

Klasifikasi Sentimen Menggunakan Metode Passive Aggressive dengan Menggunakan Model Bahasa BERT pada Dataset Kecil

Yazid Abdullah Subhi, Surya Agustian*, Muhammad Irsyad, Fitri Insani

Fakultas Sains dan Teknologi, Teknik Informatika, Universitas Islam Negeri Sultan Syarif Kasim Riau, Pekanbaru, Indonesia

Email: ¹1205010498@students.uin-suska.ac.id, ^{2,*}surya.agustian@uin-suska.ac.id,

³irsyadtech@uin-suska.ac.id, ⁴fitri.insani@uin-suska.ac.id

Email Penulis Korespondensi: surya.agustian@uin-suska.ac.id

Submitted: 03/12/2024; Accepted: 21/12/2024; Published: 25/12/2024

Abstrak—Klasifikasi teks merupakan salah satu tugas pemrosesan bahasa alami yang sangat populer, khususnya dalam konteks klasifikasi sentimen. Data latih yang terbatas menjadi permasalahan yang banyak dibahas dalam berbagai penelitian klasifikasi teks. Penelitian ini berfokus pada pengoptimalkan performa klasifikasi menggunakan algoritma Passive Aggressive (PA) dengan memanfaatkan data pelatihan yang terbatas. Penelitian ini juga membandingkan metode representasi teks konvensional, seperti TF-IDF, dengan pendekatan yang lebih modern menggunakan word embeddings seperti FastText dan BERT. Dataset utama mencakup isu sentimen yang dibahas, terkait pengangkatan Kaesang Pangarep sebagai ketua umum PSI, yang dikumpulkan melalui teknik crawling Twitter, dengan pengelompokan label sentimen menjadi positif, negatif, dan netral. Dari dua versi data train yang tersedia, masing-masing hanya terdiri atas 300 tweet dengan kelas positif, negatif dan netral yang berimbang. Data ini dibagi menjadi 80% untuk training dan 20% untuk validasi dalam mencari model yang optimal. Data eksternal dengan isu berbeda yang sudah memiliki label sentimen, digunakan untuk menambah jumlah data train. Dari eksperimen yang dilakukan, model bahasa BERT yang digunakan sebagai fitur input pada metode passive aggressive dengan penyesuaian hyperparameter, memiliki performa yang lebih baik dibandingkan fitur TF-IDF. Evaluasi terhadap data test menunjukkan fitur BERT pada Passive Aggressive menghasilkan *F1-score* sebesar 0.52, lebih baik dibandingkan dengan representasi teks konvensional seperti TF-IDF, dengan perolehan *F1-score* sebesar 0.42. Pemanfaatan model bahasa BERT memberikan kontribusi yang signifikan terhadap peningkatan performa klasifikasi teks di bidang pemrosesan bahasa alami, khususnya pada metode passive aggressive.

Kata Kunci: Klasifikasi Sentimen; Dataset Kecil; BERT; Passive Aggressive

Abstract—Text classification is one of the most popular tasks in natural language processing, especially in the context of sentiment classification. Insufficient training data poses a significant challenge in many text classification studies. This research focuses on optimizing classification performance using the Passive Aggressive (PA) algorithm, leveraging limited training data. It compares conventional text representation methods like TF-IDF with modern approaches employing word embeddings such as FastText and BERT. The primary dataset encompasses sentiment issues related to Kaesang Pangarep's appointment as the chairman of PSI, gathered through Twitter crawling, and classified into positive, negative, and neutral sentiment labels. Two versions of the training data, each containing only 300 balanced tweets for positive, negative, and neutral classes, were used. The data was split 80% for training and 20% for validation in the search for an optimal model. External data with different issues and pre-existing sentiment labels was used to augment the training data. Experimental results demonstrated that the BERT language model, used as input features for the Passive Aggressive method with hyperparameter tuning, outperformed TF-IDF features. Evaluation on the test data revealed that BERT features with Passive Aggressive achieved an *F1-score* of 0.52, surpassing the conventional TF-IDF representation with an *F1-score* of 0.42. The utilization of the BERT language model significantly contributed to improving text classification performance in the field of natural language processing, particularly for the Passive Aggressive method.

Keywords: Classification; Small Dataset; BERT; Passive Aggressive

1. PENDAHULUAN

Klasifikasi teks adalah salah satu tugas pemrosesan bahasa alami yang paling sering digunakan. Aplikasi umum dari klasifikasi teks meliputi identifikasi spam, klasifikasi teks berita, klasifikasi sentimen, analisis emosi, dan penilaian niat, dan lain-lain [1]. Namun, performa algoritma klasifikasi teks sangat bergantung pada jumlah dan kualitas data training yang tersedia.

Salah satu tantangan utama dalam klasifikasi teks adalah keterbatasan ukuran dataset. Dataset yang kecil dapat menyebabkan model pembelajaran mesin kurang terlatih, yang berpotensi mengakibatkan *overfitting* atau *underfitting*. *Overfitting* terjadi ketika model terlalu rumit dan hanya mengingat data pelatihan, sehingga gagal dalam menggeneralisasi dengan baik pada data baru [2], mengakibatkan kinerja yang buruk ketika dihadapkan dengan data di luar data pelatihan. *Underfitting* adalah keadaan ketika kinerja model bernilai buruk, baik pada training maupun testing data. *Underfitting* terjadi akibat model yang terlalu tidak fleksibel [3].

Dalam klasifikasi teks, beberapa teknik preprocessing sangat penting untuk meningkatkan kualitas data. Teknik-teknik seperti *case folding*, *stopword removal*, *stemming* dan tokenisasi dapat membantu mengurangi kompleksitas data serta meningkatkan representasi teks yang lebih relevan untuk proses klasifikasi [4][5][6]. Selain teknik-teknik tradisional ini, pendekatan yang lebih maju seperti BERT (*Bidirectional Encoder Representations from Transformers*) telah terbukti lebih efektif dalam menghasilkan representasi teks kontekstual, terutama ketika data latih terbatas. Hal ini dapat meningkatkan performa klasifikasi pada beberapa metode *machine learning*.

Masalah kurangnya data yang diangkat dalam penelitian klasifikasi teks, masih sangat jarang dieksplorasi. Padahal dalam kondisi nyata, banyak tugas-tugas klasifikasi teks yang membutuhkan kecepatan hasil, sehingga ketersediaan data latih yang mencukupi beserta label kelasnya menjadi tantangan tersendiri. Misalnya seorang tokoh

publik ingin mengetahui sentimen masyarakat terhadap dirinya. Kemudian ia menyewa sebuah perusahaan konsultan IT untuk melakukan analisis sentimen tersebut. Karena output yang membutuhkan waktu cepat, maka konsultan perlu memanfaatkan mesin untuk melakukan klasifikasi tersebut. Namun, tidak mungkin melakukan pengembangan data training yang cukup, karena memberikan label yang banyak pada data akan membutuhkan waktu dan biaya [7].

Penelitian ini menyelidiki langkah-langkah optimasi dari proses klasifikasi yang dilakukan oleh mesin dengan memanfaatkan data training yang kecil. Tujuan dilakukannya optimasi adalah untuk meningkatkan performa klasifikasi, dalam hal ini agar hasil klasifikasi yang dilakukan mesin lebih akurat lagi. Kasus yang diangkat dalam penelitian ini adalah sentimen masyarakat terhadap pengangkatan Kaesang Pangarep, sebagai ketua umum PSI tanpa melalui proses pemilihan partai. Tugas ini merupakan sebuah shared task yang diorganisasikan oleh Agustian dkk dengan set data training sangat kecil, yaitu berjumlah 300 sampel, masing-masing 100 tweet untuk kelas positif, negatif dan netral.

Passive Aggressive (PA) adalah algoritma pembelajaran online yang memperbarui modelnya setiap kali menemui data baru dan dapat menyesuaikan bobot berdasarkan data tersebut. Klasifikator ini memiliki parameter regularisasi yang menentukan apakah data telah diklasifikasikan dengan tepat atau tidak [8]. Algoritma ini terbukti efektif untuk deteksi berita palsu berdasarkan konten artikel, penelitian yang dilakukan oleh Kumar menunjukkan bahwa *Passive Aggressive* mencapai akurasi 94% lebih tinggi dibanding Naive Bayes yang hanya mencapai akurasi 89% [9]. Selain itu, Penelitian Shridhar dengan judul “Subword Semantic Hashing for Intent Classification on Small Datasets”, *Passive Aggressive* mampu mencapai kinerja yang sangat baik pada *dataset Chatbot*, dengan skor akurasi rata-rata 99,6% [10]. Oleh karena itu, algoritma ini menjadi salah satu teknik dasar yang sering diterapkan dalam berbagai aplikasi pemrosesan bahasa alami dan klasifikasi teks.

Namun, seiring dengan tantangan dalam menangani dataset yang kecil, pendekatan yang lebih canggih seperti BERT telah menunjukkan kemampuannya yang luar biasa dalam memahami konteks teks secara mendalam. Penelitian yang dilakukan Ostendorff misalnya, penggabungan representasi teks BERT dengan metadata dari penulis dan embedding dari Wikidata dapat meningkatkan hasil klasifikasi teks. Hasilnya, *F1-score* mencapai 87,20% untuk klasifikasi delapan label dan 64,70% untuk 343 label, membuktikan bahwa metode ini sangat efektif, terutama dalam kondisi data pelatihan yang terbatas [11]. Penelitian lainnya yang dilakukan oleh Sazan, menunjukkan bahwa embedding BERT yang dikombinasikan dengan arsitektur CNN-BiLSTM, secara efektif mengenali nuansa teks Bangla yang relevan dengan konten depresif, dengan *F1-score* sebesar 84%. Ini menunjukkan bahwa, BERT dapat memberikan pemahaman yang lebih mendalam tentang teks dibandingkan dengan teknik representasi teks lainnya seperti TF-IDF dan FastText [12].

Untuk mengoptimalkan performa model klasifikasi teks, penulis akan melakukan beberapa pendekatan. Berdasarkan eksperimen awal, model baseline *Passive Aggressive* dengan fitur ekstraksi TF-IDF menunjukkan indikasi overfitting, dengan *F1-score* pada data latih mencapai 0.99, namun turun drastis pada data validasi dan pengujian, masing-masing sebesar 0.58 dan 0.42. Hasil ini menyoroti keterbatasan dataset kecil dalam melatih model secara optimal. Oleh karena itu, penulis memanfaatkan data eksternal untuk memperkaya dataset utama guna mengurangi risiko overfitting. Selanjutnya, dilakukan perbandingan antara TF-IDF, FastText dan BERT dalam representasi teks. Penelitian sebelumnya menunjukkan bahwa BERT lebih unggul dalam klasifikasi dibandingkan TF-IDF, terutama pada dataset kecil, seperti yang dibuktikan oleh penelitian Ostendorff dan Sazan. Terakhir penulis melakukan Penyesuaian hyperparameter menggunakan teknik grid search. Grid search memungkinkan penentuan parameter algoritma secara sistematis dengan menguji kombinasi nilai yang sudah ditetapkan, yang dapat membantu meningkatkan akurasi model dan mengurangi risiko overfitting [13].

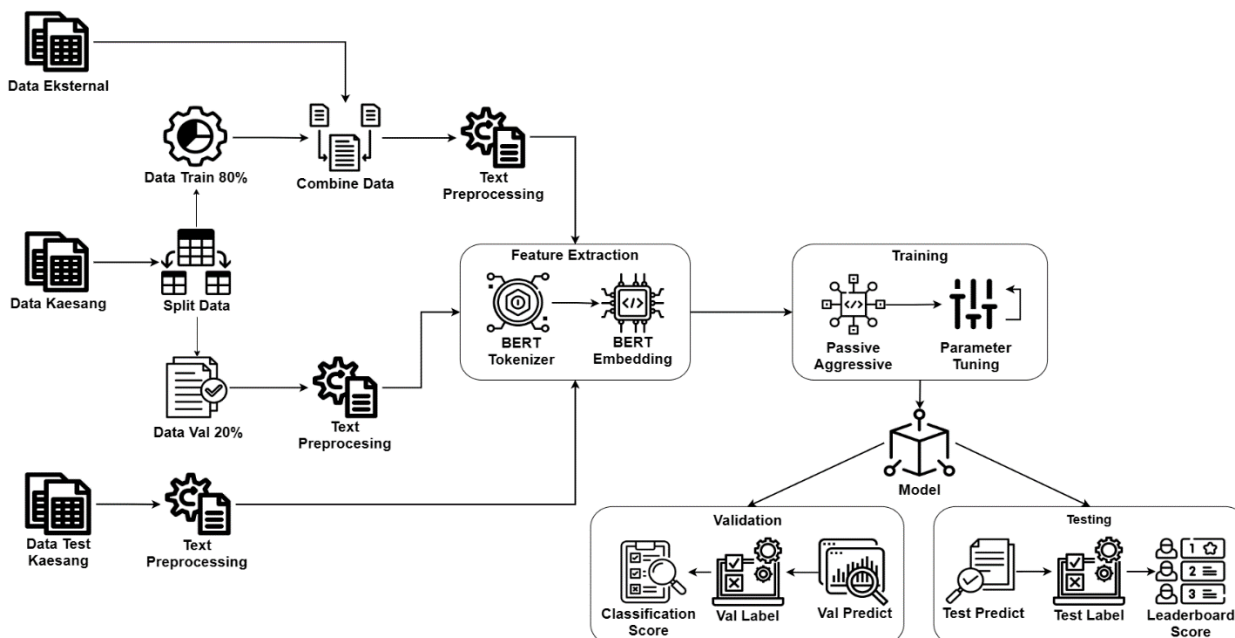
2. METODOLOGI PENELITIAN

2.1 Tahapan Penelitian

Untuk memberikan gambaran yang jelas mengenai alur dalam penelitian ini, penulis menyajikan tahapan penelitian yang dilakukan secara sistematis. Berikut ini adalah gambaran seluruh proses penelitian, mulai dari tahap pembagian data, preprocessing, pemanfaatan model BERT sebagai tokenizer dan fitur embedding, hingga evaluasi model menggunakan metode *Passive Aggressive* dengan parameter tuning. Penjelasan detail mengenai setiap tahap dapat dilihat pada Gambar 1.

Tahapan awal penelitian ini dimulai dari pembagian Data Kaesang menjadi data training (80%) dan data validasi (20%). Selanjutnya data training dapat digabungkan dengan Data Eksternal untuk memperkaya kosa kata data training. Seluruh data kemudian melalui tahap text preprocessing sebelum diproses menggunakan model BERT, yang berfungsi sebagai Tokenizer untuk mengubah teks menjadi representasi token, serta menghasilkan embedding dari token. Hasil embedding BERT kemudian digunakan sebagai input ke metode *Passive Aggressive Classifier*, dengan dilakukan parameter tuning untuk mendapatkan performa yang terbaik.

Evaluasi model dilakukan dengan menggunakan data validasi, di mana performa model diukur menggunakan metrik evaluasi seperti *F1-score* dan confusion matrix. Model optimal yang diperoleh dari proses ini kemudian digunakan untuk mengklasifikasi pada data testing, dan hasil klasifikasi akan dibandingkan dengan label sebenarnya (tes label) untuk mengevaluasi performa model secara keseluruhan.



Gambar 1. Tahapan Penelitian

2.2 Dataset

Data dikumpulkan menggunakan teknik *crawling* dari Twitter dengan kata kunci "Kaesang PSI" pada periode 25 September hingga 3 Oktober 2023. *Crawler* digunakan untuk mengotomatisasi pengumpulan data berupa tweet, re-tweet, likes, dan komentar yang relevan dengan topik tersebut. Pemberian label sentimen (positif, netral, negatif) dilakukan melalui *crowdsourcing* oleh empat anotator. Label akhir ditentukan berdasarkan mayoritas suara. Jika tidak ada kesepakatan mayoritas, tweet tersebut dihapus dari dataset. Proses ini memastikan data yang dikumpulkan relevan dan terverifikasi secara efisien.

Tabel 1. Dataset Penelitian [7]

Nama Dataset	Positif	Netral	Negatif	Jumlah
Train Kaesang v1	100	100	100	300
Train Kaesang v2	100	100	100	300
Data Covid	463	6664	873	8000
Data Open Topic	1505	3408	2656	7569
Data Test Kaesang	-	-	-	924

Dalam penelitian ini, dataset yang tersedia untuk penelitian adalah seperti yang dapat dilihat pada Tabel 1 [7] dengan angka-angka pada tabel adalah jumlah sampel untuk setiap kelas positif, negatif dan netral. Dari sekumpulan ataset ini, nantinya dibagi menjadi dua bagian yaitu data train dan data validasi. Data Kaesang v2 dengan contoh tweet seperti pada Tabel 2, displit menjadi data training sebanyak 80%, yang digunakan untuk model *baseline* dan 20% sebagai data validasi. Data validasi digunakan untuk pengujian model dari seluruh komposisi dan variasi langkah optimasi yang dilakukan untuk mendapatkan model yang paling optimal.

Tabel 2. Contoh Tweet Data Train Kaesang v2

Label	Tweet
Negatif	@adearmando61 Kaesang sebenarnya pengen bilang: Kamu itu udah tua bangka, gak cocok jadi kader PSI yg partai anak muda.
Netral	@ch_chotimah2 @ganjarpranowo @PartaiPerindo @Hary_Tanoe @DPP_PPP Kebayang kalau PSI dukung Ganjar dan Kaesang sebagai Ketum PSI posisinya setara dgn Ketum2 parpol lain.
Positif	@gibran_gen @kaesangp Kinerja mas ketum PSI mas kaesang Gercep Mirip mas Gibran

Dataset tambahan, yaitu Kaesang v1, Covid dan Open Topic [7], juga digunakan untuk memperkaya data training dan meningkatkan performa model. Penambahan data lainnya ke dalam porsi data training awal (Kaesang v2) yang 80% akan menambah daftar kosa kata untuk pembelajaran mesin yang lebih baik. Problem penelitian ini tidak membatasi bagaimana komposisi data yang dapat digunakan untuk training sampai mendapatkan model yang paling optimal. Karena itu proses penambahan data dengan topik yang berbeda dari isu utama (sentimen terhadap Kaesang), dilakukan secara empiris.

2.3 Text Preprocessing

Text preprocessing adalah tahap penting yang bertujuan mengekstraksi informasi dari teks tidak terstruktur serta menentukan relevansi dokumen dengan kebutuhan pengguna. Berikut yang digunakan dalam tahap text preprocessing:

- Case Folding* yaitu teknik untuk mengubah semua huruf menjadi huruf kecil (*lowercase*).
- Text Cleaning* merupakan proses pembersihan kata dengan menghilangkan delimiter seperti koma (,), titik (.), dan delimiter lainnya [14]. Selain itu penulis juga menghapus username, link dan karakter yang diulang.
- Stopword Removal* bertujuan mengurangi kompleksitas dokumen dengan menghilangkan kata-kata umum seperti "ini", "dan", "itu", yang tidak memiliki makna signifikan dalam penambangan teks [15].
- Stemming* adalah teknik yang mengubah kata ke bentuk dasarnya dengan menghilangkan akhiran [16].

Secara umum, analisis sentimen memerlukan penanganan khusus dalam pemrosesan awal teks sebelum proses klasifikasi. Tidak semua langkah-langkah umum dalam pemrosesan teks untuk kebutuhan information retrieval (*search engine*) bermanfaat dalam klasifikasi. Oleh karena itu, penulis juga menyelidiki komposisi langkah pemrosesan teks yang tepat dan dapat menghasilkan model paling optimal untuk kebutuhan klasifikasi sentimen.

Tahapan text preprocessing yang dilakukan dalam penelitian ini meliputi *case folding* dan *text cleaning* sebagai dasar *text preprocessing* (langkah a dan b). *Stopword removal* dan *stemming* merupakan opsi yang dapat digunakan atau tidak. Langkah normalisasi teks yang mengubah typo atau kata-kata yang berupa singkatan, tidak dilakukan dalam penelitian ini, misalnya kata “yg” tidak diubah menjadi “yang”, “gak” tidak diubah menjadi “tidak”, “udah” tidak diubah menjadi “sudah”, dan seterusnya.

Tabel 3. Penerapan Text Preprocessing

Proses	Text sebelum proses	Text sesudah proses
Text Cleaning	@adearmando61 Kaesang sebenarnya pengen bilang: Kamu itu udah tua bangka, gak cocok jadi kader PSI yg partai anak muda.	kaesang sebenarnya pengen bilang kamu itu udah tua bangka gak cocok jadi kader psi yg partai anak muda
Stopword Removal	@adearmando61 Kaesang sebenarnya pengen bilang: Kamu itu udah tua bangka, gak cocok jadi kader PSI yg partai anak muda.	kaesang pengen bilang udah tua bangka gak cocok kader PSI yg partai anak muda
Stemming	@adearmando61 Kaesang sebenarnya pengen bilang: Kamu itu udah tua bangka, gak cocok jadi kader PSI yg partai anak muda.	kaesang benar ken bilang kamu itu udah tua bangka gak cocok jadi kader psi yg partai anak muda

Simulasi dari tahapan preprocessing ini dapat dilihat pada Tabel 3 di atas. Proses yang ditunjukkan dalam Tabel 3 adalah tweet yang melalui tahapan *preprocessing* dasar (*text cleaning*), kemudian dilanjutkan dengan stopword removal, atau dilanjutkan dengan stemming.

2.4 Ekstraksi Fitur

Fitur digunakan untuk mengenali objek berdasarkan karakteristik khusus yang dimilikinya. Tujuan utama dari ekstraksi fitur menghasilkan ciri-ciri dari suatu objek, sehingga dapat dilakukan analisis dan perbandingan untuk mengklasifikasikan karakteristik yang ada dalam sebuah objek, dalam hal ini adalah data. Untuk data teks, fitur yang dapat diekstrak dapat berupa token kata, karakter n-gram, maupun frasa dan gabungan kata (*word n-gram*). Selanjutnya fitur-fitur tersebut ditransformasi menjadi vektor fitur yang merupakan nilai-nilai angka yang mewakili token fitur dengan perhitungan tertentu. Dalam penelitian ini, fitur yang diekstrak kemudian ditransformasikan menggunakan metode TF-IDF dan word embeddings (FastText dan BERT).

2.4.1 TF-IDF

TF-IDF (*Term Frequency - Inverse Document Frequency*) adalah metode untuk mengkonversi data teks menjadi data numerik dengan memberi bobot pada setiap kata atau fitur. TF-IDF adalah metrik statistik yang menilai pentingnya kata dalam sebuah dokumen. TF mengukur seberapa sering kata muncul dalam dokumen, sedangkan IDF mengukur pentingnya kata dengan memberikan bobot lebih rendah pada kata-kata yang muncul di banyak dokumen dan bobot lebih tinggi pada kata-kata yang jarang muncul, sehingga kata yang lebih signifikan dalam dokumen tertentu memiliki bobot lebih tinggi [17].

Penelitian ini mentransformasi tweet di dalam data training menjadi *bag of words*. Data train merupakan komposisi dari data kaesang setelah dikeluarkan porsi data validasi, dan data covid dalam proporsi tertentu, fleksibel sesuai dengan eksperimen yang dilakukan. Kemudian daftar kata *bag of words* menjadi dasar penghitungan nilai TF dan IDF dari setiap kata di dalam tweet, baik tweet untuk data training maupun data testing.

2.4.2 FastText

FastText adalah algoritma *word embeddings* yang dikembangkan oleh Facebook yang bertujuan untuk membuat representasi kata berbasis *character n-gram* [18]. memungkinkan algoritma ini untuk menangani kata-kata yang tidak terdapat dalam kosa kata (*out-of-vocabulary*). FastText telah terbukti efektif ketika dilatih pada dataset besar dengan

jutaan token [19], meskipun penelitian mengenai performanya pada *low resource language* seperti pada Bahasa Indonesia masih terbatas.

Penelitian ini menggunakan seluruh data tweet yang ada pada data Kaesang v1 dan v2 (termasuk data testing) dan Data Covid-19 untuk membangun lebih banyak kosa kata vektor word embeddings, sehingga konteks keberadaan kata di dalam kalimat tweet menjadi lebih terarah dalam ruang vektornya. Sementara data Open Topic tidak digunakan sama sekali. Pemilihan komposisi data ini dilakukan secara empiris.

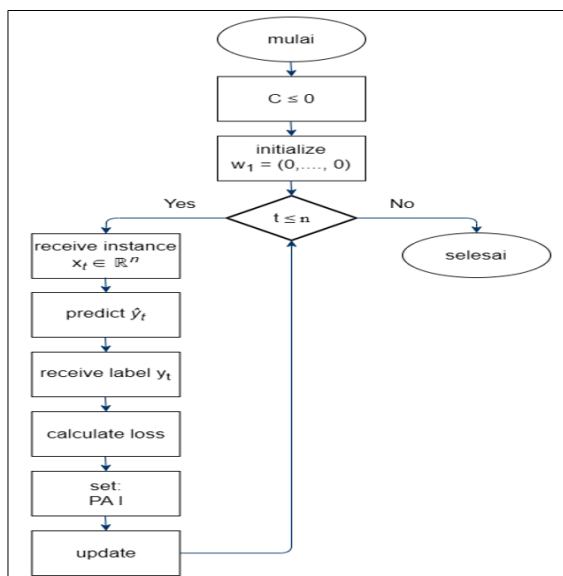
2.4.3 Bidirectional Encoder Representation from Transformers (BERT)

BERT (*Bidirectional Encoder Representations from Transformers*) adalah model yang digunakan untuk menghasilkan embedding dari teks, yaitu token [CLS] ditambahkan di awal setiap input dan berfungsi sebagai representasi keseluruhan dari kalimat, kemudian teks diurai menjadi token-token yang lebih kecil sesuai dengan aturan model BERT, lalu ditambahkan *padding* jika tweet terlalu pendek atau dipotong jika terlalu panjang. Hasil tokenisasi adalah representasi numerik dari teks yang dapat digunakan sebagai input untuk model [20], di mana embedding dari token [CLS] digunakan dalam model klasifikasi, memberikan informasi kontekstual yang kaya dan relevan. Keunggulan BERT terletak pada kemampuannya untuk memahami konteks dalam kalimat, menjadikannya lebih efektif dibandingkan metode ekstraksi fitur tradisional.

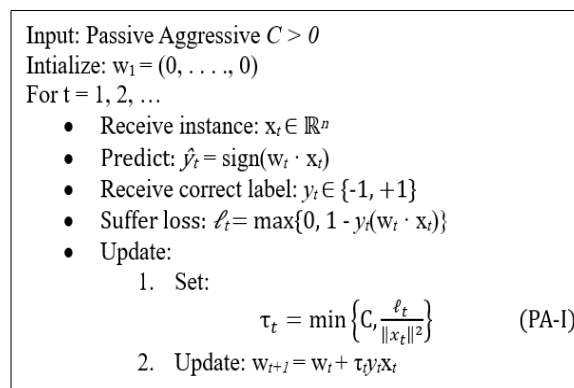
Pada penelitian ini, jenis BERT yang digunakan adalah IndoBERTtweet [21], yang dilatih menggunakan korpus tweet bahasa Indonesia, untuk menangani teks bahasa Indonesia, terutama dari media sosial seperti tweet. Model ini unggul dalam memahami teks informal, *slank*, dan gaya penulisan yang sering ditemukan di platform media sosial. Fitur input untuk metode PA adalah vektor yang dihasilkan dari transformasi tweet menggunakan word embeddings IndoBERTtweet.

2.5 Passive Aggressive Classifier

Algoritma *Passive Aggressive* adalah algoritma pembelajaran daring yang menggunakan prediktor linear untuk berbagai tugas, termasuk klasifikasi biner dan regresi [22]. Algoritma ini tidak memerlukan seluruh data dalam memori saat pelatihan, sehingga lebih efisien dalam penggunaan memori dan lebih cepat dibandingkan algoritma offline [23]. Menurut Ahmed & Rawat, klasifikasi PA bersikap pasif jika hasil yang diperoleh benar, tetapi agresif jika hasilnya salah. Algoritma ini mencari bobot baru yang paling mendekati bobot sebelumnya dan memenuhi nilai loss [24]. Berikut adalah gambar *flowchart* dan *pseudocode* dari algoritma *Passive Aggressive*.



Gambar 2. Flowchart PA



Gambar 3. Pseudocode PA

Gambar 2 dan Gambar 3 menjelaskan algoritma PA yang dimulai dengan inisialisasi bobot awal dan parameter regulasi C . Untuk setiap data x_t , algoritma mengklasifikasi label \hat{y}_t dan menghitung loss ℓ_t . Jika $loss > 0$, bobot diperbarui menggunakan nilai τ_t untuk memperbaiki kesalahan klasifikasi. Proses ini diulang hingga seluruh data selesai, memungkinkan model memperbaiki kesalahan secara adaptif dan agresif sambil tetap mempertahankan klasifikasi yang benar.

2.6 Optimasi

Dalam upaya meningkatkan performa model klasifikasi sentimen, langkah-langkah optimasi perlu dilakukan untuk mengatasi permasalahan overfitting dan meningkatkan akurasi pada dataset yang terbatas. Overfitting menjadi salah satu tantangan utama pada eksperimen awal, di mana model tidak mampu menggeneralisasi dengan baik terhadap data validasi dan testing. Langkah optimasi yang dilakukan dalam penelitian ini adalah sebagai berikut:

- Data Aggregation*: menggunakan data eksternal dari topik sentimen yang berbeda untuk memperluas data latih, sehingga dapat meningkatkan kemampuan model.
- Text Preprocessing*: menyelidiki variasi langkah text preprocessing seperti *stopword removal* dan *stemming* untuk menyederhanakan teks sekaligus mempertahankan informasi penting.
- Seleksi dan ekstraksi fitur: membandingkan teknik ekstraksi fitur seperti TF-IDF dan word embeddings FastText untuk menentukan representasi teks yang paling efektif dalam mengungkapkan informasi sentimen.
- Transfer Learning*: menggunakan model *pre-trained* seperti BERT untuk memanfaatkan pengetahuan bahasa yang telah dilatih pada dataset besar.
- Parameter tuning: Mengoptimalkan parameter model klasifikasi menggunakan grid search untuk menemukan kombinasi yang memberikan akurasi dan efisiensi terbaik pada model klasifikasi.

Dengan menerapkan berbagai variasi pada langkah di atas ini, eksperimen dilakukan untuk mencari model yang paling optimal, yaitu model yang menghasilkan nilai *F1-score* paling tinggi pada hasil prediksi terhadap data validasi. Dengan demikian, diharapkan model klasifikasi sentimen yang lebih efektif dan akurat dalam mengidentifikasi dan mengklasifikasikan sentimen pada dataset yang terbatas dapat diperoleh.

2.7 Parameter Tuning

Parameter tuning adalah proses untuk menemukan pengaturan parameter (atau konfigurasi) algoritma yang tepat sehingga kinerja algoritma tersebut dioptimalkan. Banyak pendekatan tuning parameter otomatis telah diusulkan [25]. Salah satunya adalah Grid Search, yang merupakan metode untuk mencari kombinasi parameter terbaik untuk model. Metode ini berfungsi dengan menguji semua kemungkinan kombinasi berdasarkan nilai parameter yang telah ditentukan oleh pengguna [26]. Grid search memiliki kelemahan dalam ruang berdimensi tinggi, namun seringkali dapat dengan mudah diparalelkan, karena nilai-nilai hyperparameter yang digunakan algoritma biasanya bersifat independen satu sama lain [27].

Dalam penelitian ini, untuk meningkatkan performa model, dilakukan pencarian kombinasi parameter terbaik dengan menggunakan teknik *grid search* dari *sklearn*. Parameter yang akan diuji adalah *C*, toleransi (*tol/stopping criteria*) dan iterasi maksimum dari algoritma PA dengan opsi nilai yang dipilih sesuai Table 4. Parameter lainnya menggunakan nilai default, seperti jenis perhitungan *loss*, tidak menggunakan *early stopping* dan nilai rata-rata.

Tabel 4. Parameter yang akan diuji

Parameter	Nilai
<i>C</i>	{0.001, 0.01, 0.1}
<i>Tol</i>	{0.0001, 0.001}
<i>Max_iter</i>	{3000, 4000, 5000}

Proses grid search yang dilakukan menghasilkan kombinasi parameter yang optimal untuk meningkatkan kinerja model Passive Aggressive. Dengan menemukan nilai parameter yang tepat, model menjadi lebih stabil dan dapat memberikan hasil klasifikasi yang lebih akurat pada data latih, validasi, dan testing. Hal ini menunjukkan pentingnya tuning parameter untuk memastikan model dapat bekerja dengan efisien dan memberikan performa terbaik, terutama dalam klasifikasi sentimen dengan dataset terbatas.

2.8 Evaluasi

Model paling optimal dievaluasi dengan menggunakan *F1-score* seperti persamaan (1) sebagai official score dalam penelitian ini. Kesalahan klasifikasi ditampilkan dengan confusion matrix, dan hasilnya digunakan untuk menentukan *precision*, *recall*, *accuracy*, dan *F1-score*. Setiap kelas akan dievaluasi menggunakan *F1-score*.

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (1)$$

F1-score digunakan dalam penelitian ini karena metrik ini memberikan evaluasi yang berimbang antara *precision* dan *recall*, yang merupakan aspek penting dalam mengukur kemampuan model dalam klasifikasi sentimen. Lebih khusus lagi, untuk pengukuran kinerja dengan 3 kelas seperti di dalam penelitian ini, penggunaan *F1-score* (*average*) menjadi lebih objektif karena mengukur kinerja klasifikasi berdasarkan rata-rata dari ketiga kelas. Formula untuk mengukur *precision*, *recall* dan *F1-score* pada masing-masing kelas dapat dilihat pada penelitian Yohana dkk. [28]. Sedangkan pengukuran kinerja terhadap data testing, dilakukan oleh sistem pada Leaderboard [7] sehingga peneliti dapat membandingkan hasil yang diperoleh dari metode yang diusulkan, dengan hasil yang didapat oleh peneliti lainnya.

3. HASIL DAN PEMBAHASAN

3.1 Set up Eksperimen

Untuk menemukan model klasifikasi PA yang paling optimal, serangkaian eksperimen dilakukan dengan skenario sebagai berikut:

- Penerapan model dasar (*baseline*):

Model ini menerapkan tahap pengujian awal dengan fitur dan langkah *text preprocessing* dasar saja, yaitu fitur TF-IDF dan hanya *text cleaning* dan *casefolding*, serta tanpa melakukan parameter tuning pada metode klasifikasi PA (hanya menggunakan parameter *default*). Sedangkan *dataset* yang digunakan hanya data train Kaesang v2 yang hanya terdiri atas 240 tweet (setelah displit menjadi data *train* dan validasi).

b. Pengembangan fitur model bahasa:

Model bahasa yang dipilih di antara TF-IDF, word embeddings FastText dan BERT. Fitur representasi teks ini diterapkan bersama-sama dengan kombinasi text preprocessing dan *data aggregation*. Apabila data train bertambah, maka fitur vektor TF-IDF dan FastText akan berubah pula, sesuai dengan komposisi data train yang baru. Sedangkan untuk BERT, digunakan model bahasa yang telah dilatih sebelumnya, yaitu pre-trained IndoBERTweet.

c. Variasi penerapan langkah *text preprocessing*:

Langkah text preprocessing tambahan yang diterapkan adalah penerapan *stemming* dan *stopword removal*, menyelidiki pengaruhnya terhadap hasil klasifikasi oleh metode PA. Langkah ini diterapkan setelah text tweet mengalami *text cleaning* dan *case folding*.

d. Penambahan/penggabungan data training (*data aggregation*):

e. Dari data yang tersedia untuk penelitian ini (Tabel 1), penambahan data untuk training dilakukan secara empiris (coba-coba), dengan menambahkan dengan data train Kaesang v1, dan atau secara bertahap sejumlah data dari data eksternal (data sentimen *covid* dan atau *open topic*). Tentu saja eksperimen yang dilakukan bisa banyak sekali. Oleh karena itu, penulis hanya melakukan beberapa eksperimen saja dengan jumlah penambahan tertentu (kelipatan 100 tweet setiap label pos-net-neg, sejumlah maksimal data tersedia)

f. Parameter tuning:

Dilakukan dengan cara sebagaimana diterangkan di dalam bagian 2.7.

3.2 Hasil Eksperimen Metode Dasar (Baseline)

Penelitian ini diawali dengan pengujian model PA menggunakan fitur TF-IDF dan penggunaan text preprocessing yang sederhana (hanya *cleaning* saja) dengan menggunakan data Kaesang v2 sebagai baseline. Tujuan dari baseline ini adalah untuk mengetahui kinerja awal model dalam mengklasifikasikan sentimen berdasarkan dataset yang terbatas. Model ini kemudian diaplikasikan untuk memprediksi data testing, dan hasilnya disubmit ke sistem Leaderboard dengan identifikasi RUN-1. Maksimal setiap peneliti dapat mengirimkan sampai 3 RUN hasil prediksi data testing.

Tabel 5. Hasil Model Baseline

Jenis pengujian	<i>F1-score</i>	Accuracy	Precision	Recall
Data Training	0.99	0.99	0.99	0.99
Data Validasi	0.58	0.58	0.60	0.58
Data Testing	0.42	0.49	0.47	0.49

Hasil pada Tabel 5 menunjukkan model baseline mengalami *overfitting*, dengan *F1-score* yang sangat tinggi pada data latih yaitu 0.99 namun menurun drastis pada validasi dengan *F1-score* 0.58 dan testing dengan *F1-score* 0.42. Perbedaan ini menunjukkan model terlalu menyesuaikan data latih sehingga gagal mengenali pola pada data baru, yang disebabkan oleh keterbatasan dataset kecil. Oleh karena itu, diperlukan upaya optimasi untuk meningkatkan kemampuan generalisasi model.

3.3 Penelusuran Model Optimal

Sebagaimana dijelaskan pada bagian set up eksperimen, penelusuran model optimal dilakukan dengan pengujian model hasil training pada data validasi. Hal ini bertujuan mengukur kinerja model yang telah dioptimasi pada data validasi yang sama. Pengujian ini menggunakan beberapa kombinasi model PA dengan metode ekstraksi fitur menggunakan model bahasa TF-IDF, word embeddings FastText, dan BERT. Selanjutnya, setiap kombinasi model diuji dengan berbagai kombinasi preprocessing, termasuk penerapan *stopword removal* dan *stemming*, untuk memahami pengaruhnya terhadap performa model pada data validasi. Pengujian dilakukan juga dengan mempertahankan kombinasi data tambahan untuk mengevaluasi dampak perbedaan dataset pada hasil klasifikasi. Tabel 6 berikut menunjukkan hasil pengujian data validasi untuk setiap kombinasi model, data dan text preprocessing.

Tabel 6. Penelusuran Model Optimal dan Pengujian Terhadap Data Validasi

Parameter	Dataset	<i>Stopword Removal</i>	<i>Stemming</i>	Parameter			Score	
				<i>C</i>	<i>Tol</i>	<i>Max_iter</i>	<i>F1-score</i>	acc
TF-IDF	KS2 + CV	-	-	0.01	0.001	3000	0.64	0.65
TF-IDF	KS2 + CV	✓	-	0.1	0.001	3000	0.62	0.63
TF-IDF	KS2 + CV	-	✓	0.01	0.001	3000	0.63	0.63
FastText	KS2 + KS1 + CV	-	-	0.001	0.0001	3000	0.58	0.58
FastText	KS2 + KS1	✓	-	0.001	0.001	5000	0.50	0.52
FastText	KS2 + CV	-	✓	0.001	0.0001	5000	0.55	0.58

Parameter	Dataset	Stopword Removal	Stemming	Parameter			Score	
				C	Tol	Max_iter	F1-score	acc
FastText	KS2 + KS1	✓	✓	0.001	0.001	5000	0.48	0.50
BERT	KS2 + KS1	-	-	0.001	0.001	4000	0.60	0.62
BERT	KS2 + CV	-	-	0.001	0.001	5000	0.66	0.67
BERT	KS2 + CV	✓	-	0.001	0.001	5000	0.64	0.65
BERT	KS2 + KS1 + CV	-	✓	0.001	0.001	3000	0.62	0.65
BERT	KS1 + CV + OP	✓	✓	0.001	0.001	3000	0.60	0.60

Tabel 6 memperlihatkan hasil pengujian data validasi menggunakan kombinasi model PA dengan fitur ekstraksi TF-IDF, FastText, dan BERT. Hasil menunjukkan bahwa TF-IDF memiliki performa terbaik, bila *stopword removal* dan *stemming* tidak diterapkan, dengan *F1-score* tertinggi 0.64 dan akurasi 0.65. FastText menunjukkan performa yang kurang stabil, dengan *F1-score* berkisar antara 0.48 hingga 0.55. Sementara itu, BERT mencapai *F1-score* 0.66 dan akurasi 0.67 pada data validasi dengan menggunakan data train KS2 dan CV, tanpa menggunakan *stemming* dan *stopword removal*.

Pengujian BERT lebih tinggi bila tidak menggunakan kombinasi ini, karena *pre-trained* BERT embeddings yang digunakan (IndoBERTtweet) juga dilatih tanpa menggunakan proses tersebut. Dengan demikian, kecocokan kata pada data latih yang kecil di penelitian ini, lebih besar dengan model embeddings yang telah dilatih dari 409 juta token kata dari model IndoBERTtweet [29].

3.4 Pengujian Data Test

Pengujian data test bertujuan untuk mengevaluasi performa akhir model yang telah dioptimasi pada data yang baru. Hasil pengujian ini menggambarkan perbandingan kinerja model PA dengan berbagai metode ekstraksi fitur dan optimasi data yang digunakan dalam eksperimen. Dari model yang telah dilatih dan dioptimasi, dipilih 2 model untuk diterapkan pada data testing, karena hanya tersedia 2 RUN lagi untuk hasil prediksi model yang dapat disubmit pada sistem Leaderboard. Hasil pengujian terhadap data Test menggunakan 2 model yang dipilih berdasarkan hasil penelusuran model optimal, yaitu menggunakan fitur TF-IDF dan BERT, sebagaimana pada Tabel 7 berikut.

Tabel 7. Pengujian Data Test

Run	Model	Data Train	Score			
			F1-score	Acc	Prec	Rec
Run 1	PA+TF-IDF	KS2	0.42	0.49	0.47	0.49
Run 2	PA+TF-IDF	KS2 + CV	0.42	0.51	0.44	0.49
Run 3	PA+BERT	KS2 + KS1	0.52	0.61	0.52	0.62

Pada Tabel 7, terlihat 3 hasil pengujian model PA menggunakan berbagai metode ekstraksi fitur dan optimasi data yang disubmit ke sistem Leaderboard. Model *baseline* (Run 1) menggunakan PA dengan TF-IDF tanpa penambahan data, menghasilkan *F1-score* sebesar 0.42 dan akurasi 0.49. Run kedua menggunakan kombinasi PA dengan fitur TF-IDF, dengan dataset tambahan data covid (CV), hanya berhasil meningkatkan akurasi menjadi 0.51, tetapi *F1-score* tidak meningkat. Pada Run 3, penggunaan model BERT embeddings pada gabungan data KS2 dan KS1 untuk training, menunjukkan peningkatan performa tertinggi dengan *F1-score* 0.52 dan akurasi 0.61. Hal ini menunjukkan bahwa kombinasi PA+BERT lebih efektif dalam menangkap pola sentimen dibandingkan TF-IDF.

3.5 Ranking LeaderBoard

Sistem Leaderboard bertujuan untuk mencatat score dari berbagai metode yang diusulkan para peneliti terhadap data test yang diberikan. Dengan demikian, para peneliti dapat membandingkan performa model yang dikembangkan dengan metode yang berbeda oleh penelitian lainnya. *Official score* yang digunakan adalah *F1-score*, yang dianggap sebagai indikator utama untuk menilai kemampuan model dalam melakukan klasifikasi sentimen. Selain *F1-score*, metrik tambahan seperti accuracy, precision, dan recall juga ditampilkan untuk memberikan gambaran yang lebih lengkap mengenai kualitas klasifikasi. Dengan cara ini, dapat dianalisis bagaimana model yang dikembangkan dalam penelitian ini berperilaku jika dibandingkan dengan metode-metode lain yang telah diuji pada dataset serupa. Hasil dari perbandingan penelitian ini dengan beberapa peneliti lainnya dapat dilihat pada Tabel 8.

Tabel 8. Perbandingan hasil pengujian

Peneliti	Model	F1-score	Acc	Prec	Rec
Rank 1 [30]	BERT classifier	0.60	0.70	0.59	0.68
Organizer [7]	SVM	0.51	0.61	0.53	0.57
Admin [7]	Baseline	0.40	0.45	0.49	0.49
Penelitian ini	Passive Aggressive	0.52	0.61	0.52	0.62

Dari Tabel 8 dapat dilihat bahwa model Passive Aggressive dengan fitur BERT embeddings yang diterapkan dalam penelitian ini menunjukkan hasil yang cukup baik, dengan *F1-score* 0.52 dan akurasi 0.61. Meskipun metode klasifikasi BERT sebagai fitur ekstraksi memberikan hasil yang lebih baik, model ini masih berada di bawah BERT yang memiliki *F1-score* 0.60. Sementara itu, penelitian ini dapat melampaui hasil yang dicapai oleh metode SVM

dengan fitur TF-IDF yang digunakan oleh Organizer [7], yang memberikan hasil *F1-score* sebesar 0.51. Hasil ini menunjukkan bahwa penggunaan BERT dalam kombinasi dengan model Passive Aggressive (PA) dapat memberikan hasil yang melampaui metode standar optimal yang ditetapkan organizer untuk dicapai oleh para peneliti, meskipun potensi untuk dilakukan penyempurnaan lebih lanjut masih terbuka.

4. KESIMPULAN

Untuk mengatasi permasalahan data training yang kecil, dapat dilakukan beberapa langkah optimasi, termasuk penambahan data eksternal, variasi teks preprocessing, eksplorasi teknik ekstraksi fitur lainnya seperti word embeddings FastText dan BERT, serta parameter tuning menggunakan grid search. Penelitian ini telah membuktikan bahwa langkah optimasi yang dilakukan dapat meningkatkan hasil klasifikasi sebesar 10% dibandingkan dengan metode dasarnya (*baseline*). Pendekatan transfer learning menggunakan pre-trained BERT embeddings sebagai fitur input metode Passive Aggressive menunjukkan stabilitas yang lebih baik dalam menangani data baru. Hal ini menunjukkan bahwa model BERT yang sudah dilatih terhadap jutaan token lebih mampu menangkap konteks yang lebih kompleks dalam teks, sehingga hasil klasifikasi pada data testing lebih konsisten. Ia juga menegaskan bahwa pendekatan transfer learning dengan menggunakan BERT sebagai ekstraksi fitur mampu meningkatkan kemampuan generalisasi model, khususnya pada dataset kecil, yang ditunjukkan dari hasil evaluasi *F1-score* terhadap data testing sebesar 52%, yang jauh lebih baik dibandingkan menggunakan fitur TF-IDF. Saran untuk penelitian selanjutnya, untuk memperluas dataset dengan topik yang lebih relevan (misalnya sama-sama sentimen terhadap tokoh politik), mengeksplorasi teknik fine-tuning pada model BERT, serta mempertimbangkan metode ensemble untuk menggabungkan keunggulan beberapa teknik ekstraksi fitur. Dengan pendekatan ini, diharapkan performa klasifikasi sentimen dapat lebih ditingkatkan, khususnya pada data testing.

REFERENCES

- [1] J. Cai, J. Li, W. Li, and J. Wang, "Deeplearning Model Used in Text Classification," in 2018 15th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP), IEEE, Dec. 2018, pp. 123–126. doi: 10.1109/ICCWAMTIP.2018.8632592.
- [2] S. Masripah and L. Yusuf, "Perbandingan Kriteria Decision Tree pada Pengetahuan Masyarakat pada Pemilihan Umum Presiden Indonesia," *INTI Nusa Mandiri*, vol. 18, no. 2, pp. 183–191, Feb. 2024, doi: 10.33480/inti.v18i2.5065.
- [3] D. N. Fathurrahman, A. B. Osmond, and R. E. Saputra, "Deep Neural Network untuk Pengenalan Ucapan pada Bahasa Sunda Dialek Tengah Timur (Majalengka)," vol. 5, p. 6073, Dec. 2018. [Online]. Available: <https://openlibrarypublications.telkomuniversity.ac.id/index.php/engineering/article/view/7967/7858>
- [4] H. H. Sinaga, "Perbandingan Metode Decision Tree Dan XGBOOST Untuk Klasifikasi Sentimen Vaksin Covid-19 Di Twitter," UIN Sultan Syarif Kasim, Pekanbaru, 2022, [Online]. Available: <https://repository.uin-suska.ac.id/65475/1/Habib%20Hakim%20Sinaga%20Repository.pdf>
- [5] M. M. Kusari and S. Agustian, "SVM Method with FastText Representation Feature for Classification of Twitter Sentiments Regarding the Covid-19 Vaccination Program," *Jurnal Teknologi Informasi & Komunikasi Digital Zone*, vol. 13, pp. 140–150, May 2022, doi: 10.31849/digitalzone.v13i2.11531
- [6] A. Zikri, "Penerapan Support Vector Machine Dan FastText Untuk Mendeteksi Hate Speech Dan Abusive Pada Twitter," UIN Sultan Syarif Kasim, Pekanbaru, 2023. [Online]. Available: <http://repository.uin-suska.ac.id/id/eprint/74125>
- [7] S. Agustian, M. Irfan Syah, N. Fatiara, and R. Abdillah, "New Directions in Text Classification Research: Maximizing The Performance of Sentiment Classification from Limited Data Arah Baru Penelitian Klasifikasi Teks: Memaksimalkan Kinerja Klasifikasi Sentimen dari Data Terbatas," pp. 1–10, 2024, [Online]. Available: https://github.com/s4gustian/Small_DataSet_Sentiment_Classification
- [8] A. Vatsa, A. Kumar, S. Vats, and A. Kumar, "Comparing the Performance of Classification Algorithms for Melanoma Skin Cancer," in *13th IEEE Integrated STEM Education Conference, ISEC 2023*, Institute of Electrical and Electronics Engineers Inc., Mar. 2023, pp. 375–380. doi: 10.1109/ISEC57711.2023.10402205.
- [9] R. Kumar, "Fake News Detection using Passive Aggressive and TF-IDF Vectorizer," *International Research Journal of Engineering and Technology*, pp. 902–904, Dec. 2020, [Online]. Available: <https://www.irjet.net/archives/V7/i9/IRJET->
- [10] K. Shridhar *et al.*, "Subword Semantic Hashing for Intent Classification on Small Datasets," Sep. 2019, doi: 10.1109/IJCNN.2019.8852420.
- [11] M. Ostendorff, P. Bourgonje, M. Berger, J. Moreno-Schneider, G. Rehm, and B. Gipp, "Enriching BERT with Knowledge Graph Embeddings for Document Classification," Sep. 2019, [Online]. Available: <http://arxiv.org/abs/1909.08402>
- [12] S. A. Sazan, M. H. Miraz, and A. B. M. Muntasir Rahman, "Enhancing Depressive Post Detection in Bangla: A Comparative Study of TF-IDF, BERT and FastText Embeddings," *Annals of Emerging Technologies in Computing*, vol. 8, no. 3, pp. 34–49, 2024, doi: 10.33166/AETiC.2024.03.003.
- [13] N. A. Pramudhyta and M. S. Rohman, "Perbandingan Optimasi Metode Grid Search dan Random Search dalam Algoritma XGBoost untuk Klasifikasi Stunting," *JURNAL MEDIA INFORMATIKA BUDIDARMA*, vol. 8, no. 1, pp. 19–29, Jan. 2024, doi: 10.30865/mib.v8i1.6965.
- [14] D. Satriani, L. U. Khasanah, and N. A. Rizki, "Penerapan Metode Grid-Search dalam Menentukan Parameter Model Pertumbuhan Penduduk di Kota Samarinda," *Prosiding Seminar Nasional Matematika dan Statistika*, vol. 1, pp. 65–74, 2019. [Online]. Available: <https://jurnal.fmipa.unmul.ac.id/index.php/SNMSA/article/view/528>
- [15] M. A. Rosid, A. S. Fitriani, I. R. I. Astutik, N. I. Mulloh, and H. A. Gozali, "Improving Text Preprocessing for Student Complaint Document Classification Using Sastrawi," in *IOP Conference Series: Materials Science and Engineering*, Institute of Physics Publishing, Jul. 2020, pp. 1–6. doi: 10.1088/1757-899X/874/1/012017.



- [16] A. Tabassum and R. R. Patil, "A Survey on Text Pre-Processing & Feature Extraction Techniques in Natural Language Processing," *International Research Journal of Engineering and Technology*, vol. 7, pp. 4864–4867, Jun. 2020, [Online]. Available: <https://irjet.net/archives/V7/i6/IRJET-V7I6913.pdf>
- [17] J. Andre Septian, T. Maulana Fahrudin, and A. Nugroho, "Analisis Sentimen Pengguna Twitter Terhadap Polemik Persepakbolaan Indonesia Menggunakan Pembobotan TF-IDF dan K-Nearest Neighbor," *JOURNAL OF INTELLIGENT SYSTEMS AND COMPUTATION*, pp. 43–49, Aug. 2019, [Online]. Available: <https://t.co/9Wl0aWpfD5>
- [18] N. Badri, F. Kboubi, and A. H. Chaibi, "Combining FastText and Glove Word Embedding for Offensive and Hate speech Text Detection," in *Procedia Computer Science*, Elsevier B.V., 2022, pp. 769–778. doi: 10.1016/j.procs.2022.09.132.
- [19] A. Amalia, O. S. Sitompul, E. B. Nababan, and T. Mantoro, "An Efficient Text Classification Using fastText for Bahasa Indonesia Documents Classification," in 2020 International Conference on Data Science, Artificial Intelligence, and Business Analytics (DATABIA), IEEE, Jul. 2020, pp. 69–75. doi: 10.1109/DATABIA50434.2020.9190447.
- [20] A. R. Hanum *et al.*, "Analisis Kinerja Algoritma Klasifikasi Teks BERT Dalam Mendeteksi Berita Hoaks," *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIK)*, vol. 11, no. 3, pp. 537–546, Jun. 2024, doi: 10.25126/jtiik938093.
- [21] F. Koto, J. H. Lau, and T. Baldwin, "IndoBERTweet: A Pretrained Language Model for Indonesian Twitter with Effective Domain-Specific Vocabulary Initialization," p. 1, Sep. 2021, [Online]. Available: <http://arxiv.org/abs/2109.04607>
- [22] K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer†, "Online Passive-Aggressive Algorithms," *Journal of Machine Learning Research*, vol. 7, pp. 551–585, Jun. 2006. [Online]. Available: <https://www.jmlr.org/papers/volume7/crammer06a/crammer06a.pdf>
- [23] J. Wang and S. Zhang, "PA-PseU: An incremental passive-aggressive based method for identifying RNA pseudouridine sites via Chou's 5-steps rule," *Chemometrics and Intelligent Laboratory Systems*, vol. 210, pp. 1–12, Mar. 2021, doi: 10.1016/j.chemolab.2021.104250.
- [24] N. Ahmed and M. Rawat, "Identification of Fake News using Machine Learning and Deep Learning," in 2023 *International Conference on IoT, Communication and Automation Technology (ICICAT)*, IEEE, Jun. 2023, pp. 1–5. doi: 10.1109/ICICAT57735.2023.10263681.
- [25] C. Huang, Y. Li, and X. Yao, "A Survey of Automatic Parameter Tuning Methods for Metaheuristics," Apr. 01, 2020, *Institute of Electrical and Electronics Engineers Inc.* doi: 10.1109/TEVC.2019.2921598.
- [26] M. Fajri and A. Primajaya, "Komparasi Teknik Hyperparameter Optimization pada SVM untuk Permasalahan Klasifikasi dengan Menggunakan Grid Search dan Random Search," *Journal of Applied Informatics and Computing (JAIC)*, vol. 7, no. 1, pp. 10–15, Jun. 2023
- [27] P. Liashchynskiy and P. Liashchynskiy, "Grid Search, Random Search, Genetic Algorithm: A Big Comparison for NAS," pp. 1–11, Dec. 2019, [Online]. Available: <http://arxiv.org/abs/1912.06059>
- [28] P. Yohana, S. Agustian, S.K. Gusti, "Klasifikasi Sentimen Masyarakat terhadap Kebijakan Vaksin Covid-19 pada Twitter dengan Imbalance Classes Menggunakan Naive Bayes", *Seminar Nasional Teknologi Informasi Komunikasi dan Industri 14*, 2022. [Online]. Available: <https://ejournal.uin-suska.ac.id/index.php/SNTIKI/article/view/19012/8336>
- [29] 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 (EMNLP 2021)*, Sep. 2021. [Online]. Available: <https://doi.org/10.48550/arXiv.2109.04607>
- [30] J. Pranata, S. Agustian, J. Jasril, E. Haerani, "Penggunaan Model Bahasa indoBERT pada metode Random Forest untuk Klasifikasi Sentimen dengan Dataset Terbatas," to be appear in *Building of Informatics, Technology and Science (BITS)*, Vol 6 No 3 (2024): December 2024, doi: <https://doi.org/10.47065/bits.v6i3.6335>