Building of Informatics, Technology and Science (BITS)
Volume 6, No 2, September 2024, Page: 878−889
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v6i2.5634

Perbandingan Kinerja Pre-Trained Word Embedding pada Klasifikasi Sentimen Ulasan Produk Tokopedia dengan Long Short-Term Memory (LSTM)

Naufal Angling Dirfas, Vinna Rahmayanti Setyaning Nastiti*

Teknik, Informatika, Universitas Muhammadiyah Malang, Malang, Indonesia
Email: naufalangling@webmail.umm.ac.id, vinastiti@umm.ac.id
Email Penulis Korespondensi: vinastiti@umm.ac.id
Submitted: **19/07/2024**; Accepted: **09/09/2024**; Published**: 09/09/2024**

Abstrak−Dataset ulasan produk merupakan data yang berkembang pesat dan menarik untuk dieksplorasi. Peningkatan jumlah pengguna internet dan kebiasaan berbelanja pelanggan melalui toko online berdampak signifikan terhadap pertumbuhan data ulasan produk, terutama untuk toko online di Indonesia, seperti Tokopedia. Data sampel yang digunakan berjumlah 1079. Penelitian ini bertujuan untuk mengevaluasi kinerja tiga jenis pre-trained word embeddings, yaitu FastText, GloVe, dan Word2Vec, dalam model Long Short-Term Memory (LSTM) untuk klasifikasi sentimen ulasan produk di Tokopedia. Sistem klasifikasi sentimen otomatis sangat dibutuhkan untuk mengolah sejumlah besar ulasan produk, memudahkan penjual dalam mengetahui opini konsumen terhadap produk mereka. Penelitian ini berkontribusi dengan mengevaluasi dampak berbagai pre-trained word embeddings terhadap kinerja model LSTM dalam tugas klasifikasi sentimen. Selain itu, penelitian ini juga bertujuan untuk mengukur efektivitas model LSTM yang dikombinasikan dengan beberapa pretrained word embedding. Dengan mengimplementasikan arsitektur deep learning, komputer dapat mempelajari dan mengenali data kontekstual yang tersimpan dalam kalimat ulasan. Penelitian ini dilakukan dalam tiga tahap: pemilihan model, pengaturan lapisan, dan pengoptimalan hyperparameter, untuk menampilkan pengujian mendalam terhadap arsitektur deep learning yang digunakan serta kombinasi lapisan dan parameter yang sesuai untuk mendapatkan kinerja klasifikasi sentimen yang tinggi. Hasil eksperimen menunjukkan bahwa FastText dengan LSTM memberikan performa terbaik dengan akurasi 85.08%, diikuti oleh Word2Vec dengan akurasi 84.62%, dan GloVe dengan akurasi 83.04%. Kontribusi utama dari penelitian ini adalah menyajikan pengujian mendalam terkait dataset ulasan produk dan memberikan arsitektur deep learning beserta kombinasi lapisan dan parameter yang memiliki performa terbaik dalam mengenali sentimen pada dataset ulasan produk. Arsitektur ini mencapai performa yang lebih tinggi dibandingkan dengan metode BERT dengan lapisan CNN dan BiLSTM.

Kata Kunci: LSTM; GloVe; FastText; Word2Vec; Representasi Vektor Kata yang Telah Dilatih

Abstract−The product review dataset is a rapidly growing and interesting source of data to explore. The increase in the number of internet users and customer shopping habits through online stores has a significant impact on the growth of product review data, especially for online stores in Indonesia, such as Tokopedia. The sample data used amounted to 1079. This research aims to evaluate the performance of three types of pre-trained word embeddings, namely FastText, GloVe, and Word2Vec, in the Long Short-Term Memory (LSTM) model for sentiment classification of product reviews on Tokopedia. An automated sentiment classification system is needed to process many product reviews, making it easier for sellers to know what consumers think of their products. This research contributes by evaluating the impact of various pre-trained word embeddings on the performance of LSTM models in sentiment classification tasks. In addition, this research also aims to measure the effectiveness of LSTM models combined with multiple pre-trained word embeddings. By implementing a deep learning architecture, computers can learn and recognize contextual data stored in review sentences. The research was conducted in three stages: model selection, layer setup, and hyperparameter optimization, to feature in-depth testing of the deep learning architecture used and the appropriate combination of layers and parameters to obtain high sentiment classification performance. The experimental results show that FastText with LSTM provides the best performance with 85.08% accuracy, Word2Vec with 84.62% accuracy, and GloVe with 83.04% accuracy. The main contribution of this research is to present an in-depth test of the product review dataset and provide a deep learning architecture along with a combination of layers and parameters that has the best performance in recognizing sentiment on the product review dataset. This architecture achieves higher performance than the BERT method with CNN and BiLSTM layers.

Keywords: LSTM; GloVe; FastText; Word2Vec; Pretrained Word Embedding

1. PENDAHULUAN

Marketplace adalah media *e-commerce* yang menghubungkan penjual dan pembeli serta merupakan tempat untuk melakukan kegiatan bisnis[1]. Merujuk pada data yang dihimpun dari situs *iPrice*, *Tokopedia* menempati urutan pertama sebagai *marketplace* yang paling banyak dikunjungi, yaitu mencapai 157 juta pengunjung per bulannya hingga kuartal ke-3 tahun 2021[2]. Oleh karena itu, diperlukan sistem klasifikasi sentimen otomatis untuk menganalisis produk dalam skala besar. Analisis sentimen dapat digunakan dalam *review* produk, dengan analisis sentimen penjual dapat mengetahui pendapat masyarakat yang telah membeli suatu produk, baik *review* positif maupun negatif[3]. Dengan mengintegrasikan penyematan kata yang telah dilatih sebelumnya *(pre-trained word embeddings)* ke dalam model *Long Short-Term Memory (LSTM)*, diharapkan dapat meningkatkan kinerja dan efisiensi klasifikasi sentimen ulasan produk dalam skala besar.

Penelitian sebelumnya terkait analisis sentimen ulasan produk telah menggunakan beberapa model. Antara lain pembelajaran *Naive Bayes* berkelanjutan, *Getted Recurrent Unit (GRU), Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM), Support Vector Machine (SVM), K-Nearest Neighboor (KNN), Logistic Regression, Random Forest,* dan *CatBoost Classifier*. Dari semua model tersebut, model terbaik adalah *Logistic Regression*

Building of Informatics, Technology and Science (BITS)
Volume 6, No 2, September 2024, Page: 878−889
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v6i2.5634

dengan akurasi 90%, presisi 85%, *recall* 79% dan *F1-score* 81%[4][5][6][7][8][9]. Penelitian lebih lanjut terkait analisis sentimen menunjukkan bahwa *Long Short-Term Memory (LSTM)* memiliki potensi yang signifikan. Beberapa penelitian terdahulu menyimpulkan bahwa keakuratan model ini bergantung pada parameter yang digunakan, seperti perubahan vektor dimensi, nilai *dropout*, *learning rate*, dan sebagainya[10]. Kombinasi penyematan kata *(word embeddings)* juga dapat meningkatkan kinerja model *Long Short-Term Memory* dalam klasifikasi sentimen[11][12][13]. Penyematan kata yang telah dilatih sebelumnya merupakan elemen penting dalam *Natural Language Processing (NLP)*, terutama dalam *deep learning*. Efek penggunaan penyematan kata yang telah dilatih sebelumnya berbeda tergantung pada jaringan yang digunakan. Seperti yang ditunjukkan dalam penelitian[14], akurasi *CNN* tidak berubah secara signifikan dengan atau tanpa penyematan kata yang telah dilatih sebelumnya, sementara akurasi *BiLSTM* dan *CNN-BiLSTM* sangat dipengaruhi oleh jenis penyematan kata yang digunakan. Akurasi tertinggi pada model *CNN-BiLSTM* dengan *FastText* sebesar 88,8%. Hasil eksperimen menunjukkan bahwa penyematan kata yang telah dilatih sebelumnya sebagai representasi teks pada dataset dapat sedikit meningkatkan performa model, terutama hasil *CNN* yang paling tinggi dengan akurasi 83,9% karena *CNN* secara komputasi lebih efisien[15]. Penelitian lainnya merekomendasikan penggunaan model *BERT-SPC* untuk tugas *Aspect-Based Sentiment Analysis (ABSA)*. Penelitian ini juga menunjukkan beberapa kekurangan model, terutama dalam hal mengukur opini terhadap komputer. Skor akurasi tertinggi pada model *BERT-SPC* berkisar antara 72,65% hingga 84,98%[16].

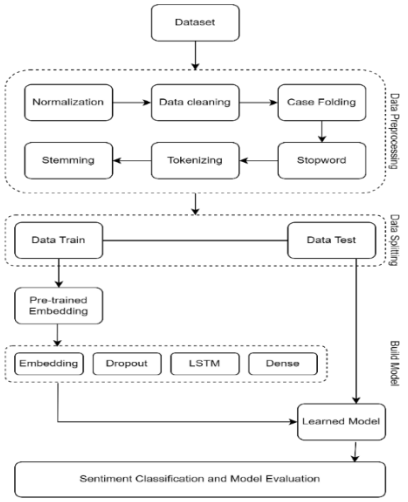
Dataset yang digunakan dalam penelitian ini adalah dataset dengan nama *PRDECT-ID*[17]. Terdapat penelitian terbaru yang menggunakan *deep learning* dengan model *CNN*, *BERT*, dan *LSTM*. Eksperimen yang dilakukan meliputi fine-tuning dengan mengkombinasikan *BERT* dengan lapisan tambahan *CNN* atau *LSTM* serta *fine-tuning hyperparameter*-nya. Penambahan lapisan *CNN* secara efektif meningkatkan nilai *F1-score* dibandingkan dengan kinerja model *BERT* tanpa lapisan tambahan. Akurasi yang didapatkan dari eksperimen tersebut sebesar 69,26%[18]. Untuk penelitian selanjutnya, penulis menyarankan untuk mengimplementasikan model *pre-trained BERT* bertumpuk dan model berbasis *ensemble BERT* untuk meningkatkan akurasi pengenalan emosi pada dataset ulasan produk.

Penelitian ini berfokus pada pentingnya sistem klasifikasi sentimen otomatis untuk *Tokopedia* dan marketplace lainnya, yang memungkinkan analisis ulasan produk dalam skala besar. Selain itu, penelitian ini memberikan kontribusi signifikan terhadap penelitian sebelumnya dengan mengevaluasi kinerja berbagai *pre-trained word embeddings* dalam model *LSTM* untuk klasifikasi sentimen. Meskipun beberapa model seperti *Logistic Regression* dan *CNN* telah menunjukkan performa yang baik, masih minim penelitian yang membandingkan kinerja berbagai model *deep learning* dengan penyematan kata yang berbeda. Dalam penelitian ini, saran penulis akan diambil dengan menggunakan penyematan kata yang telah dilatih sebelumnya untuk representasi teksnya, namun akan diimplementasikan dengan model *LSTM*. Tiga penyematan kata yang populer akan digunakan untuk perbandingan kinerja yaitu *GloVe*, *FastText*, dan *Word2Vec*[19]. Tujuan ini adalah untuk mengeksplorasi efektivitas *LSTM* dalam klasifikasi sentimen terhadap ulasan produk pembeli dari dataset *PRDECT-ID*, dengan harapan dapat mencapai peningkatan performa yang signifikan dalam akurasi dan *F1-score* dibandingkan dengan model sebelumnya.

2. METODOLOGI PENELITIAN

2.1 Tahapan Penelitian

Bagian ini menjelaskan tahapan-tahapan yang dilakukan dalam penelitian ini untuk melakukan klasifikasi sentimen ulasan produk menggunakan model *Long Short-Term Memory (LSTM)* dengan *pre-trained word embeddings*. Metodologi penelitian ini mencakup tiga tahap utama: pemrosesan data, pemisahan data, dan pembangunan model. Gambar 1 adalah *flowchart* yang menggambarkan keseluruhan proses penelitian ini.



Gambar 1. *Flowchart* penelitian

Building of Informatics, Technology and Science (BITS)
Volume 6, No 2, September 2024, Page: 878−889
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v6i2.5634

Gambar 1 menjelaskan tahapan-tahapan yang dilakukan dalam penelitian ini. Dimulai dari pengumpulan dataset sekunder ulasan produk, proses dilanjutkan dengan pemrosesan data yang mencakup normalisasi, pembersihan data, pengubahan semua huruf menjadi huruf kecil, *stemming*, tokenisasi, dan penghapusan kata-kata umum yang tidak relevan. Setelah itu, data dibagi menjadi data latih dan data uji. Tahap berikutnya adalah pembangunan model dengan menggunakan penyematan kata yang telah dilatih sebelumnya *(GloVe, FastText, Word2Vec)* dan melibatkan lapisan *embedding, dropout, LSTM*, serta *dense*. Terakhir, model yang telah dilatih digunakan untuk klasifikasi sentimen dan evaluasi kinerja model dalam mengklasifikasikan ulasan produk.

2.2 Dataset

Dataset yang digunakan dalam penelitian ini diambil dari *Mendeley Data* dengan nama *PRDECT-ID*. *PRDECT-ID* adalah singkatan dari *Product Reviews Dataset for Emotion Classification Task*[17]. Memiliki 5400 ulasan produk dan 29 kategori produk didalam dataset. Terdiri dari 11 fitur yaitu *Category, Product Name, Location, Price, Overall Rating, Number Sold, Total Review, Customer Rating, Customer Review, Sentiment* dan *Happy.* Fitur yang digunakan yaitu *Customer Review* untuk membuat prediksi dan *Sentiment* sebagai label atau target untuk diprediksi. Fitur *Sentiment* berisi 2 kelas yaitu *Positive* dan *Negative.* Total distribusi *Sentiment* dapat dilihat pada Tabel 1.

Tabel 1. Distribusi Sentimen

Sentiment	Jumlah Data
Positive	2579
Negative	2821

Beberapa contoh ulasan dengan sentimen positif antara lain: " Alhamdulillah berfungsi dengan baik. Packaging aman. Respon cepat dan ramah. Seller dan kurir amanah", " Bagus, berkualitas, sesuai gbr. Makasih Seller, Tokped dan Kurir" dan " Pengiriman super cepat, barang bagus, recommended seller, thanks." Sedangkan contoh ulasan dengan sentimen negatif mencakup: " Barang Rusak , Ga Guna Sama Sekali ... Parah", " Kecewa parah nggk berkah jualan gitu " dan " Pelayanan buruk, pengiriman lama dan respon yang lama. Jangan beli barang disini kalo ga mau kecewa. Cukup sekali ini aja"

2.3 Data Preprocessing

Dari dataset yang kotor tersebut akan dilakukan *Data Preprocessing* agar dataset tersebut layak digunakan untuk penelitian. *Preprocessing* merupakan suatu proses pembersihan data, reduksi data, dan diskritisasi data. Fase *preprocessing* tersebut akan membuat dataset lebih presisi. Dengan begitu digunakan tahapan *data preprocessing* untuk menyamaratakan data, menghapus kata yang tidak bermakna, mengeliminasi elemen dalam data, serta pengembalian kata awal[20]. Ada beberapa tahapan dalam *data preprocessing* yaitu *normalization, data cleaning, case folding, stopwords, tokenization,* dan *stemming.*

2.3.1 Normalization

Normalization adalah proses pengambilan kata dalam korpus menjadi kata yang sebenarnya yang didapatkan dari dataset *Kaggle* dengan Author Cita Tiara Hanni dan berjudul Cyberbullying Bahasa Indonesia. Seperti [‘bagusss’] menjadi [‘bagus’]. Daftar kata yang dinormalisasi ditulis secara manual untuk memperbaiki kekurangan terjemahan menggunakan *Google Translate*. Tindakan ini dapat dilakukan dengan memilih kata kunci yang tepat yang berhubungan dengan setiap aspek sentimennya[21]. Sampel *review* mengambil dari dataset, untuk hasil *normalization* dapat dilihat di Tabel 2.

Tabel 2. Sampel *Normalization*

Customer Review	Hasil Normalization
Bagus, berkualitas, sesuai gbr. Makasih Seller, Tokped dan Kurir	Bagus, berkualitas, sesuai gambar. terima kasih Seller, Tokped dan Kurir
Barang Rusak , Ga Guna Sama Sekali ... Parah	Barang Rusak , tidak Guna Sama Sekali ... Parah

Tabel 2 menunjukkan hasil proses normalisasi terhadap data ulasan produk. Proses normalisasi ini melibatkan pengubahan kata-kata yang tidak baku atau yang memiliki variasi penulisan menjadi bentuk yang standar dan sesuai. Sebagai contoh, kata "gbr." diubah menjadi "gambar", dan frasa seperti "Makasih Seller, Tokped dan Kurir" dinormalisasi menjadi "terima kasih Seller, Tokped dan Kurir".

2.3.2 Data Cleaning

Data Cleaning adalah tahapan Dimana pengolahan data original dengan cara penghapusan mention, simbol, emotikon, angka, dan spasi yang berlebihan. Sampel *review* mengambil dari Tabel 2 untuk hasil *data cleaning* dapat dilihat di Tabel 3.

Building of Informatics, Technology and Science (BITS)
Volume 6, No 2, September 2024, Page: 878−889
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v6i2.5634

Tabel 3. Sampel *Data Cleaning*

Customer Review	Hasil Normalization
Bagus, berkualitas, sesuai gambar. Terima kasih Seller, Tokped dan Kurir	Bagus berkualitas sesuai gambar terima kasih Seller Tokped dan Kurir
Barang Rusak , tidak Guna Sama Sekali ... Parah	Barang Rusak tidak Guna Sama Sekali Parah

Setelah tahap normalisasi, dilakukan tahap pembersihan data yang hasilnya ditampilkan pada Tabel 3. Tahap ini mencakup penghapusan simbol, emotikon, angka, dan spasi berlebihan yang tidak diperlukan dalam analisis sentimen. Dalam ulasan "Bagus, berkualitas, sesuai gambar. Terima kasih Seller, Tokped dan Kurir", semua karakter yang tidak relevan seperti tanda baca yang berlebihan dan angka dihilangkan, sehingga ulasan menjadi lebih bersih dan terstruktur.

2.3.3 Case Folding

Case Folding adalah proses yang paling sering digunakan sebagai tahapan selanjutnya dalam *preprocessing*. Dari tahap *data cleaning* terdapat 5 *instance* yang mengandung *Null* atau *NaN*, sehingga total data menjadi 5395. *Case Folding* bertujuan menghapus semua huruf kapital atau huruf besar yang ada pada data dokumen, atau secara lebih gampang *case folding* mengembalikan semua huruf besar ke huruf kecilnya[22]. Sampel *review* mengambil dari Tabel 3 untuk hasil *case folding* dapat dilihat di Tabel 4.

Tabel 4. Sampel *Case Folding*

Customer Review	Hasil Case Folding
Bagus berkualitas sesuai gambar terima kasih Seller Tokped dan Kurir	bagus berkualitas sesuai gambar terima kasih seller tokped dan kurir
Barang Rusak tidak Guna Sama Sekali Parah	barang rusak tidak guna sama sekali parah

Tabel 4 menampilkan hasil dari proses *case folding*, yaitu proses pengubahan semua huruf dalam teks ulasan menjadi huruf kecil. Proses ini penting untuk memastikan konsistensi dalam analisis data karena perbedaan antara huruf besar dan kecil dapat mempengaruhi hasil analisis. Misalnya, kata "Seller" dan "seller" akan dianggap berbeda jika tidak dilakukan case folding. Dengan mengubah semua huruf menjadi huruf kecil, seperti terlihat dalam hasil "Bagus berkualitas sesuai gambar" menjadi "bagus berkualitas sesuai gambar", data menjadi lebih homogen dan siap untuk tahap analisis berikutnya.

2.3.4 Stopwords

Stopwords adalah item dalam frasa yang tidak penting dalam *text mining* untuk divisi apapun. Untuk meningkatkan ketepatan penilaian, kata-kata ini biasanya diabaikan. Ada beberapa *stopwords* yang berbeda dalam berbagai format tergantung pada bidang, bahasa, dan sebagaianya[23]. Proses *stopword* menggunakan *library NLTK* dan mendownload *stopword* dalam bahasa Indonesia. Sampel *review* mengambil dari Tabel 4 untuk hasil *stopwords* dapat dilihat di Tabel 5.

Tabel 5. Sampel *Stopwords*

Customer Review	Hasil Stopwords
bagus berkualitas sesuai gambar terima kasih seller tokped dan kurir	bagus berkualitas sesuai gambar terima kasih seller tokped kurir
barang rusak tidak guna sama sekali parah	barang rusak parah

Setelah *case folding*, dilakukan proses penghapusan *stopwords* yang hasilnya ditampilkan pada Tabel 5. Stopwords adalah kata-kata umum yang tidak memberikan makna signifikan dalam analisis teks, seperti "dan", "yang", "untuk", dan sebagainya. Penghapusan stopwords bertujuan untuk meningkatkan efisiensi dan akurasi analisis dengan menghilangkan kata-kata yang tidak relevan. Sebagai contoh, ulasan "terima kasih seller tokped dan kurir" setelah dihapus stopwords menjadi "terima kasih seller tokped kurir".

2.3.5 Tokenization

Tokenization adalah proses potong memotong kalimat dari sebuah dokumen teks menjadi potongan kata kata atau karakter yang sesuai dengan keperluan. Nantinya pemecahan kalimat dan kata dilakukan berdasarkan spasi yang ada di dalam kalimat[24]. Potongan potongan tersebut dikenal dengan token. Sampel *review* mengambil dari Tabel 5 untuk hasil *tokenization* dapat dilihat di Tabel 6.

Tabel 6. Sampel *Tokenization*

Building of Informatics, Technology and Science (BITS)
Volume 6, No 2, September 2024, Page: 878−889
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v6i2.5634

Customer Review	Hasil Tokenization
bagus berkualitas sesuai gambar terima kasih seller tokped kurir	[‘bagus’, ‘berkualitas’, ‘sesuai’, ‘gambar’, ‘terima’, ‘kasih. ‘seller’ ‘tokped’, ‘kurir’]
barang rusak parah	[‘barang’, ‘rusak’, ‘parah’]

Dari Tabel 6 menampilkan hasil dari proses *tokenization*, yaitu proses memecah teks ulasan menjadi token-token yang lebih kecil, seperti kata atau karakter. Dalam proses ini, kalimat dalam ulasan dipecah berdasarkan spasi sehingga setiap kata menjadi satu token. Misalnya, ulasan "bagus berkualitas sesuai gambar" diubah menjadi ['bagus', 'berkualitas', 'sesuai', 'gambar']. Proses ini penting untuk mempersiapkan data agar dapat dianalisis lebih lanjut dalam model.

2.3.6 Stemming

Stemming adalah proses pengurangan kata menjadi bentuk dasar dari kata tersebut. Stemming bertujuan untuk mengekstrak informasi yang bermakna dalam data yang berukuran besar. Penelitian ini menggunakan *stemmer Sastrawi* yang dapat digunakan untuk dokumen berbahasa Indonesia[21]. Cara kerja algoritma ini adalah setiap kata yang ada dilakukan proses *stemming*. Jadi sebenarnya tidak semua kata diharuskan untuk di *stemming*. Oleh karena itu ada modifikasi algoritma pada proses *stemming* ini, sehingga luaran hasil yang didapatkan akan lebih optimal[25]. Sampel *review* mengambil dari Tabel 6 untuk hasil *stemming* dapat dilihat di Tabel 7.

Tabel 7. Sampel *Stemming*

Customer Review	Hasil Stemming
[‘bagus’, ‘berkualitas’, ‘sesuai’, ‘gambar’, ‘terima’, ‘kasih. ‘seller’ ‘tokped’, ‘kurir’]	bagus kualitas sesuai gambar terima kasih seller tokped kurir
[‘barang’, ‘rusak’, ‘parah’]	barang rusak parah

Tabel 7 menampilkan hasil dari proses *stemming*, yaitu proses mengurangi kata-kata menjadi bentuk dasarnya untuk mengekstrak informasi yang bermakna dalam data yang berukuran besar. Proses *stemming* menggunakan algoritma yang mengubah setiap kata menjadi bentuk dasarnya. Sebagai contoh, kata "berkualitas" diubah menjadi "kualitas" dan "penggunaan" diubah menjadi "guna". Proses ini penting dalam analisis teks karena membantu menyederhanakan dan mengurangi variasi kata yang digunakan dalam teks.

2.4 Data Splitting

Data splitting adalah teknik pembagian data menjadi 2 bagian yaitu data uji dan data latih. *Data splitting* memainkan peran penting dalam meningkatkan performa model *Natural Language Processing (NLP)*. Rasio pembagian data pada penelitian ini dengan rasio 80:20. Membagi data ke dalam beberapa set yang berbeda adalah teknik yang umum digunakan dalam pembelajaran mesin. Data biasanya dibagi menjadi set pelatihan dan validasi untuk melatih dan menemukan hiperparameter model (pemilihan model) dan memperkirakan kesalahan atau akurasi prediksi model. Tujuan dari pembagian data yang bervariasi untuk menyempurnakan algoritma model dan memilih yang terbaik dan parameter yang sesuai[26]. Pengujian pada penelitian ini dilakukan dengan menghitung *Accuracy* dan *Loss* dari hasil klasifikasi. Jumlah data pelatihan dan data uji dapat dilihat pada Tabel 8.

Tabel 8. Total Data Pelatihan dan Data Pengujian

Jenis Data	Jumlah
Data Pelatihan	4316
Data Pengujian	1079

Tabel 8 menampilkan hasil dari proses pembagian data (*data splitting*) menjadi dua bagian, yaitu data latih dan data uji. Teknik ini penting untuk meningkatkan performa model dalam *Natural Language Processing (NLP)*. Pada penelitian ini, data dibagi dengan rasio 80:20, dimana 80% data digunakan untuk melatih model dan 20% sisanya digunakan untuk menguji model. Pembagian ini memungkinkan evaluasi yang lebih akurat terhadap kemampuan model dalam memprediksi sentimen dari data yang belum pernah dilihat sebelumnya. Tabel ini menunjukkan bahwa data pelatihan berjumlah 4316 dan data pengujian berjumlah 1079.

2.5 Pre-trained Embedding

Di bidang pemrosesan bahasa alami (NLP), penggunaan *Pre-Trained Embedding* telah menjadi strategi yang diadopsi secara luas untuk meningkatkan kinerja berbagai tugas[27]. *Word embedding* adalah representasi vektor padat dari kata-kata, menangkap hubungan semantik dan sintaksis, dan biasanya dilatih pada korpus data teks yang besar menggunakan algoritme pembelajaran tanpa pengawasan, seperti *GloVe, FastText*, dan *Word2Vec*[27]. Model-model yang telah dilatih sebelumnya ini mampu menangkap informasi kontekstual dan menghasilkan *embedding* yang lebih ekspresif dan kuat daripada *word embedding* yang statis. Dengan memanfaatkan pengetahuan yang dikodekan dalam *Pre-Trained Embedding* sebelumnya, para peneliti dan praktisi dapat secara signifikan meningkatkan kinerja model

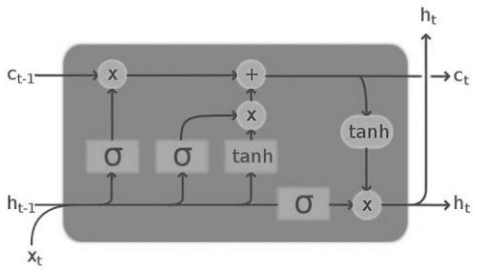
Building of Informatics, Technology and Science (BITS)
Volume 6, No 2, September 2024, Page: 878−889
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v6i2.5634

mereka, mengurangi kebutuhan akan kumpulan data khusus tugas yang besar dan waktu yang diperlukan untuk pelatihan[28]. Pendekatan ini sangat efektif dalam tugas-tugas di mana data berlabel langka, karena representasi yang telah dilatih sebelumnya dapat memberikan titik awal yang kuat untuk penyempurnaan lebih lanjut.

GloVe, *FastText*, dan *Word2Vec* yang telah dilatih sebelumnya adalah model penyematan kata yang banyak digunakan yang menawarkan representasi kata padat untuk berbagai tugas pemrosesan bahasa alami. Model-model ini telah dipelajari secara ekstensif di berbagai bahasa seperti Inggris, Hindi, Turki, dan bahkan dalam konteks hukum di India. Penelitian telah menunjukkan bahwa penyematan *FastText* dan *GloVe* cenderung berkinerja baik dalam menangkap kesamaan semantik tingkat kata, terutama dalam bahasa seperti Turki[29][30]. Selain itu, penelitian telah mengevaluasi bias yang ada dalam model pra-pelatihan ini, dengan *FastText* diidentifikasi sebagai model yang paling tidak bias dalam beberapa kasus[31]. Selain itu, penyematan *GloVe* telah digunakan secara efektif dalam tugas-tugas analisis sentimen, seperti memprediksi putusan positif dan negatif dalam kasus hukum, menampilkan aplikasi praktis mereka di luar terjemahan bahasa[32]. Model pra-pelatihan ini memainkan peran penting dalam meningkatkan efisiensi dan akurasi berbagai aplikasi *NLP* sambil juga meningkatkan kesadaran tentang potensi bias yang mungkin ada di dalamnya[33].

2.6 Long Short-Term Memory(LSTM)

Model *Long Short-Term Memory (LSTM)* adalah sebuah pendekatan dalam deep learning yang sangat relevan untuk berbagai aplikasi *Natural Language Processing (NLP)*, seperti penerjemahan teks, pengenalan suara, dan klasifikasi gambar atau teks. Banyak penelitian yang telah dilakukan dengan menggunakan *LSTM*, yang menunjukkan bahwa metode ini memiliki kinerja yang lebih baik daripada metode konvensional. *LSTM* dapat bekerja lebih baik daripada *RNN* pada deret data yang panjang, karena pada sel *LSTM* terdapat node yang memiliki *self-recurrent*[29][27].Terdapat tiga komponen penting dalam tahapan proses *LSTM*, yaitu *forget gate*, *input gate*, dan *output gate*. Gambar 2 adalah gambar arsitektur *LSTM*.



Gambar 2. Arsitektur *LSTM*

Forget Gate Informasi yang terdapat pada data masukan akan diproses dan diurutkan untuk menentukan data mana yang layak untuk disimpan atau tidak layak untuk disimpan di sel memori. Terdapat persamaan untuk fungsi forget gate, yaitu :.

$$f_t = \sigma(W_f x1 + U_f h_{t-1} + b_f) \quad (1)$$

Dua gerbang di *input gate* akan diimplementasikan; gerbang pertama menentukan nilai terbaru dengan fungsi aktivasi *sigmoid*. Karena menggunakan *sigmoid*, output t adalah "0" dan "1". Jika nilai t adalah "0" maka akan dilupakan, sedangkan nilai "1" akan disimpan. Dan nilai aktivasi tanh akan membuat vektor nilai terbaru, yang akan disimpan dalam sel memori. Persamaannya adalah :

$$i_t = \sigma(W_t x1 + U_f h_{t-1} + b_t) \quad (2)$$

$$C'_t = tanh(W_c x1 + U_f h_{t-1} + b_c) \quad (3)$$

Cell gates akan menggantikan sel memori lama dengan yang baru; nilai ini didapat dari penggabungan antara forget gate dan input gate. Berikut ini adalah persamaan dari *Cell gates* :

$$C_t = (f_t * C_{t-1} * C'_t) \quad (4)$$

Langkah terakhir yang akan dilakukan adalah *output gate*, yang akan melakukan dua langkah. Langkah pertama menentukan nilai pada sel memori yang akan dihapus menggunakan fungsi aktivasi *sigmoid*, diikuti dengan menempatkan nilai sel memori menggunakan fungsi tanh. Langkah terakhir kedua gerbang tersebut akan dijumlahkan untuk mendapatkan nilai keluaran. Dengan persamaan sebagai berikut :

$$o_t = \sigma(W_o x1 + U_o h_{t-1} + b_o) \quad (5)$$

$$h_t = o_t * \tanh(C_t) \quad (6)$$

Informasi :
f_t = *Forget gate*
i_t = *Input gate*
c_t = *Cell gate*

Building of Informatics, Technology and Science (BITS)
Volume 6, No 2, September 2024, Page: 878−889
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v6i2.5634

o_t = *Output gate*
h_t = *Hidden state*
b = *Bias*
W= *Bobot*

LSTM (Long Short-Term Memory) terdiri dari empat komponen utama yaitu *forget gate, input gate, cell gate,* dan *output gate,* yang bekerja secara sinergis untuk mengelola informasi dalam jaringan. *Forget gate* menentukan informasi mana yang harus dibuang dari sel memori dengan mengalikan *input* saat ini dan *hidden state* sebelumnya dengan bobot masing-masing, menambahkan bias, dan melewatkan hasilnya melalui fungsi aktivasi *sigmoid*. *Input gate* kemudian terdiri dari dua bagian yaitu pertama, menentukan nilai baru untuk ditambahkan ke sel memori menggunakan fungsi aktivasi *sigmoid*, dan kedua, menghasilkan vektor kandidat memori baru dengan fungsi aktivasi *tanh*. *Cell gate* menggabungkan output dari *forget gate* dan *input gate* untuk memperbarui sel memori, dimana sel memori lama dikalikan dengan *output forget gate* dan ditambahkan ke kandidat memori baru yang telah diskalakan oleh *output* dari *input gate*, menghasilkan sel memori baru. *Output gate*, sebagai langkah terakhir, menentukan nilai *output* dari sel memori dengan fungsi aktivasi *sigmoid* dan mengalikan hasilnya dengan *tanh* dari sel memori baru untuk menghasilkan *hidden state* baru yang digunakan sebagai *input* untuk *timestep* berikutnya.

2.7 Pengukuran Kinerja

Pengukuran dan evaluasi kinerja algoritma yang telah dibuat memiliki peran penting dalam menilai dan mengevaluasi kinerja algoritma. Performa sistem menggunakan metrik klasifikasi biner dari berbagai klasifikasi biner yang sudah ada sebelumnya untuk menentukan nilai akurasi[34]. Menentukan nilai optimal untuk model yang ada tergantung pada tingkat presisi yang diinginkan. Untuk evaluasi ini, kami enggunakan skala empat kategori yaitu *true positive (TP), true negative (TN), false positive (FP),* dan *false negative (FN)*[35]. *Confusion Matrix* digunakan sebagai kerangka kerja untuk investigasi dan analisis lebih lanjut.

$$Akurasi\ \% = \frac{TP+TN}{TP+FP+FP+FN} x\ 100\% \quad (7)$$

$$Presisi\ \% = \frac{TP}{TP+FP} x\ 100\% \quad (8)$$

$$Recall\ \% = \frac{TP}{TP+FN} x\ 100\% \quad (9)$$

$$F1-score\ \% = \frac{2xPresisi+Recall}{Presisi+Recall} x\ 100\% \quad (10)$$

3. HASIL DAN PEMBAHASAN

Pada pengujian ini, dibuat 3 skenario yang membentuk sistem seperti tabel 9. Setiap skenario akan membandingkan evaluasi kinerja model berdasarkan *pretrained embedding* yang berbeda, yaitu *FastText, GloVe,* dan *Word2Vec* yang digabungkan dengan *LSTM*. *Hyperparameter* yang akan digunakan pada 3 skenario dapat dilihat pada Tabel 10.

Tabel 9. Skenario Model

Skenario	Model
1	FastText + LSTM
2	GloVe + LSTM
3	Word2Vec + LSTM

Penggunaan tiga jenis *pretrained embedding* ini dimaksudkan untuk mengevaluasi pengaruh *embedding* yang berbeda terhadap performa model dalam melakukan klasifikasi sentimen. Dengan melakukan perbandingan ini, dapat ditentukan *embedding* mana yang memberikan hasil terbaik dalam konteks data dan model yang digunakan. Hal ini penting untuk memastikan bahwa model *LSTM* dapat memanfaatkan representasi teks yang paling efektif dalam meningkatkan akurasi dan efisiensi klasifikasi sentimen.

Tabel 10. Penyetelan Hyperparameter

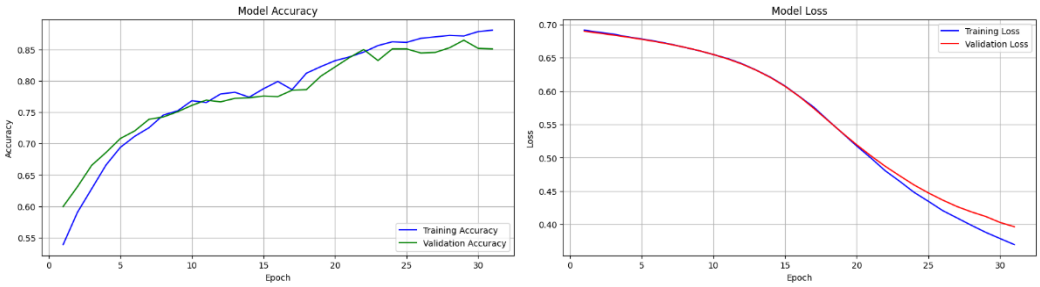
Parameter	Nilai
Embedding Dim	300
Dropout	0.5
LSTM	32
Dense	Sigmoid
Learning Rate	0.00001
Batch Size	32
Best Epoch	31

Building of Informatics, Technology and Science (BITS)
Volume 6, No 2, September 2024, Page: 878−889
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v6i2.5634

Selanjutnya, dilakukan penyetelan hyperparameter untuk masing-masing skenario model yang ditampilkan dalam Tabel 10. *Hyperparameter* yang dioptimalkan meliputi dimensi *embedding*, tingkat *dropout*, laju pembelajaran, ukuran *batch*, jumlah unit *LSTM*, dan jumlah *epoch* terbaik. Dimensi *embedding* yang digunakan adalah 300, dengan tingkat *dropout* 0,5 dan laju pembelajaran sebesar 0,00001. Ukuran *batch* yang digunakan adalah 32, dan jumlah *epoch* terbaik ditetapkan sebanyak 31. Jumlah unit *LSTM* yang digunakan adalah 32, sedangkan fungsi aktivasi untuk layer *dense* adalah *sigmoid*. *Hyperparameter* tersebut diperoleh melalui proses *fine-tuning* untuk mendapatkan konfigurasi terbaik yang dapat meningkatkan performa model *LSTM* pada masing-masing skenario yang diuji.

3.1 Skenario 1

Pada skenario pertama, model *LSTM* diintegrasikan dengan *word embedding FastText*. *FastText* adalah metode penyematan kata yang merupakan lanjutan dari pengembangan *word2vec*. Dikembangkan oleh tim riset *AI Facebook*, model ini terinspirasi oleh penelitian *Mikolov* dan rekan-rekannya. *FastText* mampu melatih 1 miliar kata dalam waktu 10 menit dengan hasil yang lebih baik dibandingkan model lainnya. Kurva pembelajaran untuk model ini dapat dilihat pada Gambar 3, sementara evaluasi model *FastText* dengan *LSTM* disajikan pada Tabel 11.



(a) (b)

Gambar 3. (a) Kurva Akurasi Skenario 1 dan (b) Kurva Kerugian Skenario 1

Analisis performa model *LSTM* dengan *embedding FastText* dalam skenario 1 disajikan pada Gambar 3. Pada Gambar 3(a), kurva akurasi untuk data pelatihan menunjukkan peningkatan signifikan selama fase awal pelatihan dan stabil setelah sekitar 15 *epoch*. Akurasi data validasi juga meningkat secara bertahap, namun terdapat sedikit variasi yang menunjukkan bahwa model mampu mempelajari pola dari data validasi tanpa mengalami *overfitting* yang signifikan. Gambar 3(b) menampilkan kurva kerugian (*loss*) yang menunjukkan penurunan secara konsisten pada data pelatihan hingga mencapai kestabilan setelah sekitar 20 *epoch*. Kurva *loss* untuk data validasi juga menunjukkan penurunan, meskipun terdapat sedikit variasi yang mencerminkan adanya perubahan kecil dalam performa model pada data validasi. Variasi ini umumnya diharapkan dalam proses pelatihan model dan menunjukkan bahwa model sedang berusaha untuk menyesuaikan diri dengan pola-pola yang ada pada data validasi.

Tabel 11. Hasil Kinerja Klasifikasi Skenario 1

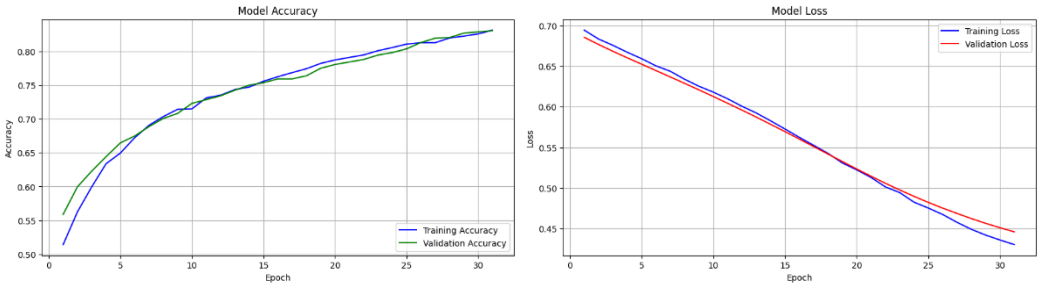
	Precission	Recall	F1-Score	Support
Negative	81%	93%	87%	561
Positive	91%	76%	83%	518
Accuracy	85,08%			

Hasil kinerja klasifikasi model dengan *embedding FastText* dalam skenario 1 dipaparkan dalam Tabel 11. Hasil ini menunjukkan bahwa model mampu mengklasifikasikan sentimen dengan baik secara keseluruhan. Namun, terdapat ketidakseimbangan dalam recall antara kelas negatif dan positif, di mana *recall* untuk kelas negatif mencapai 93% sedangkan recall untuk kelas positif hanya 76%. Hal ini mengindikasikan bahwa model lebih efisien dalam mengidentifikasi ulasan negatif, tetapi memiliki ruang untuk perbaikan dalam menangani ulasan positif.

3.2 Skenario 2

Dalam skenario kedua, model *LSTM* menggunakan *word embedding GloVe (Global Vector)*. *GloVe* menghitung frekuensi kemunculan kata-kata bersama dalam sebuah korpus, dengan tujuan mengkodekan makna melalui rasio probabilitas kemunculan kata-kata. Model *GloVe* bertujuan untuk mempelajari vektor kata sehingga hasil kali titik dari vektor-vektor tersebut sesuai dengan logaritma probabilitas kemunculan bersama kata-kata. Kurva pembelajaran untuk model ini dapat dilihat pada Gambar 4, sedangkan evaluasi model *GloVe* dengan *LSTM* disajikan pada Tabel 12

Building of Informatics, Technology and Science (BITS)
Volume 6, No 2, September 2024, Page: 878−889
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v6i2.5634



(a) (b)

Gambar 4. (a) Kurva Akurasi Skenario 2 dan (b) Kurva Kerugian Skenario 2

Pada Gambar 4, terlihat kurva akurasi model *LSTM* yang menggunakan *pretrained embedding GloVe* dalam skenario 2. Pada Gambar 4(a), terlihat kurva akurasi model LSTM yang menggunakan pretrained embedding GloVe dalam skenario 2. Grafik ini mengilustrasikan peningkatan akurasi yang signifikan pada data pelatihan, yang mencapai kestabilan setelah sekitar 15 *epoch*. Akurasi pada data validasi juga meningkat secara bertahap dan cenderung stabil setelah sekitar 20 *epoch*, menunjukkan bahwa model dapat mempelajari pola dari data validasi dengan baik tanpa *overfitting* yang berlebihan. Gambar 4(b) menampilkan kurva kerugian (*loss*) model *LSTM* dengan embedding *GloVe* pada skenario yang sama. Grafik ini menunjukkan penurunan *loss* yang konsisten pada data pelatihan dan stabil setelah sekitar 15 *epoch*. Kurva *loss* pada data validasi juga menunjukkan penurunan yang serupa, meskipun terdapat sedikit variasi yang mencerminkan adanya fluktuasi kecil dalam performa model pada data validasi. Kurva *loss* yang stabil pada data validasi mengindikasikan performa model yang baik terhadap data yang belum pernah dilihat sebelumnya.

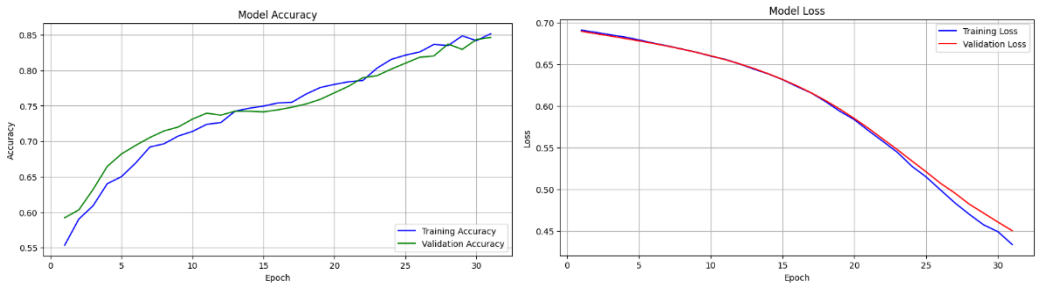
Tabel 12. Hasil Kinerja Klasifikasi Skenario 2

	Precission	Recall	F1-Score	Support
Negative	80%	89%	85%	561
Positive	86%	77%	81%	518
Accuracy	83.04%			

Tabel 12 menunjukkan kinerja klasifikasi model *LSTM* menggunakan *pretrained embedding GloVe* pada skenario 2. Model ini mencapai akurasi keseluruhan sebesar 83,04%. Hasil ini menunjukkan bahwa model secara keseluruhan efektif dalam mengklasifikasikan sentimen. Namun, terdapat perbedaan dalam kinerja antara kelas negatif dan positif, dengan *precision* yang lebih tinggi pada kelas positif dan recall yang lebih tinggi pada kelas negatif. Ini mengindikasikan bahwa meskipun model memiliki kinerja yang baik dalam mendeteksi ulasan negatif, ada ruang untuk peningkatan dalam mengidentifikasi ulasan positif dengan lebih akurat. Nilai *F1-Score* yang seimbang antara kedua kelas memperlihatkan bahwa model memiliki performa yang konsisten dalam pengklasifikasian sentimen.

3.3 Skenario 3

Skenario ketiga melibatkan penggunaan *word embedding Word2Vec* untuk model *LSTM*. *Word2Vec* adalah salah satu aplikasi pembelajaran tanpa pengawasan yang menggunakan jaringan syaraf tiruan. Model ini memanfaatkan informasi lokal dari bahasa tersebut, dengan semantik kata yang dipengaruhi oleh kata-kata di sekitarnya. *Word2Vec* menunjukkan kemampuan mempelajari pola linguistik sebagai hubungan linear antara vektor kata. Kurva pembelajaran untuk model ini dapat dilihat pada Gambar 5, sementara evaluasi model *Word2Vec* dengan *LSTM* disajikan pada Tabel 13.



(a) (b)

Gambar 5. (a) Kurva Akurasi Skenario 3 dan (b) Kurva Kerugian Skenario 3

Gambar 5 mengilustrasikan kurva akurasi dan kerugian (*loss*) dari model *LSTM* yang mengimplementasikan *embedding Word2Vec* dalam skenario 3. Pada sub-gambar (a), kurva akurasi menunjukkan bahwa akurasi pelatihan

Building of Informatics, Technology and Science (BITS)
Volume 6, No 2, September 2024, Page: 878−889
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v6i2.5634

model meningkat pesat pada beberapa *epoch* pertama dan kemudian mulai stabil mendekati nilai maksimum. Akurasi validasi, meskipun meningkat lebih lambat, menunjukkan tren peningkatan yang konsisten, mengindikasikan bahwa model mampu menggeneralisasi dengan baik terhadap data yang belum pernah dilihat sebelumnya. *Sub-gambar* (b) menampilkan kurva kerugian (*loss*) yang menunjukkan penurunan stabil pada data pelatihan seiring bertambahnya *epoch*. Kerugian validasi juga menurun secara bertahap dan stabil, yang menunjukkan bahwa model tidak mengalami *overfitting* secara signifikan. Konsistensi antara kurva kerugian pelatihan dan validasi ini mengindikasikan bahwa model *LSTM* dengan *embedding Word2Vec* mampu belajar representasi data yang baik dan efektif.

Tabel 13. Hasil Kinerja Klasifikasi Skenario 3

	Precission	Recall	F1-Score	Support
Negative	81%	92%	86%	561
Positive	90%	76%	83%	518
Accuracy	84,62%			

Tabel 13 merangkum kinerja klasifikasi model *LSTM* dengan *embedding Word2Vec* dalam skenario 3. Model ini mencapai akurasi keseluruhan sebesar 84,62%. Kinerja klasifikasi untuk masing-masing kelas juga sangat baik, dengan *precision* mencapai 81% untuk kelas negatif dan 90% untuk kelas positif. *Recall* untuk kelas negatif adalah 92%, menunjukkan bahwa model mampu mengidentifikasi sebagian besar contoh negatif dengan benar. Untuk kelas positif, *recall* adalah 76%, yang masih cukup baik dalam konteks klasifikasi sentimen. *F1-score*, yang merupakan rata-rata harmonis dari *precision* dan *recall*, menunjukkan nilai yang tinggi untuk kedua kelas, yaitu 86% untuk kelas negatif dan 83% untuk kelas positif. Hasil ini menunjukkan bahwa model mampu menjaga keseimbangan yang baik antara *precision* dan *recall*, yang penting untuk aplikasi praktis di mana baik *false positives* maupun *false negatives* dapat memiliki dampak signifikan. Secara keseluruhan, hasil evaluasi ini menegaskan bahwa penggunaan *embedding Word2Vec* memberikan kontribusi signifikan terhadap peningkatan performa model *LSTM* dalam memahami dan mengklasifikasikan konteks teks dengan akurasi yang tinggi.

3.4 Pembahasan

Hasil pengujian menunjukkan bahwa penggunaan tiga jenis *pretrained embedding*, yaitu *FastText, GloVe*, dan *Word2Vec* dengan model *LSTM*, memberikan hasil yang berbeda dalam klasifikasi sentimen. Dalam skenario pertama, embedding *FastText* menghasilkan akurasi terbaik sebesar 85.08%, dengan *precision* dan *recall* untuk kelas negatif masing-masing 81% dan 93%, serta untuk kelas positif masing-masing 91% dan 76%. *F1-score-nya* adalah 87% untuk kelas negatif dan 83% untuk kelas positif, menunjukkan performa yang lebih baik dalam mendeteksi ulasan negatif meskipun keseluruhan performanya cukup seimbang. Skenario kedua menggunakan *embedding GloVe* dengan akurasi 83.04%; *precision dan recall* untuk kelas negatif adalah 80% dan 89%, sedangkan untuk kelas positif adalah 86% dan 77%. F1-score-nya mencapai 85% untuk kelas negatif dan 81% untuk kelas positif, menandakan performa yang baik meskipun sedikit lebih rendah dibandingkan *FastText*. Pada skenario ketiga, *embedding Word2Vec* memberikan akurasi 84.62%, dengan *precision* dan *recall* untuk kelas negatif masing-masing 81% dan 92%, dan untuk kelas positif masing-masing 90% dan 76%. *F1-score-nya* adalah 86% untuk kelas negatif dan 83% untuk kelas positif, menunjukkan keseimbangan yang baik antara *precision* dan *recall* serta kemampuan klasifikasi yang solid untuk kedua jenis ulasan..

4. KESIMPULAN

Penelitian ini menyimpulkan bahwa penggunaan *pre-trained word embeddings* seperti *FastText, GloVe*, dan *Word2Vec* dalam model *LSTM* secara signifikan meningkatkan kinerja klasifikasi sentimen ulasan produk di *Tokopedia*. Hasil evaluasi menunjukkan bahwa *embedding FastText* dengan *LSTM* memberikan hasil terbaik dengan akurasi 85.08%, sedikit mengungguli *Word2Vec* dengan akurasi 84.62% dan *GloVe* dengan akurasi 83.04%. Integrasi karakter dalam *embedding FastText* memberikan keunggulan dalam menangani kata-kata yang tidak ada dalam korpus pelatihan, yang penting dalam analisis sentimen teks bebas. Namun, dalam keseluruhan metrik evaluasi, *GloVe* dan *Word2Vec* menunjukkan performa yang sangat kompetitif dan konsisten. Hasil ini menunjukkan bahwa integrasi *LSTM* dengan *pre-trained embeddings* dapat meningkatkan kinerja klasifikasi sentimen secara signifikan dibandingkan dengan metode lain. Meskipun demikian, penelitian ini memiliki keterbatasan dalam hal variasi data ulasan yang digunakan dan pengaturan *hyperparameter* yang mungkin masih bisa dioptimalkan lebih lanjut. Penelitian selanjutnya dapat memperluas cakupan data dan mengeksplorasi pengaturan *hyperparameter* yang lebih optimal serta eksperimen dengan *pre-trained embeddings* lainnya seperti *IndoBERT* untuk meningkatkan kinerja klasifikasi sentimen.

REFERENCES

[1] E. H. Muktafin, K. Kusrini, and E. T. Luthfi, "Analisis Sentimen pada Ulasan Pembelian Produk di Marketplace Shopee Menggunakan Pendekatan Natural Language Processing," *J. Eksplora Inform.*, vol. 10, no. 1, pp. 32–42, Sep. 2020, doi:

Building of Informatics, Technology and Science (BITS)
Volume 6, No 2, September 2024, Page: 878−889
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v6i2.5634

10.30864/eksplora.v10i1.390.
[2] D. Widiastuti, I. Rasal, D. Wulandari, and A. Putri, "Sentiment Analysis of Product Reviews Data on Tokopedia by Comparing The Performance of Classification Algorithms," *J. Infokum*, vol. 10, no. 2, pp. 1034–1041, 2022, [Online]. Available: http://infor.seaninstitute.org/index.php/infokum/index
[3] A. N. Rohman, R. Luviana Musyarofah, E. Utami, and S. Raharjo, "Natural Language Processing on Marketplace Product Review Sentiment Analysis," in *2020 2nd International Conference on Cybernetics and Intelligent System (ICORIS)*, IEEE, Oct. 2020, pp. 1–5. doi: 10.1109/ICORIS50180.2020.9320827.
[4] M. Loukili, F. Messaoudi, and M. El Ghazi, "Sentiment Analysis of Product Reviews for E-Commerce Recommendation based on Machine Learning," *Int. J. Adv. Soft Comput. its Appl.*, vol. 15, no. 1, pp. 1–13, 2023, doi: 10.15849/IJASCA.230320.01.
[5] Aakash, S. Gupta, and A. Noliya, "URL-Based Sentiment Analysis of Product Reviews Using LSTM and GRU," *Procedia Comput. Sci.*, vol. 235, pp. 1814–1823, 2024, doi: 10.1016/j.procs.2024.04.172.
[6] H. T. Ismet, T. Mustaqim, and D. Purwitasari, "Aspect Based Sentiment Analysis of Product Review Using Memory Network," *Sci. J. Informatics*, vol. 9, no. 1, pp. 73–83, May 2022, doi: 10.15294/sji.v9i1.34094.
[7] Hanafi, N. Suryana, and A. Basari, "Generate Contextual Insight of Product Review Using Deep LSTM and Word Embedding," *J. Phys. Conf. Ser.*, vol. 1577, no. 1, p. 012006, Jul. 2020, doi: 10.1088/1742-6596/1577/1/012006.
[8] S. Smetanin and M. Komarov, "Sentiment Analysis of Product Reviews in Russian using Convolutional Neural Networks," in *2019 IEEE 21st Conference on Business Informatics (CBI)*, IEEE, Jul. 2019, pp. 482–486. doi: 10.1109/CBI.2019.00062.
[9] F. Xu, Z. Pan, and R. Xia, "E-commerce product review sentiment classification based on a naïve Bayes continuous learning framework," *Inf. Process. Manag.*, vol. 57, no. 5, p. 102221, Sep. 2020, doi: 10.1016/j.ipm.2020.102221.
[10] P. F. Muhammad, R. Kusumaningrum, and A. Wibowo, "Sentiment Analysis Using Word2vec And Long Short-Term Memory (LSTM) For Indonesian Hotel Reviews," *Procedia Comput. Sci.*, vol. 179, pp. 728–735, 2021, doi: 10.1016/j.procs.2021.01.061.
[11] M. Khuntia and D. Gupta, "Indian News Headlines Classification using Word Embedding Techniques and LSTM Model," *Procedia Comput. Sci.*, vol. 218, pp. 899–907, 2023, doi: 10.1016/j.procs.2023.01.070.
[12] Y. KIRELLİ and Ş. ÖZDEMİR, "Sentiment Classification Performance Analysis Based on Glove Word Embedding," *Sak. Univ. J. Sci.*, vol. 25, no. 3, pp. 639–646, Jun. 2021, doi: 10.16984/saufenbilder.886583.
[13] N. K. Gondhi, Chaahat, E. Sharma, A. H. Alharbi, R. Verma, and M. A. Shah, "Efficient Long Short-Term Memory-Based Sentiment Analysis of E-Commerce Reviews," *Comput. Intell. Neurosci.*, vol. 2022, pp. 1–9, Jun. 2022, doi: 10.1155/2022/3464524.
[14] D. Nam, J. Yasmin, and F. Zulkernine, "Effects of Pre-trained Word Embeddings on Text-based Deception Detection," in *2020 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech)*, IEEE, Aug. 2020, pp. 437–443. doi: 10.1109/DASC-PICom-CBDCom-CyberSciTech49142.2020.00083.
[15] I. N. Khasanah, "Sentiment Classification Using fastText Embedding and Deep Learning Model," *Procedia Comput. Sci.*, vol. 189, pp. 343–350, 2021, doi: 10.1016/j.procs.2021.05.103.
[16] A. Chauhan, A. Sharma, and R. Mohana, "A Pre-Trained Model for Aspect-based Sentiment Analysis Task: using Online Social Networking," *Procedia Comput. Sci.*, vol. 233, pp. 35–44, 2024, doi: 10.1016/j.procs.2024.03.193.
[17] R. Sutoyo, S. Achmad, A. Chowanda, E. W. Andangsari, and S. M. Isa, "PRDECT-ID: Indonesian product reviews dataset for emotions classification tasks," *Data Br.*, vol. 44, p. 108554, 2022, doi: 10.1016/j.dib.2022.108554.
[18] A. Chowanda, R. Sutoyo, S. Achmad, E. W. Andangsari, S. M. Isa, and T. K. Chen, "Modeling Emotions Recognition on Indonesian Product Review By Combining Bert, Cnn, and Lstm Architecture," *Int. J. Innov. Comput. Inf. Control*, vol. 20, no. 3, pp. 929–944, 2024, doi: 10.24507/ijicic.20.03.929.
[19] P. Santosh Kumar, R. B. Yadav, and S. V. Dhavale, "A Comparison of Pre-trained Word Embeddings for Sentiment Analysis Using Deep Learning," 2021, pp. 525–537. doi: 10.1007/978-981-15-5113-0_41.
[20] S. R. Reddy. V., D. V. L. N. Somayajulu, and A. R. Dani, "Classification of Movie Reviews Using Complemented Naive Bayesian Classifier," *Int. J. Intell. Comput. Res.*, vol. 2, no. 3, pp. 148–153, Sep. 2011, doi: 10.20533/ijicr.2042.4655.2011.0019.
[21] A. P. P. Wardani, A. Adiwijaya, and M. D. Purbolaksono, "Sentiment Analysis on Beauty Product Review Using Modified Balanced Random Forest Method and Chi-Square," *J. Inf. Syst. Res.*, vol. 4, no. 1, pp. 1–7, Oct. 2022, doi: 10.47065/josh.v4i1.2047.
[22] I. Zulfa and E. Winarko, "Sentimen Analisis Tweet Berbahasa Indonesia Dengan Deep Belief Network," *IJCCS (Indonesian J. Comput. Cybern. Syst.*, vol. 11, no. 2, p. 187, Jul. 2017, doi: 10.22146/ijccs.24716.
[23] S. Dey, S. Wasif, D. S. Tonmoy, S. Sultana, J. Sarkar, and M. Dey, "A Comparative Study of Support Vector Machine and Naive Bayes Classifier for Sentiment Analysis on Amazon Product Reviews," in *2020 International Conference on Contemporary Computing and Applications (IC3A)*, IEEE, Feb. 2020, pp. 217–220. doi: 10.1109/IC3A48958.2020.233300.
[24] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, Sep. 1995, doi: 10.1007/BF00994018.
[25] M. U. Albab, Y. Karuniawati P, and M. N. Fawaiq, "Optimization of the Stemming Technique on Text preprocessing President 3 Periods Topic," *J. Transform.*, vol. 20, no. 2, pp. 1–10, 2023, [Online]. Available: https://journals.usm.ac.id/index.php/transformatika/■page1
[26] D. E. Birba, "A Comparative study of data splitting algorithms for machine learning model selection," *Degree Proj. Comput. Sci. Eng.*, vol. 2020, no. 1, pp. 1–23, 2020, [Online]. Available: https://www.diva-portal.org/smash/get/diva2:1506870/FULLTEXT01.pdf
[27] T. Mikolov, E. Grave, P. Bojanowski, C. Puhrsch, and A. Joulin, "Advances in Pre-Training Distributed Word Representations," Dec. 2017, [Online]. Available: http://arxiv.org/abs/1712.09405
[28] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," Oct. 2018, [Online]. Available: http://arxiv.org/abs/1810.04805

Building of Informatics, Technology and Science (BITS)
Volume 6, No 2, September 2024, Page: 878−889
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v6i2.5634

[29] Sitender, Sangeeta, N. S. Sushma, and S. K. Sharma, "Effect of GloVe, Word2Vec and FastText Embedding on English and Hindi Neural Machine Translation Systems," 2023, pp. 433–447. doi: 10.1007/978-981-19-7615-5_37.
[30] C. Tulu, "Experimental Comparison of Pre-Trained Word Embedding Vectors of Word2Vec, Glove, FastText for Word Level Semantic Text Similarity Measurement in Turkish," *Adv. Sci. Technol. Res. J.*, vol. 16, no. 4, pp. 147–156, Oct. 2022, doi: 10.12913/22998624/152453.
[31] E. Sesari, M. Hort, and F. Sarro, "An Empirical Study on the Fairness of Pre-trained Word Embeddings," in *Proceedings of the 4th Workshop on Gender Bias in Natural Language Processing (GeBNLP)*, Stroudsburg, PA, USA: Association for Computational Linguistics, 2022, pp. 129–144. doi: 10.18653/v1/2022.gebnlp-1.15.
[32] V. Vaissnave and P. Deepalakshmi, "Comparative Analysis: Sentiment Analysis for Legal Judgment Text in India's Supreme Court Based on GloVe Pretrained Word Embedding and Deep Learning Models," 2022, pp. 33–44. doi: 10.1007/978-981-19-0707-4_4.
[33] G. Curto, M. F. Jojoa Acosta, F. Comim, and B. Garcia-Zapirain, "Are AI systems biased against the poor? A machine learning analysis using Word2Vec and GloVe embeddings," *AI Soc.*, vol. 39, no. 2, pp. 617–632, Apr. 2024, doi: 10.1007/s00146-022-01494-z.
[34] V. M. Patro and M. Ranjan Patra, "Augmenting Weighted Average with Confusion Matrix to Enhance Classification Accuracy," *Trans. Mach. Learn. Artif. Intell.*, vol. 2, no. 4, Aug. 2014, doi: 10.14738/tmlai.24.328.
[35] D. P. Putra and E. B. Setiawan, "Hoax Detection Using Long Short-Term Memory (LSTM) and Gate Recurrent Unit (GRU) on Social Media," *Build. Informatics, Technol. Sci.*, vol. 4, no. 4, Mar. 2023, doi: 10.47065/bits.v4i4.3084.