding* indoBERT dan parameter tuning berhasil meningkatkan performa model dalam klasifikasi sentimen. hasil optimasi pada random forest menunjukkan peningkatan f1-score dari 0.48 pada baseline menjadi 0.54, ini membuktikan bahwa strategi optimasi yang diterapkan mampu memberikan pengaruh positif terhadap performa model, khususnya dalam menangkap pola sentimen yang lebih akurat. Penggunaan model BERT, khususnya IndoBERT sebagai metode *embedding*, juga memberikan kontribusi signifikan dalam meningkatkan kualitas representasi kata. Kombinasi *embedding* IndoBERT dengan *Random Forest* sebagai model klasifikasi menunjukkan hasil yang kompetitif, mengungguli beberapa metode lain yang terdaftar di leaderboard, seperti SVM, *Random forest* dengan TF-IDF dan *Naive Bayes*. Secara keseluruhan, penelitian ini membuktikan bahwa pendekatan optimasi yang mencakup penambahan data eksternal, pemilihan metode *embedding* yang tepat, dan parameter *tuning* dapat menghasilkan model yang lebih akurat dan *robust* untuk klasifikasi sentimen pada data terbatas. Hasil penelitian ini dapat menjadi referensi bagi penelitian-penelitian serupa yang memerlukan metode analisis sentimen dengan keterbatasan data. Sebagai saran untuk penelitian selanjutnya, penggunaan model BERT lainnya yang lebih baru atau dengan penambahan teknik *ensemble* yang lebih kompleks dapat dieksplorasi untuk melihat potensi peningkatan lebih lanjut dalam akurasi dan kemampuan generalisasi model.

REFERENCES

[1] S. A. Salahudeen *et al.*, "HausaNLP at SemEval-2023 Task 12: Leveraging African Low Resource TweetData for Sentiment Analysis," in *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, A. Kr. Ojha, A. S. Doğruöz, G. Da San Martino, H. Tayyar Madabushi, R. Kumar, and E. Sartori, Eds., Toronto, Canada: Association for Computational Linguistics, Jul. 2023, pp. 50–57. doi: 10.18653/v1/2023.semeval-1.6.

[2] R. Vindua and A. U. Zailani, "Analisis Sentimen Pemilu Indonesia Tahun 2024 Dari Media Sosial Twitter Menggunakan Python," *JURIKOM (Jurnal Riset Komputer)*, vol. 10, no. 2, p. 479, Apr. 2023, doi: 10.30865/jurikom.v10i2.5945.

[3] E. Hokijuliandy, H. Napitupulu, and Firdaniza, "Application of SVM and Chi-Square Feature Selection for Sentiment Analysis of Indonesia's National Health Insurance Mobile Application," *Mathematics*, vol. 11, no. 17, p. 3765, Sep. 2023, doi: 10.3390/math11173765.

[4] M. Riekert, M. Riekert, and A. Klein, "Simple Baseline Machine Learning Text Classifiers for Small Datasets," *SN Comput Sci*, vol. 2, no. 3, May 2021, doi: 10.1007/s42979-021-00480-4.

[5] M. Ravil, S. Agustian, M. Fikry, and F. Insani, "Peningkatan Performa Klasifikasi Sentimen Tweet Kaesang Menggunakan Naïve Bayes dengan PSO pada Dataset Kecil," *KLIK: Kajian Ilmiah Informatika dan Komputer*, vol. 4, no. 6, pp. 2909–2917, 2024, doi: 10.30865/klik.v4i6.1939.

[6] 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.

[7] O. Rasyid, "Klasifikasi Sentimen Kaesang Sebagai Ketua PSI Menggunakan Chi-Square Dengan Fitur TF-IDF Metode Random Forest," Skripsi, Fakultas Sains dan Teknologi, Universitas Sultan Syarif Kasim Riau, Pekanbaru, 2024.

[8] S. D. Lestari and E. B. Setiawan, "Sentiment Analysis Based on Aspects Using FastText Feature Expansion and NBSVM Classification Method," *Journal of Computer System and Informatics (JoSYC)*, vol. 3, no. 4, pp. 469–477, Sep. 2022, doi: 10.47065/josyc.v3i4.2202.

[9] J. J. Sanchez-Medina, "Sentiment analysis and random forest to classify LLM versus human source applied to Scientific Texts," *ArXiv*, Apr. 2024.

Building of Informatics, Technology and Science (BITS)
Volume 6, No 3, Desember 2024 Page: 1668−1676
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v6i3.6335

[10] P. Sayarizki and H. Nurrahmi, "Implementation of IndoBERT for Sentiment Analysis of Indonesian Presidential Candidates," *Journal on Computing*, vol. 9, no. 2, pp. 61–72, 2024, doi: 10.34818/indojc.2024.9.2.934.
[11] 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," *ArXiv*, Jul. 2024.
[12] S. Gustian, "Small_DataSet_Sentiment_Classification," Github. Accessed: Dec. 09, 2024. [Online]. Available: https://github.com/s4gustian/Small_DataSet_Sentiment_Classification
[13] S. Khairunnisa, A. Adiwijaya, and S. Al Faraby, "Pengaruh Text Preprocessing terhadap Analisis Sentimen Komentar Masyarakat pada Media Sosial Twitter (Studi Kasus Pandemi COVID-19)," *JURNAL MEDIA INFORMATIKA BUDIDARMA*, vol. 5, no. 2, p. 406, Apr. 2021, doi: 10.30865/mib.v5i2.2835.
[14] F. Rahutomo and A. R. T. H. Ririd, "Evaluasi Daftar Stopword Bahasa Indonesia," *Jurnal Teknologi Informasi dan Ilmu Komputer*, vol. 6, no. 1, pp. 41–48, Jan. 2019, doi: 10.25126/jtiik.2019611226.
[15] A. Nurdin, B. Anggo Seno Aji, A. Bustamin, and Z. Abidin, "PERBANDINGAN KINERJA WORD EMBEDDING WORD2VEC, GLOVE, DAN FASTTEXT PADA KLASIFIKASI TEKS," *Jurnal Tekno Kompak*, vol. 14, no. 2, p. 74, Aug. 2020, doi: 10.33365/jtk.v14i2.732.
[16] F. Koto, J. H. Lau, and T. Baldwin, "IndoBERTweet: A Pretrained Language Model for Indonesian Twitter with Effective Domain-Specific Vocabulary Initialization," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, M.-F. Moens, X. Huang, L. Specia, and S. W. Yih, Eds., Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 10660–10668. doi: 10.18653/v1/2021.emnlp-main.833.
[17] B. Wilie *et al.*, "IndoNLU: Benchmark and Resources for Evaluating Indonesian Natural Language Understanding," in *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, K.-F. Wong, K. Knight, and H. Wu, Eds., Suzhou, China: Association for Computational Linguistics, Dec. 2020, pp. 843–857. doi: 10.18653/v1/2020.aacl-main.85.
[18] L. Breiman, "Random Forests," *Mach Learn*, vol. 45, no. 1, pp. 5–32, 2001, doi: 10.1023/A:1010933404324.
[19] I. Afdhal *et al.*, "Penerapan Algoritma Random Forest Untuk Analisis Sentimen Komentar Di YouTube Tentang Islamofobia," *Jurnal Nasional Komputasi dan Teknologi Informasi*, vol. 5, no. 1, 2022.
[20] J. Zeffora and S. Shobarani, "Optimizing random forest classifier with Jenesis-index on an imbalanced dataset," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 26, no. 1, p. 505, Apr. 2022, doi: 10.11591/ijeecs.v26.i1.pp505-511.
[21] Scikit-learn Developers, "sklearn.ensemble.RandomForestClassifier," Scikit-learn. Accessed: Dec. 09, 2024. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html
[22] P. Probst, M. Wright, and A.-L. Boulesteix, "Hyperparameters and Tuning Strategies for Random Forest," Apr. 2018, doi: 10.1002/widm.1301.
[23] R. R. Sani, Y. A. Pratiwi, S. Winarno, E. D. Udayanti, and F. Alzami, "Analisis Perbandingan Algoritma Naive Bayes Classifier dan Support Vector Machine untuk Klasifikasi Berita Hoax pada Berita Online Indonesia," *Jurnal Masyarakat Informatika*, vol. 13, no. 2, pp. 85–98, Nov. 2022, doi: 10.14710/jmasif.13.2.47983.
[24] P. Yohana, S. Agustian, and S. Kurnia Gusti, "Klasifikasi Sentimen Masyarakat terhadap Kebijakan Vaksin Covid-19 pada Twitter dengan Imbalance Classes Menggunakan Naive Bayes," *Seminar Nasional Teknologi Informasi, Komunikasi dan Industri (SNTIKI)*, vol. 26, p. 2022, Oct. 2022.
[25] S. Satapara *et al.*, "Overview of the HASOC Subtracks at FIRE 2023: Detection of Hate Spans and Conversational Hate-Speech," in *Proceedings of the 15th Annual Meeting of the Forum for Information Retrieval Evaluation*, New York, NY, USA: ACM, Dec. 2023, pp. 10–12. doi: 10.1145/3632754.3633277.