Jurnal Sistem Komputer dan Informatika (JSON)
Volume 7, Nomor 4, Juni 2026
e-ISSN 2685-998X
DOI 10.30865/json.v7i4.9774

Perbandingan Kinerja dan Efisiensi Model NLP pada Analisis Sentimen Ulasan Aplikasi Layanan Publik Digital

Irfan Setiawan*, Widyastuti Andriyani
Teknologi Informasi, Magister Teknologi Informasi, Universitas Teknologi Digital Indonesia, Yogyakarta, Indonesia
Email: [1,*]irfnse@gmail.com, [2]widya@utdi.ac.id
Email Penulis Korespondensi: irfnse@gmail.com*
Submitted: **04/05/2026**; Accepted: **17/06/2026**; Published: **30/06/2026**

Abstrak–Transformasi digital layanan publik di Indonesia menghasilkan volume ulasan pengguna yang besar pada Google Play Store, khususnya untuk aplikasi layanan publik digital seperti Identitas Kependudukan Digital (IKD), BPJS Kesehatan Mobile, dan MyPertamina. Penelitian ini bertujuan untuk membandingkan kinerja dan efisiensi komputasi lima model dari tiga generasi pendekatan *Natural Language Processing* (NLP), yaitu *Naive Bayes*, *Support Vector Machine* (SVM), *Bidirectional Long Short-Term Memory* (BiLSTM), IndoBERT, dan IndoBERTweet dalam tugas analisis sentimen ulasan berbahasa Indonesia. Evaluasi dilakukan pada performa klasifikasi menggunakan *accuracy*, *macro precision*, *macro recall*, dan *macro F1-Score* serta efisiensi komputasi melalui waktu pelatihan, waktu inferensi, dan penggunaan memori. Dataset dikumpulkan melalui *scraping* ulasan Google Play Store dengan strategi pelabelan otomatis berbasis rating bintang (*weak labels*), yang keandalannya divalidasi dengan *subset* sampel menggunakan Cohen's Kappa. Penggunaan label lemah ini merupakan keterbatasan yang perlu dipertimbangkan dalam interpretasi hasil, mengingat hanya sebagian kecil data yang divalidasi secara manual. Penelitian ini mengisi gap literatur pada domain aplikasi layanan publik digital Indonesia yang masih kurang dieksplorasi dalam konteks komparasi model NLP lintas generasi, sekaligus menghasilkan implikasi praktis pemilihan model berdasarkan kondisi infrastruktur komputasi yang tersedia, dengan mempertimbangkan keterbatasan generalisasi hasil pada domain dan skala dataset yang berbeda. Hasil eksperimen menunjukkan IndoBERTweet mencapai performa tertinggi dengan *macro F1-Score* 0,8957, sementara *Naive Bayes* dan SVM dapat dijalankan dalam waktu di bawah 0,15 detik tanpa GPU dengan *macro F1-Score* masing-masing 0,8345 dan 0,8402.

Kata Kunci: Analisis Sentimen; Aplikasi Layanan Publik Digital; IndoBERT; IndoBERTweet; Machine Learning; Natural Language Processing; Studi Komparatif

Abstract–The digital transformation of public services in Indonesia has generated a large volume of user reviews on the Google Play Store, particularly for digital public service applications such as the Identitas Kependudukan Digital (IKD), BPJS Kesehatan Mobile, and MyPertamina. This study aims to compare the performance and computational efficiency of five models from three generations of Natural Language Processing (NLP) approaches, namely Naive Bayes, Support Vector Machine (SVM), Bidirectional Long Short-Term Memory (BiLSTM), IndoBERT, and IndoBERTweet in sentiment analysis tasks on Indonesian-language reviews. Evaluation was conducted on classification performance using accuracy, macro precision, macro recall, and macro F1-Score, as well as computational efficiency through training time, inference time, and memory usage. The dataset was collected by scraping Google Play Store reviews using an automatic star-rating-based labeling strategy (*weak labels*), whose reliability was validated on a sample subset using Cohen's Kappa. The use of weak labels constitutes a limitation to be considered when interpreting the results, given that only a small portion of the data was manually validated. This study addresses a gap in the literature within the domain of Indonesian digital public service applications, which remains underexplored in the context of NLP model comparison, while deriving practical implication for model selection based on available computational infrastructure. Experiment results show that IndoBERTweet achieves the highest performance with macro F1-Score of 0,8957, while Naive Bayes and SVM complete running in under 0.15 seconds without GPU, achieving macro F1-Score of 0,8345 and 0,8402 respectively.

Keywords: Sentiment Analysis; Digital Public Service Application; IndoBERT; IndoBERTweet; Machine Learning; Natural Language Processing; Comparative Study

1. PENDAHULUAN

Transformasi digital pada sektor layanan publik di Indonesia telah mengalami percepatan yang signifikan, khususnya sejak pemerintah mendorong implementasi program Sistem Pemerintahan Berbasis Elektronik (SPBE) melalui Perpres No. 95 Tahun 2018 [1]. Berbagai instansi pemerintah dan Badan Usaha Milik Negara (BUMN) yang mendapat mandat pelayanan publik telah meluncurkan aplikasi *mobile* resmi yang dapat diakses masyarakat melalui *platform* Google Play Store. Di antaranya adalah aplikasi Identitas Kependudukan Digital (IKD) dari Dirjen Dukcapil Kemendagri, aplikasi BPJS Kesehatan Mobile sebagai portal layanan jaminan kesehatan nasional, serta aplikasi MyPertamina milik PT Pertamina (Persero) yang ditugaskan pemerintah mendistribusikan subsidi bahan bakar minyak berdasarkan Kepmen ESDM No. 37.K/HK.02/MEM.M/2022 [2]. Ketiganya menjadi bagian dari ekosistem layanan publik digital sebagaimana didefinisikan dalam Undang-Undang No. 25 Tahun 2009 tentang Pelayanan Publik [3] yang mengakui BUMN sebagai penyelenggara layanan publik.

Pemilihan ketiga aplikasi tersebut didasarkan pada tiga pertimbangan. Pertama, ketiganya mewakili ragam penyelenggara layanan publik digital, yaitu instansi pemerintah pusat (IKD oleh Ditjen Dukcapil), badan penyelenggara jaminan sosial (BPJS Kesehatan), dan BUMN penerima mandat layanan publik (Pertamina). Kedua, ketiganya menyentuh kebutuhan dasar warga yang berbeda, yakni identitas kependudukan, jaminan kesehatan, dan akses energi bersubsidi, sehingga karakteristik keluhan maupun apresiasi penggunanya beragam.

Jurnal Sistem Komputer dan Informatika (JSON)
Volume 7, Nomor 4, Juni 2026
e-ISSN 2685-998X
DOI 10.30865/json.v7i4.9774

Ketiga, ketiganya memiliki basis pengguna berskala nasional dengan volume ulasan yang besar di Google Play Store, sehingga memadai untuk pelatihan dan evaluasi model. Keragaman ini diharapkan menghasilkan dataset yang lebih representatif terhadap karakteristik teks ulasan layanan publik digital Indonesia dibandingkan jika hanya menggunakan satu aplikasi tunggal.

Ulasan pengguna yang terkumpul dalam jumlah besar pada ketiga aplikasi tersebut memiliki potensi besar sebagai sumber umpan balik untuk meningkatkan kualitas layanan publik digital. Namun, menganalisis ulasan dalam skala ribuan hingga jutaan secara manual jelas tidak efisien dan tidak praktis, sehingga diperlukan pendekatan otomatis melalui analisis sentimen berbasis NLP yang mampu mengklasifikasikan opini pengguna ke dalam kategori sentimen positif maupun negatif secara sistematis [4], [5]. Permasalahan yang kemudian muncul adalah belum adanya panduan berbasis bukti empiris mengenai model NLP mana yang paling sesuai untuk menganalisis sentimen ulasan aplikasi publik digital Indonesia, baik dari sisi performa klasifikasi maupun efisiensi penggunaan sumber daya komputasi. Untuk menjawab permasalahan tersebut, penelitian ini mengusulkan studi komparatif yang mengevaluasi lima model dari tiga generasi pendekatan NLP, tidak hanya dari sisi performa klasifikasi tetapi juga efisiensi komputasi secara sistematis.

Penelitian di bidang analisis sentimen terus berkembang seiring munculnya berbagai pendekatan model. Pendekatan awal didominasi oleh model *Machine Learning* klasik seperti *Naive Bayes* dan *Support Vector Machine* (SVM) yang memanfaatkan representasi fitur berbasis TF-IDF [6] maupun *Word2Vec* [7]. Seiring berkembangnya teknologi, pendekatan *Deep Learning* seperti *Bidirectional Long Short-Term Memory* (BiLSTM) mulai banyak digunakan karena kemampuannya menangkap konteks dari dua arah secara bersamaan, sehingga lebih efektif dalam memahami dependensi jangka panjang dalam teks ulasan [8], [9]. Perkembangan terkini ditandai oleh model berbasis *Transformer* seperti BERT. Untuk bahasa Indonesia, tersedia IndoBERT yang dilatih pada korpus Indonesia berskala besar [10], [11], serta IndoBERTweet yang secara khusus dilatih pada 26 juta cuitan Twitter Indonesia untuk menangani teks informal dan singkatan khas media sosial [12].

Berbagai penelitian sebelumnya telah membandingkan model-model tersebut pada beragam domain. Ashbaugh dan Zhang [4] membandingkan *machine learning* dan *deep learning* pada ulasan pelanggan dan menemukan bahwa *Random Forest* dan *Logistic Regression* mencapai akurasi 0,99 dengan *macro F1-Score* 0,99. Kusuma et al. [13] menerapkan *Naive Bayes* pada ulasan aplikasi DANA dan mendapatkan akurasi 74,60%. Andriyani et al. [7] mengintegrasikan SVM dengan *Word2Vec* untuk ulasan produk Amazon dengan akurasi 91,3%. Aulia et al. [14] menerapkan SVM dengan kernel *linear* pada ulasan aplikasi Get Contact Bahasa Indonesia dan memperoleh akurasi 95,50%, menunjukkan efektivitas SVM untuk klasifikasi sentimen pada teks ulasan aplikasi *mobile* Bahasa Indonesia. Rahman Isnain et al. [15] membandingkan LSTM dan *Naive Bayes* pada data Twitter terkait kebijakan *new normal*, di mana LSTM unggul dengan nilai *F1-Score* 83,33% dibandingkan *Naive Bayes* yang hanya memperoleh akurasi 82%. Mutmainah et al. [9] menerapkan BiLSTM pada ulasan penggunaan aplikasi *telemedicine* di Google Play Store Bahasa Indonesia menunjukkan efektivitas BiLSTM dalam menganalisis sentimen teks dengan Bahasa Indonesia pada domain aplikasi *mobile*. Halim et al. [10] membandingkan model *machine learning*, *deep learning*, dan IndoBERT untuk mendeteksi komentar spam Instagram, di mana *indobert-large-p1* menjadi model terbaik dengan nilai 0,85 pada semua metrik evaluasi. Selain itu Nugroho et al. [11] menunjukkan bahwa *fine-tuning* IndoBERT pada ulasan aplikasi *mobile* Bahasa Indonesia menghasilkan akurasi lebih tinggi dibandingkan dengan model BERT multibahasa. Selanjutnya Jayadianti et al. [16] mengonfirmasi efektivitas IndoBERT *fine-tuning* pada dataset Bahasa Indonesia dengan skor akurasi mencapai 95,16% dengan menggabungkan arsitektur IndoBERT dan *Recurrent Convolutional Neural Network* (RCNN).

Dari kajian penelitian-penelitian tersebut, dapat diamati beberapa keterbatasan. Pertama, sebagian studi hanya membandingkan model dalam satu generasi yang sama, seperti LSTM *vs Naive Bayes* [15], sehingga tidak dapat menjawab pada generasi mana peningkatan performa benar terjadi. Kedua, sebagian akurasi tinggi dicapai pada konteks yang berbeda dari domain penelitian ini, baik dari sisi bahasa maupun domain. Ulasan produk komersial berbahasa Inggris [4], maupun domain bahasa Indonesia di luar layanan publik seperti deteksi spam Instagram [10] dan ulasan aplikasi *mobile* [11], [16], sehingga temuannya belum tentu dapat digeneralisasi ke domain layanan publik digital yang teksnya cenderung informal, emosional, dan didominasi keluhan. Ketiga, evaluasi model sebagian besar hanya mencakup metrik performa klasifikasi seperti akurasi dan *F1-Score* [4], [9], [10], [11], [16], sementara aspek efisiensi komputasi seperti waktu pelatihan, waktu inferensi, dan penggunaan memori jarang dilaporkan secara sistematis. Kondisi yang sama juga ditemukan pada penelitian yang secara khusus menggunakan domain aplikasi layanan publik digital. Maulana et al. [17] yang meneliti ulasan aplikasi MyPertamina dan Tanggraeni & Sitokdana [18] yang meneliti aplikasi *e-government* Sentuh Tanahku, serta Tarwoto et al. [19] yang menganalisis ulasan aplikasi Mobile JKN, semuanya hanya menggunakan satu model klasifikasi tunggal tanpa melakukan perbandingan lintas generasi model maupun mengukur efisiensi komputasi. Studi komprehensif oleh Dhendra dan Utomo [20] membandingkan lima model (IndoBERT, mBERT, XLM-R, CNN, dan BiLSTM) pada ulasan layanan publik NEWSAKPOLE dengan pelabelan *hybrid* berbasis rating dengan validasi manual, namun studi tersebut terbatas pada performa klasifikasi, tidak menyertakan model *Classical* ML, serta tanpa menyertakan analisis efisiensi komputasi. Hal ini menunjukkan bahwa penelitian komparatif yang menyeluruh pada domain aplikasi layanan publik digital Indonesia, yang mencakup model dari *Classical* ML

Jurnal Sistem Komputer dan Informatika (JSON)
Volume 7, Nomor 4, Juni 2026
e-ISSN 2685-998X
DOI 10.30865/json.v7i4.9774

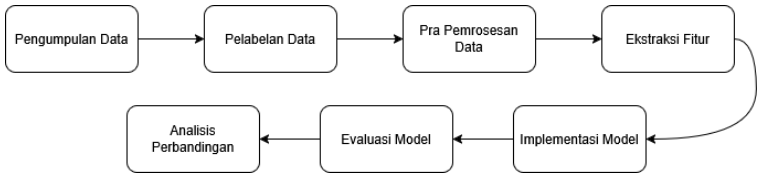
hingga *Transformer* sekaligus mengukur efisiensi komputasinya, sejauh literatur yang dilakukan dalam penelitian ini, belum ditemukan.

Berdasarkan gap tersebut, penelitian ini bertujuan untuk membandingkan kinerja klasifikasi dan efisiensi komputasi dari model *Naive Bayes*, SVM, BiLSTM, IndoBERT, dan IndoBERTweet dalam tugas analisis sentimen ulasan pengguna tiga aplikasi layanan publik digital Indonesia di Google Play Store, serta menghasilkan implikasi praktis pemilihan model berdasarkan kondisi infrastruktur komputasi yang tersedia. Implikasi tersebut bersifat kontekstual terhadap kondisi eksperimen dan memerlukan validasi lebih lanjut sebelum diterapkan pada domain atau skala dataset yang berbeda. Penelitian ini diharapkan dapat memberikan kontribusi teoritis berupa studi komparatif lintas generasi model NLP pada domain yang berdasarkan tinjauan literatur belum banyak dieksplorasi.

2. METODOLOGI PENELITIAN

2.1 Tahapan Penelitian

Penelitian ini dilakukan melalui tujuh tahapan mulai dari pengumpulan data, pelabelan data, pra-pemrosesan data, ekstraksi fitur, implementasi model, evaluasi model, dan analisis perbandingan sesuai pada Gambar 1.



Gambar 1. Tahapan Penelitian

2.2 Pengumpulan Data

Pada tahap pertama data dikumpulkan dengan cara *scraping* ulasan Google Play Store pada tiga aplikasi layanan publik digital Indonesia dengan menggunakan *library* google-play-scraper berbasis Python, dengan target dataset 9.000 ulasan atau 3.000 ulasan pada masing-masing aplikasi. Pengambilan data dilakukan dengan parameter sort=Sort.NEWEST dengan mekanisme paginasi *continuation_token*. Ulasan yang terkumpul mencakup rentang publikasi dari 11 November 2025 hingga 20 April 2026. Seluruh ulasan akan digabung menjadi satu dataset sebelum dilakukan pembagian data dengan *stratified split* dengan rasio 80:20 untuk data latih dan data uji. Pembagian ini menggunakan *train_test_split* scikit-learn dengan *stratify=y* dan *random_state=42* dan digunakan secara identik untuk seluruh model agar setiap model dilatih dan diuji pada data yang sama. Khusus untuk model *transformer*, sebesar 10% dari data latih disisihkan sebagai *validation set* guna keperluan *early stopping*.

2.3 Pelabelan Data

Pelabelan data pada penelitian ini menggunakan pendekatan *weak supervision*, yakni menggunakan rating sebagai label tanpa anotasi manual. Strategi yang digunakan bersifat *hybrid* dengan basis *rating* bintang sebagai pelabelan awal. Bintang 4-5 sebagai sentimen Positif dan bintang 1-2 sebagai sentimen Negatif. Ulasan dengan bintang 3 tidak diikutsertakan untuk menghindari ambiguitas. Karena label yang dihasilkan bersifat lemah dan berpotensi mengandung noise, label perlu divalidasi. Validasi dilakukan secara manual pada 300 ulasan (100 ulasan per aplikasi) oleh dua anotator independen. Proses anotasi dilakukan dengan *blind annotation*. Anotator hanya menerima nomor dan teks ulasan tanpa informasi rating bintang, nama aplikasi, maupun label hasil rating. Urutan sampel diacak agar distribusi aplikasi tidak dapat dikenali anotator. Anotator diberi instruksi untuk melabeli ulasan sebagai positif jika mengekspresikan kepuasan atau pengalaman baik dan negatif jika mengekspresikan keluhan atau pengalaman buruk. Kemudian menggunakan Cohen's Kappa (κ) dengan target $\kappa \geq 0{,}6$ untuk mencapai tingkat *substantial agreement* [21]. Formula Cohen's Kappa didefinisikan sebagai berikut:

$$K = \frac{P_o - P_e}{1 - P_e} \quad (1)$$

P_o adalah proporsi kesepakatan yang teramati antara kedua anotator terhadap total seluruh observasi, sedangkan P_e adalah proporsi kesepakatan yang diharapkan secara acak. Nilai P_e diestimasi melalui kalkulasi probabilitas marginal dari preferensi masing-masing anotator terhadap suatu kelas label.

2.4 Pra-pemrosesan Data

Pra-pemrosesan data adalah tahapan untuk membersihkan dan menyederhanakan data mentah agar model lebih fokus dalam mempelajari pola kata yang membentuk sentimen positif dan negatif. Tahap ini meliputi: (1) *case folding* untuk mengubah teks menjadi huruf kecil; (2) pembersihan teks dari angka, simbol, dan karakter khusus; (3) deduplikasi dengan menggunakan df.drop_duplicates() (4) normalisasi kata tidak baku menggunakan Kamus

Jurnal Sistem Komputer dan Informatika (JSON)
Volume 7, Nomor 4, Juni 2026
e-ISSN 2685-998X
DOI 10.30865/json.v7i4.9774

Alay [22]; (5) tokenisasi memecah kalimat menjadi potongan kata tunggal; (6) *stopword removal* menghilangkan kata hubung atau kata tugas yang tidak memiliki makna dengan *library* sastrawi; (7) *stemming* mengubah kata berimbuhan kembali ke kata dasar. Khusus model IndoBERT dan IndoBERTweet, tahapan *stemming* tidak diterapkan karena tokenizer BERT mempunyai mekanisme pemrosesan tersendiri [12].

2.5 Ekstraksi Fitur

Tahap ekstraksi fitur merupakan proses mengubah teks menjadi representasi vektor (baik semantik maupun numerik) agar siap untuk dilakukan klasifikasi oleh model. Pada model *Naive Bayes* dan SVM ekstraksi fitur menggunakan metode TF-IDF dengan *n-gram range* (1,2) dan *sublinear_tf = True* dengan jumlah fitur maksimal ditentukan melalui *hyperparameter tuning*. TF-IDF mengukur bobot kepentingan sebuah kata berdasarkan frekuensi kemunculannya dalam sebuah dokumen relatif terhadap keseluruhan korpus [23], dengan rumusan sebagai berikut:

$$TF - IDF(t,d) = TF(t,d) \times log\frac{N}{df(t)} \quad (2)$$

TF(t,d) adalah frekuensi kemunculan *term* (t) dalam dokumen (*d*), *N* adalah jumlah total dokumen, dan *df(t)* adalah jumlah dokumen yang mengandung *term* (t). Penggunaan *sublinear_tf = True* diterapkan untuk menskalakan nilai mentah (*TF*) dengan 1+*log(TF)*. Pendekatan ini guna menekan dominasi *term* yang sering muncul dalam suatu dokumen.

Representasi teks pada arsitektur BiLSTM memanfaatkan *pretrained word embedding FastText* bahasa Indonesia (cc.id.300.bin, 300 dimensi). Setiap teks ulasan diseragamkan panjangnya dengan metode *post-padding* hingga 128 token. Penggunaan *FastText* dipilih karena kemampuannya dalam menangani *Out-of-Vocabulary* (OOV) melalui mekanisme *subword n-grams*, sehingga model tetap dapat menangani kosa kata yang tidak baku atau salah ketik [24].

Pada model IndoBERT dan IndoBERTweet ekstraksi fitur dilakukan dengan menggunakan *tokenizer* bawaan berbasis metode *WordPiece*. Panjang sekuens input dibatasi maksimal 256 token untuk menjaga efisiensi komputasi, sedangkan klasifikasi sentimen menggunakan representasi token [CLS] pada lapisan terakhir [10], [12], [23].

2.6 Implementasi Model

Lima model dari tiga generasi pendekatan NLP diimplementasikan dan dibandingkan sebagaimana ditunjukkan pada Tabel 1. Model *Naive Bayes* dan SVM bekerja atas representasi TF-IDF sebagaimana dijelaskan pada bagian 2.5, sehingga tidak memerlukan rancangan arsitektur jaringan khusus. Arsitektur BiLSTM disusun atas lapisan *Embedding* dengan bobot *FastText* (cc.id.300.bin, 300 dimensi) yang dibekukan (*non-trainable*), diikuti satu lapisan *Bidirectional LSTM*, lapisan *Dropout*, lapisan *Dense 64* unit dengan aktivasi ReLU, lapisan *Dropout* kedua, dan lapisan *output Dense* satu unit dengan aktivasi *sigmoid*. Model dikompilasi menggunakan *optimizer* Adam dengan *loss function binary cross-entropy*, dan *restore_best_weights* untuk mengembalikan bobot terbaik. Jumlah unit, laju *dropout*, *learning rate*, dan *batch size* tidak ditetapkan di awal melainkan dicari melalui *Grid Search* pada rentang yang disajikan di Tabel 2. Model IndoBERT dan IndoBERTweet di-*fine-tune* menggunakan optimizer AdamW dengan *mixed-precision training* (fp16) untuk efisiensi memori GPU. Panjang sekuens dibatasi 256 token, dan model terbaik dipilih berdasarkan *macro F1-Score* pada *validation set (load_best_model_at_end)*. *Learning rate* dan *batch size* optimal diperoleh melalui pencarian pada *validation set* sebagaimana disajikan pada Tabel 2. Seluruh eksperimen dilakukan pada Google Colaboratory dengan GPU NVIDIA Tesla T4. *Hyperparameter tuning* untuk *Classical Machine Learning* dan BiLSTM menggunakan *Grid Search* dengan *5-fold-cross-validation*, sementara model *Transformer* menggunakan pencarian manual dengan *validation set* dan *early stopping* (*patience = 3*). Detail rentang *hyperparameter* yang digunakan disajikan pada Tabel 2.

Konsistensi eksperimen dijaga dengan menetapkan *random_seed* bernilai 42 secara seragam di seluruh *library* yang digunakan (Python, NumPy, TensorFlow, dan PyTorch), serta konfigurasi *StratifiedKFold* yang identik (5-*fold*, *shuffle*, *random_state=42*) pada setiap proses *Grid Search*. Dengan kondisi eksperimen yang seragam, setiap model diuji pada situasi yang sama.

Tabel 1. Konfigurasi Model

Generasi	Model	Representasi Fitur	Library
Classical ML	*Naive Bayes*	TF-IDF, n-gram(1,2)	Scikit-learn
Classical ML	SVM (kernel *linear*)	TF-IDF, n-gram(1,2)	Scikit-learn
Deep Learning	BiLSTM	FastText *pretrained* cc.id.300.bin	TensorFlow/Keras
Transformer	IndoBERT	*Indobert-base-p1*	HuggingFace Transformers
Transformer	IndoBERTweet	*Indobertweet-base-uncased*	HuggingFace Transformers

Jurnal Sistem Komputer dan Informatika (JSON)
Volume 7, Nomor 4, Juni 2026
e-ISSN 2685-998X
DOI 10.30865/json.v7i4.9774

Tabel 2. Rentang *Hyperparameter*

Model	Hyperparameter	Rentang Nilai	Metode
Naive Bayes	*alpha*	{0,01; 0,1; 0,5; 1,0; 2,0}	*Grid Search 5-Fold*
	max_features	{10K; 20K; 50K}	
SVM	*C*	{0,01; 0,1; 1; 10; 100}	*Grid Search 5-Fold*
	max_features	{10K; 20K; 50K}	
BiLSTM	*units*	{64; 128; 256}	*Grid Search 5-Fold*
	dropout	{0,2; 0,3; 0,5}	
	learning rate	{1e-3; 5e-4; 1e-4}	
	batch	{32; 64; 128}	
IndoBERT / IndoBERTweet	*learning rate*	{1e-5; 2e-5; 3e-5; 5e-5}	Manual + Val Set
	batch size	{16; 32}	
	max epoch	5 (patience=3)	*Early Stopping*

2.7 Evaluasi Model

Tahap evaluasi model dilakukan pada dua aspek yaitu performa dan efisiensi model. Pendekatan ganda ini diharapkan model yang direkomendasikan tidak hanya akurat dalam klasifikasi sentimen, namun juga optimal dalam efisiensi penggunaan sumber daya komputasi.

2.7.1 Performa Klasifikasi

Performa klasifikasi diukur menggunakan *accuracy*, *macro precision*, *macro recall*, dan *macro F1-Score* berserta *confusion matrix* pada tiap model. *Confusion matrix* adalah tabel yang merangkum nilai aktual dan nilai prediksi. Pada klasifikasi biner (positif dan negatif), *confusion matrix* memiliki empat komponen: *True Positive* (TP), yaitu ulasan positif yang diprediksi benar sebagai positif; *True Negative* (TN), yaitu ulasan negatif yang diprediksi benar sebagai negatif; *False Positive* (FP), yaitu ulasan negatif yang salah diprediksi sebagai positif; *False Negative* (FN), yaitu ulasan positif yang salah diprediksi sebagai negatif. Struktur *confusion matrix* untuk klasifikasi biner ditunjukkan pada Tabel 3.

Tabel 3. *Confusion matrix*

	Prediksi Positif	Prediksi Negatif
Aktual Positif	*True Positive* (TP)	*False Negative* (FN)
Aktual Negatif	*False Positive* (FP)	*True Negative* (TN)

Berdasarkan keempat komponen tersebut, *precision* mengukur ketepatan prediksi positif, *recall* mengukur proporsi sampel positif yang berhasil dideteksi, *F1-Score* menggabungkan keduanya melalui rata-rata harmonik, dan *accuracy* mengukur proporsi keseluruhan prediksi yang benar [25], sebagaimana dirumuskan pada persamaan (3)-(6).

$$Precision = \frac{TP}{TP + FP} \tag{3}$$

$$Recall = \frac{TP}{TP + FN} \tag{4}$$

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{5}$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{6}$$

Karena distribusi kelas tidak seimbang, *precision*, *recall*, dan *F1-Score* dihitung menggunakan *macro-averaging*, yaitu rata-rata nilai metrik pada setiap kelas dengan bobot yang sama sebagaimana ditunjukkan pada persamaan (7). Di mana nilai *N* adalah jumlah kelas dan $Metric_i$ adalah nilai metrik pada kelas ke-*i*.

$$Macro\ Metric = \frac{1}{N}\sum Metric_i \tag{7}$$

Kemudian untuk menguji signifikansi performa antar model, digunakan McNemar's test [26], [27] uji statistik non-parametrik yang direkomendasikan untuk perbandingan model klasifikasi pada satu dataset uji. Pengujian dilakukan pada empat kombinasi model yang berdekatan secara generasi dengan tingkat signifikansi α = 0,05. Statistik dihitung dari tabel kontingensi 2×2 yang merangkum jumlah sampel di mana kedua model

Jurnal Sistem Komputer dan Informatika (JSON)
Volume 7, Nomor 4, Juni 2026
e-ISSN 2685-998X
DOI 10.30865/json.v7i4.9774

menghasilkan prediksi yang berbeda. b menunjukkan Model A benar dan model B salah dan c menunjukkan Model A salah dan Model B benar. Tingkat signifikansi hasil pengujian dilambangkan sebagai berikut: * $p < 0,05$; ** $p < 0,01$; *** $p < 0,001$.

Selain evaluasi kuantitatif, dilakukan analisis *error* kualitatif untuk memahami pola kesalahan klasifikasi antar generasi model. Identifikasi kasus dilakukan secara programatik dengan membandingkan vektor prediksi kelima model pada seluruh data uji, kasus di mana model ML klasik salah namun IndoBERTweet benar, dan sebaliknya. Kemudian diekstrak dan dianalisis secara manual untuk mengidentifikasi pola yang representatif.

2.7.2 Efisiensi Komputasi

Efisiensi komputasi diukur melalui empat indikator: waktu pelatihan, waktu inferensi per 1000 sampel, penggunaan memori RAM, dan penggunaan memori GPU. Agar pengukuran konsisten antar *framework*, seluruh RAM diukur menggunakan *Resident Set Size* (RSS) pada *level* proses dengan modul psutil. Cara ini membuat hasilnya setara, meskipun model dibangun pada *framework* yang berbeda.

Pengukuran diseragamkan dengan tiga cara: (1) jumlah sampel inferensi disamakan menjadi 1.000 sampel untuk semua model; (2) setiap model diukur pada sesi *runtime* terpisah agar memori tidak terakumulasi antar model; (3) setiap pengukuran diawali dengan satu *warm-up run* untuk menghilangkan pengaruh inisialisasi. Nilai RAM yang dilaporkan adalah selisih RSS setelah model dimuat terhadap kondisi awal proses, sehingga mencerminkan kebutuhan memori tiap model. Waktu pelatihan diukur dengan modul *time*, waktu inferensi diambil sebagai rata-rata dari lima percobaan, dan penggunaan memori GPU diukur dengan menggunakan modul *torch.cuda.max_memory_allocated()* untuk model PyTorch.

Pemilihan keempat indikator efisiensi ini dipilih dengan pendekatan yang digunakan pada studi komparatif efisiensi model serupa [4], [28] dan didasarkan pada relevansinya terhadap skenario *deployment* pada sistem layanan publik. Waktu pelatihan menggambarkan biaya komputasi yang dibutuhkan untuk membangun model, sedangkan waktu inferensi menentukan kecepatan respon sistem saat melayani respon pengguna. Penggunaan RAM dan GPU menentukan kelayakan *deployment* pada infrastruktur yang tersedia [29].

3. HASIL DAN PEMBAHASAN

3.1 Distribusi Dataset

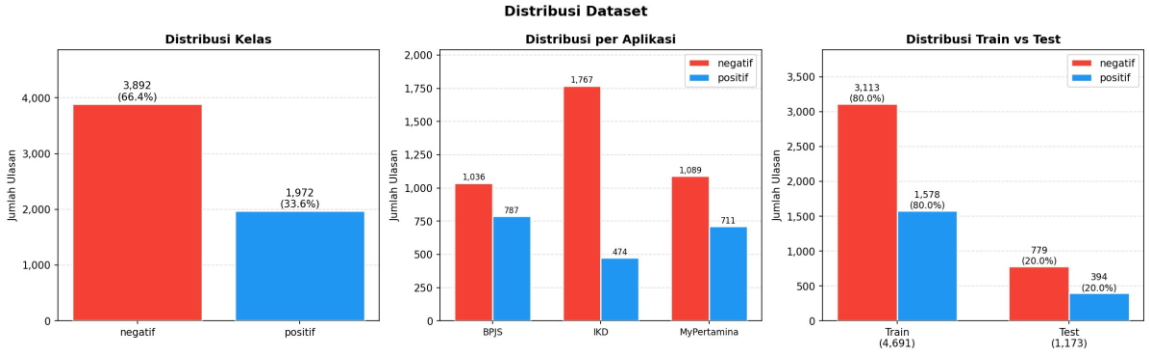
Proses pengumpulan data melalui *scraping* Google Play Store menghasilkan total 5.864 ulasan valid berbahasa Indonesia setelah *filtering* yang mencakup penghapusan konten kosong, ulasan dengan kurang dari tiga kata, serta ulasan bintang 3 yang dikeluarkan untuk menghindari ambiguitas label. Distribusi dataset per aplikasi setelah proses pelabelan dan validasi disajikan pada Tabel 4.

Tabel 4. Distribusi Dataset per Aplikasi

Aplikasi	Total	Positif	Negatif
BPJS Kesehatan Mobile	1.823	787 (43,2%)	1.036 (56,8%)
IKD	2.241	474 (21,2%)	1.767 (78,8%)
MyPertamina	1.800	711 (39,5%)	1.089 (60,5%)
Total	5.864	1.972 (33,6%)	3.892 (66,4%)

Secara jumlah, rasio kelas dataset adalah 66,4% negatif berbanding 33,6% positif. Rasio ini di bawah ambang 70:30 yang ditetapkan dalam metodologi, sehingga *class weight* tidak diterapkan. *Macro F1-Score* dipilih sebagai metrik utama evaluasi agar tetap adil terhadap kedua kelas tersebut [4], [21]. *Stratified_split* 80:20 dengan *random_state* = 42 menghasilkan 4.691 data latih dan 1.173 data uji. Distribusi kelas menunjukkan perbedaan yang cukup jauh antar aplikasi. IKD memiliki 78,8% ulasan negatif, jauh lebih tinggi dibandingkan dengan aplikasi BPJS (56,8%) dan MyPertamina (60,5%). Dominasi kelas negatif pada dataset IKD dapat dipahami. Tinjauan singkat pada sampel data langsung memperlihatkan keluhan pengguna seperti: "*ngapain daftar akun harus offline?apa fungsi nya aplikasi*", "*ribet aplikasinya nggak bisa efisien*", "*data saya sring tidak tersimpan dan harus login ulang, payah*". Perbedaan distribusi ini membuat model sulit untuk mengenali kelas positif sehingga menahan performa model terutama pada *recall*.

Jurnal Sistem Komputer dan Informatika (JSON)
Volume 7, Nomor 4, Juni 2026
e-ISSN 2685-998X
DOI 10.30865/json.v7i4.9774



Gambar 2. Distribusi Dataset per Aplikasi dan Agregat

Validasi pelabelan dilakukan pada 300 sampel oleh dua anotator independen. Nilai κ dihitung dari label mentah sebelum proses resolusi, sehingga menggambarkan kesepakatan alami tanpa intervensi. Pada Tabel 5 ketiga kombinasi menghasilkan $\kappa \geq 0{,}83$, melebihi ambang batas minimum $\kappa \geq 0{,}6$, ini menjadikan indikator yang kuat bahwa rating dapat menggambarkan sentimen teks ulasan. Meskipun demikian, validasi ini hanya mencakup 300 sampel (5% dari total dataset), sehingga nilai κ yang tinggi merupakan estimasi reliabilitas pada subset tersebut, bukan jaminan bahwa seluruh label rating pada dataset bebas *noise*.

Tabel 5. Hasil Validasi Cohen's Kappa

Kombinasi	Cohen's Kappa	Tingkat Kesepakatan
Anotator 1 vs Anotator 2	0,8822	*Almost Perfect Agreement*
Rating vs Anotator 1	0,8300	*Almost Perfect Agreement*
Rating vs Anotator 2	0,8413	*Almost Perfect Agreement*

Tingkat ketidaksepakatan antar anotator tergolong sangat minim, hanya ditemukan 15 kasus (5,0%) dari 300 sampel. Resolusi untuk anomali ini diselesaikan langsung dengan merujuk rating ulasan sebagai penengah, dikarenakan pada ke-15 kasus tersebut selalu ada satu anotator yang sepakat dengan rating. Selain itu, kami juga mengidentifikasi 14 kasus *rating-text mismatch*, di mana kedua anotator sepakat, tetapi bersebrangan dengan rating dari ulasan pengguna. Beberapa contoh ditunjukkan pada Tabel 6.

Tabel 6. Contoh Kasus Rating-Text Mismatch

Rating	Teks Ulasan	Label Rating	Label Anotator	Keterangan
5/5	"*APLIKASI IKD NYIKSA*"	positif	negatif	Teks negatif, rating tinggi
5/5	"*nyusahin, masuk sana masuk sini gabisa*"	positif	negatif	Teks negatif, rating tinggi
1/5	"*aplikasinya mantap, sangat mudah, menarik…*"	negatif	positif	Teks positif, rating rendah

Keempat belas kasus dikoreksi mengikuti kesepakatan antar anotator, sehingga menghasilkan dataset final 1.972 ulasan positif dan 3.892 ulasan negatif. Fenomena ini menunjukkan bahwa rating bintang tidak selalu menggambarkan isi teks ulasan. Terkadang memberikan bintang tinggi dengan menulis keluhan ataupun sebaliknya. Temuan ini sekaligus menjadi bukti bahwa label berbasis rating bersifat lemah, ketidakselarasan antara rating dan isi teks merupakan *noise* yang inheren pada strategi pelabelan ini. Karena validasi manual hanya mencakup 300 sampel, *noise* serupa diperkirakan masih tersisa pada bagian dataset yang tidak divalidasi. Hal ini perlu dipertimbangkan dalam menginterpretasikan nilai absolut performa model.

3.2 Pra-Pemrosesan Teks

Pra-pemrosesan menghasilkan dua versi dataset dengan karakteristik yang berbeda, *pipeline* ML/DL mereduksi panjang teks secara signifikan, rata-rata dari 16,0 token pada teks asli menjadi 8,3 token. Ini dikarenakan penghapusan *stopword* dan *stemming* menghilangkan banyak kata. *Pipeline* BERT mempertahankan panjang teks yang hampir identik dengan teks asli (rata-rata 16,0 token) karena hanya menerapkan normalisasi tanpa ada pengurangan kosa kata, seperti yang ditunjukkan pada Tabel 7 dan Tabel 8.

Tabel 7. Statistik Panjang Token

Pipeline	*Min* Token	*Max* Token	*Mean*	*Median*
Teks Asli	3	94	16,0	11
ML/DL	1	50	8,3	6
BERT	1	95	16,0	11

Jurnal Sistem Komputer dan Informatika (JSON)
Volume 7, Nomor 4, Juni 2026
e-ISSN 2685-998X
DOI 10.30865/json.v7i4.9774

Tabel 8. Hasil Pra-Pemrosesan Teks

Teks Asli	ML/DL	BERT
"aneh banget ini apk padahal udah terdaftar, tapi kenapa pas mau login keterangan blm terdaftar, malah otp nya ga muncul"	*"aneh banget apk daftar login terang daftar otp muncul"*	"*aneh banget ini apk padahal sudah terdaftar tapi kenapa pas mau login keterangan belum terdaftar malah otp nya enggak muncul*"
"*Sistemnya lelet, sudah ke Dukcapil untuk Daftar Luring, lamat email aktif sudah bener, tapi sampai sekarang belum juga dikirim kode Verifikasi*"	"*lamat lelet dukcapil daftar luring lamat email aktif kirim kode verifikasi*"	"*sistemnya lelet sudah ke dukcapil untuk daftar luring lamat email aktif sudah benar tapi sampai sekarang belum juga dikirim kode verifikasi*"
"*terimakasih my pertamina, kami jadi lebih mudah*"	"*terimakasih my pertamina mudah*"	"*terimakasih my pertamina kami jadi lebih mudah*"

3.3 Hasil Perbandingan Performa Klasifikasi

Sebelum performa dilaporkan, Tabel 9 merangkum *hyperparameter* optimal yang diperoleh dari proses *tuning* masing-masing model. Model IndoBERT berhenti pada epoch ke-1 dan IndoBERTweet berhenti pada *epoch* ke-2 dikarenakan kriteria *early stopping* terpenuhi.

Tabel 9. *Hyperparameter* Optimal

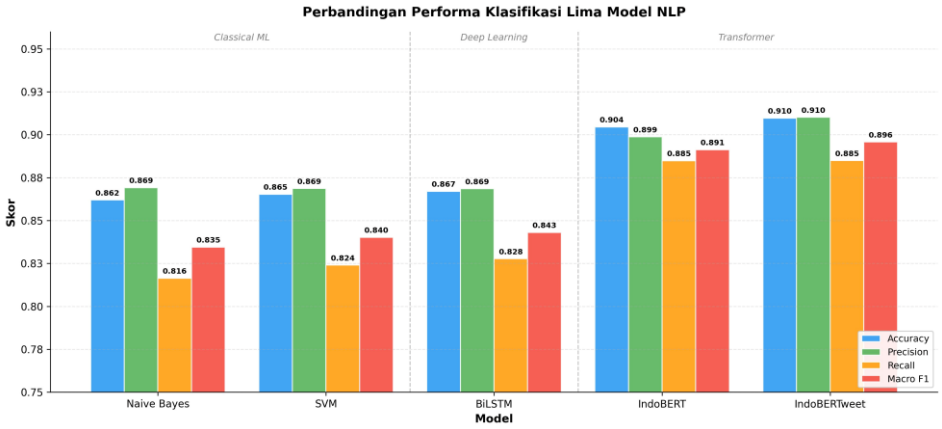
Model	*Hyperparameter* Optimal
Naive Bayes	*alpha=0,1; max_features=10K (Grid Search 5-fold)*
SVM	*C=1; max_features=20K (Grid Search 5-fold)*
BiLSTM	*units=256; dropout=0,3; learning rate=0,001; batch size=64 (Grid Search 5-fold)*
IndoBERT	*learning rate=2e-5; batch size=16; berhenti epoch ke-1 (early stopping patience=3)*
IndoBERTweet	*learning rate=3e-5; batch size=16; berhenti epoch ke-2 (early stopping patience=3)*

Evaluasi performa dilakukan pada 1.173 data uji menggunakan *accuracy*, *macro precision*, *macro recall*, dan *macro F1-Score.* Hasil ditunjukkan pada Tabel 10 dan Gambar 3.

Tabel 10. Perbandingan Performa Klasifikasi Lima Model

Model	Generasi	Accuracy	Macro Precision	Macro Recall	Macro F1
Naive Bayes	*Classical* ML	0,8619	0,8691	0,8164	0,8345
SVM	*Classical* ML	0,8653	0,8686	0,8240	0,8402
BiLSTM	*Deep Learning*	0,8670	0,8685	0,8277	0,8430
IndoBERT	*Transformer*	0,9045	0,8987	0,8848	0,8912
IndoBERTweet	*Transformer*	0,9096	0,9101	0,8849	0,8957

Catatan: precision, recall, dan F1-Score dihitung menggunakan macro-averaging, yaitu rata-rata metrik dari kedua kelas dengan bobot setara, untuk mengakomodasi distribusi kelas yang tidak seimbang.



Gambar 3. Perbandingan Performa Klasifikasi Lima Model NLP

Tabel 10 dan Gambar 3 menunjukkan perbandingan keempat metrik evaluasi pada kelima model. Selisih *accuracy* dan *macro precision* antar model relatif kecil, sehingga *macro F1-Score* menjadi metrik yang lebih informatif karena memperhitungkan keseimbangan performa kedua kelas tanpa bias terhadap salah satu kelas. Dari

Jurnal Sistem Komputer dan Informatika (JSON)
Volume 7, Nomor 4, Juni 2026
e-ISSN 2685-998X
DOI 10.30865/json.v7i4.9774

sisi *macro F1-Score* model IndoBERTweet mendapatkan nilai 0,8957 dan IndoBERT 0,8912. Kedua model transformer mengungguli model generasi sebelumnya dengan jarak yang cukup jelas, *Naive Bayes* (0,8345), SVM (0,8402), dan BiLSTM (0,8430) berada dalam rentang yang berdekatan. Pada Gambar 3 memperlihatkan pola ini secara visual, ada lompatan yang cukup terlihat antara *Deep Learning* dengan *Transformer*. Sementara itu kelompok *Classical* ML dan *Deep Learning* hampir sulit dibedakan secara grafis.

Naive Bayes dan SVM mendapatkan *macro F1-Score* dengan nilai yang kompetitif untuk model yang sederhana. Representasi TF-IDF dengan *n-gram(1,2)* cukup efektif dalam menangkap pola leksikal dan *bigram* yang informatif dari teks ulasan [23], [25]. SVM mengungguli *Naive Bayes* di seluruh metrik, konsisten dengan temuan Husada & Paramita [6] pada teks Bahasa Indonesia.

BiLSTM memperoleh *macro F1-Score* 0,8430 hanya selisih 0,0028 di atas SVM (0,8402). Peningkatan yang kecil ini tidak sebanding dengan kompleksitas modelnya. Temuan ini sejalan dengan penelitian sebelumnya pada dataset Bahasa Indonesia. Mutmainah et al. [9] melaporkan BiLSTM efektif namun membutuhkan sumber daya yang lebih besar dibandingkan dengan ML klasik.

Kedua model *transformer* berada di atas model generasi sebelumnya. IndoBERTweet mendapatkan nilai *macro F1-Score* 0,8957 sedikit lebih tinggi dibandingkan dengan IndoBERT (0,8912) selisih 0,0045. Meskipun IndoBERTweet dilatih dengan 26 juta cuitan Twitter Indonesia [12], model ini tidak memberikan keunggulan yang berarti atas IndoBERT pada domain ini.

Tabel 11. *Macro F1* per Aplikasi

Model	IKD	BPJS	MyPertamina
Naive Bayes	0,7937	0,8759	0,8053
SVM	0,7977	0,9005	0,7953
BiLSTM	0,8059	0,9011	0,7962
IndoBERT	0,8705	0,9467	0,8418
IndoBERTweet	0,8713	0,9344	0,8629

Tabel 11 menyajikan *macro F1-Score* tiap model per aplikasi (440 ulasan IKD, 341 BPJS Kesehatan Mobile, 392 MyPertamina). Kedua model *transformer* konsisten unggul pada seluruh aplikasi. MyPertamina menjadi domain paling menantang (*macro F1* tertinggi 0,8629). Keunggulan *transformer* atas *Classical ML* makin besar pada aplikasi yang sulit diklasifikasikan, menandakan model ini lebih tangguh menghadapi data yang kompleks.

Selanjutnya pada sisi *macro recall*, seluruh model menghasilkan nilai yang lebih rendah dibandingkan dengan *macro precision*. Ini menunjukkan kecenderungan model untuk lebih berhati-hati dalam memprediksi kelas positif. Kecenderungan ini berkaitan dengan distribusi dataset yang didominasi dengan kelas negatif (66,4%). Hasil *macro recall* per kelas ditunjukkan pada Tabel 12.

Tabel 12. *Recall* per Kelas Sentimen

Model	Generasi	*Recall* Negatif	*Recall* Positif	Selisih
Naive Bayes	*Classical* ML	0,9551	0,6777	0,2774
SVM	*Classical* ML	0,9499	0,6980	0,2519
BiLSTM	*Deep Learning*	0,9474	0,7081	0,2393
IndoBERT	*Transformer*	0,9448	0,8249	0,1199
IndoBERTweet	*Transformer*	0,9602	0,8071	0,1531

Pada Tabel 12 menunjukkan selisih antara *recall* kelas negatif dan *recall* kelas positif. *Naive Bayes* dan SVM menjadi model yang paling besar selisihnya, ini dikarenakan representasi TF-IDF kurang menangkap pola positif yang beragam pada teks. BiLSTM sedikit mempersempit selisih di angka 0,2393. Model Transformer pada domain ini lebih efektif dalam mendeteksi sentimen positif, IndoBERT dengan nilai selisih 0,1199 dan IndoBERTweet 0,1531.

Tabel 13. *Confusion matrix*

Model	TN	FP	FN	TP
Naive Bayes	744	35	127	267
SVM	740	39	119	275
BiLSTM	738	41	115	279
IndoBERT	736	43	69	325
IndoBERTweet	748	31	76	318

Pola yang konsisten ditunjukkan Tabel 13, nilai FN menurun dari 127 pada *Naive Bayes* menjadi 69 pada IndoBERT dan 76 pada IndoBERTweet. Ini mengonfirmasi bahwa representasi kontekstual BERT lebih mampu menangkap pola positif yang lebih beragam pada aplikasi Bahasa Indonesia [10], [12]. Sebaliknya FP lebih stabil di semua model, berkisar di angka 31 hingga 43, menunjukkan bahwa seluruh model andal dalam menangkap pola negatif, di mana kelas negatif lebih dominan pada dataset ini.

Jurnal Sistem Komputer dan Informatika (JSON)
Volume 7, Nomor 4, Juni 2026
e-ISSN 2685-998X
DOI 10.30865/json.v7i4.9774

Untuk memahami sumber perbedaan performa antar generasi, dilakukan analisis kualitatif terhadap kasus-kasus yang diklasifikasikan berbeda oleh model. Dari 1.173 data uji, ditemukan 63 kasus di mana *Naive Bayes* dan SVM salah, namun IndoBERTweet benar, dan 26 kasus sebaliknya. Tabel 14 menunjukkan tiga pola kesalahan representatif. Pada pola pertama, ML klasik gagal menangkap sentimen pada ulasan multi klausa, apresiasi diikuti permintaan fitur. Ini dikarenakan TF-IDF bekerja pada *level* token independen tanpa memahami urutan dan relasi antar klausa. Pola kedua, negasi lokal seperti "*bukan lebih mudah*" tidak tertangkap, karena "*mudah*" memiliki bobot TF-IDF positif yang kuat. IndoBERTweet, melalui representasi kontekstual, mampu menangkap kedua pola tersebut dengan lebih baik. Sebaliknya, pada pola ketiga, IndoBERTweet justru terkecoh pada ulasan dengan sentimen campuran, frasa pembuka positif diikuti oleh keluhan. Ini menunjukkan bahwa model *transformer* memiliki batas kemampuan pada kasus sentimen ambigu. Pola-pola ini menjadi *insight* penting bagi praktisi yang mempertimbangkan implementasi sistem analisis sentimen pada domain layanan publik dengan karakteristik teks ulasan yang serupa.

Tabel 14. Analisis Error Kualitatif

Teks Ulasan	Konteks	Aktual	NB	SVM	IBT	Pola
"saya sangat menyukai aplikasi ini namun bisakah menambah aplikasi…"	Apresiasi diikuti dengan permintaan fitur	Positif	Negatif	Negatif	Positif	Konteks multi-klausa
"*bukan lebih mudah tapi tambah ribet*"	Negasi eksplisit	Negatif	Positif	Positif	Negatif	Negasi lokal
"*aplikasi sudah bagus, hanya saja saat hari libur tidak bisa di akses..*"	Frasa pembuka positif diikuti keluhan	Negatif	Negatif	Negatif	Positif	Sentimen campuran

Catatan: NB = Naive Bayes, SVM = Support Vector Machine, IBT = IndoBERTweet

Selanjutnya, untuk menguji apakah perbedaan performa antar model signifikan secara statistik, dilakukan McNemar's Test [26], [27] pada empat kombinasi model yang berdekatan secara generasi. Perbandingan antar kombinasi yang tidak diuji tidak dapat disimpulkan signifikansinya dari hasil ini. Dari keempat kombinasi yang diuji, hanya lompatan dari BiLSTM ke IndoBERT yang signifikan (p = 0,0001). Perbedaan dalam satu generasi, *Naive Bayes* dengan SVM (p = 0,6778), SVM dengan BiLSTM (p = 0,9161), dan IndoBERT dengan IndoBERTweet (p = 0,5758) tidak signifikan secara statistik.

Tabel 15. Uji Signifikansi McNemar's Test

Kombinasi Model	b	c	p-value	Signifikan
Naive Bayes vs SVM	24	28	0,6778	Tidak
SVM vs BiLSTM	44	46	0,9161	Tidak
BiLSTM vs IndoBERT	36	80	0,0001	Ya***
IndoBERT vs IndoBERTweet	23	28	0,5758	Tidak

*Catatan: *** = p < 0,001*

Tabel 15 menampilkan nilai b dan c, di mana nilai b adalah jumlah sampel yang diprediksi benar oleh model A tetapi salah oleh model B, kemudian nilai c adalah kebalikan dari nilai b.

3.4 Hasil Perbandingan Efisiensi Komputasi

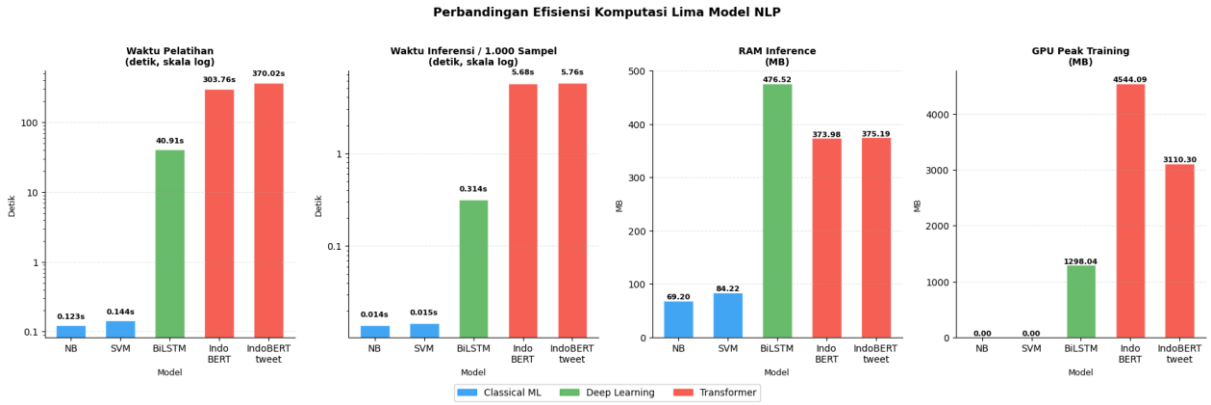
Selain performa klasifikasi, efisiensi komputasi merupakan evaluasi yang tidak kalah penting, terutama dalam konteks *deployment* sistem analisis sentimen pada infrastruktur yang memiliki keterbatasan sumber daya. Bagian ini akan membahas hasil pengukuran empat indikator efisiensi: waktu pelatihan, waktu inferensi per 1.000 sampel, konsumsi RAM saat inferensi, serta penggunaan GPU. Hasil selengkapnya ditunjukkan pada Tabel 16.

Tabel 16. Perbandingan Efisiensi Komputasi

Model	Waktu Pelatihan	Waktu Inferensi	RAM Inferensi	GPU Puncak Pelatihan
Naive Bayes	0,12 dtk	0,014 dtk	69,20 MB	0,00 MB
SVM	0,14 dtk	0,015 dtk	84,22 MB	0,00 MB
BiLSTM	40,91 dtk	0,314 dtk	476,52 MB	1.298,04 MB
IndoBERT	303,76 dtk	5,677 dtk	373,98 MB	4.544,09 MB
IndoBERTweet	370,02 dtk	5,756 dtk	375,19 MB	3.110,30 MB

Catatan: RAM inferensi diukur sebagai Resident Set Size (RSS) pada level proses menggunakan psutil, pada 1.000 sampel yang seragam untuk seluruh model. GPU puncak diukur pada tahap pelatihan menggunakan torch.cuda.max_memory_allocated() untuk model PyTorch dan nvidia-smi untuk BiLSTM (TensorFlow).

Jurnal Sistem Komputer dan Informatika (JSON)
Volume 7, Nomor 4, Juni 2026
e-ISSN 2685-998X
DOI 10.30865/json.v7i4.9774



Gambar 4. Perbandingan Efisiensi Komputasi

Tabel 16 merangkum hasil pengukuran komputasi dengan waktu pelatihan berkisar dari 0,12 detik (*Naive Bayes*) hingga 370,02 detik (IndoBERTweet) dengan kebutuhan GPU dari 0 MB hingga 4.544 MB (IndoBERT).

Peningkatan *macro F1-Score* sebesar 5,55 poin dari SVM (0,8402) ke IndoBERTweet (0,8957) diperoleh dengan waktu pelatihan 2.643 kali lebih lama (0,14 detik vs 370,02 detik) dan waktu inferensi 384 *kali* lebih lambat (0,015 detik vs 5,756 detik). Ghatora et al. [30] mencatat pola serupa, model *pre-trained* membawa biaya komputasi yang jauh lebih besar, dan ini menjadikan salah satu faktor penentu dalam keputusan *deployment*.

Naive Bayes dan SVM selesai dilatih dengan waktu di bawah 0,15 detik dan inferensi di bawah 0,02 detik per 1000 sampel, tanpa GPU. TF-IDF hanya menghitung statistik frekuensi *term*, tidak ada optimasi gradien, tidak ada bobot yang diperbarui secara berkala [23]. Ini berbeda dengan *transformer* yang memproses setiap token melalui *self-attention* pada 12 lapisan secara berurutan dengan kompleksitas $O(n^2)$ terhadap panjang sekuens [28].

BiLSTM berada pada posisi tengah: 40,91 detik pelatihan, 0,314 detik inferensi, GPU pelatihan 1.298 MB. RAM inferensi BiLSTM tercatat 476,52 MB, melampaui IndoBERT (373,98 MB) dan IndoBERTweet (375,19 MB). Tingginya kebutuhan RAM BiLSTM berkaitan dengan pemuatan matriks *embedding FastText* berdimensi 300 beserta arsitektur LSTM. Temuan ini menunjukkan bahwa model *deep learning* berbasis *embedding* pra-terlatih tidak selalu lebih ringan dari sisi memori proses dibandingkan model *transformer*, meskipun perbandingan ini terbatas pada lingkungan pengukuran yang digunakan.

Di antara dua model *transformer*, IndoBERTweet lebih efisien dibandingkan dengan IndoBERT pada dua sisi sekaligus: *macro F1-Score* lebih tinggi (0,8957 vs 0,8912) dan GPU pelatihan yang lebih rendah (3.110 MB vs 4.544 MB). Selisih 1.434 MB cukup berarti pada lingkungan dengan keterbatasan sumber daya komputasi.

3.5 Implikasi Penelitian

Hasil dari sub bagian 3.3 dan 3.4 menggambarkan bahwa tidak ada satu model yang unggul dalam kedua evaluasi. Model dengan *macro F1-Score* tertinggi membutuhkan sumber daya komputasi terbesar dan model tercepat mendapatkan performa klasifikasi terendah. Jim et al. [31] menyebutkan kondisi ini sebagai *trade-off* inheren dalam pemilihan model NLP untuk aplikasi nyata, di mana akurasi, waktu inferensi, dan kebutuhan sumber daya harus dipertimbangkan secara bersamaan. Jiang et al. [29] dalam tinjauan sistematis NLP di sektor pemerintahan mencatat bahwa keterbatasan infrastruktur bukan keterbatasan algoritma yang paling sering menghambat adopsi sistem analisis sentimen di instansi publik.

Temuan ini memiliki beberapa implikasi yang perlu dicatat, dengan catatan bahwa seluruh implikasi didasarkan pada kondisi eksperimen yang spesifik, 5.864 ulasan Bahasa Indonesia, domain layanan publik digital, lingkungan Google Colab GPU T4, sehingga validasi pada konteks yang berbeda tetap diperlukan.

Pertama, dari sisi performa, SVM menghasilkan *macro F1-Score* 0,8402 dengan waktu pelatihan 0,14 detik dan RAM 84,22 MB tanpa GPU, selisihnya hanya 0,0028 terhadap nilai *macro F1-Score* BiLSTM, dan tidak signifikan secara statistik ($p = 0,9161$) yang mengimplikasikan bahwa kompleksitas BiLSTM tidak memberikan manfaat performa yang sepadan pada dataset ini. Kedua, di antara pasangan model berdekatan yang diuji, lompatan performa yang signifikan secara statistik hanya terjadi dari BiLSTM ke IndoBERT ($p = 0,0001$) mengimplikasikan bahwa perpindahan ke *transformer* memberikan perubahan kualitatif, bukan sekadar peningkatan bertahap. Ketiga, IndoBERTweet mencapai performa tertinggi (*macro F1-Score* 0,8957) dengan kebutuhan GPU pelatihan yang lebih rendah dibandingkan dengan IndoBERT mengimplikasikan bahwa IndoBERTweet lebih efisien secara komputasi dan lebih akurat pada domain ini.

4. KESIMPULAN

Penelitian ini membandingkan lima model dari tiga generasi pendekatan NLP pada analisis sentimen ulasan tiga aplikasi layanan publik digital Indonesia (IKD, BPJS Kesehatan Mobile, MyPertamina), menggunakan 5.864

ulasan berbahasa Indonesia yang divalidasi dengan Cohen's Kappa (κ = 0,8822). Penelitian ini berkontribusi dengan membandingkan kelima model tidak hanya dari sisi performa klasifikasi, tetapi juga efisiensi komputasi yang sering diabaikan dalam studi sejenis. Dari sisi performa, urutan antar generasi konsisten: *Classical ML* terendah, diikuti *Deep Learning*, dan *Transformer* tertinggi. IndoBERTweet mencapai *macro F1-Score* tertinggi (0,8957), namun selisihnya hanya 0,0045 dengan IndoBERT. BiLSTM (0,8430) hanya unggul 0,0028 atas SVM (0,8402), tidak sebanding dengan kompleksitasnya, konsisten dengan temuan Mutmainah et al. [9] dan Rahman Isnain et al. [15]. Dari sisi efisiensi, perbedaan antar generasi jauh lebih besar daripada perbedaan performa: peningkatan 5,55 poin *macro F1-Score* dari SVM ke IndoBERTweet menuntut waktu pelatihan 2.643 kali lebih lama, dan pengukuran RAM secara konsisten menggunakan RSS menunjukkan BiLSTM (476,52 MB) justru lebih besar dari kedua *transformer* pada lingkungan eksperimen ini. Hasil penelitian bersifat kontekstual terhadap tiga aplikasi, klasifikasi biner, pelabelan berbasis rating (*weak labeling*), dan lingkungan Google Colaboratory GPU T4, sehingga generalisasi memerlukan validasi lanjutan. Penelitian selanjutnya dapat memperluas cakupan domain aplikasi, mengeksplorasi *aspect-based sentiment analysis* untuk menangkap nuansa sentimen yang lebih kaya, serta menguji model ringan seperti DistilBERT untuk skenario *deployment* dengan sumber daya terbatas. Secara keseluruhan, pemilihan model yang tepat bergantung pada ketersediaan infrastruktur, bukan semata-mata pada angka performa.

REFERENCES

[1] Republik Indonesia, *Peraturan Presiden Nomor 95 Tahun 2018 tentang Sistem Pemerintahan Berbasis Elektronik*. Jakarta, Indonesia, 2018.

[2] Kementerian Energi dan Sumber Daya Mineral, *Keputusan Menteri ESDM Nomor 37.K/HK.02/MEM.M/2022 tentang Jenis Bahan Bakar Minyak Khusus Penugasan (JBKP)*. Jakarta, Indonesia, 2022.

[3] Republik Indonesia, *Undang-Undang Nomor 25 Tahun 2009 tentang Pelayanan Publik*. Jakarta, Indonesia, 2009.

[4] L. Ashbaugh and Y. Zhang, "A Comparative Study of Sentiment Analysis on Customer Reviews Using Machine Learning and Deep Learning," *Computers*, vol. 13, no. 12, pp. 1–16, Dec. 2024, doi: 10.3390/computers13120340.

[5] A. Patel, P. Oza, and S. Agrawal, "Sentiment Analysis of Customer Feedback and Reviews for Airline Services using Language Representation Model," *Procedia Comput. Sci.*, vol. 218, pp. 2459–2467, 2023, doi: 10.1016/j.procs.2023.01.221.

[6] H. C. Husada and A. S. Paramita, "Analisis Sentimen Pada Maskapai Penerbangan di Platform Twitter Menggunakan Algoritma Support Vector Machine (SVM)," *Teknika*, vol. 10, no. 1, pp. 18–26, Feb. 2021, doi: 10.34148/teknika.v10i1.311.

[7] W. Andriyani, Y. Astuti, B. A. Wisesa, and D. Hengki, "Analisis Sentimen pada Ulasan Produk dengan SVM dan Word2Vec," *JIKO (Jurnal Informatika dan Komputer)*, vol. 8, no. 1, pp. 173–185, Feb. 2024, doi: 10.26798/jiko.v8i1.1498.

[8] L. Xiaoyan, R. C. Raga, and S. Xuemei, "GloVe-CNN-BiLSTM Model for Sentiment Analysis on Text Reviews," *J. Sens.*, vol. 2022, 2022, doi: 10.1155/2022/7212366.

[9] S. Mutmainah, D. H. Fudholi, and S. Hidayat, "Analisis Sentimen dan Pemodelan Topik Aplikasi Telemedicine Pada Google Play Menggunakan BiLSTM dan LDA," *Jurnal Media Informatika Budidarma*, vol. 7, no. 1, p. 312, Jan. 2023, doi: 10.30865/mib.v7i1.5486.

[10] J. M. Halim, P. N. Hoshe, S. Soenarto, and R. Sutoyo, "Comparative Analysis of Machine Learning, Deep Learning, and IndoBERT Models Using SPAMID-PAIR Dataset," in *Proceedings - 2024 International of Seminar on Application for Technology of Information and Communication: Smart And Emerging Technology for a Better Life, iSemantic 2024*, Institute of Electrical and Electronics Engineers Inc., 2024, pp. 253–259. doi: 10.1109/iSemantic63362.2024.10762382.

[11] K. S. Nugroho, A. Y. Sukmadewa, H. Wuswilahaken DW, F. A. Bachtiar, and N. Yudistira, "BERT Fine-Tuning for Sentiment Analysis on Indonesian Mobile Apps Reviews," in *Proceedings of the 6th International Conference on Sustainable Information Engineering and Technology*, in SIET '21. New York, NY, USA: Association for Computing Machinery, 2021, pp. 258–264. doi: 10.1145/3479645.3479679.

[12] 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*, Association for Computational Linguistics, Nov. 2021, pp. 10660–10668. doi: 10.18653/v1/2021.emnlp-main.833.

[13] I. N. Kusuma *et al.*, "Analisis Sentimen Pada Pengguna Aplikasi Dana Menggunakan Algoritma Naive Bayes," *Jurnal Mahasiswa Teknik Informatika*, vol. 8, no. 2, p. 1470, Apr. 2024, doi: 10.36040/jati.v8i2.9041.

[14] N. Aulia, S. N. Sari, and N. Wakhidah, "Analisis Sentimen Aplikasi Get Contact di APP Store Menggunakan Metode SVM (Support Vector Machine)," *Jurnal Informatika: Jurnal pengembangan IT*, vol. 10, no. 1, pp. 139–148, 2025, doi: 10.30591/jpit.v10i1.8057.

[15] A. Rahman Isnain, H. Sulistiani, B. Miftaq Hurohman, A. Nurkholis, and Styawati, "Analisis Perbandingan Algoritma LSTM dan Naive Bayes untuk Analisis Sentimen," *Jurnal Edukasi dan Penelitian Informatika*, vol. 8, no. 2, pp. 299–303, Aug. 2022, doi: 10.26418/jp.v8i2.54704.

[16] H. Jayadianti, W. Kaswidjanti, A. Utomo, S. Saifullah, F. A. Dwiyanto, and R. Drezewski, "Sentiment analysis of Indonesian reviews using fine-tuning IndoBERT and R-CNN," *ILKOM Jurnal Ilmiah*, vol. 14, no. 3, pp. 348–354, 2022, doi: 10.33096/ilkom.v14i3.1505.348-354.

[17] R. Maulana, A. Voutama, and T. Ridwan, "Analisis Sentimen Ulasan Aplikasi MyPertamina pada Google Play Store menggunakan Algoritma NBC," *Jurnal Teknologi Terpadu*, vol. 9, no. 1, pp. 42–48, Jul. 2023, doi: 10.54914/jtt.v9i1.609.

[18] A. I. Tanggraeni and M. N. N. Sitokdana, "Analisis Sentimen Aplikasi E-Government Pada Google Play Menggunakan Algoritma Naïve Bayes," *JATISI*, vol. 9, no. 2, pp. 785–795, Jun. 2022, doi: 10.35957/jatisi.v9i2.1835.

[19] Tarwoto, R. Nugroho, N. Azka, and W. S. R. Graha, "Analisis Sentimen Ulasan Aplikasi Mobile JKN di Google PlayStore Menggunakan IndoBERT," *Jurnal JTIK (Jurnal Teknologi Informasi dan Komunikasi)*, vol. 9, no. 2, pp. 495–505, Apr. 2025, doi: 10.35870/jtik.v9i2.3340.

[20] Dhendra and V. G. Utomo, "Benchmarking IndoBERT and Transformer Models for Sentiment Classification on Indonesian E-Government Service Reviews," *Jurnal Transformatika*, vol. 23, no. 1, pp. 86–95, Jun. 2025, doi: 10.26623/transformatika.v23i1.12095.

[21] A. A. Chamid, Widowati, and R. Kusumaningrum, "Labeling Consistency Test of Multi-Label Data for Aspect and Sentiment Classification Using the Cohen Kappa Method," *Ingenierie des Systemes d'Information*, vol. 29, no. 1, pp. 161–167, Feb. 2024, doi: 10.18280/isi.290118.

[22] N. Aliyah Salsabila, Y. Ardhito Winatmoko, A. Akbar Septiandri, and A. Jamal, "Colloquial Indonesian Lexicon," in *2018 International Conference on Asian Language Processing (IALP)*, Nov. 2018, pp. 226–229. doi: 10.1109/IALP.2018.8629151.

[23] D. Jurafsky and J. H. Martin, *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition, with Language Models*, 3rd ed. 2026. Accessed: Apr. 16, 2026. [Online]. Available: https://web.stanford.edu/~jurafsky/slp3/

[24] M. R. Haziq, Y. Sibaroni, and S. S. Prasetyowati, "Word Embedding Optimization In Sentiment Analysis Of Reviews On Mytelkomsel App Using Long Short-Term Memory And Synthetic Minority Over-Sampling Technique," *Jurnal Teknik Informatika (Jutif)*, vol. 5, no. 6, pp. 1581–1589, Dec. 2024, doi: 10.52436/1.jutif.2024.5.6.2498.

[25] J. Eisenstein, *Introduction to Natural Language Processing*. Cambridge, MA, USA: The MIT Press, 2019.

[26] Q. McNemar, "Note on the sampling error of the difference between correlated proportions or percentages," *Psychometrika*, vol. 12, no. 2, pp. 153–157, 1947, doi: 10.1007/BF02295996.

[27] O. Rainio, J. Teuho, and R. Klén, "Evaluation metrics and statistical tests for machine learning," *Sci. Rep.*, vol. 14, no. 1, p. 6086, 2024, doi: 10.1038/s41598-024-56706-x.

[28] F. Wang, "Comparative Evaluation of Sentiment Analysis Methods: From Traditional Techniques to Advanced Deep Learning Models," *Applied and Computational Engineering*, vol. 105, pp. 23–29, Apr. 2024, doi: 10.54254/2755-2721/105/2024TJ0056.

[29] Y. Jiang, P. C.-I. Pang, D. Wong, and H. Y. Kan, "Natural Language Processing Adoption in Governments and Future Research Directions: A Systematic Review," *Applied Sciences*, vol. 13, no. 22, 2023, doi: 10.3390/app132212346.

[30] P. S. Ghatora, S. E. Hosseini, S. Pervez, M. J. Iqbal, and N. Shaukat, "Sentiment Analysis of Product Reviews Using Machine Learning and Pre-Trained LLM," *Big Data and Cognitive Computing*, vol. 8, no. 12, pp. 1–18, Dec. 2024, doi: 10.3390/bdcc8120199.

[31] J. R. Jim, M. A. R. Talukder, P. Malakar, M. M. Kabir, K. Nur, and M. F. Mridha, "Recent advancements and challenges of NLP-based sentiment analysis: A state-of-the-art review," *Natural Language Processing Journal*, vol. 6, p. 100059, 2024, doi: 10.1016/j.nlp.2024.100059.