Pendeteksian berita palsu menggunakan RoBERTa dengan Optimalisasi Word Embedding

Adisaputra Nur Arminta
Fakultas Informatika
Universitas Telkom, Bandung
adsputraaa@student.telkomuniversity.ac.id

Yuliant Sibaroni
Fakultas Informatika
Universitas Telkom, Bandung
yuliantsibaroni@telkomuniversity.ac.id

Abstrak

Penyebaran berita palsu (*hoax*) telah menjadi permasalahan serius yang mempengaruhi opini publik dan menciptakan polarisasi di masyarakat. Penelitian ini bertujuan untuk mendeteksi berita palsu menggunakan model *RoBERTa* yang dioptimalkan dengan tiga teknik *word embedding*. *Word embedding* yang digunakan adalah *RoBERTa*, Word2Vec, dan GloVe. Dataset yang digunakan adalah "Indonesian fact and hoax political news" yang diambil dari Kaggle, Dataset ini memerlukan tahap pre-processing untuk membersihkan ketidakkonsistenan data, seperti mengubah singkatan menjadi kata lengkap dan menghapus tanda baca. Selanjutnya, dilakukan representasi teks menggunakan tiga metode word embedding yaitu Word2Vec, GloVe, dan RoBERTa. Proses pelatihan model dilakukan dengan validasi silang K-Fold untuk meningkatkan generalisasi model. Hasil penelitian menunjukkan bahwa embedding RoBERTa mencapai akurasi terbaik 96%, sedangkan word embedding Word2Vec mendapatkan akurasi 94%. Word Embedding Glove menunjukkan performa paling rendah dengan akurasi 51%. Penelitian ini membuktikan bahwa pemilihan teknik word embedding yang tidak tepat untuk model RoBERTa dapat mengurangi akurasi dan efektivitas model dalam mendeteksi berita palsu. Diharapkan bahwa temuan dalam penelitian ini dapat memberikan kontribusi terhadap peningkatan sistem deteksi berita palsu di masa mendatang.

Kata kunci: hoax, RoBERTa, GloVe, Word2Vec

1. PENDAHULUAN

1.1 Latar Belakang

Hoax berasal dari *hocus* yaitu untuk menipu sering kali muncul pada topik yang sedang hangat dibicarakan. Tujuannya adalah untuk membujuk atau memanipulasi orang dan kemudian melakukan tindakan yang telah ditetapkan sebelumnya, biasanya menggunakan ancaman atau membuat mereka mempercayai hal-hal yang tidak nyata [1]. *Hoax* adalah informasi palsu yang beredar di berbagai platform dan saluran komunikasi, contohnya seperti media sosial dan situs web di internet.

Dengan berkembangnya teknologi digital, pengguna internet dapat dengan mudah membagikan informasi dan berinteraksi sesama pengguna, namun di sisi lain, informasi tersebut juga bisa disalahgunakan untuk menyebarkan berita palsu yang dapat memberikan dampak negatif kepada masyarakat [2]. Oleh karena itu, penyebaran hoaks di dunia maya perlu segera ditangani, karena dapat memicu ketegangan sosial, meningkatkan rasa permusuhan, dan bahkan menyebabkan konflik antar kelompok [3]. Dengan demikian, pengembangan sistem deteksi hoaks otomatis yang mampu mengidentifikasi serta menangani berita palsu secara cepat menjadi hal yang krusial sebelum informasi tersebut menyebar lebih luas.

Dalam upaya mengatasi penyebaran *hoax*, penggunaan salah satu metode yang menjanjikan untuk mengklasifikasi berita hoax adalah menggunakan *transformer* [4]. Salah satu metode *transformer* yaitu adalah *RoBERTa* (*Robustly Optimized BERT Approach*), *RoBERTa* adalah modifikasi dari *BERT* yang sederhana namun efektif [18]. Modifikasi tersebut meliputi melatih model lebih lama dengan *batch* yang lebih besar, menggunakan lebih banyak data, menghapus objektif prediksi kalimat berikutnya, melatih pada urutan yang lebih panjang, dan secara dinamis mengubah pola *masking* yang diterapkan pada data latih. Melalui modifikasi ini, *RoBERTa* dapat menghasilkan hasil terbaik pada berbagai tugas pemrosesan bahasa seperti *GLUE, RACE,* dan *SQuAD* [18].

Karena dalam beberapa penelitian sebelumnya telah dilakukan upaya untuk mendeteksi *hoax* menggunakan RoBERTa, Word2vec, dan GloVe. Salah satunya adalah Penelitian yang dilakukan oleh Vladislav Kolev et al. menggunakan model RoBERTa untuk mendeteksi berita palsu berdasarkan analisis emosi dalam judul berita. Model ini mencapai akurasi 88% pada Kaggle Fake and Real News Dataset [7]. Di sisi lain, penelitian oleh Bankar S memanfaatkan pendekatan *hybrid* dengan menggabungkan Word2Vec dan model LSTM untuk menganalisis hubungan semantik antar kata dalam teks berita palsu [8]. Lebih lanjut, penelitian yang dilakukan oleh Ryan Adipradana et al. menggunakan *embedding GloVe* dalam berbagai model jaringan saraf seperti LSTM dan GRU untuk mendeteksi berita palsu dalam bahasa Indonesia. Hasilnya menunjukkan bahwa *embedding GloVe* mampu memberikan performa yang baik dengan akurasi mencapai lebih dari 81% pada tugas klasifikasi berita [9]. Penelitian oleh Shafna et al. menunjukkan bahwa model *RoBERTa* mampu mendeteksi berita palsu dengan akurasi 83,3% pada dataset berbahasa Indonesia [10].

Maka dari itu, penulis mengusulkan penggunaan metode RoBERTa karena kemampuannya dalam menangkap konteks linguistik yang kompleks melalui pelatihan yang lebih intensif dan penggunaan data yang lebih besar [15]. Selain itu, untuk memperkuat pemahaman konteks dalam kalimat, penulis mengintegrasikan teknik Word Embedding seperti Word2Vec dan GloVe. Word2Vec dipilih karena kemampuannya dalam memetakan hubungan semantik antar kata berdasarkan distribusi kata dalam korpus [6], sementara GloVe dipilih karena kemampuannya dalam menangkap hubungan global antar kata melalui faktorisasi matriks ko-okurensi [21]. Integrasi ini diharapkan dapat meningkatkan performa model dalam mengidentifikasi berita hoax dan penerapan integrasi ini

diharapkan menjadi solusi yang relevan dan penting dalam mendeteksi hoax, khususnya di era media sosial yang mendorong percepatan penyebaran informasi palsu.

1.2 Topik dan Batasannya

Dalam penelitian ini, penulis merancang dan mengembangkan sebuah sistem untuk melakukan deteksi berita hoax berdasarkan data berita di Indonesia Menggunakan model *RoBERTa* dan *Word Embedding*. Penelitian ini memiliki beberapa Batasan, yakni: (1) Data yang digunakan berasal dari *dataset kaggle* yaitu "Indonesia Fact and Hoax Political News", dalam dataset tersebut data - data pada teks sudah diklasifikasian ke dalam dua kategori yaitu *Hoax* dan *Non-Hoax* ;(2) Data yang digunakan dalam penelitian ini adalah sebanyak 5000 dari total keseluruhan *dataset* tersebut.

1.3 Tujuan

Penelitian ini memiliki tujuan untuk mengembangkan sistem yang dapat mengenali berita hoaks dengan menggunakan model deep learning RoBERTa serta optimalisasi teknik word embedding. Selain itu, penelitian ini juga mengeksplorasi bagaimana penggunaan berbagai jenis word embedding, seperti *Word2Vec* dan *GloVe*, dapat memengaruhi kinerja model RoBERTa. Penilaian kinerja sistem akan dilakukan dengan menerapkan metrik evaluasi yang sesuai, seperti akurasi, presisi, recall, dan F1-score, guna menganalisis efektivitas masing-masing metode word embedding yang diterapkan.

2. KAJIAN TEORI

2.1 Penelitian Terkait

Sudah ada beberapa penelitian sebelumnya yang berfokus pada deteksi *hoax* di Indonesia dengan menggunakan berbagai metode pendekatan pemrosesan bahasa alami. Seperti penelitian yang dilakukan oleh Muliono et al. yang berjudul "Hoax Classification in Imbalanced Datasets Based on Indonesian News Title using RoBERTa" [19], menunjukkan bahwa penggunaan model RoBERTa untuk mengklasifikasikan berita hoaks berdasarkan judulnya saja menghasilkan akurasi yang tinggi, yaitu mencapai 99,52%. Crisanadenta et al. juga telah meneliti sebuah model deteksi hoax di twitter menggunakan metode *feed-forward* dan *back-propagation neural networks* [11]. Dengan menggunakan dua metode pembobotan fitur yaitu TF-IDF dan Word2Vec. Hasil penelitian menunjukkan bahwa model *neural networks* dengan TF-IDF mencapai akurasi tertinggi sebesar 78,76% [11].

Selanjutnya, penelitian yang dilakukan oleh Irena et al. menerapkan metode *Decision Tree* untuk mengidentifikasi berita palsu (*hoax*) di Twitter. Hasil penelitian menunjukkan bahwa penggunaan fitur n-gram dapat meningkatkan akurasi dalam mengklasifikasikan *hoax*. Peneliti menunjukkan bahwa penggunaan kombinasi unigram + bigram dan unigram + bigram + trigram memberikan nilai akurasi yang tinggi. Nilai akurasi tertinggi yang dicapai dalam penelitian tersebut adalah 72.91% dengan menggunakan metode *Decision Tree* C4.5, TF-IDF *weighting*, dan *Information Gain feature selection*, dengan rasio data latih 90% dan data uji 10% (90:10) serta menggunakan 5000 fitur unigram [12]. Adrian dan Prasetyowati juga melakukan penelitian yang dimana mereka membandingkan teknik *Word Embedding* yaitu Word2Vec dan GloVe. Hasil penelitian tersebut menunjukkan bahwa LSTM dengan Word2Vec memiliki kinerja yang lebih baik dibanding LSTM dengan GloVe dalam mendeteksi berita hoax dalam bahasa Indonesia. LSTM dengan Word2Vec mencapai nilai akurasi rata-rata sebesar 95%, sementara LSTM dengan GloVe hanya mencapai nilai akurasi rata rata sebesar 90% [13].

Mallik dan Kumar, menunjukan bahwa Word2Vec lebih unggul dari teknik *word embedding* lainnya karena dapat menghasilkan representasi vektor kata yang bebas konteks dan agnostik data, yang memungkinkan model untuk mengeksplorasi fitur tersembunyi dan hubungan di seluruh korpus teks. Word2Vec juga memungkinkan model untuk mendapatkan *embedding* yang lebih umum yang dapat diproses lebih lanjut untuk mengekstrak fitur khusus tugas [14].

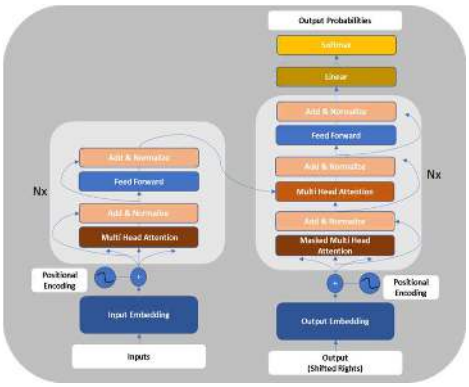
Dan yang terakhir, ada beberapa contoh kekurangan pada metode *BERT* saat digunakan pada kasus deteksi hoax menurut penelitian yaitu adalah kesulitan dalam menangani teks panjang, mengoptimalkan berbagai parameter, dan kebutuhan untuk menangkap hubungan antar entitas dalam format teks panjang dengan lebih baik untuk mendeteksi *hoax* [15]. Dengan modifikasi-modifikasi yang dilakukan pada *RoBERTa*, seperti pelatihan yang lebih lama dengan *batch* yang lebih besar dan lebih banyak data, penghapusan tujuan prediksi kalimat berikutnya yang tidak efektif, penggunaan sekuens yang lebih panjang serta penerapan pola *masking* dinamis selama pelatihan [19]. Maka dari itu penulis mengusulkan penggunaan metode *RoBERTa* yang mampu mengatasi keterbatasan BERT dan menggunakan teknik *Word Embedding* untuk lebih memperkuat pemahaman konteks dalam kalimat.

2.2 Hoax

Informasi yang tersebar di sosial media sangat beragam dan mudah diakses oleh jutaan pengguna di seluruh dunia. Namun, di balik kemudahan akses tersebut terdapat juga risiko penyebaran informasi palsu atau *hoax*. *Hoax* merupakan salah satu konsep yang berkaitan dengan informasi yang tidak benar. Secara umum, *hoax* mengacu pada berita yang dibuat atau disebarluaskan dengan tujuan untuk menyesatkan atau mengelabui para pembacanya. [16]. Dampak dari penyebaran *hoax* melalui sosial media sangat luas dan seringkali merugikan masyarakat. Hal ini dapat berdampak pada proses pengambilan keputusan, baik dalam skala pribadi maupun dalam lingkungan sosial yang lebih luas, karena informasi yang tidak akurat dapat mengganggu ingatan *semantic*. *Semantic* memori atau ingatan *semantic* mengacu pada pengetahuan umum atau pengetahuan tentang dunia yang disimpan dalam pikiran seseorang [17], oleh sebab itu peran *hoax* dalam sosial media sangat berbahaya karena bisa merusak integritas ingatan *semantic* individu dan masyarakat secara luas, ketika informasi palsu atau tidak akurat tersebar melalui sosial media hal tersebut dapat menyebabkan pergeseran atau pergantian pengetahuan yang benar dengan informasi yang salah. Hal ini dapat mengakibatkan individu masyarakat mempercayai hal-hal yang tidak benar tentang isu-isu yang relevan.

2.3 RoBERTa

RoBERTa (Robustly Optimized BERT Pretraining Approach) adalah pendekatan yang dihasilkan dari studi replikasi terhadap *BERT* (*Bidirectional Encoder Representations from Transformers*) oleh Yinhan et al. [18]. Dari hasil studi *RoBERTa* yang bertujuan untuk mengukur dampak variasi *hyperparameter* dan ukuran data latihan terhadap *pre-training* BERT, hasilnya menunjukkan bahwa BERT tidak sepenuhnya terlatih dengan baik [18]. Modifikasi utama pada *RoBERTa* melibatkan peningkatan proses *pretraining* untuk mengatasi keterbatasan *BERT* yang dianggap belum sepenuhnya optimal. Beberapa perubahan utama yang diterapkan pada *RoBERTa* adalah peningkatan jumlah data latih, *Batch Size* lebih besar, dan *training* lebih panjang [18]. RoBERTa hanya menggunakan bagian *encoder* dari arsitektur Transformer, sama seperti BERT. Hal ini menandakan bahwa RoBERTa tidak menggunakan bagian *decoder*, yang biasanya digunakan untuk tugas generasi teks seperti terjemahan atau pembuatan kalimat. Alasan RoBERTa hanya memakai *encoder* adalah karena model ini dirancang khusus untuk tugas-tugas pemahaman bahasa (*language understanding*) [18]. Gambar arsitektur *transformer RoBERTa* dapat dilihat pada gambar 1 [20].



GAMBAR 1
Arsitektur *Transformer*

Gambar 1 menunjukkan struktur umum dari arsitektur *transformer*, yang menjadi dasar dari model *RoBERTa*. *Transformer* terdiri dari *encoder* dan *decoder*. Setiap lapisan *encoder* memiliki komponen *Multi-Head Attention* yang memungkinkan model untuk mempelajari hubungan antara kata dalam satu urutan input dan *Feed Forward Network* (FFN).

- *Input Embedding*

Langkah pertama dalam arsitektur *transformer* adalah memproses barisan input yang terdiri dari token-token yang akan diproses. Token-token ini kemudian diubah menjadi representasi vektor dalam ruang fitur yang lebih abstrak menggunakan matriks *embedding*. Proses ini memungkinkan model untuk bekerja dengan representasi vektor yang lebih kompleks, abstrak, mudah diproses, dan dipahami oleh model. Representasi vektor ini menangkap makna semantik dari setiap token dan hubungannya dengan token lain dalam barisan input. Dengan mengubah barisan input menjadi representasi vektor, *transformer* dapat mempelajari hubungan antar kata dan makna kontekstualnya [20].

- *Positional Encoding*

Positional encoding layer bertanggung jawab untuk menambahkan informasi posisi relatif dari token dalam urutan data ke representasi vektor yang dihasilkan oleh input *embedding layer*. Dengan demikian, positional encoding layer memungkinkan model transformer untuk memperhatikan urutan token dalam data input, sehingga model dapat memahami konteks dan hubungan antara token dalam urutan data. Hal ini membantu dalam memperkuat representasi vektor dengan informasi posisi relatif yang diperlukan untuk mendukung proses transformasi data secara efektif [20].

- *Encoder Stack*

Saat memasuki encoder stack, data input melalui serangkaian lapisan encoder yang identik (N = 6). Perlu diingat bahwa di setiap sub-lapisan dalam encoder stack terdapat juga *mekanisme residual connections* [22] dan layer *normalization* [23]. Residual connections memungkinkan informasi untuk mengalir lebih lancar melalui lapisan-lapisan, sementara layer *normalization* membantu dalam menjaga stabilitas dan percepatan pelatihan model. Dengan demikian, saat memasuki *encoder stack* data input akan melalui serangkaian lapisan *encoder* yang masing-masing terdiri dari sub-lapisan *self-attention* dan *position-wise feed-forward networks*, yang kedua hal tersebut bekerja bersama untuk memproses dan memahami informasi dalam urutan data dengan cara yang efisien dan efektif. Sub-lapisan encoder itu sendiri ada dua yaitu sebagai berikut.

- *Multi-Head Self Attention*

Pada sub-lapisan ini, setiap token dalam urutan data dapat menghadap token lain dalam urutan yang sama. Dengan mekanisme *self-attention*, setiap token dapat memperhatikan informasi yang relevan dari token lain, memungkinkan model untuk memahami hubungan antara token dalam urutan data [20].

- *Position-wise Feed-Forward Networks*

Setelah melalui sub-lapisan *self-attention*, data kemudian diproses melalui jaringan feed- forward yang diterapkan pada setiap posisi secara terpisah dan identik. Hal ini memungkinkan model untuk memproses informasi lokal pada setiap posisi dalam urutan data dan membantu dalam ekstraksi fitur yang lebih kompleks [20].

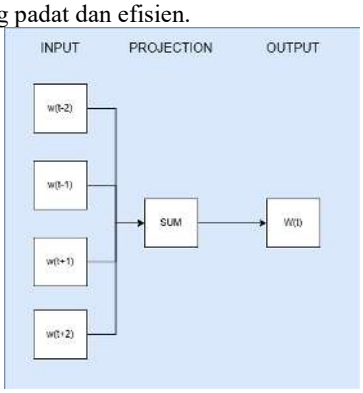
2.4 Word Embedding

Word embedding adalah teknik yang digunakan dalam pemrosesan bahasa alami (*NLP*) untuk memetakan kata-kata ke dalam vektor-vektor numerik berdimensi rendah. Proses ini memungkinkan model *machine learning* dan *deep learning* untuk memproses teks secara lebih efektif karena model *machine learning* memiliki keterbatasan dalam memahami bahasa alami. Model hanya dapat memproses informasi yang telah diubah menjadi format yang dapat mereka baca, yaitu angka [6]. Salah satu aspek penting dalam word embedding adalah jumlah dimensi vektor yang digunakan. Pemilihan dimensi yang optimal, seperti 300 dimensi, sangat berpengaruh terhadap kualitas representasi kata. Dimensi ini cukup tinggi untuk menangkap hubungan semantik dan sintaksis antar kata, namun tidak terlalu besar sehingga tetap efisien dalam pemrosesan komputasi. Dengan 300 dimensi, model dapat

menangkap lebih banyak konteks tanpa menyebabkan overfitting atau kebutuhan daya komputasi yang berlebihan [6].

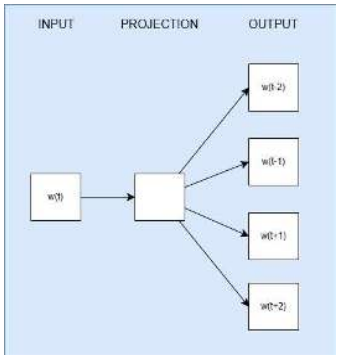
- *Word2Vec*

Word2Vec adalah salah satu metode populer untuk menghasilkan *Word embedding*. Menurut jurnal yang diteliti oleh S Sivakumar [24], Word2Vec memiliki dua arsitektur utama, yakni *Continuous Bag of Words* (CBOW) serta Skip-Gram. Model CBOW berfungsi untuk memperkirakan suatu kata berdasarkan kata-kata yang ada di sekelilingnya, sedangkan Skip-Gram bekerja dengan cara memprediksi kata-kata di sekitarnya menggunakan kata utama sebagai acuan. Kedua arsitektur ini memanfaatkan jaringan saraf untuk mempelajari representasi vektor dari kata- kata yang mempertahankan makna semantik dan hubungan kontekstual antar kata. Keunggulan utama dari Word2Vec dibandingkan teknik tradisional seperti *Bag of Words* (BOW) dan TF-IDF adalah kemampuannya untuk menghasilkan representasi vektor yang padat dan efisien.



GAMBAR 2
Tahapan CBOW

CBOW, seperti yang ditampilkan pada Gambar 2, adalah model yang didasarkan pada *Neural Network*. Algoritma *Continuous Bag of Words* (CBOW) merupakan salah satu model Neural Network sederhana namun sangat efisien. Model ini bertujuan untuk menghasilkan representasi vektor dari kata-kata dalam suatu bahasa, sehingga informasi dari setiap kata dapat direpresentasikan dalam bentuk ruang vektor. Seperti yang terlihat pada Gambar 3, model CBOW memprediksi kata tertentu w(t)w(t) berdasarkan konteks di sekitarnya, yaitu w(t−2),w(t−1),w(t+1),w(t+2)w(t-2), w(t- 1), w(t+1), w(t+2).



GAMBAR 3
Tahapan *Skip-gram*

Model skip-gram, seperti yang ditunjukkan pada Gambar 3, adalah pendekatan yang banyak digunakan untuk menghasilkan representasi kata dengan memanfaatkan hubungan antara suatu kata dan kata-kata di sekitarnya. Secara khusus, model ini dirancang untuk memprediksi konteks di sekitar kata tertentu w(t)w(t) dengan tujuan memaksimalkan akurasi prediksi. Model ini menghasilkan keluaran berupa kata-kata di sekitar w(t)w(t), yaitu w(t−2),w(t−1),w(t+1),w(t+2)w(t- 2), w(t-1), w(t+1), w(t+2), sebagaimana digambarkan pada Gambar 4. Dalam penelitian ini, salah satu arsitektur skip-gram akan digunakan untuk memprediksi kata-kata di sekitar kata masukan, yang dibatasi oleh ukuran jendela tertentu. Singkatnya, model skip-gram mampu memprediksi kata-kata yang berada di sekitar kata target.

- *GloVe*

GloVe (Global Vector) merupakan salah satu metode dalam *word embedding* yang berfungsi untuk mentransformasikan setiap kata dalam dokumen ke dalam bentuk vektor [21]. Teknik ini digunakan dengan tujuan memperoleh keterkaitan semantik antar kata berdasarkan matriks *co-occurrence*. *Word Embedding* GloVe membuat matriks kemunculan (*occurrence matrix*) dan akan membentuk *corpus* yang akan merepresentasikan hubungan antar kata. GloVe memudahkan juga akan memudahkan dalam pemrosesan pelatihan data dalam jumlah yang besar [21]. GloVe bekerja dengan membentuk sebuah matriks yang mencatat frekuensi kemunculan kata dalam berbagai konteks. Model ini kemudian menghasilkan output berupa kata-kata yang memiliki kemiripan makna, yang selanjutnya digunakan dalam proses pengembangan vektor. Glove memiliki rumus seperti persamaan (1) dan (2).

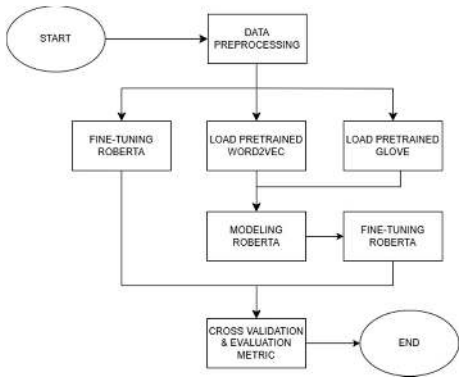
$$w^{r} + w^{\rightarrow\rightarrow\rightarrow}{}_{k}\rightarrow + b_{i} + b^{\rightarrow\rightarrow\rightarrow}\rightarrow = log(X_{ik}) \quad (1)$$

Keterangan
W = word vector
$w^{\rightarrow}\rightarrow$ = word context vector
b_i = major scalar bias
b_k = word context scalar biar X = emergence matrix

Dari persamaan di atas, kita akan mendapatkan fungsi pembobotan ke dalam fungsi biaya, yang menghasilkan persamaan

3. METODE

Dalam merancang sistem pendeteksi berita palsu, pengujian kinerja metode *embedding* menggunakan *RoBERTa* dilakukan melalui beberapa tahapan, seperti terlihat pada Gambar 4.

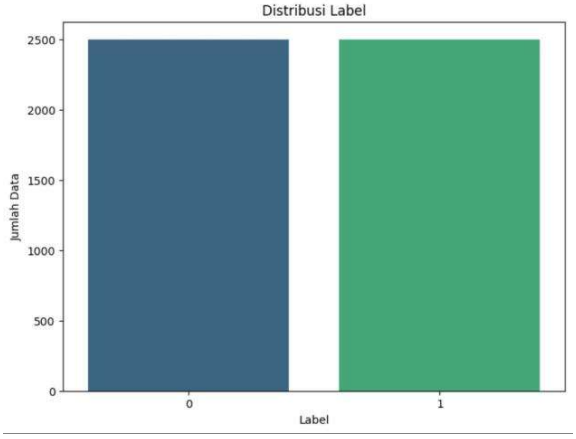


GAMBAR 4
Rancangan Sistem

Penelitian ini diawali dengan tahap pengolahan data untuk memastikan kesesuaian data dengan kebutuhan model. Setelah data siap, penelitian dilanjutkan dengan tiga skenario utama. Pada skenario pertama, model *RoBERTa* digunakan tanpa modifikasi. Skenario kedua mengganti *embedding RoBERTa* dengan *embedding* yang dihasilkan dari *pre-trained* Word2Vec, kemudian *embedding* tersebut diintegrasikan ke dalam model RoBERTa. Skenario ketiga juga mengganti *embedding RoBERTa*, tetapi menggunakan *embedding* dari *pre-trained* GloVe. Setelah salah satu skenario diterapkan, proses dilanjutkan dengan pemodelan menggunakan *RoBERTa* untuk mendeteksi berita palsu. Model yang telah dibangun kemudian divalidasi menggunakan *cross-validation* guna memastikan konsistensi hasil. Tahap akhir penelitian adalah pengukuran performa ketiga skenario tersebut menggunakan matriks evaluasi seperti *Accuracy*, *Precision*, *Recall*, dan *F1-Score* untuk menilai kemampuan model dalam membedakan berita asli dan berita palsu.

3.1 Dataset

Dataset yang digunakan dalam penelitian ini adalah "*Indonesia False News (Hoax) Dataset*" yang diambil dari *Kaggle*. *Dataset* ini terdiri dari 5000 data berita dalam bahasa Indonesia, yang terbagi menjadi 2500 data *hoax* dan 2500 data non-hoax seperti pada gambar 1. *Dataset* ini tidak memerlukan proses pelabelan tambahan karena seluruh data sudah diberi label "*hoax*" atau "*non-hoax*" oleh penyedia *dataset*.



GAMBAR 5
Distribusi Dataset

3.2 Preprocessing

Proses preprocessing data sangat penting dalam persiapan data untuk model pembelajaran mesin. Ini membantu meningkatkan kualitas data dan memastikan bahwa model dapat mempelajari pola dengan lebih efektif. Berikut adalah langkah-langkah dalam preprocessing data untuk penelitian pendeteksian berita palsu.

1. Pembersihan Data

Pada tahap pembersihan data, tujuan utamanya adalah menghilangkan bagian-bagian yang tidak diperlukan dari teks. Pertama, tanda baca seperti titik, koma, tanda seru, dan lainnya dihapus karena tidak memberikan nilai tambah untuk memahami isi teks. Selain itu, angka-angka panjang, misalnya nomor telepon, juga dibuang karena tidak ada kaitannya dengan analisis teks. Terakhir, jika ada spasi berlebihan yang muncul setelah pembersihan, spasi tersebut disederhanakan menjadi satu spasi saja agar teks terlihat lebih rapi. Semua langkah ini dilakukan untuk memastikan data yang digunakan bersih dan hanya berisi informasi yang benar-benar penting untuk analisis selanjutnya.

2. Tokenisasi

Tokenisasi adalah proses memecah teks menjadi bagian-bagian kecil yang disebut token, biasanya berupa kata-kata individual. Tujuannya adalah untuk memisahkan teks menjadi unit-unit yang lebih mudah diproses. Misalnya, dalam kalimat "anjing asal indonesia ikut selamatkan korban gempa turki", tokenisasi akan memecahnya menjadi ["anjing", "asal", "indonesia", "ikut", "selamatkan", "korban", "gempa", "turki"]. Dengan memisahkan teks menjadi kata-kata terpisah, model atau sistem dapat menganalisis setiap kata secara lebih detail.

3. Normalisasi Teks

Normalisasi teks adalah proses untuk menyamakan format teks dan memperbaiki kata-kata yang tidak standar. Pertama, semua huruf diubah menjadi huruf kecil agar seragam. Misalnya, kata "Pemilu" dan "pemilu" akan dianggap sama setelah proses ini. Selain itu, kata-kata yang mengandung angka atau penulisan tidak baku, seperti “4n4k” atau “kmu”, diperbaiki menjadi bentuk yang benar, seperti “anak” atau “kamu”. Ini dilakukan dengan menggunakan daftar kata yang sudah disiapkan, di mana setiap kata tidak baku dipetakan ke kata yang standar. Dengan langkah-langkah ini, teks menjadi lebih rapi dan siap untuk dianalisis lebih lanjut.

3.3 Pretrained Word Embedding

Word embedding berfungsi dalam mentransformasikan teks ke dalam bentuk vektor numerik sehingga dapat diolah lebih lanjut oleh model. Dalam tugas ini, digunakan tiga metode *embedding*, yaitu *RoBERTa*, Word2Vec, dan *GloVe*. Pada *pre-trained* Word2Vec dan GloVe, *embedding* digunakan untuk menghasilkan vektor dengan 300 dimensi. Dalam prosesnya, teks terlebih dahulu ditokenisasi menjadi kata-kata, kemudian setiap kata dicocokkan dengan *vocabulary* dari model yang digunakan. Jika kata ditemukan dalam *vocabulary* Word2Vec atau *dictionary* GloVe, maka vektor yang sesuai akan diambil. Setelah semua kata dalam teks dikonversi ke vektor, vektor-vektor tersebut kemudian dirata-ratakan untuk membentuk satu representasi vektor dari teks tersebut. Yang terakhir adalah *word embedding RoBERTa*, di mana *word embedding* sudah dilakukan

secara otomatis oleh model melalui tokenisasi dan representasi kontekstual. Berbeda dengan Word2Vec dan GloVe yang menghasilkan *embedding* statis, *RoBERTa* menghasilkan *embedding* berdimensi 768 untuk setiap token, dengan vektor yang berubah sesuai konteks dalam teks. Representasi kontekstual ini membuat *RoBERTa* lebih unggul dalam memahami hubungan antar kata dan makna kalimat secara keseluruhan. Hasil dari masing–masing teknik *word embedding* dapat dilihat pada tabel 1, 2, dan 3.

TABEL 1
Output Word2Vec

Teks	WordPair	Caption
anjing asal indonesia ikut selamatkan korban gempa turki	(anjing, asal), (anjing, indonesia)	Input : anjing Target : asal, indonesia
anjing asal indonesia ikut selamatkan korban gempa turki	(asal, anjing), (asal, indonesia), (asal, ikut)	Input : asal Target : anjing indonesia, ikut
anjing asal indonesia ikut selamatkan korban gempa turki	(indonesia, anjing), (indonesia, asal), (indonesia, ikut), (indonesia, selamatkan)	Input : indonesia Target : anjing, asal. ikut, Selamatkan
anjing asal indonesia ikut selamatkan korban gempa turki	(ikut, asal), (ikut,indonesia), (ikut,selamatkan), (ikut,korban)	Input : ikut Target : anjing, asal, indonesia, selamatkan, Korban
anjing asal indonesia ikut selamatkan korban gempa turki	(selamatkan, indonesia), (selamatkan, ikut), (selamatkan, korban) (selamatkan, gempa)	Input : indonesia Target : anjing, asal. ikut, Selamatkan
anjing asal indonesia ikut selamatkan korban gempa turki	(korban, ikut), (korban, selamatkan), (korban, gempa), (korban, turki)	Input : korban Target : anjing, asal, indonesia, ikut, selamatkan, gempa, turki
anjing asal indonesia ikut selamatkan korban gempa turki	(gempa, selamatkan), (gempa, korban), (gempa, turki)	Input : gempa Target : selamatkan, korban, turki
anjing asal indonesia ikut selamatkan korban gempa turki	(turki, korban), (turki, gempa),	Input : turki Target : korban, gempa

Word2Vec bekerja dengan menebak kata berdasarkan kata-kata di sekitarnya menggunakan dua metode, yaitu CBOW (*Continuous Bag of Words*) dan Skip-gram. Dalam hasil tabel 1, terlihat bahwa kata "anjing" sering dipasangkan dengan kata "asal" dan "indonesia", karena dalam *dataset* yang digunakan, kata-kata ini sering muncul bersama dan memiliki makna yang saling terkait. Word2Vec dapat memahami hubungan antara kata-kata yang berdekatan dalam sebuah kalimat atau paragraf, tetapi karena vektor yang dihasilkan selalu tetap, makna kata tidak bisa berubah sesuai dengan konteksnya. Ini membuat Word2Vec bagus dalam menangkap hubungan kata dalam skala kecil, tetapi kurang fleksibel dalam memahami makna kata dalam berbagai situasi atau kalimat yang lebih kompleks.

TABEL 2
Output GloVe

Teks	Label	Output
anjing asal indonesia ikut selamatkan korban gempa turki	1	[6.24440014e-02 - 1.83347508e-01 1.82855517e-01…]
pesan whatsapp dari bmkg yang kabarkan gunung sinabung meletus	1	[-0.345855 0.0992025 0.11425751...]
pensiunan jenderal tni al darajat hidayat gabung pks jelang pemilu	0	[-4.36100513e-02 4.33248840e-03 1.64178997e-01...]
anies dijadwalkan jadi pembicara acara partai ummat	0	[0.2029255 - 0.2755925 0.38128…]

Berbeda dengan Word2Vec, GloVe memberikan angka yang jelas untuk setiap kata, termasuk "anjing". Misalnya, angka pertama dalam vektor "anjing" (0.0624) menunjukkan bahwa kata ini memiliki hubungan yang cukup kuat dengan konsep tertentu, seperti "hewan". Sementara itu, angka negatif di dimensi lain (-0.1833) bisa berarti bahwa kata "anjing" kurang terkait atau bahkan tidak berhubungan dengan konsep lain, seperti "politik" atau "ekonomi". Cara kerja GloVe adalah dengan menghitung seberapa sering kata-kata muncul bersama dalam jumlah besar, sehingga hasilnya selalu tetap. Artinya, tidak peduli dalam kalimat apa kata "anjing" muncul, representasi angkanya akan selalu sama dan tidak berubah sesuai konteks.

TABEL 3
Output Word Embedding RoBERTa

Teks	Label	Output
anjing asal indonesia ikut selamatkan korban gempa turki	1	[-7.31544197e-02 3.55973132e-02 1.06571177e-02]
pesan whatsapp dari bmkg yang kabarkan gunung sinabung meletus	1	[-3.45693342e-02 4.39395867e-02 2.05307435e-02...]
pensiunan jenderal tni al darajat hidayat	0	[-3.59291807e-02 2.97025219e-02 1.63022727e-02...]

gabung pks jelang pemilu		
anies dijadwalkan jadi pembicara acara partai ummat	0	[-4.05475162e-02 5.62006533e-02 1.77119914e-02…]

RoBERTa memberikan arti kata yang bisa berubah tergantung pada kalimatnya. Misalnya, jika kata "anjing" muncul dalam kalimat "anjing laut berenang di pantai", maka maknanya akan lebih dekat dengan kata "laut". Namun, jika kata "anjing" ada dalam berita penyelamatan, seperti "anjing membantu selamatkan korban gempa", maka hubungannya akan lebih kuat dengan kata "selamatkan". Berbeda dengan Word2Vec dan GloVe yang selalu memberi angka tetap untuk setiap kata, RoBERTa bisa menyesuaikan bobotnya sesuai dengan kata-kata lain di sekitarnya. Model ini menggunakan teknik khusus yang disebut self-attention, yang memungkinkan mesin memahami bagaimana kata-kata dalam satu kalimat saling berhubungan. Jadi, dalam konteks biologi, "anjing" mungkin lebih dekat dengan kata "mamalia" atau "karnivora", sementara dalam konteks penyelamatan, "anjing" lebih berkaitan dengan "menolong" atau "bertindak".

3.4 Modeling RoBERTa

Pada tahap Modeling *RoBERTa*, model dikembangkan untuk mengklasifikasikan berita palsu dengan memanfaatkan *embedding* dari Word2Vec atau *GloVe* sebagai input. Tidak seperti *RoBERTa* pada umumnya yang melakukan tokenisasi dan *embedding* secara otomatis, di sini model menerima *embedding* statis berdimensi 300, yang kemudian diubah ke 768 dimensi menggunakan lapisan linear agar sesuai dengan arsitektur *RoBERTa*. *Embedding* yang telah ditransformasikan ini dimasukkan ke dalam *RoBERTa* melalui *inputs embeds*, sehingga model dapat mengekstraksi fitur dari teks tanpa melakukan tokenisasi sendiri. Hasil keluaran *RoBERTa* kemudian diproses melalui lapisan klasifikasi, yang terdiri dari transformasi dimensi, aktivasi ReLU, *Dropout* untuk mencegah *overfitting*, dan lapisan output yang menghasilkan prediksi akhir untuk dua kelas (*hoax* atau tidak *hoax*). Pendekatan ini memungkinkan *RoBERTa* tetap berfungsi sebagai *feature extractor*, sementara *embedding* eksternal seperti Word2Vec atau *GloVe* digunakan sebagai representasi awal teks.

3.5 Fine-Tuning RoBERTa

Fine-tuning adalah proses penyesuaian model pembelajaran mesin yang sudah dilatih sebelumnya dengan tujuan meningkatkan kemampuannya dalam menangani tugas tertentu. Dalam konteks deteksi hoax, fine-tuning dilakukan dengan cara melatih model yang sudah ada menggunakan data berita yang spesifik dan relevan. Proses ini dimulai dengan menginisialisasi model baru untuk memastikan hasil pelatihan tidak terpengaruh oleh iterasi sebelumnya. Selama pelatihan, model diperbaiki dengan cara mempelajari pola dan karakteristik dari data yang diberikan, sehingga ia dapat menjadi lebih akurat dalam mengidentifikasi berita yang benar atau *hoax*. Setelah model dilatih, langkah penting berikutnya adalah evaluasi untuk mengukur seberapa baik model tersebut dapat bekerja pada data yang belum pernah dilihat sebelumnya. Dengan melakukan *fine-tuning* secara sistematis, model dapat ditingkatkan performanya, sehingga lebih efektif dalam mendeteksi berita *hoax* dan memberikan hasil yang lebih dapat diandalkan.

3.6 Cross Validation & Evaluation Metric

Evaluasi model dimulai dengan memilih metrik yang sesuai, seperti akurasi, presisi, *recall*, dan F1-score. Berikut ini adalah penjelasan mengenai masing-masing metrik beserta rumusnya. Akurasi berfungsi untuk mengukur seberapa sering model menghasilkan prediksi yang tepat terhadap seluruh kelas yang ada. Pengukuran ini mencakup hasil prediksi yang benar baik untuk kategori positif maupun negatif. Dalam hal ini, TP mengacu pada *True Positive*, yaitu kondisi ketika model memprediksi positif dengan benar, sedangkan TN merujuk pada *True Negative*, yakni saat model mengidentifikasi negatif dengan benar. Sementara itu, FP atau *False Positive* terjadi ketika model secara keliru mengklasifikasikan negatif sebagai positif, dan FN atau *False Negative* muncul saat model salah mengklasifikasikan positif sebagai negatif.

TABEL 4
Counfussion Matrix

Klasifikasi	Positive Prediction	Negative Prediction
Actual Positive	True Positive (TP)	False Negative (FN)
Actual Negative	False Positive (FP)	True Negative (TN)

Keterangan Tabel 4 :
TP = menunjukkan jumlah contoh positif yang diklasifikasikan secara akurat. TN = menunjukkan jumlah contoh negatif yang diklasifikasikan secara akurat. FP = jumlah contoh negatif aktual yang diklasifikasikan sebagai positif.
FN = adalah jumlah contoh positif aktual yang diklasifikasikan sebagai negatif.
Pada pengukuran *confussion matrix* ini menggunakan tingkat akurasi, presisi, recall, dan nilai F1-Score. Akurasi adalah variabel yang akan menggambarkan seberapa akurat model klasifikasi. Rumus akurasi yang akan digunakan adalah seperti persamaan (3).

$$\text{Akurasi} = \frac{(TP+TN)}{(TP+FP+TN+F\ )} \quad (3)$$

Presisi mengukur seberapa banyak dari prediksi positif yang benar-benar positif, dapat dilihat pada persamaan (4). Ini memberi tahu seberapa sering model benar ketika memprediksi kelas positif. Sedangkan Recall adalah metrik yang mengukur kemampuan model dalam menemukan semua positif yang sebenarnya. Recall dapat dilihat pada persamaan (5).

$$Precision = \frac{TP}{TP+F} \quad (4)$$

$$Recall = \frac{TP}{TP+FN} \quad (5)$$

F1-Score adalah rata-rata harmonis dari presisi dan *recall*, memberikan skor tunggal yang menggabungkan kedua metrik tersebut, dapat dilihat pada persamaan (6). Ini mengambil kedua metrik ke dalam pertimbangan dan berguna ketika ingin mencari keseimbangan antara presisi dan *recall*.

$$f1 - Score = \frac{2(precision*recall)}{(precision+rec)} \quad (6)$$

Semakin tinggi nilai akurasi, presisi, *recall*, dan *F1-Score*, semakin baik kinerja modelnya. Namun, perlu memeriksa setiap metrik secara terpisah karena terkadang terdapat keseimbangan antara presisi dan *recall*.

Cross-validation adalah teknik evaluasi model yang memungkinkan kita untuk lebih memahami bagaimana model akan berfungsi pada data yang belum pernah dilihat sebelumnya. Proses kerja *cross-validation* melibatkan beberapa iterasi, di mana setiap iterasi menggunakan partisi data yang berbeda sebagai data pengujian, sementara partisi lainnya digunakan sebagai data pelatihan. Dalam 10 *fold cross-validation*, dataset dibagi menjadi 10 bagian (fold). Pada iterasi pertama, *fold* pertama digunakan sebagai data pengujian, sementara 9 *fold* lainnya digunakan untuk pelatihan. Penggunaan 10 *fold cross-validation* dipilih karena memberikan keseimbangan yang baik antara bias dan varians, menghasilkan estimasi kinerja model yang lebih stabil dan akurat dibandingkan metode pembagian data secara langsung [5].
Proses ini berlanjut hingga setiap fold telah bergantian berfungsi sebagai data pengujian satu kali, dengan total 10 iterasi. Hal ini memastikan bahwa model dilatih dan dievaluasi secara menyeluruh melalui seluruh dataset, sehingga hasil akhir dari evaluasi model lebih robust dan dapat diandalkan. Dalam setiap *fold*, model dilatih selama 10 epoch. Setiap epoch mengacu pada satu siklus lengkap dari pemrosesan data pelatihan, di mana model melalui semua batch data pelatihan dan melakukan pembaruan bobot berdasarkan loss yang dihitung. Dari sini, setiap *fold* menghasilkan evaluasi metrik yang berbeda, seperti akurasi, *precision*, *recall*, dan *f1-score*, yang kemudian dapat dirata-rata untuk mendapatkan gambaran umum tentang performa model di seluruh dataset. Dengan cara ini, cross-validation dapat membantu dalam menghindari overfitting dan memberikan estimasi yang lebih akurat tentang efektivitas model saat diterapkan pada data baru.

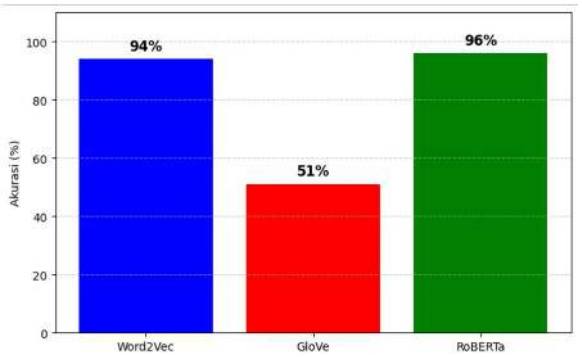
TABEL 5
Cross Validation

Fold 1	Test	Train	Train	Train	Train	Train	Train	Train	Train	Train
Fold 2	Train	Test	Train	Train	Train	Train	Train	Train	Train	Train
Fold 3	Train	Train	Test	Train	Train	Train	Train	Train	Train	Train
Fold 4	Train	Train	Train	Test	Train	Train	Train	Train	Train	Train
Fold 5	Train	Train	Train	Train	Test	Train	Train	Train	Train	Train
Fold 6	Train	Train	Train	Train	Train	Test	Train	Train	Train	Train
Fold 7	Train	Train	Train	Train	Train	Train	Test	Train	Train	Train
Fold 8	Train	Train	Train	Train	Train	Train	Train	Test	Train	Train
Fold 9	Train	Train	Train	Train	Train	Train	Train	Train	Test	Train
Fold 10	Train	Train	Train	Train	Train	Train	Train	Train	Train	Test

4. HASIL DAN PEMBAHASAN

4.1 Hasil Pengujian

Pada penelitian ini, model utama yang digunakan adalah *RoBERTa*, yang diuji dengan tiga skenario berbeda untuk mengevaluasi pengaruh metode *word embedding* terhadap performa *RoBERTa*. Skenario pertama adalah dengan hanya menggunakan RoBERTa., yang secara otomatis menghasilkan *embedding* kontekstual untuk setiap token. Skenario kedua menggunakan *embedding* dari *pre-trained* Word2Vec, dimana teks diubah menjadi vektor berdimensi 300, lalu dipetakan ke dimensi 768 agar sesuai dengan arsitektur *RoBERTa*. Skenario ketiga menggunakan *embedding* dari *pre-trained* GloVe, yang memiliki proses serupa dengan Word2Vec dalam mengonversi teks menjadi vektor numerik.
Ketiga metode *embedding* ini digunakan dalam *Natural Language Processing* (NLP) untuk membantu model memahami teks dalam bentuk numerik. Perbandingan ketiga skenario bertujuan untuk menganalisis bagaimana penggunaan teknik *word embedding* yang berbeda mempengaruhi performa RoBERTa dalam mengenali pola dan hubungan semantik dalam teks. Hasil akurasi *RoBERTa* berdasarkan metode *embedding* yang digunakan dapat dilihat pada Tabel 6.



GAMBAR 6
Perbandingan Akurasi

Pada gambar 6. Terlihat bahwa akurasi tertinggi diperoleh oleh RoBERTa, mencapai nilai akurasi 96%. Sementara itu, model GloVe mencatatkan nilai akurasi terendah yaitu 51%. Di sisi lain, akurasi yang diperoleh menggunakan word2vec adalah 94%, yang hampir menyentuh akurasi RoBERTa.
Dapat dilihat pada Tabel 6. *K-Fold* pada setiap model

memiliki berbagai hasil akurasi. Akurasi yang diperoleh dengan *word embedding* model Word2Vec pada *fold* kesembilan sebesar 96%. Sedangkan akurasi yang diperoleh dengan menggunakan *word embedding* model *Glove* mendapatkan hasil tertinggi pada fold kelima, kedelapan dan sembilan sebesar yaitu 90% dan akurasi tertinggi dengan menggunakan *word embedding* bawaan *RoBERTa*, yaitu sebesar 98%.

TABEL 6
Akurasi RoBERTa dengan *Word Embedding*

Fold	Akurasi		
	Word2Vec embedding	Glove embedding	Roberta embedding
1	91%	53%	94%
2	94%	50%	96%
3	96%	63%	98%
4	93%	46%	96%
5	95%	50%	96%
6	95%	48%	95%
7	94%	52%	95%
8	95%	52%	95%
9	95%	50%	97%
10	94%	45%	95%
Rata-rata	94%	51%	96%

4.2 Analisis Hasil Pengujian

Setiap skema penelitian menggunakan tiga metode word embedding didapatkan hasil *confusion matrix* seperti yang tertera pada tabel 5. Temuan evaluasi yang diperoleh dari *Confusion Matrix* ini memberikan nilai *True Positive* (TP), *True Negative* (TN), *False Negative* (FN), dan *False Positive* (FP) dari setiap skema EAT yang dijalankan.

TABEL 7
Hasil Pengujian

Model Embedding	Akurasi	F1-Score	Recall	Precision
Word2Vec	94%	94%	93%	95%
GloVe	51%	47%	70%	47%
RoBERTa	96%	96%	95%	96%

Pada Tabel 5, Hasil evaluasi menunjukkan bahwa penggunaan word embedding RoBERTa menghasilkan performa terbaik, dengan akurasi 96%, precision 96%, recall 95%, dan F1-score 96%. Keunggulan ini terjadi karena RoBERTa dirancang dengan embedding kontekstual yang dinamis, yang memungkinkan model menangkap makna kata berdasarkan konteksnya dalam teks. Hal ini sangat penting dalam tugas deteksi berita palsu, di mana pemahaman konteks memainkan peran besar dalam membedakan antara berita asli dan berita palsu.

Sebaliknya, penggunaan GloVe menghasilkan performa terendah, dengan akurasi hanya 51%, precision 47%, recall 70%, dan F1-score 47%. Performa yang rendah ini dapat dijelaskan oleh beberapa faktor. Pertama, sifat embedding GloVe yang statis menyebabkan setiap kata memiliki representasi tetap, sehingga tidak dapat menyesuaikan maknanya berdasarkan konteks dalam kalimat. Akibatnya, model kesulitan menangkap variasi makna kata yang bergantung pada konteks. Kedua, terdapat ketidakcocokan antara embedding statis dan model berbasis transformer seperti RoBERTa. RoBERTa bekerja dengan embedding kontekstual, sehingga menggantinya dengan embedding statis seperti GloVe dapat menyebabkan kehilangan informasi penting yang biasanya ditangkap oleh mekanisme self-attention. Ketiga, embedding GloVe kemungkinan memasukkan noise atau informasi yang tidak relevan, yang justru mengurangi akurasi model dalam mendeteksi berita palsu.

Sementara itu, penggunaan Word2Vec menghasilkan akurasi 94%, precision 95%, recall 93%, dan F1-score 94%, yang lebih baik dibandingkan GloVe tetapi masih di bawah embedding bawaan RoBERTa. Word2Vec memiliki beberapa keunggulan dibandingkan GloVe dalam konteks ini. Word2Vec lebih mampu menangkap hubungan semantik antar kata, karena metode pembelajarannya berbasis prediksi kata berdasarkan konteks sekitarnya, baik melalui CBOW maupun Skip-gram. Selain itu, embedding yang dihasilkan Word2Vec lebih fleksibel dibandingkan GloVe, meskipun tetap bersifat statis. Representasi kata dalam Word2Vec lebih mencerminkan distribusi kata dalam korpus pelatihannya, membuatnya lebih kompatibel dengan model seperti RoBERTa dibandingkan GloVe. Namun, meskipun Word2Vec lebih unggul dibandingkan GloVe, ia tetap kalah dari embedding bawaan RoBERTa, karena tidak bersifat kontekstual. Word2Vec masih menggunakan vektor tetap untuk setiap kata, sedangkan RoBERTa menyesuaikan makna kata sesuai dengan konteksnya dalam kalimat.

5. KESIMPULAN

Penelitian ini membandingkan tiga metode *embedding word* yaitu Word2Vec, GloVe dan *RoBERTa*, yang telah dicoba pada model RoBERTa, dengan menggunakan 3 skenario *word embedding* yang berbeda-beda. Kami telah mencoba ketiga metode tersebut. Dalam metode GloVe, hasil yang didapat mengindikasikan bahwa pendekatan *word embedding* ini menunjukkan kinerja paling rendah jika dibandingkan dengan kedua metode lainnya. Hal ini terlihat dari nilai akurasi sebesar 51%, precision 47%, recall 70%, dan f1-score sebesar 47%. Sementara itu, metode Word2Vec memberikan hasil kedua terbaik dengan akurasi sebesar 94%, precision 95%, recall 93%, dan f1-score sebesar 94%. Adapun metode *embedding* bawaan dari model *RoBERTa* menunjukkan performa terbaik dengan akurasi sebesar 96%, precision 96%, *recall* 95%, dan f1-score sebesar 96%.

Berdasarkan penelitian ini, dapat disimpulkan bahwa pemilihan metode *word embedding* sangat mempengaruhi kinerja model dalam mendeteksi berita palsu. GloVe menunjukkan performa paling rendah, mengindikasikan bahwa metode ini kurang efektif dalam menangkap hubungan semantik yang kompleks dalam teks berita. Sebaliknya, Word2Vec menunjukkan performa yang lebih baik dibandingkan GloVe, meskipun masih kalah dibandingkan dengan *embedding* bawaan *RoBERTa*. Metode *embedding* bawaan dari model *RoBERTa*

memberikan hasil terbaik, menunjukkan keunggulan pendekatan berbasis *transformer* dalam memahami konteks secara lebih mendalam. Secara keseluruhan, temuan ini menunjukkan bahwa penggunaan *embedding* yang lebih canggih, seperti *RoBERTa*, dapat meningkatkan keakuratan model dalam tugas pemrosesan bahasa alami, khususnya dalam mendeteksi berita palsu**.**

Untuk penelitian selanjutnya, disarankan untuk mengeksplorasi berbagai metode embedding lainnya seperti FastText, ELMo, dan BERT guna menemukan pendekatan yang lebih efektif dalam mendeteksi berita palsu. Penelitian juga dapat mempertimbangkan fine-tuning pada model RoBERTa serta menerapkannya pada dataset yang berbeda untuk menguji robustnes. Selain itu, penting untuk menganalisis fitur tambahan, seperti ciri-ciri linguistik dan metadata, yang dapat meningkatkan akurasi model. Melakukan studi kualitatif juga dapat membantu memahami kekuatan dan kelemahan dari masing-masing metode embedding melalui analisis kesalahan. Evaluasi terhadap waktu proses dan efisiensi sumber daya juga harus diperhatikan seiring dengan peningkatan kompleksitas model. Terakhir, perluasan penelitian untuk mencakup pengujian dalam berbagai bahasa dan konteks budaya akan memberi wawasan lebih luas mengenai tantangan global dalam mendeteksi berita palsu.

REFERENSI

[1] J. C. Hernández dkk, "A first step towards automatic hoax detection," dalam IEEE Annual International Carnahan Conference on Security Technology, Proceedings, 2002, hlm. 102–114. doi: 10.1109/ccst.2002.1049234.

[2] S. Zannettou dkk, "The Web of False Information: Rumors, Fake News, Hoaxes, Clickbait, and Various Other Shenanigans," Apr. 2018, doi: 10.1145/3309699.

[3] K. Pol dkk, "Pencegahan Hoax di Media Sosial Guna Memelihara Harmoni Sosial," 2019.

[4] W. C. Chang dkk, "Taming Pretrained Transformers for Extreme Multi-label Text Classification," dalam Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Association for Computing Machinery, Agu. 2020, hlm. 3163–3171. doi: 10.1145/3394486.3403368.

[5] Wijiyanto dkk., "Teknik K-Fold Cross Validation untuk Mengevaluasi Kinerja Mahasiswa," Jurnal Algoritma, vol. 21, no. 1, hlm. 239–248, Mei 2024. doi: 10.33364/algoritma/v.21-1.1618.

[6] S. D. Oktaviani dkk, "Perbandingan Kinerja Word Embedding Word2Vec, GloVe, dan FastText dalam Klasifikasi Teks," Jurnal Teknokompak, vol. 14, no. 2, hlm. 45–52, 2022. Tersedia di: https://ejurnal.teknokrat.ac.id/index.php/teknokompak/article/download/732/462.

[7] V. Kolev dkk, "FOREAL: RoBERTa Model for Fake News Detection based on Emotions," dalam International Conference on Agents and Artificial Intelligence, Science and Technology Publications, Lda, 2022, hlm. 429–440. doi: 10.5220/0010873900003116.

[8] S. Bankar dan S. Gupta, "Fake News Detection Using LSTM-Based Deep Learning Approach and Word Embedding Feature Extraction," 2023, hlm. 129–141. doi: 10.1007/978-981-99-1699-3_8.

[9] R. Adipradana dkk, "Hoax Analyzer for Indonesian News Using RNNs with FastText and GloVe Embeddings," Bulletin of Electrical Engineering and Informatics, vol. 10, no. 4, hlm. 2130-2136, Agu. 2021. doi: 10.11591/eei.v10i4.2956.

[10] S. F. N. Azizah dkk, "Performance Analysis of Transformer Based Models (BERT, ALBERT and RoBERTa) in Fake News Detection," dalam Proceedings of Universitas Sebelas Maret, 2023, hlm. 1–6. Tersedia di: https://github.com/Shafna81/fakenewsdetection.git.

[11] C. W. Kencana dkk, "Sistem Deteksi Hoax pada Twitter dengan Metode Klasifikasi Feed-Forward dan Back-Propagation Neural Networks," Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi), vol. 4, no. 4, 2020, hlm. 655–663. doi: 10.29207/resti.v4i4.2038.

[12] B. Irena dan E. B. Setiawan, "Identifikasi Berita Palsu (Hoax) pada Media Sosial Twitter dengan Metode Decision Tree C4.5," Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi), vol. 4, no. 4, hlm. 711–716, 2020. doi: 10.29207/resti.v4i4.2125.

[13] M. G. Adrian dkk, "Effectiveness of Word Embedding GloVe and Word2Vec within News Detection of Indonesian using LSTM," Jurnal Media Informatika Budidarma, vol. 7, no. 3, hlm. 1180, Jul. 2023. doi: 10.30865/mib.v7i3.6411.

[14] A. Mallik dan S. Kumar, "Word2Vec and LSTM based deep learning technique for context-free fake news detection," Multimedia Tools and Applications, vol. 83, no. 1, hlm. 919–940, 2024. doi: 10.1007/s11042-023-15364-3.

[15] W. Shishah, "Fake News Detection Using BERT Model with Joint Learning," Arab Journal of Science and Engineering, vol. 46, no. 9, hlm. 9115–9127, 2021. doi: 10.1007/s13369-021-05780-8.

[16] C. C. Wang, "Fake news and related concepts: Definitions and recent research development," *Contemporary Management Research*, vol. 16, no. 3, pp. 145–174, Sep. 2020, doi: 10.7903/CMR.20677.

[17] N. Arbiyah dkk "The Danger of Hoax: The Effect of Inaccurate Information on Semantic Memory," *Makara Human Behavior Studies in Asia*, vol. 24, no. 1, p. 80, Jul. 2020, doi: 10.7454/hubs.asia.1020719.

[18] Y. Liu dkk, "RoBERTa: A Robustly Optimized BERT Pretraining Approach," Jul. 2019. Tersedia di: http://arxiv.org/abs/1907.11692.

[19] Y. Muliono dkk, "Hoax Classification in Imbalanced Datasets Based on Indonesian News Title using RoBERTa," 2022 3rd International Conference on Artificial Intelligence and Data Sciences (AiDAS), IPOH, Malaysia, 2022, pp. 264-268, doi: 10.1109/AiDAS56890.2022.9918747.

[20] A. Vaswani et al., "Attention Is All You Need."

[21] J. Pennington dkk, "GloVe: Global Vectors for Word Representation," in Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, Oct. 2014, pp. 1532–1543.

[22] K. He dkk "Deep Residual Learning for Image Recognition." [Online]. Available: http://image-net.org/challenges/LSVRC/2015/.

[23] J. L. Ba dkk, "Layer Normalization," Jul. 2016, [Online]. Available: http://arxiv.org/abs/1607.06450.

[24] S. Sivakumar dkk, "Review on Word2Vec Word Embedding Neural Net," dalam 2020 International

Conference on Smart Electronics and Communication (ICOSEC), 2020, hlm. 282–290. doi: 10.1109/ICOSEC49089.2020.9215319.
[25]