Jurnal Komputer, Informasi dan Teknologi Vol: 5, No 1, 2025, Page: 1-15

Klasifikasi *Hoax* Vs *Non-Hoax* Pada Berita Bencana Alam Berbahasa Indonesia Menggunakan *Word Embedding*

Rangga Saputra Hari Pratama*, Setio Basuki

Program Studi Informatika, Fakultas Teknik, Universitas Muhammadiyah Malang

Abstrak: Hoaks atau berita palsu terkait bencana alam dapat menyebabkan kepanikan dan disinformasi yang berdampak luas pada masyarakat. Oleh karena itu, diperlukan metode otomatis untuk mengklasifikasikan berita hoax dan non-hoax secara efektif. Penelitian ini mengimplementasikan metode Word Embedding dan algoritma Long Short-Term Memory (LSTM) dalam klasifikasi berita hoax bencana alam berbahasa Indonesia. Tiga model Word Embedding yang digunakan adalah Word2Vec, FastText, dan GloVe. Proses penelitian melibatkan tahap preprocessing data, pembagian dataset, implementasi model LSTM, hingga analisis kinerja model dengan menggunakan metrik akurasi, precision, recall, serta F1-score. Hasil dari penelitian ini menyatakan bahwa model FastText dengan LSTM memberikan akurasi tertinggi sebesar 99%, diikuti oleh Word2Vec-LSTM dan GloVe-LSTM. Model FastText mampu menangkap informasi dari kata-kata yang jarang muncul, meningkatkan efektivitas dalam mendeteksi berita hoax. Selain itu, teknik augmentasi data menggunakan metode Random Synonym Replacement terbukti meningkatkan variasi dan keseimbangan dataset, yang berdampak positif pada performa model. Dengan penelitian ini, diharapkan dapat menjadi acuan bagi peneliti selanjutnya dalam pengembangan sistem deteksi berita hoax yang lebih akurat dan efisien, khususnya dalam konteks berita bencana alam.

Kata Kunci: *Hoaks, Word Embedding, LSTM, FastText, Word2Vec, GloVe,* Klasifikasi Berita

DOI: https://doi.org/10.53697/jkomitek.v5i1.233
*Correspondence: Rangga Saputra Hari Pratama
Email: ranggashp17@gmail.com

Received: 25-04-2025
Accepted: 25-05-2025
Published: 25-06-2025

Abstract: *Hoaxs or fake news related to natural disasters can cause panic and widespread misinformation among the public. Therefore, an automated method is needed to effectively classify hoax and non-hoax news. This study implements Word Embedding methods and the Long Short-Term Memory (LSTM) algorithm to classify hoax news on Indonesian-language natural disaster reports. Three Word Embedding models were used: Word2Vec, FastText, and GloVe. The research process involves data preprocessing, dataset partitioning, LSTM model implementation, and performance evaluation using accuracy, precision, recall, and F1-score metrics. The findings indicate that the FastText model combined with LSTM achieved the highest accuracy of 99%, followed by Word2Vec-LSTM and GloVe-LSTM.FastText effectively captures information from rarely occurring words, improving its ability to detect hoax news. Additionally, the data augmentation technique using the Random Synonym Replacement method has proven to enhance dataset diversity and balance, positively impacting model performance. With these findings, this study is expected to serve as a reference for future researchers in developing more accurate and efficient hoax detection systems, particularly in the context of natural disaster news.*

Keywords*: Hoax, Word Embedding, LSTM, FastText, Word2Vec, GloVe, News Classification*

Pendahuluan

Perkembangan teknologi informasi dan komunikasi telah membawa dampak signifikan terhadap cara manusia mengakses, menyebarkan, dan menerima informasi (Amalia et al, 2022). Di era digital ini, media sosial dan platform digital lainnya menjadi saluran utama dalam penyebaran informasi (Fauzy & Erwin Budi Setiawan, 2023). Namun,

Informasi palsu yang beredar melalui media sosial tidak hanya menimbulkan keresahan tetapi juga dapat memicu kepanikan massal dan menghambat proses penanganan bencana secara efektif (Kominfo, 2024). Hoax semacam ini sering menggunakan narasi sensasional untuk menarik perhatian, seperti istilah "gempa dahsyat" atau "tsunami raksasa," yang dapat memperburuk situasi krisis (Ginting, 2024). Dalam kondisi tersebut, masyarakat membutuhkan informasi yang cepat, akurat, dan dapat dipercaya (Zahra & Fauzan, 2022).Oleh karena itu, kemampuan untuk mendeteksi dan membedakan berita hoax dari berita yang valid menjadi sangat krusial, khususnya pada situasi bencana alam yang memerlukan respons cepat dan terkoordinasi (Bhoir, 2020).

Dalam mengatasi penyebaran hoax, diperlukan sistem deteksi otomatis yang akurat untuk memastikan masyarakat menerima informasi yang valid (Sastrawan et al., 2022). Salah satu pendekatan yang menjanjikan adalah penerapan Natural Language Processing (NLP) menggunakan teknik word embedding seperti Word2Vec, FastText, dan GloVe (Pilehvar & Camacho-Collados, 2020). Teknik ini mengubah kata-kata menjadi representasi vektor numerik sehingga hubungan semantik antar kata dapat dianalisis secara lebih mendalam (Shaker & Dhannoon, 2024). Kombinasi word embedding dengan algoritma Long Short-Term Memory (LSTM) telah menunjukkan hasil yang baik dalam berbagai penelitian sebelumnya untuk klasifikasi berita hoax, khususnya pada bahasa Indonesia (Rhman, 2021).

Namun, penelitian terdahulu lebih banyak berfokus pada berita politik, dengan sedikit perhatian terhadap berita bencana alam (Shaker & Dhannoon, 2024b). Selain itu, variasi metode embedding yang digunakan masih terbatas, dan evaluasi performa model sering kali kurang komprehensif (Sabri et al, 2024a).Oleh karena itu, penelitian ini berfokus pada penerapan model Word2Vec, FastText, dan GloVe yang dikombinasikan dengan algoritma LSTM untuk mendeteksi berita hoax pada topik bencana alam berbahasa Indonesia.

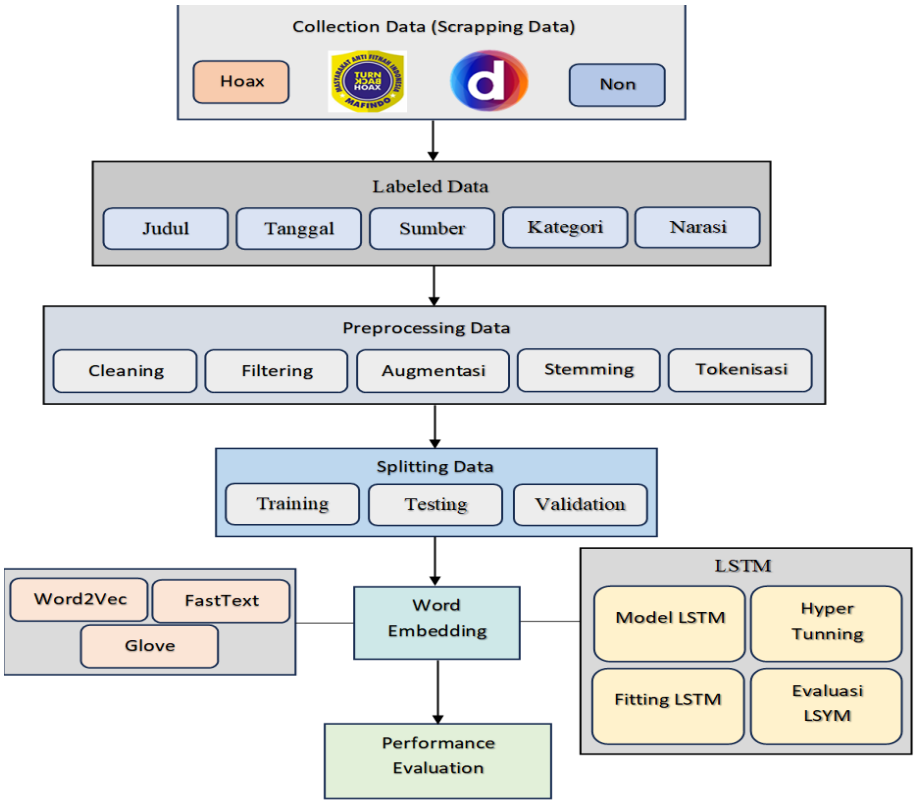
(Sunan et al, 2024) berhasil menunjukkan bahwa kombinasi word embedding GloVe dengan algoritma Long Short-Term Memory (LSTM) memberikan akurasi tinggi dalam klasifikasi berita hoax politik. (Adrian et al, 2023) juga mengatakan bahwa Word2Vec lebih unggul dibandingkan GloVe dalam mendeteksi hoax berbahasa Indonesia di media social. Namun, sebagian besar penelitian tersebut masih berfokus pada topik berita politik, sementara penelitian pada domain bencana alam, khusus dalam konteks bahasa Indonesia, masih sangat terbatas (Nurdin et al., 2020).

Penelitian ini bertujuan untuk mengatasi keterbatasan tersebut melalui penerapan model klasifikasi hoax yang lebih akurat dan efisien pada berita bencana alam berbahasa Indonesia. Dengan menggunakan pendekatan ini juga diharapkan memberikan kontribusi signifikan dalam mengurangi dampak negatif dari penyebaran informasi palsu, khususnya dalam situasi darurat yang membutuhkan informasi yang cepat dan akurat.

Metodologi

Penelitian ini berfokus pada klasifikasi berita hoax dan non-hoax terkait bencana alam menggunakan metode word embedding dan algoritma Long Short-Term Memory (LSTM). Populasi penelitian adalah berita bencana alam berbahasa Indonesia yang tersebar di platform daring. Sampel penelitian diambil dari dua sumber utama, yaitu berita hoax dari Turnbackhoax.id dan berita non-hoax dari Detik.com. Total dataset yang digunakan adalah 4.032 data, terdiri dari 2.016 data hoax dan 2.016 data non-hoax, yang telah diseimbangkan jumlahnya untuk menghindari bias klasifikasi.

Penelitian ini memiliki beberapa tahapan sistematis untuk meningkatkan akurasi deteksi berita hoax terkait berita bencana alam di Indonesia menggunakan metode Word Embedding. Setiap langkah dari penelitian ini dirancang untuk memberikan analisis yang mendalam mengenai penggunaan embedding berbasis kata dan konteks untuk membedakan berita hoax dari berita valid. Rencana penelitian ini terdiri dari beberapa tahapan utama, yaitu:



Gambar 1. Kerangka Kerja Penelitian Klasifikasi Hoax vs Non-Hoax

Tahap awal dalam penelitian ini adalah preprocessing data, yang bertujuan untuk menghilangkan bagian-bagian yang tidak relevan dari teks. Tahapan awal berupa *cleaning*, yaitu proses menghapus URL, angka, tanda baca, serta elemen lain yang tidak diperlukan, sehingga menghasilkan data yang lebih rapi dan terstruktur. Setelah itu, dilakukan augmentasi data menggunakan metode *Random Synonym Replacement*. Metode ini mengganti beberapa kata dalam teks dengan sinonimnya, sehingga variasi data meningkat tanpa mengubah makna asli dari teks tersebut(Setiawan & Lestari, 2021) .

Tahap berikutnya adalah filtering, yaitu penghapusan stopwords menggunakan pustaka Sastrawi, diikuti dengan stemming untuk mengubah kata ke bentuk dasarnya. Hal ini dilakukan untuk mengurangi redundansi kata dan memastikan model hanya memproses informasi yang relevan. Langkah berikutnya adalah proses tokenisasi, yaitu membagi teks menjadi bagian-bagian kata yang lebih kecil. agar dapat diproses oleh model machine learning. Data kemudian dibagi menjadi tiga bagian: data latih (60%), data validasi (20%), dan data uji (20%). Proses ini penting untuk memastikan model dapat belajar secara optimal dari data latih dan dievaluasi pada data uji yang tidak pernah dilihat sebelumnya.

Penelitian ini menggunakan tiga teknik *word embedding* untuk merepresentasikan teks dalam bentuk vektor numerik, yaitu Word2Vec, FastText, dan GloVe. Word2Vec bekerja dengan dua pendekatan, yakni CBOW (Continuous Bag of Words) dan Skip-Gram (Raheem & Chong, 2024.CBOW memprediksi kata target berdasarkan kata-kata di sekitarnya, sedangkan Skip-Gram menggunakan kata target untuk memprediksi kata-kata konteksnya . FastText memperluas kemampuan dengan mempertimbangkan elemen sub-kata, sehingga lebih efektif dalam menangani kata-kata yang jarang ditemukan atau tidak dikenal dalam korpus data. Sedangkan GloVe, memanfaatkan statistic co-occurrence global untuk menghasilkan vektor yang dapat merepresentasikan hubungan semantik antar kata secara mendalam (Ellaky et al, 2024).

Setelah teks diubah menjadi vektor numerik, model *Long Short-Term Memory (LSTM)* diterapkan untuk melakukan proses klasifikasi. LSTM dirancang untuk data sekuensial dengan mekanisme yang memungkinkan penyimpanan informasi penting dan penghapusan informasi yang tidak relevan melalui *input gate, forget gate, dan output gate*(Reddy et al, 2024). Model ini dilatih menggunakan data latih yang telah diproses, dengan optimasi parameter seperti *learning rate* dan jumlah unit pada setiap lapisan untuk mencapai performa terbaik (Farhan et al, 2024).

Evaluasi model dilakukan dengan mengukur akurasi, precision, recall, dan F1-score, yang bertujuan untuk menilai kemampuan model dalam mengklasifikasikan berita hoax dan non-hoax secara akurat (Sabri et al, 2024b). Selain itu, visualisasi performa model selama pelatihan menggunakan learning curve, yang membantu mengidentifikasi stabilitas dan efektivitas pembelajaran model. Evaluasi lebih lanjut dilakukan menggunakan confusion matrix yang menunjukkan jumlah prediksi benar dan salah untuk masing-masing kelas (*hoax* dan *non-hoax*). Kombinasi teknik *word embedding* dan LSTM diharapkan dapat menghasilkan model yang andal untuk mendeteksi berita hoax dalam konteks bencana alam.

Hasil Dan Pembahasan

Hasil *Prepocessing* Data

Pada penelitian ini pemrosesan data akan dilakukan menggunakan pemrograman python melalui Google Colaboratory. Beberapa proses yang dilakukan seperti cleaning

data, augmentasi random synonym replacement (penggantian sinonim acak), filtering, stemming, tokenization.

a. Cleaning data

Tabel 1. Proses Pembersihan

Original Teks	Hasil Pembersihan
Hujan deras selama berjam-jam menyebabkan banjir di beberapa wilayah Jakarta. Ribuan rumah terendam air dengan ketinggian mencapai 1,5 meter. Pemerintah setempat bersama BPBD dan relawan telah mengevakuasi warga ke tempat yang lebih aman. (Sumber: www.detik.com, Twitter @BPBDJakarta). Narasi: Banjir kali ini disebut sebagai yang terparah dalam lima tahun terakhir.	Hujan deras selama berjam-jam menyebabkan banjir di beberapa wilayah Jakarta. Ribuan rumah terendam air dengan ketinggian mencapai 1,5 meter. Pemerintah setempat bersama BPBD dan relawan telah mengevakuasi warga ke tempat yang lebih aman. Banjir kali ini disebut sebagai yang terparah dalam lima tahun terakhir.

Tabel 1. menunjukkan hasil proses pembersihan teks. Pada kolom Original Teks, teks masih mentah dengan struktur tidak tertata, mengandung pengulangan kata, dan informasi yang kurang jelas. Setelah proses pembersihan, kolom Hasil Pembersihan menampilkan teks yang lebih terstruktur, jelas, dan bebas dari elemen tidak relevan, seperti kata berulang dan kalimat tidak efektif. Tahapan ini juga menambahkan kategori seperti KATEGORI Disinformasi dan PENJELASAN untuk memberikan konteks lebih jelas, sehingga teks siap untuk dianalisis dalam proses *Natural Language Processing (NLP).*

b. Augmentasi *random synonym replacement* dan *filtering data*

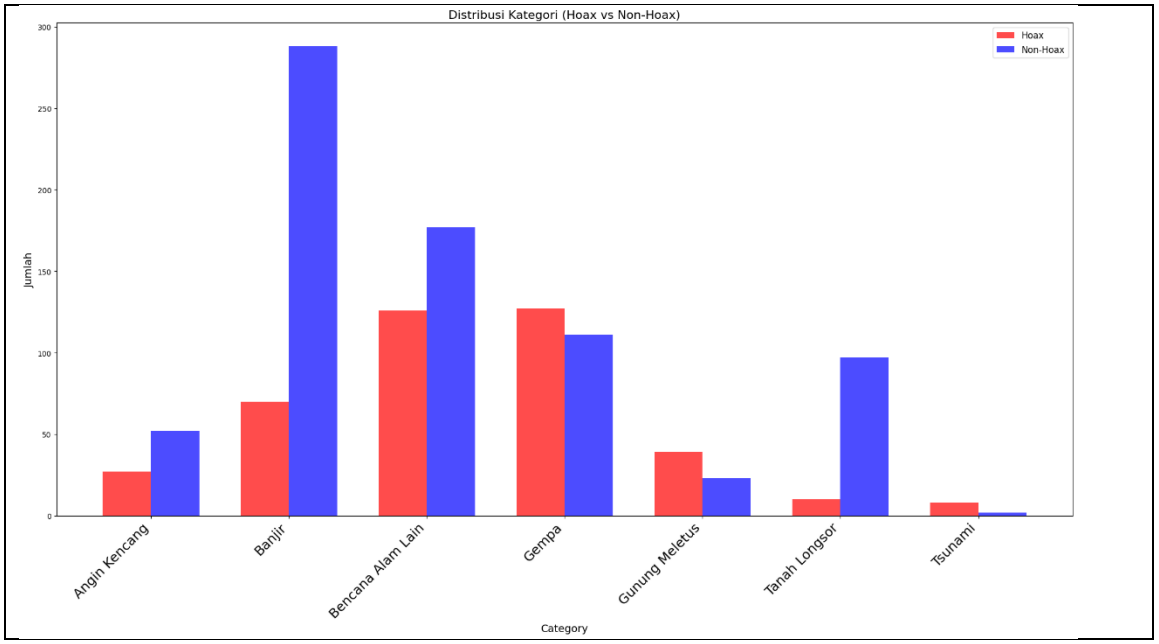
Tabel 2. Proses Penggantian Sinonim Acak Dan Penyaringan

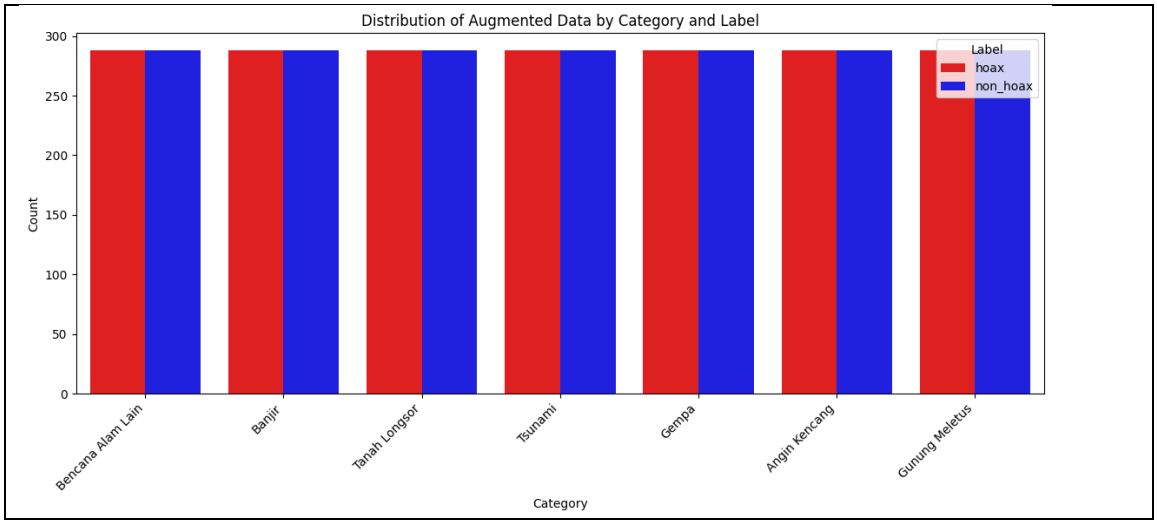
Hasil Pembersihan	Hasil Penggantian Sinonim Acak	Hasil Penyaringan
Hujan deras selama berjam-jam menyebabkan banjir di beberapa wilayah Jakarta. Ribuan rumah terendam air dengan ketinggian mencapai 1,5 meter. Pemerintah setempat bersama BPBD dan relawan telah mengevakuasi warga ke tempat yang lebih aman. Banjir kali ini disebut sebagai yang terparah	Curah hujan tinggi selama berjam-jam mengakibatkan genangan air di sejumlah daerah Jakarta. Ribuan rumah terendam dengan ketinggian air hingga 1,5 meter. Otoritas setempat bersama tim penyelamat telah membantu warga untuk mengungsi ke lokasi lebih aman. Bencana ini disebut	Hujan deras sebabkan banjir di Jakarta. Ribuan rumah terendam. Pemerintah dan relawan evakuasi warga. Banjir terparah lima tahun terakhir.

Hasil Pembersihan	Hasil Penggantian Sinonim Acak	Hasil Penyaringan
dalam lima tahun terakhir.	sebagai yang paling buruk dalam lima tahun terakhir.	

Tabel 2. menjelaskan tahapan pengolahan teks yang terdiri dari pembersihan, penggantian sinonim, dan penyaringan akhir. Pada tahap pembersihan, teks dibersihkan dari kata-kata yang tidak relevan untuk menghasilkan informasi yang lebih ringkas dan padat. Selanjutnya, dilakukan penggantian kata dengan sinonim menggunakan teknik *Random Synonym Replacement*. Misalnya, kata "gelombang" diganti dengan "ombak" tanpa mengubah maknanya. Tahap terakhir adalah penyaringan untuk memastikan teks tetap terstruktur dan hanya berisi informasi penting.

Teknik *Random Synonym Replacement* juga digunakan sebagai metode augmentasi data untuk menangani ketidakseimbangan kelas pada dataset. Proses ini menambah variasi teks sehingga distribusi data menjadi lebih merata. Hasilnya, model memiliki peluang lebih baik dalam mengenali pola dan meningkatkan performa klasifikasi. Visualisasi hasil augmentasi data dapat dilihat pada Gambar 2. berikut :





Gambar 2. Hasil Augmentasi Data Dengan *Random Synonym Replacement*

Gambar 2 menunjukkan distribusi data sebelum dan setelah augmentasi menggunakan teknik penggantian sinonim acak. Grafik pertama memperlihatkan peningkatan jumlah data pada beberapa kategori setelah augmentasi, menambah variasi teks dan mengurangi bias terhadap kategori tertentu. Grafik kedua menampilkan distribusi data yang lebih seimbang setelah augmentasi, mengatasi ketidakseimbangan yang sebelumnya ada. Proses augmentasi ini berhasil meningkatkan variasi dan keseimbangan data, memungkinkan model belajar lebih optimal tanpa bias terhadap kategori tertentu. Hasil ini diharapkan dapat meningkatkan akurasi dan kemampuan model dalam mengklasifikasikan teks dengan lebih baik.

c. Stemming data dan tokenisasi data

Tabel 1. Proses *Stemming* dan Tokenisasi

Hasil Stemming	Hasil Tokenasasi	Hasil Indeks Numerik
hujan deras sebab banjir jakarta rumah rendam pemerintah relawan evakuasi warga banjir parah lima tahun	["hujan", "deras", "sebab", "banjir", "jakarta", "rumah", "rendam", "pemerintah", "relawan", "evakuasi", "warga", "banjir", "parah", "lima", "tahun"]	[20, 5, 25, 95, 60, 70, 30, 35, 40, 15, 1, 45, 50, 55, 10, 60, 65, 15, 70, 35, 45, 30]

Tabel 3. menunjukkan hasil proses stemming dan tokenisasi dalam pengolahan teks. Stemming dilakukan untuk mengubah kata ke bentuk dasar dengan menghapus imbuhan, sehingga kata dengan makna serupa direpresentasikan dalam bentuk yang sama, seperti "ungsikan" menjadi "ungsi" dan "rusaknya" menjadi "rusak". Setelah stemming, dilakukan tokenisasi, yaitu memecah teks menjadi kata-kata individual, seperti "bencana", "alam", dan "banjir", agar lebih mudah dianalisis oleh model.

Tahap akhir dari proses ini adalah konversi teks ke indeks numerik, di mana setiap kata direpresentasikan sebagai angka berdasarkan urutan atau nilai dalam kamus kata

(*vocabulary*). Representasi numerik ini mempermudah pemrosesan oleh model machine learning atau deep learning, sehingga pola dalam data dapat dikenali lebih efisien. Dengan proses ini, teks yang semula berupa kalimat panjang diubah menjadi format terstruktur, mendukung efisiensi pemrosesan dan meningkatkan akurasi analisis.

Hasil Splitting Data

Tabel 2. Total Pembagian Dataset

Data Latih	Data Uji	Data Validasi
2419	807	806

Tabel 4. menunjukkan hasil pembagian dataset ke dalam tiga kategori utama, yaitu data latih (training data), data uji (testing data), dan data validasi (validation data). Pembagian dataset ini bertujuan untuk memastikan bahwa model dapat belajar dengan baik serta menghindari risiko overfitting, yaitu kondisi di mana model terlalu menyesuaikan diri dengan data latih tetapi gagal bekerja dengan baik pada data baru. Dari tabel tersebut, Jumlah data train, test, dan validation dihitung berdasarkan total dataset dengan pembagian 60% untuk training (2.419 data), 20% untuk testing (807 data), dan 20% untuk validation (806 data).

Implementasi Word Embedding dengan LSTM (Long Short-Term Memory)

Tabel 3. Perbandingan Implementasi *Word Embedding*

Metode	Implementasi	Keterangan
Word2Vec (Bahasa Indonesia)	Word2Vec(sentences, vector_size=100, window=5, min_count=2, workers=4)	- Melatih model menggunakan teks dalam Bahasa Indonesia. - Menghasilkan embedding berdasarkan kemunculan kata dalam konteks (CBOW atau Skip-gram). - Model disimpan dalam file word2vec_indonesia.model.
FastText	FastText(sentences, vector_size=100, window=5, min_count=2, workers=4)	- Mirip dengan Word2Vec, tetapi mempertimbangkan subword (karakter n-gram), sehingga lebih baik dalam menangani kata yang tidak dikenal (OOV). - Model disimpan dalam file fasttext_indonesia.model.
Glove	train_glove_model(corpus, embedding_dim=100, window=5, min_count=2, iterations=50)	- Melatih model GloVe dari awal menggunakan data sendiri. - Menggunakan matriks co-occurrence dan faktoralisasi matriks untuk membangun

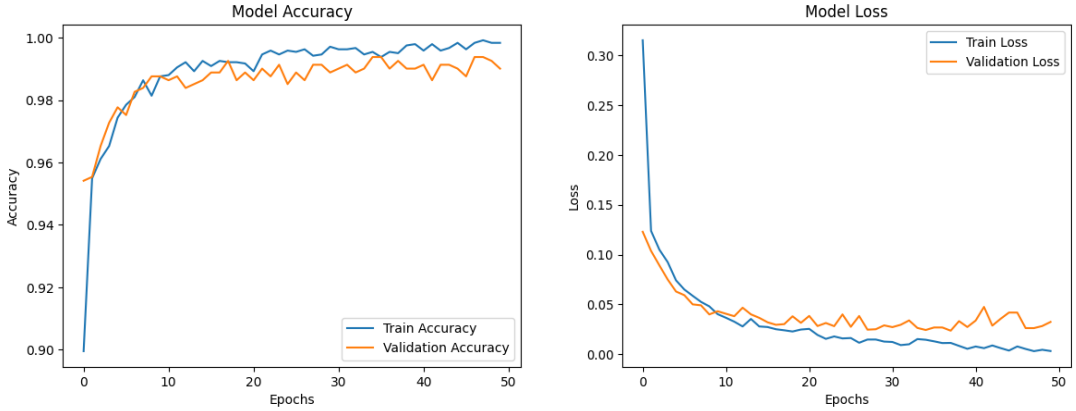
Metode	Implementasi	Keterangan
		embedding. - Memerlukan beberapa iterasi untuk konvergensi. - Model disimpan dalam file glove_indonesia.model.

Tabel 5. membandingkan tiga metode Word Embedding: Word2Vec, FastText, dan GloVe. Word2Vec mempelajari konteks kata menggunakan metode CBOW atau Skip-gram, menghasilkan model word2vec_indonesia.model yang dapat digunakan untuk berbagai tugas NLP. FastText, serupa dengan Word2Vec, lebih unggul dalam menangani kata yang jarang muncul (*Out-of-Vocabulary*) dengan mempertimbangkan n-gram, dan disimpan dalam *fasttext_indonesia.model*.

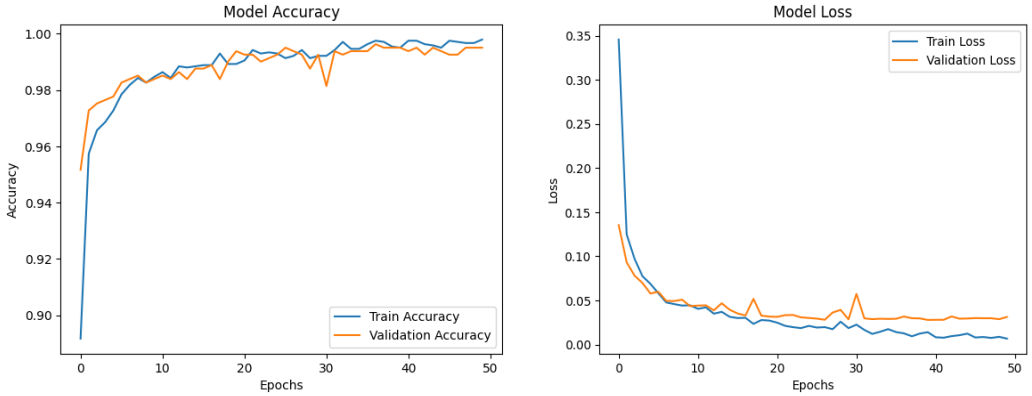
Sementara itu, GloVe menggunakan pendekatan matriks co-occurrence untuk menghasilkan vektor kata yang representatif, meskipun memerlukan iterasi lebih banyak. Hasilnya disimpan dalam *glove_indonesia.model*. Word2Vec dan FastText berfokus pada konteks kata dalam kalimat, sedangkan GloVe lebih menitikberatkan pada hubungan kata secara global berdasarkan statistik korpus.

Analisa Performance Evaluation

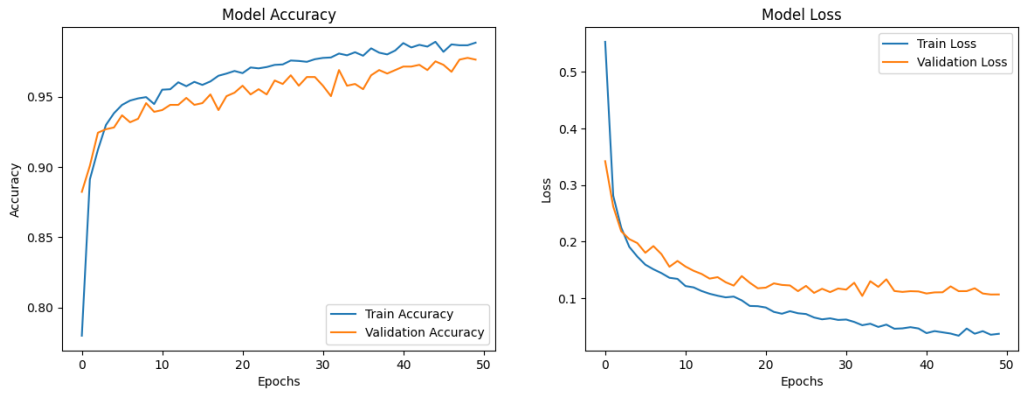
a. Perbandingan learning curve



Gambar 3. Learning Curve untuk Word2Vec dan LSTM



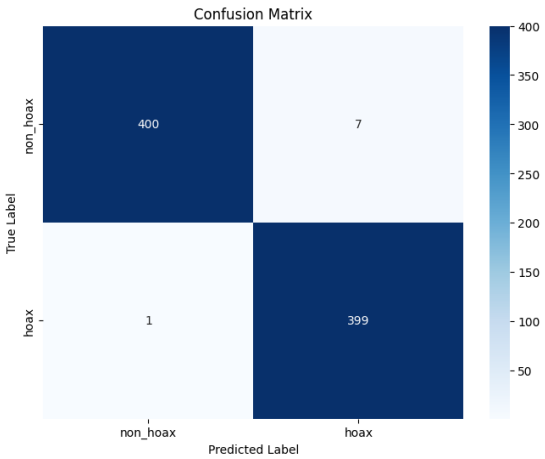
Gambar 4. *Learning Curve* untuk *FastText* dan LSTM



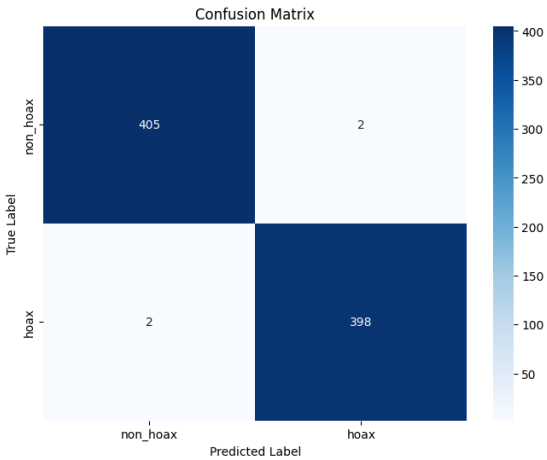
Gambar 5. *Learning Curve* untuk *Glove* dan LSTM

Learning curve untuk berbagai kombinasi metode embedding dan LSTM menunjukkan hasil yang beragam. Pada Gambar 3 menggambarkan Word2Vec-LSTM dengan akurasi mencapai 0.95 dan kurva loss yang menurun secara bertahap tanpa tanda overfitting, menunjukkan model mampu merepresentasikan data dengan baik. Sedangkan Gambar 4 menampilkan performa terbaik dari FastText-LSTM dengan akurasi 0.98 dan loss yang lebih rendah, menunjukkan keunggulan FastText dalam menangkap informasi, termasuk kata-kata yang jarang muncul. Sebaliknya, Gambar 5 menunjukkan hasil GloVe-LSTM dengan akurasi lebih rendah, yaitu 0.93, dan loss lebih tinggi, yang mengindikasikan keterbatasan GloVe dalam memahami hubungan antar kata dibandingkan Word2Vec dan FastText. Secara keseluruhan, FastText-LSTM menunjukkan performa terbaik dalam klasifikasi teks.

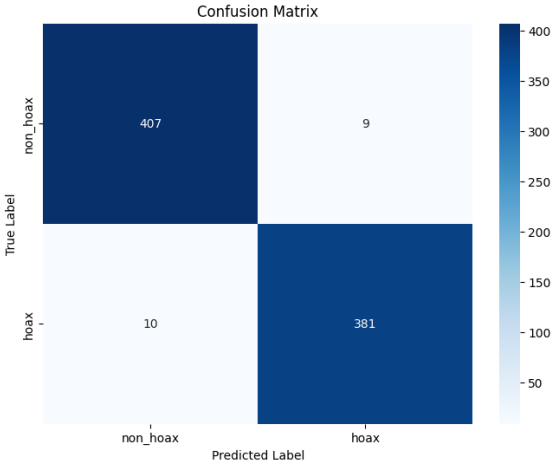
b. Perbandingan Confussion Matrix



Gambar 6. *Confussion Matrix Word2Vec* dan LSTM



Gambar 7. *Confussion Matrix FastText* dan LSTM



Gambar 8. *Confussion Matrix Glove* dan LSTM

Dari ketiga confusion matrix yang ditampilkan, terlihat perbandingan kinerja klasifikasi untuk Word2vec+LSTM (400 true positive, 399 true negative), FastText+LSTM (425 true positive, 398 true negative), dan Glove+LSTM (407 true positive, 391 true negative). FastText+LSTM menunjukkan performa terbaik dengan jumlah prediksi benar tertinggi di kedua kelas dan false positive/negative terendah (2 dan 3), diikuti oleh Word2vec+LSTM (7 false positive, 1 false negative), sementara Glove+LSTM menunjukkan tingkat kesalahan prediksi yang lebih tinggi (9 false positive, 10 false negative). Hal ini konsisten dengan hasil learning curve sebelumnya dimana FastText+LSTM menunjukkan performa klasifikasi yang superior dibandingkan dua model lainnya.

Classification Report

Tabel 6. *Performance Comparison of Word Embedding Models with* LSTM

Model	Class	Precision	Recall	F1-score	Support
Word2vec + LSTM	Non-hoax	1.00	0.98	0.99	407
	Hoax	0.98	1.00	0.99	400
	Accuracy			0.99	807
FastText + LSTM	Non-hoax	0.98	1.00	0.99	407
	Hoax	1.00	0.98	0.99	400
	Accuracy			0.99	807
Glove + LSTM	Non-hoax	0.98	0.98	0.98	416
	Hoax	0.98	0.97	0.98	391
	Accuracy			0.98	807

Hasil menunjukkan bahwa Word2Vec+LSTM dan FastText+LSTM memiliki performa terbaik dengan akurasi 0.99 dan F1-score 0.99 untuk kedua kelas (hoax dan non-hoax). FastText memiliki presisi sempurna (1.00) untuk kelas non-hoax, yang menunjukkan bahwa model ini sangat akurat dalam mengidentifikasi berita yang bukan hoaks. Namun, Word2Vec+LSTM juga unggul dalam keseimbangan antara precision dan recall. Sebaliknya, GloVe+LSTM memiliki performa sedikit lebih rendah dengan akurasi 0.98 dan F1-score 0.98, menunjukkan bahwa metode ini masih cukup baik tetapi tidak sebaik dua metode

lainnya. Support yang seimbang antara kelas hoax dan non-hoax menunjukkan dataset yang well-balanced, memberikan validitas tambahan pada hasil evaluasi.

Analisa perbandingan dengan penelitian sebelumnya

Tabel 7. Perbandingan Performa Hasil Klasifikasi dengan Penelitian Sebelumnya

Penelitian	Model	Akurasi
Muhammad Ghifari Adrian [10]	Word2Vec + LSTM	95%
Penelitian Ini	Word2Vec + LSTM dengan Augmentasi *random synonym replacement*	99%

Tabel ini menyajikan perbandingan antara hasil penelitian ini dengan studi sebelumnya yang dilakukan oleh Muhammad Ghifari Adrian. Model yang digunakan dalam kedua penelitian sama, yaitu Word2Vec+LSTM, namun dalam penelitian ini ditambahkan teknik augmentasi random synonym replacement. Dari hasil yang diperoleh, model dalam penelitian Muhammad Ghifari Adrian memiliki akurasi 95%, sedangkan model yang dikembangkan dalam penelitian ini meningkat hingga 99%. Peningkatan akurasi sebesar 4% menunjukkan bahwa teknik augmentasi yang diterapkan efektif dalam meningkatkan kemampuan model untuk mengklasifikasikan berita hoaks dan non-hoaks dengan lebih baik. Hal ini mengindikasikan bahwa strategi peningkatan data dapat membantu model memahami variasi bahasa dengan lebih optimal.

Testing Model

Tabel 8. Hasil Pengujian *Model Word Embedding*

Metode	Berita Singkat	Actual Label	Predicted Label
Word2Vec (Bahasa Indonesia)	"Banjir melanda 8 desa di 2 kecamatan Kabupaten Demak. Imbasnya, ada 5.065 KK dan 21.301 jiwa yang terdampak banjir." (Detik.com)	Non-hoax	Non-hoax
FastText	"Sukabumi dilanda banjir bandang setinggi delapan meter, menyebabkan kerusakan parah." (TrunBackHoax.Id)	Hoax	Hoax
Glove	"Puluhan rumah warga di Situbondo terendam air luapan Sungai Meraan. Ketinggian air mencapai setinggi dada orang dewasa." (Detik.com)	Non-hoax	Non-hoax

Tabel 8. Menunjukkan hasil pengujian model Word2Vec, FastText, dan GloVe dalam mengklasifikasikan berita banjir sebagai hoax atau non-hoax. Word2Vec dan GloVe berhasil

mengidentifikasi berita dari sumber terpercaya (detik.com) sebagai non-hoax, sementara FastText juga mampu mengenali berita hoaks tentang banjir di Bandung yang telah diverifikasi oleh turnbackhoax.id. Ini menunjukkan bahwa ketiga model dapat mengklasifikasikan berita dengan baik, terutama FastText yang berhasil membedakan hoax dari fakta, kemungkinan karena kemampuannya mengenali kata-kata yang jarang muncul melalui subword representation.

Simpulan

Penelitian ini berhasil menunjukkan bahwa penggunaan metode *Word Embedding* dan *Long Short-Term Memory (LSTM)* dalam klasifikasi berita hoax pada berita bencana alam berbahasa Indonesia memberikan hasil yang optimal. Dari berbagai metode yang diuji, kombinasi FastText dan LSTM menghasilkan akurasi tertinggi, menunjukkan bahwa FastText memiliki kemampuan yang lebih baik dalam menangani kata-kata yang jarang muncul serta memahami makna kontekstual dari suatu berita. Hasil penelitian ini juga menunjukkan bahwa proses preprocessing data yang mencakup pembersihan teks, tokenisasi, stemming, serta augmentasi data berkontribusi terhadap peningkatan performa model dalam mengklasifikasikan berita hoax dan non-hoax. Dibandingkan dengan penelitian sebelumnya yang tidak menggunakan teknik augmentasi, penelitian ini mampu meningkatkan akurasi secara signifikan. Penelitian ini diharapkan dapat dijadikan referensi dalam pengembangan sistem deteksi berita hoax secara lebih luas, khususnya untuk berita yang berkaitan dengan bencana alam.

References

Adrian, M. G., Prasetyowati, S. S., & Sibaroni, Y. (2023). Effectiveness of Word Embedding GloVe and Word2Vec within News Detection of Indonesian uUsing LSTM. *Jurnal Media Informatika Budidarma*, *7*(3), 1180. https://doi.org/10.30865/mib.v7i3.6411

Amalia, J., Pakpahan, J., Pakpahan, M., Panjaitan, Y., Informatika dan Teknik Elektro, F., & Teknologi Del, I. (2022). Model Klasifikasi Berita Palsu Menggunakan Bidirectional LSTM Dan Word2Vec Sebagai Vektorisasi. *Jurnal Teknik Informatika Dan Sistem Informasi*, *9*(4).

David Rhman, A. D. dan F. M. (2021). *Penerapan Weighted Word Embedding pada Pengklasifikasian Teks Berbasis Recurrent Neural Network untuk Layanan Pengaduan Perusahaan Transportasi*.

Ellaky, Z., Benabbou, F., Matrane, Y., & Qaqa, S. (2024). A Hybrid Deep Learning Architecture for Social Media Bots Detection Based on BiGRU-LSTM and GloVe Word Embedding. *IEEE Access*, *12*, 100278–100294. https://doi.org/10.1109/ACCESS.2024.3430859

Farhan, S., Shoukat, R., & Aslam, A. (2024). Automatic Sarcasm Detection on Cross-Platform Social Media Datasets: A GLoVe and Bi-LSTM Based Approach. *Journal of Universal Computer Science*, *30*(5), 674–693. https://doi.org/10.3897/jucs.104790

Fauzy, A. R. I., & Erwin Budi Setiawan. (2023). Detecting Fake News on Social Media Combined with the CNN Methods. *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, *7*(2), 271–277. https://doi.org/10.29207/resti.v7i2.4889

Ginting, I. (2024). *Pentingnya Daya Kritis Masyarakat Tangkal HOAX* . Https://Babelprov.Go.Id/Artikel_detil/Pentingnya-Daya-Kritis-Masyarakat-Tangkal-Hoax%C2%A0.

Kominfo. (2024). *Siaran Pers No. 02/HM/KOMINFO/01/2024 tentang Hingga Akhir Tahun 2023, Kominfo Tangani 12.547 Isu Hoaks*. Https://Www.Kominfo.Go.Id/Berita/Siaran-Pers/Detail/Siaran-Pers-No-02-Hm-Kominfo-01-2024-Tentang-Hingga-Akhir-Tahun-2023-Kominfo-Tangani-12-547-Isu-Hoaks.

Nurdin, A., Anggo, B., Aji, S., Bustamin, A., & Abidin, Z. (2020). Perbandingan Kinerja Word Embedding Word2vec, Glove, Dan Fasttext Pada Klasifikasi Teks. *Jurnal TEKNOKOMPAK*, *14*(2), 74.

Pilehvar, M. T., & Camacho-Collados, J. (2020). *Embeddings in Natural Language Processing Theory and Advances in Vector Representation of Meaning*. Morgan publisher.

Raheem, M., & Chong, Y. C. (2024). E-Commerce Fake Reviews Detection Using LSTM with Word2Vec Embedding. *Journal of Computing and Information Technology*, *32*(2), 65–80. https://doi.org/10.20532/cit.2024.1005803

Reddy, J., Mundra, S., & Mundra, A. (2024). Ensembling Deep Learning Models for Fake News Classification. *Procedia Computer Science*, *235*, 2766–2774. https://doi.org/10.1016/j.procs.2024.04.261

Sabri, T., Bahassine, S., El Beggar, O., & Kissi, M. (2024a). An improved Arabic text classification method using word embedding. *International Journal of Electrical and Computer Engineering*, *14*(1), 721–731. https://doi.org/10.11591/ijece.v14i1.pp721-731

Sabri, T., Bahassine, S., El Beggar, O., & Kissi, M. (2024b). An improved Arabic text classification method using word embedding. *International Journal of Electrical and Computer Engineering*, *14*(1), 721–731. https://doi.org/10.11591/ijece.v14i1.pp721-731

Sastrawan, I. K., Bayupati, I. P. A., & Arsa, D. M. S. (2022). Detection of fake news using deep learning CNN–RNN based methods. *ICT Express*, *8*(3), 396–408. https://doi.org/10.1016/j.icte.2021.10.003

Setiawan, E., & Lestari, I. (2021). Stance Classification Pada Berita Berbahasa Indonesia Berbasis Bidirectional LSTM. *Journal Of Intelligent Systems And Computation*.

Shaker, N. H., & Dhannoon, B. N. (2024a). Word embedding for detecting cyberbullying based on recurrent neural networks. *IAES International Journal of Artificial Intelligence*, *13*(1), 500–508. https://doi.org/10.11591/ijai.v13.i1.pp500-508

Shaker, N. H., & Dhannoon, B. N. (2024b). Word embedding for detecting cyberbullying based on recurrent neural networks. *IAES International Journal of Artificial Intelligence*, *13*(1), 500–508. https://doi.org/10.11591/ijai.v13.i1.pp500-508

Sunan, R. A., K., H. F. E., & Aditya, C. S. K. (2024). Klasifikasi Hoax Berita Politik Menggunakan Algoritma Long Short-Term Memory (LSTM) dengan Penambahan Fitur Embedding Global Vector (GloVe). *Jurnal Edukasi Dan Penelitian Informatika (JEPIN)*, *10*(2), 287. https://doi.org/10.26418/jp.v10i2.76042

Vinit, B. S. (2020). An efficient fake news detector. *2020 International Conference on Computer Communication and Informatics, ICCCI 2020*. https://doi.org/10.1109/ICCCI48352.2020.9104177

Zahra, A., & Fauzan, M. N. (2022). Sistem Identifikasi "Fake News" menggunakan Metode Multinomial Naïve Bayes. *Jurnal Sistem Dan Teknologi Informasi (JustIN)*, *10*(4), 489. https://doi.org/10.26418/justin.v10i4.52441