Building of Informatics, Technology and Science (BITS)
Volume 7, No 2, September 2025 Page: 1456−1466
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v7i2.8034

Klasifikasi Spam Bahasa Indonesia dengan IndoBERT dan XLM-RoBERTa: Evaluasi Pooling, Stride, dan Late-Fusion

Darmono Darmono*, Rujianto Eko Saputro, Azhari Shouni Barkah

Fakultas Ilmu Komputer, Program Studi Magister Ilmu Komputer, Universitas Amikom Purwokerto, Banyumas, Indonesia
Email: [1,*]23ma41d027@students.amikompurwokerto.ac.id, [2]rujianto@amikompurwokerto.ac.id,
[3]azhari@amikompurwokerto.ac.id
Email Penulis Korespondensi: 23ma41d027@students.amikompurwokerto.ac.id
Submitted: **17/07/2025**; Accepted: **30/09/2025**; Published**: 30/09/2025**

Abstrak−Spam pada pesan pendek berbahasa Indonesia seperti SMS dan email tetap menantang karena variasi leksikal, obfusikasi karakter, dan distribusi kelas yang tidak seimbang. Penelitian ini menyajikan evaluasi sistematis untuk menjawab pertanyaan: konfigurasi apa yang paling seimbang antara akurasi dan efisiensi pada deteksi spam Indonesia. Kami membandingkan dua backbone pralatih (IndoBERT dan XLM RoBERTa) serta strategi representasi (truncation dibandingkan chunking), skema peringkasan (pooling), dan strategi penggabungan fitur (fusion). Sistem dirancang secara feature based dengan penekanan pada kesederhanaan, lalu dievaluasi menggunakan F1 Macro, recall kelas spam, AUPRC (Area Under the Precision Recall Curve), dan metrik efisiensi berupa waktu pembentukan embedding serta latensi pelatihan. Hasil menunjukkan bahwa IndoBERT unggul pada keputusan biner dengan kinerja tinggi sekaligus efisiensi komputasi, sementara XLM RoBERTa sedikit lebih baik pada AUPRC sehingga relevan untuk skenario pemeringkatan risiko. Strategi truncation dan mean pooling konsisten memberikan hasil stabil. Late fusion memang hanya memberi kenaikan marginal, tetapi tetap penting karena menunjukkan potensi integrasi sinyal domain dalam meningkatkan ketahanan model pada kondisi obfusikasi tinggi. Rekomendasi akhir untuk produksi adalah IndoBERT dengan konfigurasi truncation, mean pooling, dan embedding only. Keterbatasan penelitian meliputi fokus pada pesan pendek dan belum diuji pada skenario obfusikasi ekstrem. Penelitian lanjutan diarahkan pada augmentasi karakter, evaluasi lintas domain, dan penalaan ambang berbasis biaya kesalahan.

Kata Kunci: Deteksi Spam; IndoBERT; XLM-RoBERTa; Bahasa Indonesia; Truncation; Chunking; Mean Pooling

Abstract−Spam detection for Indonesian short messages such as SMS and email remains challenging due to lexical variation, character obfuscation, and class imbalance. This study provides a systematic evaluation to determine the most balanced configuration between accuracy and efficiency for Indonesian spam filtering. We compare two pretrained backbones (IndoBERT and XLM RoBERTa), along with representation strategies (truncation versus chunking), summarization schemes (pooling), and feature fusion approaches. The system follows a feature based design with an emphasis on simplicity, and is assessed using F1 Macro, spam class recall, AUPRC (Area Under the Precision Recall Curve), and efficiency metrics in terms of embedding build time and training latency. Results indicate that IndoBERT achieves superior binary classification performance with high efficiency, while XLM RoBERTa slightly outperforms on AUPRC, making it more suitable for risk ranking scenarios. Truncation combined with mean pooling consistently yields stable results. Although late fusion only provides marginal improvements, it remains relevant as it highlights the potential of domain specific signals to enhance robustness under heavy obfuscation. The final recommendation for production is IndoBERT with truncation, mean pooling, and embedding only. Limitations include the focus on short messages and the lack of evaluation under extreme obfuscation. Future work should explore character level augmentation, cross domain evaluation, and cost sensitive threshold tuning.

Keywords: Spam Detection; IndoBERT; XLM-RoBERTa; Indonesian; Truncation; Chunking; Mean Pooling

1. PENDAHULUAN

Spam digital merupakan salah satu bentuk gangguan yang semakin berkembang seiring dengan pesatnya pertumbuhan konektivitas digital dan adopsi platform komunikasi daring. Pada berbagai kanal komunikasi seperti SMS, email, dan media sosial, spam tidak hanya menurunkan kenyamanan pengguna, tetapi juga sering kali berperan sebagai vektor penyebaran malware, penipuan daring (phishing), serta eksploitasi data pribadi. Fenomena ini tidak mengenal batas geografis, dan Indonesia sebagai salah satu negara dengan populasi pengguna internet terbanyak di dunia juga turut terdampak. Menurut data Asosiasi Penyelenggara Jasa Internet Indonesia (APJII), lebih dari 210 juta penduduk Indonesia telah terhubung dengan internet, menciptakan ekosistem digital yang sangat aktif namun sekaligus rentan terhadap penyalahgunaan pesan digital, termasuk dalam bentuk spam. Deteksi otomatis terhadap pesan spam dalam Bahasa Indonesia menjadi semakin penting, namun juga menantang. Bahasa Indonesia memiliki kekhasan leksikal dan morfologis yang kompleks, termasuk fleksibilitas struktur kalimat, penggunaan slang atau serapan lokal, serta obfusikasi karakter oleh pengirim spam untuk menghindari deteksi sistematis. Tandra et al. (2021) menunjukkan bahwa pendekatan berbasis pembelajaran mendalam telah berhasil diterapkan untuk deteksi spam dalam Bahasa Indonesia, meskipun masih menghadapi hambatan seperti panjang teks dan variasi struktur bahasa [1].

Ancaman spam tidak terbatas pada email atau SMS saja, namun juga meluas ke media sosial seperti Twitter dan Instagram. Komentar spam yang menyerupai interaksi wajar kini kerap ditemukan di kolom komentar akun publik figur maupun instansi resmi. Studi oleh Alhaura dan Budi (2020) menyoroti pola aktivitas akun spam di Twitter Indonesia yang memiliki kesamaan perilaku, seperti frekuensi posting dan penggunaan tautan promosi, meskipun deteksi semantik tetap menjadi tantangan utama [2]. Di ranah email, pendekatan klasik seperti Naïve Bayes dan n-gram masih digunakan, tetapi terbukti terbatas dalam memahami konteks dan makna teks yang tersirat [3]. Di sinilah teknologi berbasis pemrosesan bahasa alami (natural language processing/NLP) modern menjadi relevan.

Building of Informatics, Technology and Science (BITS)
Volume 7, No 2, September 2025 Page: 1456−1466
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v7i2.8034

Model Transformer seperti BERT telah menjadi tulang punggung dalam berbagai tugas NLP, termasuk klasifikasi teks. Untuk Bahasa Indonesia, beberapa varian seperti IndoBERT, IndoBERTweet, dan IndoRoBERTa telah dikembangkan dan digunakan dalam beragam studi seperti deteksi ujaran kebencian[4], klasifikasi konten eksplisit [5], dan sarkasme [6]. Model multibahasa seperti XLM-RoBERTa juga terbukti kompetitif dalam menangani data dari berbagai bahasa sekaligus, termasuk Bahasa Indonesia (Yulianti et al., 2024). Meski demikian, sebagian besar studi sebelumnya masih menitikberatkan pada fine-tuning end-to-end dan belum mengkaji secara sistematis pendekatan feature-based, yaitu penggunaan representasi vektor dari model pralatih tanpa pelatihan ulang. Pendekatan ini lebih ringan secara komputasi dan fleksibel untuk berbagai pipeline klasifikasi. Studi Pal (2025) menunjukkan bahwa metode feature-based tetap mampu menghasilkan performa kompetitif dalam klasifikasi teks Bahasa Indonesia, terutama untuk tugas deteksi disinformasi[7] .

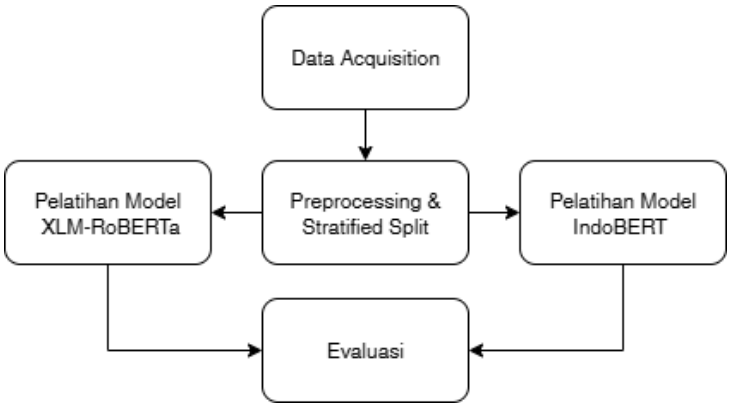
Namun, keterbatasan teknis model Transformer juga perlu dipertimbangkan. Batas maksimal input sepanjang 512 token menjadi hambatan utama dalam memproses pesan spam yang panjang. Truncation sebagai metode pemangkasan teks cenderung menghilangkan informasi penting yang sering muncul di akhir teks, seperti tautan atau ajakan. Sebagai solusi, teknik chunking berbasis stride yaitu pemecahan teks menjadi potongan yang saling tumpang tindih dapat mempertahankan konteks secara lebih utuh, meskipun menambah biaya komputasi [1]. Di sisi lain, metode pooling yang digunakan untuk merangkum representasi dokumen, seperti CLS pooling dan mean pooling, juga memiliki dampak terhadap kinerja akhir model. Beberapa studi mengindikasikan bahwa mean pooling dapat menghasilkan representasi yang lebih stabil pada teks panjang dengan distribusi semantik yang beragam [7]. Namun, belum ada kajian empiris yang membandingkan kedua metode ini secara sistematis dalam konteks deteksi spam Bahasa Indonesia.

Selain representasi semantik, spam juga sering mengandung sinyal eksplisit seperti simbol mata uang (Rp), URL, pola angka mencurigakan, atau frasa promosi. Informasi ini tidak selalu tertangkap oleh model Transformer murni. Untuk mengatasi hal ini, pendekatan late-fusion digunakan, yaitu menggabungkan fitur berbasis aturan (rule-based) dengan representasi vektor model pralatih. Pendekatan ini terbukti meningkatkan kinerja deteksi konten manipulatif dalam studi Latifah (2024), khususnya dalam konteks obfusikasi dan data berlabel terbatas [8]. Sebagian besar penelitian terdahulu hanya mengevaluasi satu aspek teknis seperti jenis model atau metode pemotongan teks secara terpisah. Padahal, integrasi berbagai strategi seperti pemilihan backbone model, pendekatan representasi input, metode pooling, dan penggabungan sinyal domain dapat memberikan hasil yang lebih holistik. Oleh karena itu, penelitian ini mengambil pendekatan evaluatif dan preskriptif yang menggabungkan seluruh komponen tersebut dalam satu kerangka kerja klasifikasi spam Bahasa Indonesia berbasis Transformer. Tujuannya tidak hanya untuk mengidentifikasi kombinasi konfigurasi terbaik, tetapi juga menghasilkan panduan yang dapat langsung diadopsi dalam pengembangan sistem deteksi spam yang andal dan efisien di lingkungan produksi..

2. METODOLOGI PENELITIAN

2.1 Tahapan Penelitian

Kami mengadopsi pipeline evaluasi yang lazim pada studi klasifikasi teks berbasis model bahasa pralatih. Alur dirancang dua cabang yang simetris IndoBERT dan XLM-RoBERTa untuk memastikan perbandingan yang adil di bawah tiga sumbu faktor yang menjadi fokus: pooling, penanganan panjang masukan (truncation vs sliding-window chunking dengan stride), serta late-fusion. Seluruh eksperimen menjaga tahap keputusan hilir tetap sama (konfigurasi dikunci), sehingga variasi kinerja merefleksikan kualitas representasi dan strategi input, bukan perbedaan pengklasifikasi.



Gambar 1. Flowchart tahapan penelitian

Alur pada Gambar 1 dimulai dari Data Acquisition, yaitu pemuatan korpus berbahasa Indonesia berlabel spam/ham dan pemeriksaan integritas. Tahap Preprocessing & Stratified Split melakukan pembersihan ringan, normalisasi label, deduplikasi, dan pembagian train/test dengan seed tetap. Dari titik ini proses bercabang: pada cab.

Building of Informatics, Technology and Science (BITS)
Volume 7, No 2, September 2025 Page: 1456−1466
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v7i2.8034

IndoBERT dan cab. XLM-RoBERTa, teks di-encode dalam mode feature-based (frozen) untuk menghasilkan document embedding. Di masing-masing cabang kami melakukan ablasi pooling (mean vs CLS) dan penanganan panjang masukan, membandingkan truncation dengan chunking berbasis sliding window pada stride 0/32/64; embedding antarpotongan kemudian diagregasi menjadi satu representasi dokumen. Selanjutnya, kami membentuk dua skema penyusunan fitur: embedding-only serta late-fusion (penggabungan embedding dengan fitur domain ringkas seperti indikator URL, token "Rp/IDR", pola nomor "08…", dan nominal bertitik yang telah dinormalisasi). Kedua cabang bermuara pada Evaluasi dengan protokol yang sama untuk semua kombinasi; hasil dilaporkan menggunakan F1-macro, recall kelas spam, dan AUPRC, serta indikator efisiensi (waktu pembentukan embedding dan latensi inferensi). Desain ini mengikuti praktik standar di literatur: satu pipeline, dua backbone, faktor input/representasi dikontrol eksplisit, sehingga efek pooling, stride, dan late-fusion dapat dipisahkan dan ditafsirkan dengan jelas.

2.2 Data Acquisition

Dataset yang digunakan berasal dari kaggle.com/datasets/bobsteward/dataset-sms-spam-indonesia. Korpus berisi pesan berbahasa Indonesia yang telah diberi label biner spam dan ham. Berkas diunduh dalam format tabular (CSV) dan dimuat apa adanya tanpa penambahan korpus eksternal agar evaluasi tetap terikat pada satu sumber data.

Prosedur pemuatan mengikuti praktik standar dalam studi klasifikasi teks berbasis model pralatih. Pertama, sistem memetakan kolom teks dan kolom label dari berkas sumber; skema kolom yang berbeda (misalnya label, kategori, class untuk label dan sms, pesan, text, message untuk teks) didukung dengan deteksi otomatis, dan dipastikan teridentifikasi secara konsisten sebelum melanjutkan. Praktik ini penting karena klasifikasi yang andal memerlukan penetapan kolom label yang eksplisit, dan penanganan variasi skema data secara fleksibel merupakan hal yang krusial dalam sistem klasifikasi berbasis transfer learning [9].

Agar nomenklatur seragam di seluruh eksperimen, seluruh variasi label pada berkas sumber dinormalisasi ke himpunan {spam, ham} (misalnya pemetaan dari non-spam, not spam, legit, 0/1, dan padanannya). Kami tidak melakukan pembersihan semantik agresif pada tahap ini (misalnya penghapusan tautan atau angka) karena informasi tersebut justru penting bagi tugas deteksi spam. Untuk menjaga transparansi, random seed ditetapkan sejak tahap pemuatan, dan sumber dataset (tautan Kaggle di atas) dicantumkan pada bagian Acknow-ledgement/References naskah.

2.3 Preprocessing & Stratified Split

Tahap ini menyiapkan korpus agar siap dievaluasi tanpa menghilangkan sinyal yang relevan untuk deteksi spam. Proses dimulai dengan normalisasi label ke himpunan {spam, ham}misalnya memetakan label seperti non-spam, legit, atau angka 0/1 ke dua kelas tersebut. Proses ini umum dilakukan dalam klasifikasi teks berbasis supervised learning untuk memastikan konsistensi antar eksperimen [10].Teks dibersihkan secara ringan dengan memangkas spasi di awal/akhir, menyatukan spasi ganda, dan menghapus karakter kontrol. Namun, informasi yang lazim menjadi indikator spam seperti URL, angka Rupiah, pola nomor ponsel lokal, huruf kapital, serta tanda baca tetap dipertahankan. Pendekatan ini sejalan dengan prinsip domain-aware preprocessing di mana fitur eksplisit yang relevan dibiarkan utuh agar tidak menghilangkan sinyal yang dapat dimanfaatkan oleh model.

Untuk mencegah data leakage, duplikasi dihapus berdasarkan isi pesan sebelum pembagian data dilakukan. Praktik deduplikasi global ini penting agar pesan yang identik tidak muncul di kedua himpunan train dan test, sebagaimana direkomendasikan dalam studi tentang efisiensi dan validitas klasifikasi berbasis teks klinis dan umum. Entri kosong atau sangat pendek dibuang karena tidak memberikan cukup sinyal semantik. Setelah pembersihan, disusun ringkasan korpus: ukuran total, proporsi kelas, serta distribusi panjang token terhadap max_seq_len (acuan 256 token). Estimasi proporsi pesan yang melebihi batas ini menjadi dasar keputusan untuk penggunaan teknik chunking pada tahap berikutnya [11].

Pembagian data dilakukan secara stratified agar proporsi spam/ham pada train dan test tetap konsisten. Skema yang digunakan adalah 80:20 (train:test) dengan random seed tetap untuk memastikan reprodusibilitas eksperimen, sebagaimana dipraktikkan dalam eksperimen klasifikasi berbasis pretrained model untuk menjamin evaluasi yang adil antar konfigurasi [12]. Indeks baris hasil split disimpan, dan himpunan uji tersebut digunakan konsisten di seluruh kombinasi konfigurasi (backbone, pooling, truncation/chunking–stride, embedding-only/late-fusion) guna memastikan keadilan perbandingan performa.

2.4 Pelatihan Model IndoBERT

Dalam penelitian ini, IndoBERT digunakan sebagai feature extractor dalam konfigurasi frozen (tanpa fine-tuning), bertujuan untuk menghasilkan representasi dokumen yang stabil. Pendekatan ini memungkinkan evaluasi adil terhadap pengaruh berbagai strategi input processing seperti teknik pooling, penanganan panjang teks (truncation vs. chunking), serta kontribusi late fusion tanpa bias akibat pelatihan end-to-end [13].

Tokenisasi dilakukan menggunakan kosakata dari model indobenchmark/indobert-base-p2, dengan panjang maksimum urutan tetap (misalnya max_seq_len = 256) untuk menjaga konsistensi antar eksperimen. Untuk teks pendek yang muat dalam batas ini, vektor dokumen diekstraksi dari hidden states menggunakan dua teknik utama: (i) mean pooling, yang merata-ratakan seluruh vektor token isi, dan (ii) CLS pooling, yang hanya mengambil vektor dari

Building of Informatics, Technology and Science (BITS)
Volume 7, No 2, September 2025 Page: 1456−1466
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v7i2.8034

token spesial [CLS] di awal urutan. Studi sebelumnya menunjukkan bahwa mean pooling cenderung memberikan ringkasan yang lebih stabil untuk teks pendek–menengah, sementara CLS pooling sering kali mengandung informasi ringkasan yang dipelajari secara eksplisit oleh model [14].

Untuk menangani pesan panjang, dua strategi diuji. Truncation memangkas teks di luar batas token maksimum, yang umum digunakan karena efisiensi komputasi [13]. Alternatifnya, chunking dengan sliding window mempertahankan konteks penuh dengan membagi teks menjadi segmen tumpang tindih menggunakan variasi stride (misalnya 0, 32, dan 64). Tiap segmen diolah oleh IndoBERT, lalu diringkas menggunakan teknik pooling yang sama, kemudian diaggregasi menjadi satu vektor dokumen dalam studi ini dengan average over segments.

2.5 Pelatihan Model XLM-RoBERTa (Feature-Based)

XLM-RoBERTa juga dimanfaatkan sebagai feature extractor dalam mode dibekukan (frozen) tanpa fine-tuning, untuk memastikan bahwa perbedaan hasil sepenuhnya berasal dari desain masukan dan bukan dari pembaruan bobot model. Rancangan pipeline-nya diselaraskan dengan cabang IndoBERT agar valid untuk perbandingan langsung, mencakup skema tokenisasi, batas panjang urutan tetap (max_seq_len = 256), teknik pooling, serta strategi penanganan teks panjang dan penggabungan fitur domain.

Teks awal ditokenisasi menggunakan kosakata XLM-RoBERTa, lalu diolah untuk memperoleh hidden states. Dua metode peringkasan dievaluasi: (i) mean pooling, yang menghitung rata-rata vektor token isi, dan (ii) CLS pooling, yang mengambil representasi dari token khusus [CLS] pada awal urutan. Perbedaan kedua metode ini penting karena CLS pooling lebih dipengaruhi oleh pembelajaran global model, sementara mean pooling lebih stabil pada teks pendek-menengah [15]. Untuk teks yang melampaui panjang maksimum, dua pendekatan dibandingkan: truncation sebagai baseline yang efisien namun berisiko kehilangan konteks akhir, dan sliding-window chunking dengan stride 0, 32, dan 64 untuk mempertahankan konteks penuh. Setiap segmen hasil chunking diproses secara terpisah lalu diringkas menggunakan metode pooling yang sama. Embedding dari segmen-segmen ini kemudian diagregasi menggunakan rata-rata antarsegmen untuk menghasilkan satu document embedding per pesan [11].

Setelah embedding dokumen diperoleh, dua skema input untuk klasifikasi disiapkan. Pertama, pada skema embedding-only, vektor dokumen digunakan secara langsung tanpa modifikasi. Kedua, pada skema late-fusion, vektor embedding dikombinasikan dengan domain-specific features yang relevan untuk spam berbahasa Indonesia, seperti: keberadaan atau jumlah URL, token mata uang seperti “Rp” atau “IDR”, pola nomor telepon lokal berawalan “08…”, angka bertitik (misalnya “1.000.000”), serta ciri struktural seperti panjang teks, rasio huruf kapital, digit, dan tanda baca. Fitur domain ini dinormalisasi terlebih dahulu agar setara secara skala sebelum digabungkan, sesuai prinsip feature scaling dalam sistem gabungan [16].

Akhirnya, tahap klasifikasi atau keputusan hilir dijaga tetap konstan: baik arsitektur, parameter, maupun anggaran pelatihan tidak berubah antar percobaan. Ini dilakukan untuk memastikan bahwa setiap perbedaan kinerja dapat dikaitkan dengan kualitas representasi yang dihasilkan oleh XLM-RoBERTa di bawah variasi pooling, strategi panjang (truncation vs chunking), dan metode penggabungan fitur domain, bukan karena pengaruh pengklasifikasi itu sendiri. Seluruh proses juga mencatat waktu pembentukan embedding dan latensi inferensi per dokumen sebagai bagian dari evaluasi efisiensi, yang hasilnya akan dibahas pada bagian evaluasi. Output tahap ini berupa dua matriks: document embedding untuk skema embedding-only dan gabungan embedding-fitur untuk late-fusion—seluruhnya siap untuk tahap evaluasi kuantitatif terstandar.

2.6 Metrik Evaluasi

Performa sistem klasifikasi spam dilakukan secara menyeluruh dengan menggabungkan metrik kualitas dan efisiensi. Untuk aspek kualitas, digunakan metrik-metrik berbasis probabilitas dan konfusi biner. Model menghasilkan skor kontinu:

$$s(x_i) \in [0,1] \tag{1}$$

Bagi setiap dokumen x_i, yang ditafsirkan sebagai probabilitas kelas positif (spam). Skor ini merefleksikan keyakinan model; nilai mendekati 1 menandakan spam, mendekati 0 menandakan ham. Jika diperlukan interpretasi probabilistik yang lebih akurat, kalibrasi (mis. Platt/temperature scaling) dapat diterapkan tanpa mengubah prosedur evaluasi. Prediksi kelas diperoleh dengan menerapkan ambang τ, secara default $\tau = 0.5$, untuk membentuk label prediksi:

$$\hat{y}_i(\tau) = I![\, s(x_i) \geq \tau\,] \tag{2}$$

Di sini I[·] (ditulis sebagai $I!$ pada Pers. (2)) adalah fungsi indikator yang bernilai 1 bila kondisi di dalamnya benar dan 0 bila salah. Nilai τ dapat disetel (mis. untuk memaksimalkan F1 atau memenuhi batas minimal recall). Untuk setiap τ, dari pasangan $(yi(\tau), yi)$ dihitung komponen matriks konfusi: TP, FP, FN, TN.

Evaluasi berbasis threshold ini melibatkan perhitungan TP, FP, FN, dan TN, dari mana Precision, Recall, dan F1-score untuk masing-masing kelas dapat dihitung:

$$[\text{Precision} = \frac{TP}{TP+FP},\ \text{Recall} = \frac{TP}{TP+FN},\ F1 = \frac{2\cdot\text{Precision}\cdot\text{Recall}}{\text{Precision}+\text{Recall}}] \tag{3}$$

Bila penyebut nol (mis. TP+FP=0 atau TP+FN=0), metrik terkait didefinisikan 0 untuk stabilitas. Pada skenario biner, metrik dapat dihitung per kelas dengan memperlakukan masing-masing kelas sebagai “positif” secara bergantian

Building of Informatics, Technology and Science (BITS)
Volume 7, No 2, September 2025 Page: 1456−1466
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v7i2.8034

Dalam kondisi data miring kelas, metrik macro-F1 digunakan sebagai agregasi global karena memperlakukan setiap kelas secara seimbang tanpa dipengaruhi ukuran dominan:

$$\text{Macro-F1} = \frac{1}{K}\sum_{k=1}^{K} F1_k \quad (4)$$

Untuk kasus biner, K=2 dan $F1_k$ adalah F1 ketika kelas k diposisikan sebagai positif. Berbeda dari micro-average (yang menggabungkan TP/FP/FN lintas kelas dan dapat bias ke kelas mayoritas), macro-F1 memberi bobot sama per kelas sehingga lebih adil pada data tak seimbang [17].

Selain itu, Recall untuk kelas spam dilaporkan secara eksplisit sebagai indikator sensitivitas sistem terhadap kasus positif. Hal ini penting karena spam classification termasuk kategori high-risk classification, di mana kesalahan deteksi spam (FN) dapat menyebabkan kerugian signifikan bagi pengguna akhir. Dalam konteks ini, Recall menjadi komponen krusial yang tak dapat diabaikan [18].

Untuk mengevaluasi performa prediksi sepanjang seluruh rentang ambang, digunakan Area Under the Precision–Recall Curve (AUPRC). Estimasi numerik AUPRC diperoleh melalui pendekatan Average Precision (AP):

$$AP = \sum_n (R_n - R_{n-1}) P_n \quad (5)$$

dengan P_n dan R_n masing-masing presisi dan recall pada threshold ke-n. AUPRC dipilih alih-alih AUROC karena lebih informatif pada dataset dengan distribusi kelas tidak seimbang [19].

Dari sisi efisiensi, dua metrik utama digunakan. Pertama, waktu pembentukan embedding (T_{embed}) yang mengukur total waktu untuk menghasilkan representasi dokumen dari seluruh himpunan uji. Kedua, latensi inferensi per dokumen (L_{inf})dihitung sebagai rata-rata waktu prediksi per sampel individual:

$$L_{\text{inf}} = \frac{T_{\text{total\_inf}}}{N} \quad (6)$$

dengan N jumlah dokumen pada himpunan uji. Di samping itu, throughput sistem Θ dihitung sebagai:

$$\Theta = \frac{N}{T_{\text{embed}} + T_{\text{inf}}} \quad (7)$$

yang merepresentasikan jumlah dokumen yang dapat diproses per detik. Evaluasi kinerja model dalam penelitian ini dilakukan dengan mempertimbangkan keseimbangan antara efektivitas klasifikasi dan efisiensi komputasi, agar sistem layak digunakan dalam lingkungan nyata dengan batasan sumber daya [20].

3. HASIL DAN PEMBAHASAN

3.1 IndoBERT (feature-based)

Bagian ini menyajikan hasil model IndoBERT ketika digunakan sebagai feature extractor yang dibekukan. Seluruh pengujian menggunakan himpunan uji yang sama dan decision layer yang dikunci agar variasi kinerja merefleksikan pengaruh rancangan representasi dan strategi masukan. Tiga aspek dievaluasi secara berurutan: (i) mekanisme peringkasan token pada level dokumen (mean vs CLS), (ii) penanganan panjang masukan melalui truncation dibanding sliding-window chunking dengan variasi stride, dan (iii) skema penyusunan fitur embedding-only versus late-fusion yang menggabungkan embedding dengan fitur domain. Setiap subbagian melaporkan F1-macro, recall kelas spam, dan AUPRC, serta indikator efisiensi pembuatan embedding. Konfigurasi terbaik pada satu aspek digunakan sebagai dasar pengujian aspek berikutnya.

3.1.1 Ablasi Pooling

Untuk menilai pengaruh mekanisme peringkasan token terhadap representasi dokumen IndoBERT, kami menguji mean pooling dan CLS pooling pada kondisi terkendali: masukan diproses dengan truncation (tanpa chunking), skema fitur embedding-only, dan stride diatur 0 (tidak relevan pada truncation). Konfigurasi tahap keputusan (downstream) dikunci agar variasi kinerja semata-mata merefleksikan efek pooling. Hasil ablasi Poolinh model IndoBERT dijelaskan dalam Tabel 1.

Tabel 1. Ablasi pooling IndoBERT

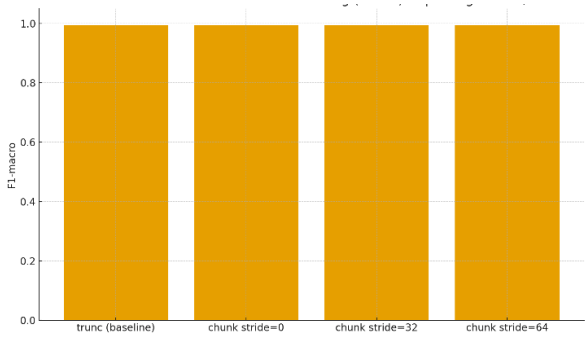
Pooling	F1-macro	Recall (spam)	AUPRC (spam)	Embed time (s)	Train time (s)
Mean	0.994	1.000	0.9993	7.289	0.230
CLS	0.988	0.970	0.9998	7.245	0.383

Pada Tabel 1 dijelaskan mean pooling memberikan F1-macro dan *recall* kelas *spam* yang lebih tinggi dibanding CLS pooling dalam kondisi terkendali, dengan waktu pembentukan *embedding* yang setara (≈7,2–7,3 s). Meskipun AUPRC keduanya sangat tinggi (≈0,999), *recall* yang unggul pada *mean pooling* mengindikasikan sensitivitas yang lebih baik terhadap kelas positif pada ambang baku. Hasil ini mendukung pemilihan *mean pooling* sebagai default untuk IndoBERT ketika masukan tidak melebihi panjang urutan maksimum.

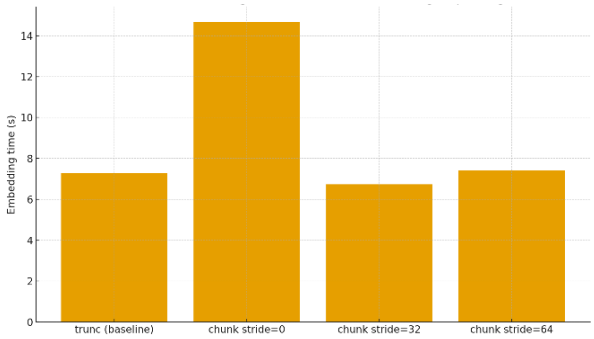
Building of Informatics, Technology and Science (BITS)
Volume 7, No 2, September 2025 Page: 1456−1466
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v7i2.8034

3.1.2 Truncation vs chunking (stride)

Perbandingan dilakukan antara truncation dan sliding-window chunking dengan stride = {0, 32, 64}. Faktor lain dikendalikan tetap untuk memastikan atribusi yang tepat: pooling = mean, skema fitur = embedding-only, dan decision layer dikunci. Untuk chunking, setiap segmen dienkode dan diringkas pada level token, kemudian diagregasi dengan perataan antar-segmen menjadi satu document embedding. F1-macro untuk truncation dan tiga varian chunking ditampilkan dalam Gambar 2. dan waktu pembentukan embedding pada konfigurasi yang sama ditampilkan dalam Gambar 3.



Gambar 2. F1-macro IndoBERT: Truncation vs Chunking

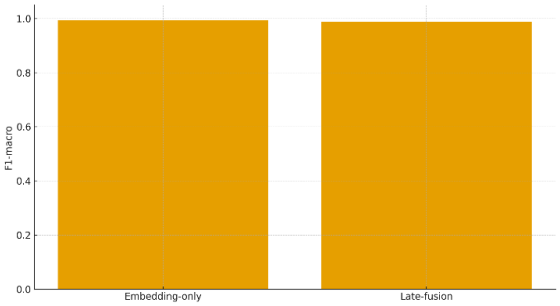


Gambar 3. Waktu Pembentukan Embedding (s): Truncation vs Chunking

Hasil pada Gambar 2 dan Gambar 3 menunjukkan kualitas prediksi identik pada semua konfigurasi: F1-macro = 0,994, recall kelas spam = 1,000, dan AUPRC ≈ 0,9993 baik untuk truncation maupun chunking dengan stride 0/32/64. Dengan demikian, pada korpus ini chunking tidak memberikan keuntungan akurasi terhadap baseline truncation ketika pooling=mean dan embedding-only digunakan. Dari sisi efisiensi (Gambar 9), chunking dengan stride=0 menimbulkan overhead tertinggi (~14,7 s) karena jumlah segmen lebih banyak; stride=32/64 mendekati baseline truncation (~6,7–7,4 s). Temuan ini menyiratkan bahwa, selama mayoritas pesan tidak melampaui max_seq_len, truncation sudah memadai untuk IndoBERT pada konfigurasi ini, sedangkan chunking baru relevan ketika proporsi teks panjang meningkat atau ketika informasi kritis sering muncul di bagian yang akan terpotong.

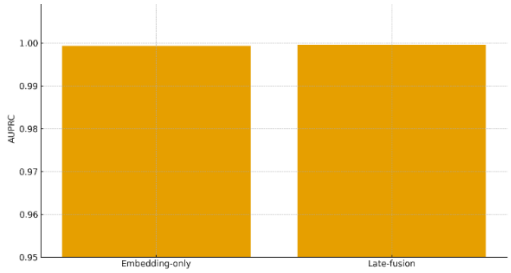
3.1.3 Embedding-only vs late-fusion

Perbandingan dilakukan antara embedding-only dan late-fusion pada setelan dasar yang sudah ditetapkan: pooling = mean, strategi panjang = truncation, stride = 0, dan decision layer dikunci. Pada late-fusion, document embedding digabung (concatenate) dengan fitur domain terstandardisasi (indikator URL, token “Rp/IDR”, pola nomor “08…”, nominal bertitik, panjang/rasio kapitalisasi, digit, tanda baca). F1-macro untuk kedua skema disajikan pada Gambar 4, sedangkan AUPRC (kelas spam) disajikan pada Gambar 5.



Gambar 4. F1-macro IndoBERT, Embedding-only vs Late-fusion

Building of Informatics, Technology and Science (BITS)
Volume 7, No 2, September 2025 Page: 1456−1466
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v7i2.8034



Gambar 5. AUPRC (kelas spam) — Embedding-only vs Late-fusion

Gambar 4 menampilkan bahwa embedding-only mencapai F1-macro 0,994, sementara late-fusion mencapai 0,988 dan Gambar 5 menampilkan AUPRC yang sangat tinggi pada kedua skema—0,9993 (embedding-only) dan 0,9996 (late-fusion). Secara keseluruhan, late-fusion memberi kenaikan AUPRC yang sangat kecil, tetapi tidak meningkatkan F1/recall dibanding embedding-only. Pada korpus ini, representasi IndoBERT sudah cukup menangkap sinyal domain, sehingga embedding-only merupakan pilihan yang lebih sederhana dan efektif pada konfigurasi terbaiknya.

3.2 XLM-RoBERTa (feature-based)

Bagian ini menyajikan hasil model XLM-RoBERTa ketika digunakan sebagai feature extractor yang dibekukan. Protokol uji dibuat identik dengan IndoBERT: himpunan uji yang sama, decision layer dikunci, serta urutan evaluasi yang sama agar perbedaan kinerja dapat diatribusikan pada rancangan representasi dan strategi masukan. Seperti pada 3.1, analisis disusun berlapis: pertama mengevaluasi pooling (mean vs CLS), kemudian penanganan panjang masukan (truncation vs sliding-window chunking dengan variasi stride), dan terakhir penyusunan fitur (embedding-only vs late-fusion). Konfigurasi yang terbaik pada satu langkah digunakan sebagai dasar untuk langkah berikutnya.

3.2.1 Ablasi Pooling XLM-RoBERTa

Pengujian ini menilai pengaruh mekanisme peringkasan token terhadap kualitas representasi XLM-RoBERTa. Dua opsi dievaluasi, mean pooling dan CLS pooling, pada kondisi terkendali: truncation (tanpa chunking), embedding-only, dan stride = 0; decision layer dijaga tetap. F1-macro untuk kedua opsi ditampilkan pada Tabel 2.

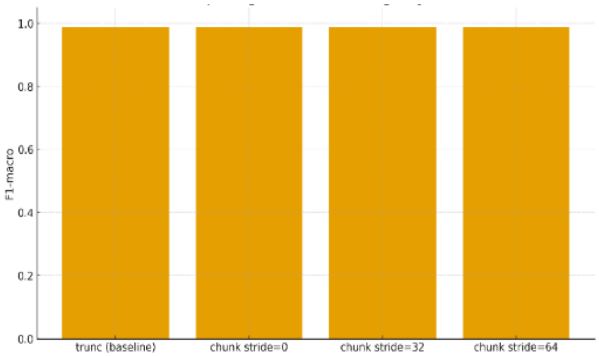
Tabel 2. Ringkasan ablasi pooling XLM-RoBERTa

Pooling	F1-macro	Recall (spam)	AUPRC (spam)	Embed time (s)	Train time (s)
mean	0.988	0.985	0.9996	7.012	1.571
CLS	0.982	0.970	0.9989	7.062	1.101

Pada Tabel 2 dijelaskan mean pooling memberikan F1-macro dan recall kelas spam yang lebih tinggi dibanding CLS pooling dalam kondisi terkendali, sementara AUPRC keduanya sangat tinggi (>0.998). Waktu pembentukan embedding hampir identik (~7,0–7,1 s). Hasil ini menunjukkan bahwa, untuk XLM-RoBERTa pada skenario truncation + embedding-only, mean pooling lebih disarankan sebagai pilihan default.

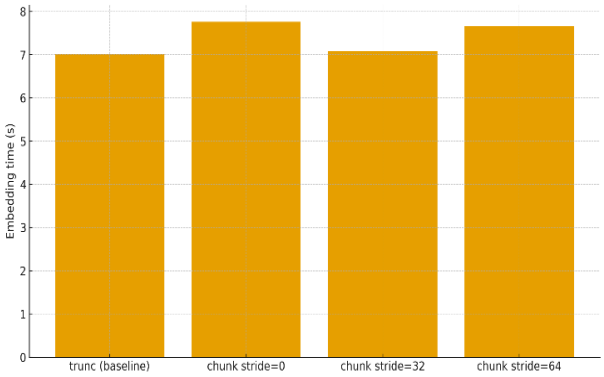
3.2.2 Truncation vs chunking (stride)

Perbandingan dilakukan antara truncation dan sliding-window chunking dengan stride = {0, 32, 64} untuk XLM-RoBERTa. Faktor lain dikendalikan tetap: pooling = mean, skema fitur = embedding-only, dan decision layer dikunci. Pada chunking, setiap segmen diproses dan diringkas pada level token, kemudian diagregasi (perataan antarsegmen) menjadi satu document embedding. F1-macro untuk truncation dan tiga varian chunking disajikan pada Gambar 7. Waktu pembentukan embedding pada konfigurasi yang sama disajikan pada Gambar 8.



Gambar 7. F1-macro XLM-RoBERTa: Truncation vs Chunking

Building of Informatics, Technology and Science (BITS)
Volume 7, No 2, September 2025 Page: 1456−1466
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v7i2.8034

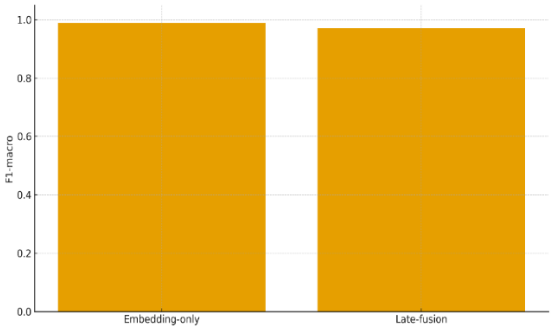


Gambar 8. Waktu Pembentukan Embedding (s): XLM-RoBERTa Truncation vs Chunking

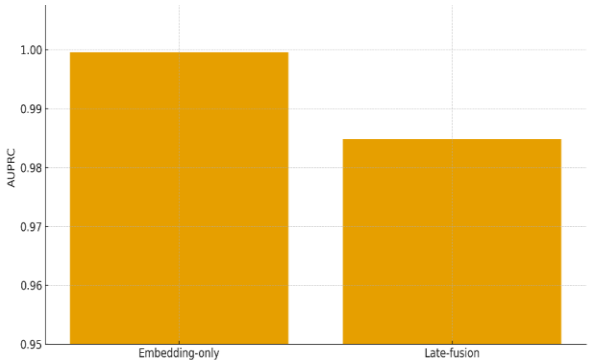
Gambar 7 menampilkan F1-macro yang tetap pada semua konfigurasi (≈0,988) sementara Gambar 8 memperlihatkan waktu pembentukan embedding berada pada rentang ~7,0–7,8 s, dengan *chunking* stride=0 sebagai yang paling mahal. Kualitas prediksi identik antara truncation dan chunking (F1-macro ≈ 0,988, recall spam ≈ 0,985, AUPRC ≈ 0,9996). Dari sisi efisiensi, chunking menambah overhead moderat terutama pada stride=0, sedangkan stride=32 sangat dekat dengan baseline truncation. Temuan ini menyiratkan bahwa, pada korpus dan setelan ini, truncation sudah memadai untuk XLM-RoBERTa; chunking baru relevan bila proporsi teks yang melampaui max_seq_len tinggi atau informasi kritis sering muncul di bagian yang akan terpotong.

3.2.3 Embedding-only vs late-fusion

Perbandingan dilakukan antara em1bedding-only dan late-fusion pada setelan dasar: pooling = mean, strategi panjang = truncation, stride = 0, dan decision layer dikunci. Pada late-fusion, document embedding XLM-RoBERTa digabung (concatenate) dengan fitur domain terstandardisasi (indikator URL, token “Rp/IDR”, pola nomor “08…”, nominal bertitik, panjang/rasio kapitalisasi, digit, tanda baca). F1-macro untuk kedua skema disajikan pada Gambar 9, sedangkan AUPRC (kelas spam) disajikan pada Gambar 10.



Gambar 9. F1-macro XLM-RoBERTa — Embedding-only vs Late-fusion



Gambar 10. AUPRC (kelas spam) — Embedding-only vs Late-fusion

Pada konfigurasi dasar XLM-RoBERTa, embedding-only unggul pada F1-macro dan recall kelas spam, sekaligus mempertahankan AUPRC yang lebih tinggi daripada late-fusion. Hal ini mengindikasikan bahwa representasi XLM-RoBERTa sudah memadai dalam menangkap sinyal domain yang relevan; penambahan fitur buatan justru tidak meningkatkan, bahkan menurunkan, kinerja pada metrik ambang tetap. Mengingat kompleksitas ekstra yang dibawa oleh rekayasa fitur, embedding-only merupakan pilihan yang lebih sederhana dan efektif untuk XLM-RoBERTa pada setelan terbaik ini.

Building of Informatics, Technology and Science (BITS)
Volume 7, No 2, September 2025 Page: 1456−1466
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v7i2.8034

3.3 Perbandingan Hasil Terbaik IndoBERT dan XLM-RoBERTa

Bagian ini membandingkan performa terbaik dua backbone, yakni IndoBERT (indobenchmark/indobert-base-p2) dan XLM-RoBERTa (xlm-roberta-base), pada himpunan uji dengan konfigurasi lain dikendalikan tetap (skema fitur, pooling, dan mekanisme keputusan). Evaluasi berfokus pada F1-Macro, Precision/Recall kelas spam, AUPRC kelas spam, serta efisiensi berupa waktu pembentukan embedding dan waktu pelatihan; ringkasan hasilnya disajikan pada Tabel 3.

Tabel 3. Perbandingan hasil terbaik untuk model indobert dan xlmroberta pada himpunan uji.

Model	Mode	Stride	Pooling	Fusi	F1-Macro	Precision (spam)	Recall (spam)	AUPRC (spam)	Waktu embed (s)	Waktu latih (s)
indobert	trunc	64	mean	emb_only	0.9941	0.9853	1.0000	0.9993	6.71	0.37
xlmroberta	trunc	0	mean	emb_only	0.9882	0.9851	0.9851	0.9996	7.01	1.57

Sebagaimana terlihat pada Tabel 3, IndoBERT menunjukkan kualitas keseluruhan yang lebih baik dengan F1-Macro 0,9941, Recall (spam) 1,0000, dan Precision (spam) 0,9853, sementara XLM-RoBERTa meraih F1-Macro 0,9882, Recall (spam) 0,9851, dan Precision (spam) 0,9851. Dari sisi pemeringkatan probabilitas, XLM-RoBERTa sedikit unggul pada AUPRC (≈0,9996 vs 0,9993), namun IndoBERT lebih efisien dengan waktu embed ≈ 6,71 s (vs 7,01 s) dan waktu latih ≈ 0,37 s (vs 1,57 s). Konfigurasi terbaik keduanya konsisten pada mode=truncation dan pooling=mean dengan fusion=emb_only; nilai stride yang tercantum tidak berpengaruh karena parameter tersebut tidak aktif pada mode truncation.

3.4 Pembahasan

Penilaian terhadap backbone model menunjukkan bahwa IndoBERT memiliki keunggulan marginal atas XLM-RoBERTa dalam metrik F1-Macro dan recall spam, yang konsisten dengan temuan oleh Koto et al. (2020) bahwa model monolingual seperti IndoBERT menunjukkan performa unggul pada berbagai tugas IndoLEM dibandingkan model multibahasa seperti mBERT atau MALAYBERT [21]. Hal ini menegaskan kecocokan linguistik yang lebih baik dari model pralatih monolingual terhadap struktur dan distribusi Bahasa Indonesia dalam korpus . Di sisi lain, XLM-RoBERTa sedikit unggul dalam AUPRC, mengindikasikan kemampuan pemeringkatan probabilitas yang lebih halus sejalan dengan hasil oleh Sirusstara et al. (2022), yang menunjukkan efektivitas XLM-R dalam memproses teks campuran dan tugas-tugas berisiko tinggi seperti deteksi konten clickbait, terutama ketika interpretasi kontinu skor menjadi penting dalam sistem rekomendasi atau filter berita otomatis [22] .

Dari segi efisiensi komputasi, IndoBERT terbukti lebih ringan pada pembentukan embedding dan latensi inferensi. Keunggulan ini penting dalam konteks aplikasi berlatensi rendah, seperti sistem filter real-time, sebagaimana juga disarankan oleh Latifah et al. (2024) yang menunjukkan bahwa IndoBERT cukup efisien bahkan dalam tugas klasifikasi multiclass setelah augmentasi data berbasis Easy Data Augmentation (EDA), sekaligus menunjukkan skalabilitas yang baik untuk aplikasi industri [8].

Eksperimen representasi menunjukkan bahwa strategi truncation tetap optimal untuk korpus SMS/email pendek. Temuan ini konsisten dengan distribusi panjang token yang cenderung pendek, sehingga chunking tidak menambah nilai informasi dan justru memperbesar variansi, seperti dilaporkan pula oleh Ridho & Yulianti (2024), di mana kombinasi embedding IndoBERT dan arsitektur BiLSTM tetap unggul pada teks ringkas dibandingkan pendekatan segmentasi kompleks yang sering kali membebani tanpa manfaat yang signifikan pada teks pendek .

Terkait metode pooling dan fusion, pooling rata-rata (mean) menghasilkan stabilitas lebih baik dibandingkan CLS atau max pooling, sementara late-fusion dengan domain features memberikan keuntungan marginal. Fenomena ini juga tercermin dalam studi Muftie & Haris (2023), yang menekankan efektivitas representasi dense transformer tanpa tambahan fitur luar untuk teks pendek Bahasa Indonesia, serta mencatat bahwa representasi semantik yang dihasilkan secara end-to-end oleh model seperti IndoBERT sudah cukup mengakomodasi sinyal-sinyal sintaktik yang umum ditemukan dalam spam pendek [23].

Jika dibandingkan dengan penelitian terdahulu, seperti studi Syahputra et al. (2023) pada deteksi clickbait [24] dan Ridho & Yulianti (2024) pada hoaks [24], capaian F1-Macro pada penelitian ini yang mendekati satuan menempatkan sistem dalam kategori performa sangat tinggi, bahkan dengan skema tugas yang lebih sederhana (biner). Ini menggarisbawahi relevansi IndoBERT dalam berbagai domain spam/abusif di Indonesia, serta konsistensi model ini melintasi task dan jenis input yang bervariasi .

Augmentasi data juga menjadi faktor penting. Latifah et al. (2024) menunjukkan bahwa Easy Data Augmentation (EDA) berhasil meningkatkan F1 minoritas dalam klasifikasi multiclass SMS spam hingga 12% [8]. Hal ini memperkuat temuan bahwa augmentasi sederhana pun dapat berdampak signifikan terutama pada distribusi tidak seimbang, relevan untuk konteks AUPRC dan false negative, karena noise terkontrol dalam augmentasi dapat memperkuat generalisasi model terhadap variasi gaya bahasa spam .

Akhirnya, tantangan seperti code-switching, obfusikasi karakter, dan implikatur pragmatis tetap menjadi sumber error diperlukan strategi berlapis seperti thresholding adaptif dan human-in-the-loop. Studi oleh Hidayatullah et al. (2024) menyarankan pendekatan berbasis pemutakhiran data berkala dan integrasi sinyal adversarial-aware

Building of Informatics, Technology and Science (BITS)
Volume 7, No 2, September 2025 Page: 1456−1466
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v7i2.8034

sebagai langkah preventif terhadap serangan berbasis bahasa implisit yang sering kali lolos dari filter deterministik berbasis kata kunci[25].

4. KESIMPULAN

Penelitian ini menegaskan bahwa pendekatan feature-based dengan backbone pralatih mampu memberikan kinerja yang sangat tinggi sekaligus efisien untuk klasifikasi spam berbahasa Indonesia. Hasil pengujian menunjukkan bahwa model monolingual seperti IndoBERT lebih sesuai untuk konteks lokal dibandingkan model multibahasa, terutama dalam pengambilan keputusan biner, sementara XLM-RoBERTa tetap relevan pada skenario yang menekankan ranking probabilistik dan teks campuran. Dari sisi strategi representasi, truncation terbukti sudah cukup untuk menangani korpus dengan dominasi pesan pendek, sehingga penggunaan chunking hanya layak dipertimbangkan jika terdapat proporsi pesan panjang atau obfusikasi tinggi. Mean pooling lebih stabil dibandingkan alternatif lain, memberikan ringkasan dokumen yang konsisten, sementara late fusion dengan fitur domain hanya memberikan keuntungan terbatas sehingga tidak selalu diperlukan. Temuan ini memiliki implikasi praktis penting yaitu sistem produksi untuk penapisan spam di Indonesia dapat mengadopsi konfigurasi sederhana namun efektif tanpa menanggung biaya komputasi tambahan yang besar. Dengan demikian, penelitian ini bukan hanya bersifat deskriptif, tetapi juga preskriptif, memberikan rekomendasi konfigurasi yang seimbang antara akurasi dan efisiensi. Adapun keterbatasan studi ini mencakup penggunaan korpus tunggal dengan dominasi pesan pendek, belum dieksplorasinya strategi fine tuning parameter efisien, serta belum adanya evaluasi terhadap robustness pada obfusikasi atau serangan adversarial. Hal ini membuka ruang penelitian lanjutan, khususnya pada aspek kalibrasi probabilitas lintas domain, thresholding adaptif, dan integrasi strategi pertahanan terhadap variasi spam yang lebih kompleks.

REFERENCES

[1] V. Tandra, Y. Yowen, R. Tanjaya, W. L. Santoso, and N. N. Qomariyah, “Short Message Service Filtering With Natural Language Processing in Indonesian Language,” in 2021 Int. Conf. on ICT for Smart Society (ICISS), 2021, pp. 1–7, doi: 10.1109/ICISS53185.2021.9532503.

[2] L. Alhaura and I. Budi, “Malicious Account Detection on Indonesian Twitter Account,” in 2020 3rd Int. Conf. on Computer and Informatics Engineering (IC2IE), 2020, pp. 176–181, doi: 10.1109/IC2IE50715.2020.9274682.

[3] Y. Vernanda, S. Hansun, and M. B. Kristanda, “Indonesian Language Email Spam Detection Using N-Gram and Naïve Bayes Algorithm,” Bulletin of Electrical Engineering and Informatics, vol. 9, no. 5, pp. 2012–2019, 2020, doi: 10.11591/eei.v9i5.2444.

[4] F. Astari and Devianty, “Transfer Learning Methods for Hate Speech Detection in Bahasa Indonesia,” J. Inf. Technol. Comput. Sci., 2025, doi: 10.25126/jitecs.2025101542.

[5] A. F. Hidayatullah, R. Apong, D. T. C. Lai, and A. Qazi, “Adult Content Detection on Indonesian Tweets by Fine-Tuning Transformer-Based Models,” in 2023 6th Int. Conf. on Applied Computer and Information Systems (ACIIS), 2023, pp. 1–6, doi: 10.1109/ACIIS59385.2023.10367283.

[6] R. A. Fitrianto, A. Editya, M. M. Alamin, A. L. Pramana, and A. K. Alhaq, “Classification of Indonesian Sarcasm Tweets on X Platform Using Deep Learning,” in 2024 7th Int. Conf. on Informatics and Computational Sciences (ICICOS), 2024, pp. 388–393, doi: 10.1109/icicos62600.2024.10636904.

[7] A. A. Pal, S. Mondal, C. Kumar, and C. Kumar, “A Transformer-Based Approach for Fake News and Spam Detection in Social Media Using RoBERTa,” in 2025 Int. Conf. on Multi-Agent Systems and Collaborative Intelligence (ICMSCI), 2025, pp. 1256–1263, doi: 10.1109/ICMSCI62561.2025.10894342.

[8] N. Latifah, R. Dwiyansaputra, and G. S. Nugraha, “Multiclass Text Classification of Indonesian Short Message Service (SMS) Spam Using Deep Learning Method and Easy Data Augmentation,” MATRIK: J. Manajemen, Teknik Informatika dan Rekayasa Komputer, vol. 23, no. 3, pp. 663–676, 2024, doi: 10.30812/matrik.v23i3.3835.

[9] L. Yelamanchili, C.-S. M. Wu, C. Pollett, and R. Chun, “Multi-Label Text Classification With Transfer Learning,” in 2024 IEEE/ACIS 9th Int. Conf. on Big Data, Cloud Computing, and Data Science (BCD), 2024, pp. 21–26, doi: 10.1109/BCD61269.2024.10743077.

[10] M. Benballa, S. Collet, and R. Picot-Clémente, “Saagie at SemEval-2019 Task 5: From Universal Text Embeddings and Classical Features to Domain-Specific Text Classification,” in Proc. 13th Int. Workshop on Semantic Evaluation (SemEval 2019), Minneapolis, MN, USA, Jun. 6–7, 2019, pp. 469–475, doi: 10.18653/v1/S19-2083.

[11] M. Z. Koufi, Z. Guessoum, A. Keziou, I. Yahiaoui, C. Martineau, and W. Domin, “Text Chunking to Improve Website Classification,” 2023, pp. 197–216, doi: 10.1007/978-3-031-53025-8_15.

[12] O. Coban, M. Yağanoğlu, and F. Bozkurt, “Domain Effect Investigation for BERT Models Fine-Tuned on Different Text Categorization Tasks,” Arab. J. Sci. Eng., 2023, doi: 10.1007/s13369-023-08142-8.

[13] F. Indriani, R. A. Nugroho, M. Faisal, and D. Kartini, “Comparative Evaluation of IndoBERT, IndoBERTweet, and mBERT for Multilabel Student Feedback Classification,” J. RESTI (Rekayasa Sistem dan Teknologi Informasi), 2024, doi: 10.29207/resti.v8i6.6100.

[14] T. I. Z. M. Putra, S. Suprapto, and A. F. Bukhori, “Model Klasifikasi Berbasis Multiclass Classification Dengan Kombinasi IndoBERT Embedding dan Long Short-Term Memory Untuk Tweet Berbahasa Indonesia,” J. Ilmu Siber dan Teknologi Digital, 2022, doi: 10.35912/jisted.v1i1.1509.

[15] Z. Chi et al., “XLM-E: Cross-Lingual Language Model Pre-Training via ELECTRA,” 2021, pp. 6170–6182, doi: 10.18653/v1/2022.acl-long.427.

[16] G. Subhashini, M. G., H. Ashik, and B. Duresh, “Advanced SMS Spam Detection Using Integrated Feature Extraction,” in 2024 4th Int. Conf. on Ubiquitous Computing, Intelligence and Information Systems (ICUIS), 2024, pp. 780–785, doi:

Building of Informatics, Technology and Science (BITS)
Volume 7, No 2, September 2025 Page: 1456−1466
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v7i2.8034

10.1109/ICUIS64676.2024.10867034.
[17] S. Riyanto, I. Sitanggang, T. Djatna, and T. Atikah, "Comparative Analysis Using Various Performance Metrics in Imbalanced Data for Multi-Class Text Classification," Int. J. Adv. Comput. Sci. Appl. (IJACSA), 2023, doi: 10.14569/ijacsa.2023.01406116.
[18] J. Opitz, "A Simple but Thorough Primer on Metrics for Multi-Class Evaluation Such as Micro F1, Macro F1, Kappa and MCC," 2022. [Online]. Available: https://consensus.app/papers/a-simple-but-thorough-primer-on-metrics-for-multiclass-opitz/33ab5081d43a5db6bb29ff8bade3b77d/
[19] J. Leevy, T. Khoshgoftaar, and J. Hancock, "Evaluating Performance Metrics for Credit Card Fraud Classification," in 2022 IEEE 34th Int. Conf. on Tools With Artificial Intelligence (ICTAI), 2022, pp. 1336–1341, doi: 10.1109/ICTAI56018.2022.00202.
[20] J. Hancock, T. Khoshgoftaar, and J. Johnson, "Informative Evaluation Metrics for Highly Imbalanced Big Data Classification," in 2022 21st IEEE Int. Conf. on Machine Learning and Applications (ICMLA), 2022, pp. 1419–1426, doi: 10.1109/ICMLA55696.2022.00224.
[21] F. Koto, A. Rahimi, J. H. Lau, and T. Baldwin, "IndoLEM and IndoBERT: A Benchmark Dataset and Pre-Trained Language Model for Indonesian NLP," in Proc. 28th Int. Conf. on Computational Linguistics (COLING), Dec. 2020, pp. 757–770, doi: 10.18653/v1/2020.coling-main.66.
[22] J. Sirusstara, N. Alexander, A. Alfarisy, S. Achmad, and R. Sutoyo, "Clickbait Headline Detection in Indonesian News Sites Using Robustly Optimized BERT Pre-Training Approach (RoBERTa)," in 2022 3rd Int. Conf. on Artificial Intelligence and Data Sciences (AiDAS), 2022, pp. 1–6, doi: 10.1109/AiDAS56890.2022.9918678.
[23] F. Muftie and M. Haris, "IndoBERT-Based Data Augmentation for Indonesian Text Classification," in 2023 Int. Conf. on Information Technology Research and Innovation (ICITRI), 2023, pp. 128–132, doi: 10.1109/ICITRI59340.2023.10250061.
[24] M. Y. Ridho and E. Yulianti, "From Text to Truth: Leveraging IndoBERT and Machine Learning Models for Hoax Detection in Indonesian News," J. Ilm. Tek. Elektro Komput. dan Inform., vol. 10, no. 3, pp. 544–555, 2024, doi: 10.26555/jiteki.v10i3.29450.
[25] A. F. Hidayatullah, R. A. Apong, D. T. C. Lai, and A. Qazi, "Word Level Language Identification in Indonesian–Javanese–English Code-Mixed Text," Procedia Comput. Sci., vol. 244, pp. 105–112, 2024, doi: 10.1016/j.procs.2024.10.183.