

BAB II

KAJIAN LITERATUR

2.1. Tinjauan Pustaka

Bagian ini akan berisikan landasan teori-teori yang digunakan dan penelitian terdahulu untuk mendukung penyelesaian penelitian yang akan dilakukan.

2.1.1. Berita Palsu

Berita palsu didefinisikan sebagai sebuah judul dan cerita yang secara sengaja dipalsukan, ditulis dan diterbitkan di sebuah situs web yang dirancang untuk terlihat seperti situs berita sungguhan, dan saat ini penyebarannya masif di media sosial [10]. Karena dipicu oleh media sosial, membuatnya tersedia secara instan di seluruh dunia dengan kecepatan penyebaran yang sebelumnya belum pernah terjadi. Berita palsu saat ini menghasilkan "ketidakberesan informasi" di mana satir berita, parodi, fabrikasi, manipulasi, iklan, dan propaganda digabungkan dan digambarkan dengan membagi menjadi tiga jenis yang berbeda: (1) miskomunikasi, yang merupakan informasi yang salah tanpa disengaja; (2) disinformasi, yang merupakan pembuatan dan/atau penyebaran sengaja informasi palsu; (3) mal-informasi, yang merupakan publikasi sengaja informasi pribadi/rahasia yang benar dengan mengubah konteksnya (memilih informasi secara selektif) [11], [12].

Dampak fenomena berita palsu sangat merugikan. Berita palsu dapat menghancurkan kepercayaan masyarakat terhadap media dan lembaga berita serta mempengaruhi keputusan politik dan pemilihan umum. Masyarakat menjadi rentan terhadap manipulasi informasi, yang dapat memicu ketegangan sosial, merusak hubungan antarpribadi, dan mengancam stabilitas demokratis. Dampaknya juga dapat meluas ke sektor ekonomi, mengganggu stabilitas pasar dan merugikan perusahaan atau individu yang menjadi sasaran berita palsu [1], [13].

Selain itu, di bidang politik, penyebaran berita palsu sangat marak, terutama mendekati periode Pemilihan Umum dan Daerah. Para politisi menggunakan propaganda; dimana fakta diputar sesuai keinginan mereka sendiri untuk memenangkan dukungan. Orang lain mungkin sengaja menyebarkan informasi yang salah dengan tujuan menghasilkan uang dari jumlah tayangan yang mereka dapatkan di media sosial [1], [14].

Hal ini memunculkan kekhawatiran bahwa masyarakat umum mungkin rentan percaya pada informasi palsu yang disusun dengan tujuan menyesatkan mereka. Dalam

konteks ini, berita palsu dapat menjadi ancaman serius bagi Indonesia jika orang-orang terbujuk untuk percaya pada informasi palsu yang berkaitan dengan kekacauan dalam suasana politik yang tegang menjelang dan setelah pemilihan umum [14], [15].

2.1.1. Pendeteksian Berita Palsu

Pendeteksian berita palsu memiliki peranan yang signifikan, karena dapat membantu menekan tingkat penyebarannya sehingga masyarakat tidak lagi mempercayai dan dimanipulasi [16]. Namun, pendeteksian berita palsu menjadi sulit karena memiliki jenis yang cukup beragam. Dalam penyebarannya pun sering sekali kita kesulitan dalam membedakan berita yang asli dengan yang palsu. Tingkat kemiripan yang hampir sama persis, penggunaan dan pemilihan kata yang emosional membuat masyarakat tergiring dan berpihak pada pembuat berita palsu. Selain itu, faktor dari politisi yang juga ikut menyebarkan berita palsu memperburuk keadaan.

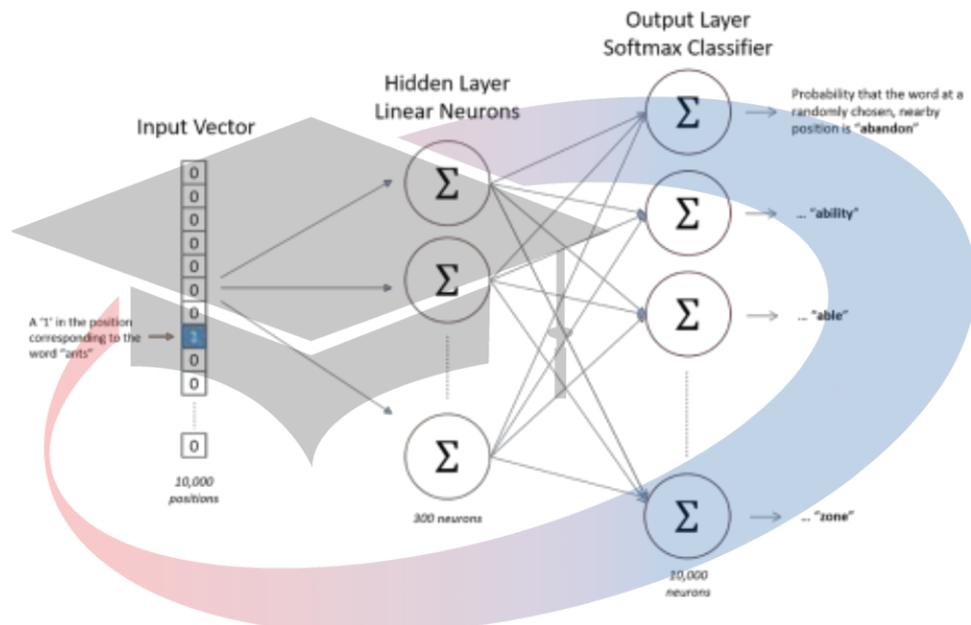
Berita palsu dapat dideteksi dengan menggunakan judul berita. Judul atau *headline* berita saat ini cukup banyak yang menggunakan kalimat-kalimat kontroversial dan terkadang mengandung *clickbait*. Dalam hal ini, masyarakat sering terprovokasi akibat hanya membaca judul dari berita saja, tanpa beserta narasinya. Salah satu tantangan dalam pendeteksian berita palsu berdasarkan judul adalah penggunaan diksi yang bermakna negatif mengarah ke misinformasi [17] dan cenderung bias karena kekurangan informasi untuk dikelola.

Selain itu, berita palsu juga dapat dideteksi dengan menggunakan narasi berita, di mana narasi memberikan keterangan yang lebih kompleks sehingga pembaca umumnya tidak akan mudah terperdaya dengan pemilihan kata-kata berkonotasi negatif. Namun, dalam pendeteksian ini juga ditemukan kesulitan, yakni penangkapan makna dan representasi kata yang cenderung bervariasi sehingga membutuhkan ketelitian dalam membaca berita.

2.1.2. Word Embedding

Word embedding, juga dikenal sebagai representasi vektor kata, adalah metode yang memungkinkan kata-kata dengan makna yang serupa memiliki representasi yang serupa pula. Teknik ini penting dalam menciptakan vektor kata yang menyimpan informasi semantik dan sintaksis berdasarkan konteks dalam sebuah korpus besar. *Word embedding* dalam studi *Natural Language Processing* adalah studi untuk mencari pendekatan efektif dalam pemodelan bahasa [18], [19], [20].

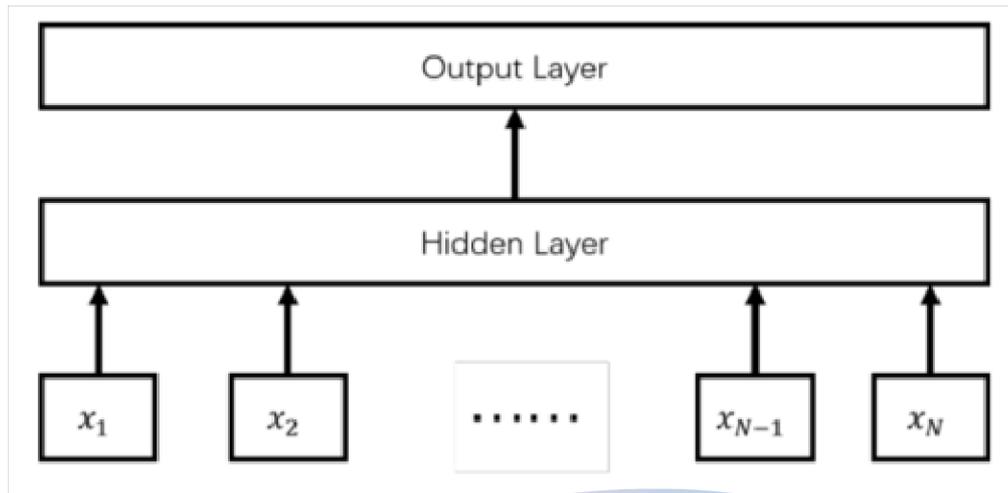
Word2Vec adalah salah satu model *word embedding* yang terkenal, menciptakan vektor kata sehingga kata-kata dengan makna serupa dipetakan ke titik-titik yang berdekatan dalam ruang vektor. Word2Vec dan model *word embedding* lainnya yang berbasis kamus menggunakan kata-kata sebagai entitas *atomic*, sehingga mereka tidak memanfaatkan fitur morfologis. Pendekatan ini menyederhanakan pembelajaran, membuatnya mudah untuk menangani data berskala besar. Selain itu, memiliki keunggulan dalam menciptakan *word embedding* untuk jenis bahasa apapun. Cara kerja Word2Vec dapat dilihat pada Gambar 3.1



Gambar 3. 1 Neural Network Training Word2Vec Model[21]

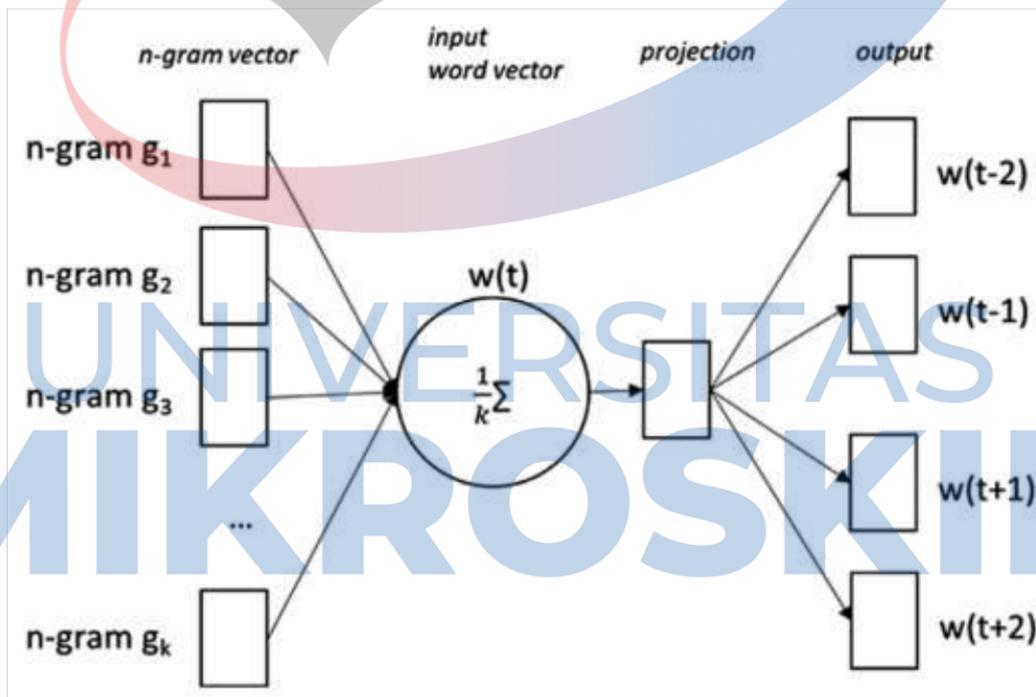
Dengan demikian, Word2Vec mampu menangani kata-kata secara semantis dan menciptakan vektor kata yang nyaman untuk interpretasi semantik teks. Namun, karena mengabaikan morfologi, model ini tidak dapat menciptakan vektor kata untuk kata-kata baru yang tidak muncul dalam data pelatihan [19].

Sementara itu, FastText, *library* yang dikembangkan oleh Facebook yang dapat digunakan untuk *word embedding*. FastText merupakan pengembangan dari Word2Vec yang lebih dahulu dikenal sebagai metode untuk proses *word embedding* [18]. FastText adalah solusi untuk masalah yang ada pada Word2Vec, model ini menyertakan karakteristik morfologis dengan memproses *subword* dari setiap kata. Sebuah "*subword*" adalah n-gram pada tingkat karakter dari kata tersebut. Model arsitektur dari FastText dapat dilihat pada Gambar 2.1.



Gambar 2. 1 Model Arsitektur FastText[22]

Cara kerja dari FastText adalah dengan mempelajari vektor dari setiap n-gram; kemudian, ia memperoleh vektor kata dari jumlah vektor n-gram. Metode pembelajaran FastText yang menggunakan n-gram ditunjukkan pada Gambar 2.2.

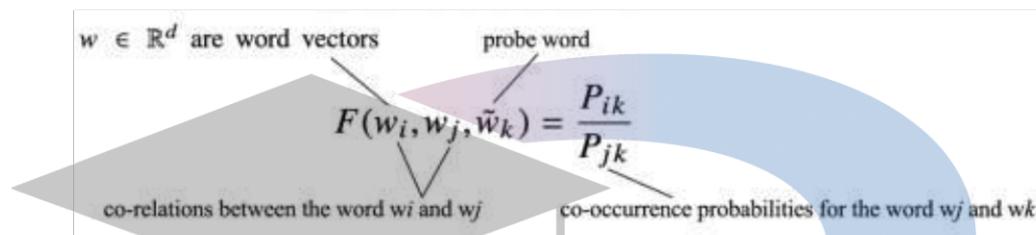


Gambar 2. 2 FastText Learning Method (window size = 5)[19]

Metode ini menciptakan *word embedding* pada tingkat karakter dan mencerminkan fitur morfologis. Oleh karena itu, FastText dapat menciptakan vektor kata untuk kata-kata baru yang tidak muncul dalam data pelatihan. Bahkan untuk kata yang sebelumnya tidak terlihat, jumlah n-gram dapat mewakili vektor untuk kata ini. Selain itu, morfologis

tercermin, yang meningkatkan kinerja dalam evaluasi sintaksis dibandingkan dengan algoritma *word embedding* berbasis kamus [19].

GloVe (*Global Vectors for Word Representation*) adalah algoritma pembelajaran tanpa pengawasan untuk mendapatkan representasi vektor kata. Algoritma ini menggabungkan statistik ko-okurensi kata global dari korpus untuk menghasilkan representasi vektor yang mencerminkan makna dan hubungan semantik kata. GloVe terkenal dengan efisiensi komputasi dan kemampuannya menghasilkan representasi vektor yang berkinerja baik pada berbagai tugas pemrosesan bahasa alami.



Gambar 2. 3 Rasio Ko-okurensi GloVe

GloVe didasarkan pada model *log-bilinear* regresi untuk mempelajari representasi vektor kata. Model ini meminimalkan fungsi biaya yang mengukur kesalahan dalam memprediksi ko-okurensi kata.

GloVe lebih efisien secara komputasi dibandingkan Word2Vec karena tidak memerlukan iterasi negatif *sampling*. GloVe (*Global Vectors for Word Representation*) adalah algoritma *word embedding* yang menghasilkan representasi vektor kata berdasarkan ko-okurensi kata dalam korpus teks. Algoritma ini terkenal karena efisiensi komputasi dan kemampuannya menghasilkan representasi vektor yang berkinerja baik pada berbagai tugas pemrosesan bahasa alami, seperti klasifikasi sentimen, terjemahan mesin, dan jawaban pertanyaan. GloVe unggul dibandingkan Word2Vec dalam hal efisiensi, kemampuan menangani kata langka, pemanfaatan informasi semantik, dan kinerja pada tugas NLP.

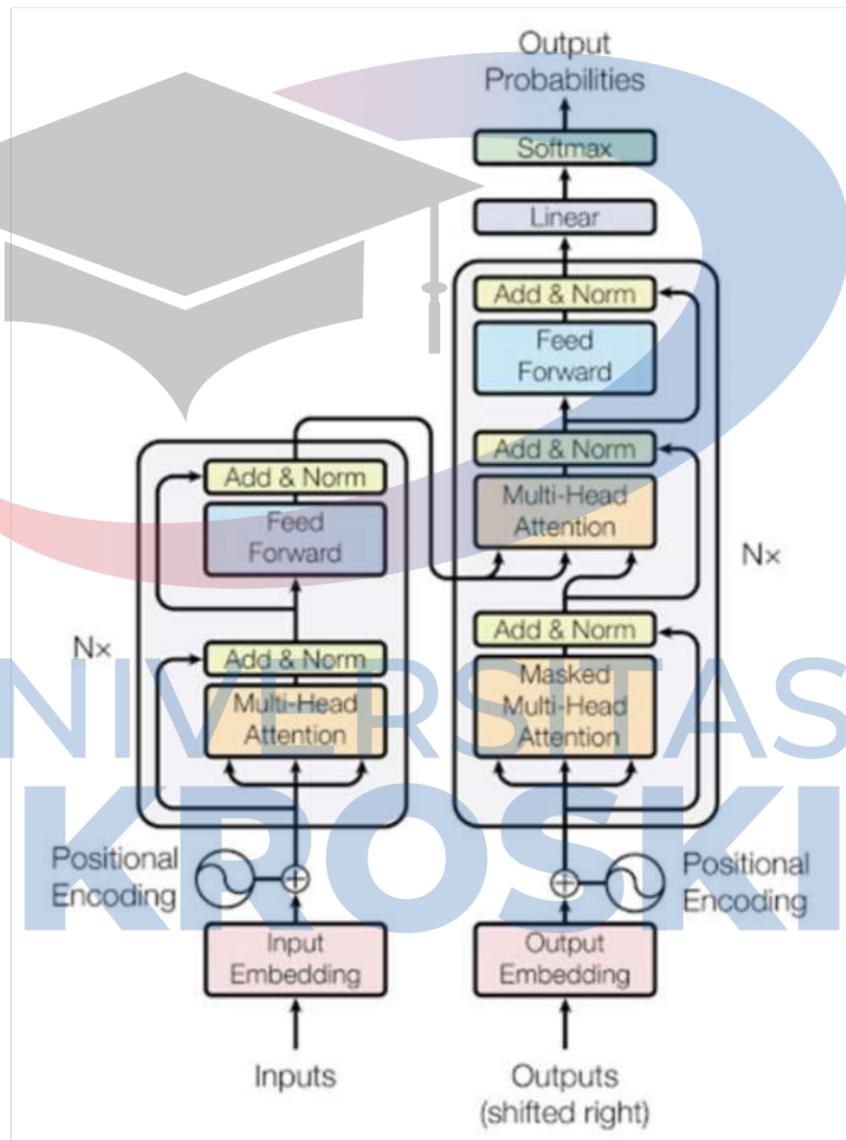
2.1.1. IndoBERT

IndoBERT adalah model berbasis *transformer* yang menyerupai BERT [23], tetapi dilatih secara eksklusif sebagai *masked language model* menggunakan *framework* Huggingface. IndoBERT mengikuti konfigurasi default BERT-Base (*uncased*).

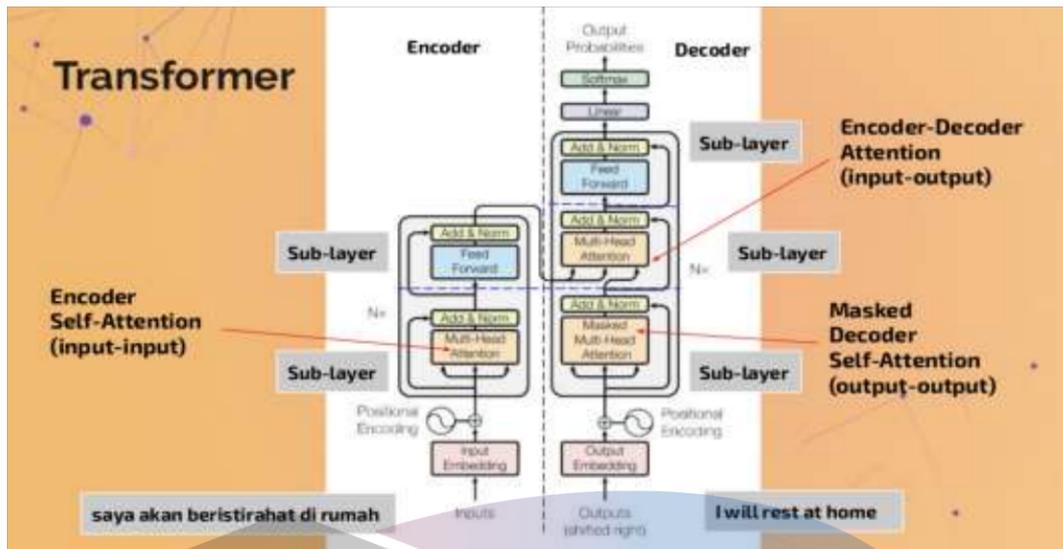
IndoBERT memiliki 12 lapisan tersembunyi masing-masing dengan dimensi 768, 12 *attention heads* dan *hidden layer feed-forward* dengan dimensi 3.072 [24]. Fajri Koto et al (2021) memodifikasi *framework* Huggingface untuk membaca *text stream* yang terpisah untuk blok dokumen yang berbeda, dan menetapkan pelatihan untuk menggunakan 512

token *per batch*. Pelatihan IndoBERT diterapkan dengan kosakata WordPiece Bahasa Indonesia berukuran 31.923. Secara keseluruhan, IndoBERT dilatih lebih dari 220 juta kata, yang diambil dari tiga sumber utama : (1) Wikipedia bahasa Indonesia (74 juta kata); (2) artikel berita dari Kompas, Tempo, dan Liputan6 (total 55 juta kata); dan (3) sebuah Korpus Web Indonesia (90 juta kata)[24], [25], [26].

Arsitektur model IndoBERT dapat dilihat pada Gambar 2.3 dan lebih detailnya telah ditambahkan oleh Ruben Stefanus (2021) dalam klasifikasi berita Indonesia dengan IndoBERT [27]. pada Gambar 2.4.



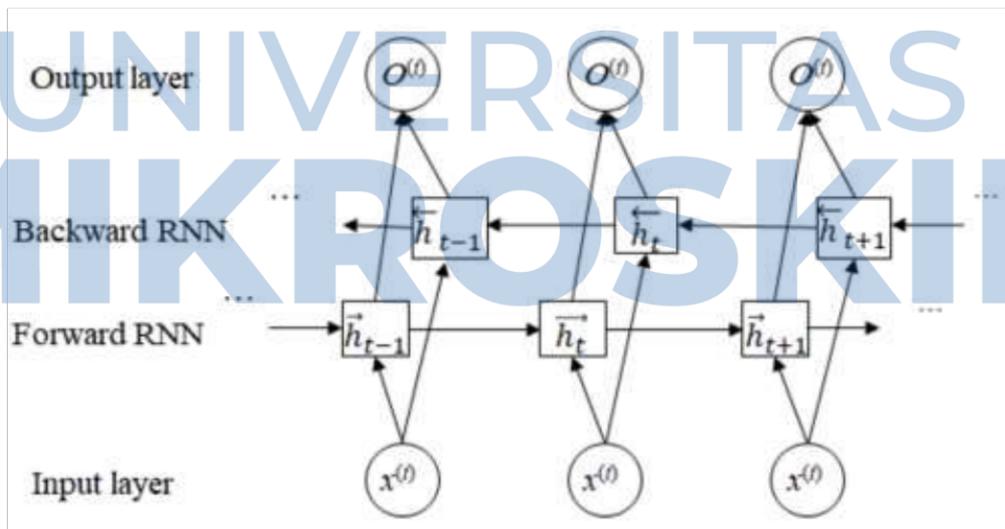
Gambar 2. 4 Arsitektur Model IndoBERT[28]



Gambar 2. 5 Detail Arsitektur Model IndoBERT[27]

2.1.1. Bidirectional Gated Recurrent Unit

Bi-GRU (*Bidirectional Gated Recurrent Unit*) adalah arsitektur *Recurrent Neural Network* (RNN) yang dimodifikasi untuk memproses data sekuensial secara dua arah. Arsitektur ini menggabungkan dua GRU (*Gated Recurrent Units*) yang mengalir ke arah yang berlawanan untuk menangkap informasi kontekstual dari kata-kata di sekitarnya. Bi-GRU terkenal dengan kemampuannya dalam menangkap dependensi jarak jauh dan menghasilkan representasi kontekstual yang kaya untuk tugas-tugas pemrosesan bahasa alami seperti pengenalan entitas bernama, analisis sentimen, dan terjemahan mesin.



Gambar 2. 6 Arsitektur Bidirectional GRU

Bi-GRU terdiri dari dua GRU yang terhubung secara paralel. GRU pertama memproses data sekuensial dari awal ke akhir, sedangkan GRU kedua memproses data

sekuensial dari akhir ke awal. Output dari kedua GRU digabungkan untuk menghasilkan representasi kontekstual akhir untuk setiap kata dalam urutan.

GRU menggunakan mekanisme *gating* untuk mengontrol aliran informasi melalui jaringan. Mekanisme *gating* ini memungkinkan GRU untuk fokus pada informasi yang relevan dan mengabaikan informasi yang tidak relevan. Hal ini penting untuk tugas-tugas pemrosesan bahasa alami di mana konteks kata-kata sangat penting.

Umumnya Bi-GRU lebih efisien secara komputasi dibandingkan RNN lain seperti Bi-LSTM karena memiliki parameter yang lebih sedikit, serta lebih baik dalam menangkap dependensi jarak jauh dalam data sekuensial dibandingkan Bi-LSTM karena dapat memproses informasi kontekstual dari kedua arah.

2.1.1. Attention Mechanism

Attention Mechanism (mekanisme atensi) merupakan mekanisme yang dapat memodelkan ketergantungan antara target output dengan urutan input. Mekanisme atensi membantu model berbasis data untuk lebih fokus pada segmen data yang informatif dan mengabaikan segmen yang kurang berkontribusi pada hasil akhir. Hasilnya berupa pengetahuan yang didapatkan diinterpretasikan sehingga meningkatkan potensial kinerja algoritma [29].

Adapun mekanisme atensi yang digunakan bersumber dari Bahdanau et al. Berikut langkah-langkah serta rumus mekanisme atensi yang akan digunakan dalam penelitian [9], [30], [31]:

1. Hitung Skor Penjajaran (*Alignment Scores*)

$$e_i = \tanh(X_i W + b) \dots\dots\dots (2.2)$$

Untuk setiap *timestep* i dalam input, skor penjajaran e dihitung dengan mengalikan input X_i dengan bobot W dan menambahkan bias b , lalu menerapkan fungsi aktivasi tanh.

2. Hitung Bobot Atensi (Softmax)

$$\alpha_i = \frac{\exp(e_i)}{\sum_j \exp(e_j)} \dots\dots\dots (2.3)$$

Fungsi softmax mengonversi skor penjajaran e menjadi bobot atensi α . Bobot ini adalah probabilitas yang menunjukkan seberapa penting setiap *timestep* dalam input.

3. Hitung Vektor Konteks (*Context Vector*)

$$C = \sum_i (X_i \cdot \alpha_i) \dots\dots\dots (2.4)$$

Vektor konteks C adalah hasil dari penjumlahan tertimbang dari input X berdasarkan bobot atensi α ini menggabungkan informasi dari semua timestep dengan memperhatikan pentingnya masing-masing timestep.

1. Mengaplikasikan Fungsi GELU

$$GELU(C) = 0.5 \cdot C \cdot \left(1 + \operatorname{erf}\left(\frac{C}{\sqrt{2}}\right)\right) \dots\dots\dots (2.5)$$

Fungsi *Gaussian Error Linear Unit* (GELU) diterapkan pada vektor konteks C . Fungsi ini dapat memperkenalkan non-linearitas yang membantu model menangkap hubungan kompleks dalam data. Fungsi ini memperkenalkan mekanisme yang lebih halus dalam memutuskan apakah suatu neuron harus diaktifkan atau tidak.

2. Formula Final

$$C = GELU\left(\sum_i (X_i \cdot \alpha_i)\right) = 0.5 \cdot \left(\sum_i (X_i \cdot \alpha_i)\right) \cdot \left(1 + \operatorname{erf}\left(\frac{\sum_i (X_i \cdot \alpha_i)}{\sqrt{2}}\right)\right) \dots\dots\dots (2.6)$$

Dimana :

X : input tensor dengan bentuk $(batch_size, timesteps, input_dim)$

W : bobot atensi dengan bentuk $(input_dim, 1)$

b : bias atensi dengan bentuk $(timesteps, 1)$

e : skor penjumlahan

α : bobot atensi (*softmax* dari skor penjumlahan)

C : vektor konteks

exp : fungsi eksponensial

erf : fungsi eror

2.1.1. Confusion Matrix

Proses evaluasi pembelajaran mesin penting dilakukan untuk mengetahui apakah model yang digunakan memiliki performa yang baik. Di dalam *text mining*, evaluasi model yang digunakan adalah *Confusion Matrix*. *Confusion Matrix* adalah pengukuran performa untuk masalah klasifikasi dengan machine learning di mana keluaran dapat berupa dua kelas atau lebih. *Confusion Matrix* adalah tabel dengan 4 kombinasi berbeda dari nilai prediksi dan nilai aktual. Nilai prediksi adalah keluaran dari program yang nilainya Positif dan Negatif sementara Nilai Aktual adalah nilai sebenarnya dimana nilainya *True* dan *False*. Representasi hasil proses klasifikasinya terbagi menjadi 4, yakni *True Positive*, *True Negative*, *False Positive* dan *False Negative* [32].

		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

Gambar 2. 7 Confusion Matrix [28]

Adapun metrik yang digunakan dalam *Confusion Matrix* sebagai berikut [32], [33]:

1. *Accuracy*, menggambarkan seberapa akurat model dalam mengklasifikasikan dengan benar.

$$Accuracy = \frac{TP+TN}{TP+FP+FN+TN} \dots\dots\dots (2.6)$$

2. *Precision*, menggambarkan akurasi antara data yang diminta dengan hasil yang diprediksi oleh model.

$$Precision = \frac{TP}{TP+FP} \dots\dots\dots (2.7)$$

3. *Recall* atau *Sensitivity*, menggambarkan keberhasilan model dalam menemukan kembali sebuah informasi.

$$Recall = \frac{TP}{TP+FN} \dots\dots\dots (2.8)$$

4. *F1-Score*, menggambarkan perbandingan rata-rata *precision* dan *recall* yang dibobotkan. *Accuracy* tepat digunakan sebagai acuan performansi algoritma jika dataset terdiri dari jumlah data *False Negative* dengan *False Positive* yang sangat mendekati (*symmetric*). Namun jika tidak, *F1-Score* lebih baik digunakan sebagai acuan.

$$F1 - Score = \frac{2*Recall*Precision}{Recall+Precision} \dots\dots\dots (2.9)$$

2.1.1. Adam optimizer

Adam (*Adaptive Moment Estimation*) adalah algoritma optimisasi yang menggabungkan kelebihan dari dua metode sebelumnya, yaitu AdaGrad dan RMSprop. Adam menggunakan rata-rata momentum dari gradien serta kuadrat gradien untuk menyesuaikan laju pembelajaran setiap parameter dengan cara yang adaptif. Algoritma ini memperhitungkan dua momen penting: momen pertama yang merupakan rata-rata gradien bergerak, dan momen kedua yang adalah rata-rata kuadrat gradien bergerak. Dengan mempertimbangkan kedua momen ini, Adam dapat menangani masalah gradien yang terlalu kecil atau besar, serta memastikan pembaruan parameter yang lebih stabil dan efektif[34].

Dalam pelaksanaannya, Adam memerlukan dua parameter hiperparameter utama, yaitu β_1 dan β_2 , yang mengatur laju peluruhan momen pertama dan kedua. β_1 biasanya ditetapkan pada 0.9 dan β_2 pada 0.999, meskipun nilai-nilai ini dapat disesuaikan tergantung pada kebutuhan model dan data. Selama proses pelatihan, Adam memperbarui parameter menggunakan estimasi yang sudah dikoreksi untuk mengatasi bias dari momen pertama dan kedua. Koreksi ini membantu mengurangi masalah bias yang terjadi pada awal pelatihan saat estimasi momen masih belum akurat [35].

Adam sangat populer dalam berbagai aplikasi pembelajaran mesin karena kemampuannya yang baik untuk mengatasi berbagai jenis masalah dan data. Algoritma ini sering digunakan dalam pelatihan model neural network dan deep learning, berkat kemampuannya untuk mengonvergensi lebih cepat dan mengurangi waktu pelatihan secara keseluruhan.

2.1.2. Binary Crossentropy

Dalam model klasifikasi biner, fungsi crossentropy biner yang paling umum digunakan menghitung perbedaan antara prediksi model dan label sebenarnya yang diinginkan. Keluaran dari fungsi ini adalah probabilitas antara 0 dan 1, dan secara matematis mengukur entropi antara distribusi prediksi dan distribusi sebenarnya. Dalam hal ini, prediksi dianggap sebagai probabilitas dari satu kelas, sementara label sebenarnya dianggap sebagai probabilitas dari satu kelas.

Fungsi ini memberikan penalti yang tinggi untuk prediksi yang jauh dari label sebenarnya, yang berarti jika model memprediksi probabilitas yang sangat rendah untuk

kelas yang sebenarnya benar, atau sebaliknya, *loss* yang dihasilkan akan tinggi. Hal ini mendorong model untuk membuat prediksi yang lebih akurat dan probabilistik sesuai dengan data sebenarnya.

Aplikasi pembelajaran mesin seperti deteksi anomali, klasifikasi gambar, dan prediksi hasil biner lainnya membutuhkan penggunaan *Binary Crossentropy Function loss* ini memiliki keunggulan utama dalam menangani ketidakseimbangan kelas dengan baik, mengimbangi kesalahan prediksi kedua kelas. Ini menjadikannya pilihan utama untuk model klasifikasi biner di banyak bidang, seperti visi komputer, pemrosesan bahasa alami, dan bioinformatika[36].

2.1. Penelitian Terdahulu

Penelitian ini mengacu pada penelitian Antonius Kurniawan et al (2020) yang telah menerapkan model *deep learning* CNN dan LSTM. Penelitian ini menggunakan dataset sebanyak 1788 data yang terdiri dari 804 data fakta dan 984 data hoaks. Selain dengan menggunakan model *deep learning*, penelitian tersebut menambahkan proses *word embedding* dengan Word2Vec untuk mengubah kata-kata dalam teks berita pada dataset menjadi kumpulan vektor. Selain dengan menggunakan model *deep learning*, penelitian tersebut menambahkan proses *word embedding* dengan Word2Vec untuk mengubah kata-kata dalam teks berita pada dataset menjadi kumpulan vektor. Hasil dari penelitian menyatakan CNN unggul dengan akurasi mencapai 88% dan LSTM mencapai akurasi 84%. Terdapat kekurangan pada penelitian ini seperti *overfit* karena akurasi data validasi tidak sebaik data latih dan kemungkinan untuk memaksimalkan dataset serta *pre-processing* serta *word embedding* khusus Bahasa Indonesia [3]. Penelitian Junita Amalia et al (2022) [3] mengklasifikasikan berita fakta dan palsu dengan menggabungkan Bi-LSTM dengan Word2Vec sebagai *word embedding*. Arsitektur yang digunakan adalah CBOW. Performa model yang dihasilkan memiliki akurasi baik dan bergantung pada parameter *windows size*, *embedding size* dan unit Bi-LSTM yang digunakan. Namun dalam pengujian dataset judul, hasil akurasi tidak sebaik narasi karena cenderung mengarah bias bila hanya mendeteksi kalimat yang singkat. Performa model dengan *window size* 5, *embedding size* 300, dan *units* 128 memiliki performa terbaik untuk data judul dengan akurasi 79.18%. Sementara untuk data konten model dengan *windows size* 5, *embedding size* 300, dan *units* 256 memiliki performa terbaik dengan akurasi 92.8%.

Pada penelitian Ryan et al (2021) yang menganalisis berita palsu menggunakan *Recurrent Neural Network* (RNN) dengan FastText dan GloVe. Model RNN yang digunakan

adalah *Long Short Term Memory* (LSTM), *Gated Recurrent Unit* (GRU), Bidirectional LSTM dan Bidirectional GRU. Masing-masing model digabungkan dengan FastText dan GloVe, dan ditemukan bahwa kombinasi yang terbaik adalah Bi-GRU dengan FastText dimana akurasi yang didapatkan adalah 93.163% dan F1-Score 94.298%. Penelitian ini menyatakan bahwa di seluruh model yang diteliti, kombinasi dengan FastText *embedding* menghasilkan akurasi yang lebih baik dibandingkan dengan GloVe. Kesimpulannya adalah FastText mampu menangkap vektor kata yang lebih kompleks daripada GloVe. Akan tetapi, penelitian ini masih bisa ditingkatkan karena dataset yang digunakan berasal dari Bahasa Inggris yang diterjemahkan melalui Google Translate, yang mana hasil terjemahan kata kadang tidak sesuai dengan maksud dalam Bahasa Indonesia [5]. Untuk hasil keseluruhan dari setiap kombinasi model dapat dilihat pada Gambar 2.8.

Model Name (fastText)	Accuracy (%)	Precision (%)	Recall (%)	F1 Macro Score (%)
LSTM	81.153 (± 3.168)	89.184 (± 3.168)	81.023 (± 3.033)	84.700 (± 2.525)
BI-LSTM	87.825 (± 1.455)	91.034 (± 1.195)	87.786 (± 1.482)	89.234 (± 1.255)
GRU	89.460 (± 1.694)	93.523 (± 1.542)	89.439 (± 1.821)	91.400 (± 1.611)
BI-GRU	93.163 (± 1.334)	95.710 (± 1.082)	93.083 (± 1.403)	94.298 (± 1.245)

Model Name (GloVe)	Accuracy (%)	Precision (%)	Recall (%)	F1 Macro Score (%)
LSTM	81.767 (± 2.418)	88.328 (± 2.043)	81.725 (± 2.502)	84.858 (± 2.245)
BI-LSTM	86.767 (± 2.066)	90.534 (± 2.145)	86.766 (± 2.028)	88.104 (± 2.201)
GRU	85.000 (± 2.399)	88.814 (± 2.248)	84.921 (± 2.448)	86.613 (± 2.393)
BI-GRU	88.967 (± 1.531)	91.805 (± 1.459)	88.929 (± 1.551)	89.987 (± 1.457)

Gambar 2. 8 Hasil Penelitian Ryan Pradana et al (2021)

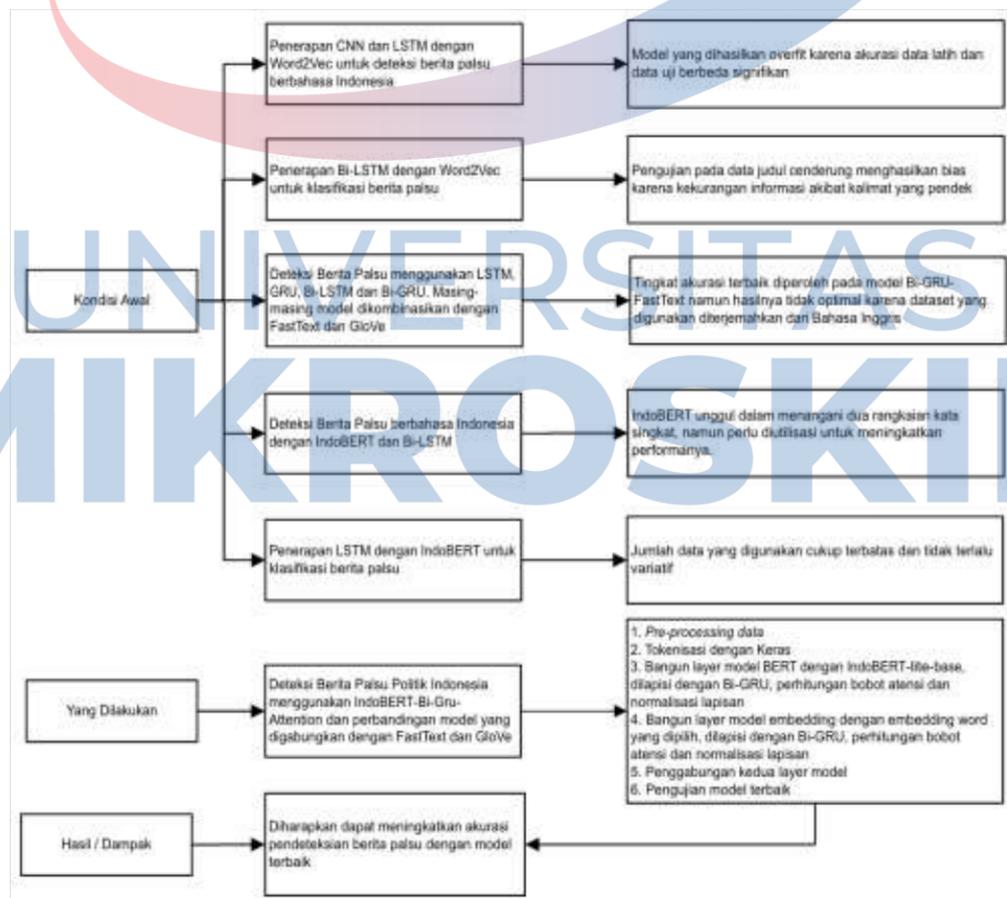
Pendeteksian berita palsu Indonesia juga dilakukan oleh Angeline Karen et al(2023)[6] dengan membandingkan Bi-LSTM dan IndoBERT sebagai model *deep learning* klasifikasi berdasarkan judul (*headline*) dan narasi (isi) berita. Penelitian tersebut menyatakan bahwa pendeteksian dengan Bi-LSTM dengan dataset judul berita mendapatkan akurasi 81.29% dan F1-Score 81.17%, Bi-LSTM dengan dataset isi berita mendapatkan akurasi 97.33% dan F1-Score 97.46%. Sementara itu, IndoBERT dengan dataset judul berita mendapatkan akurasi 94.93% dan F1-Score 94.81% dan berdasarkan dataset isi berita, IndoBERT mencapai akurasi 93.05% dan F-1 Score 93.51%. Penggunaan IndoBERT terbukti mendukung pendeteksian berita palsu dengan dataset Bahasa Indonesia. Penelitian ini menyatakan IndoBERT dapat meningkatkan performansi bagian yang membutuhkan data yang berpasangan dan memiliki kemampuan untuk memahami relasi di antara bagian-bagian kalimat yang berbeda. Namun masih terbatas dalam menangkap ketergantungan data jangka panjang dan adanya kemungkinan tidak bekerja dengan baik pada tugas yang sangat berbeda

dengan yang sudah dilatih sebelumnya. IndoBERT masih perlu dilakukan utilisasi untuk meningkatkan performansinya.

Penelitian terkait IndoBERT terbaru dilakukan oleh Rifky Maulana et al (2023) [6], berbeda dengan Angeline Karen, penelitian ini justru menggabungkan IndoBERT dengan model *deep learning* LSTM. Penelitian ini menggunakan dataset yang berisi 3309 berita hoaks dan non-hoaks. Modelnya dapat memprediksi berita palsu dengan tingkat akurasi sebesar 94.2%, *precision* 0.944, *recall* 0.940 dan *f1-score* sebesar 0.94. Meskipun sudah menerapkan *optimizer*, model yang dihasilkan masih dapat ditingkatkan dengan *deep learning* lain, serta pengubahan nilai *hyperparameter* untuk melihat hasil terbaik. Dataset yang digunakan juga masih terlalu sedikit dan kurang bervariasi.

2.1. Kerangka Konsep

Kerangka konsep pemecahan masalah menggambarkan bagaimana masalah penelitian dapat diselesaikan melalui solusi-solusi yang diusulkan sehingga solusi tersebut diharapkan memiliki dampak yang dapat menyelesaikan permasalahan penelitian. Kerangka konsep pemecahan masalah pada penelitian ini dapat dilihat pada Gambar 2.9.



Gambar 2. 9 Kerangka Konsep Pemecahan Masalah

Pada gambar 2.9 di atas, kondisi awal dimulai dari adanya penelitian yang menerapkan model CNN, LSTM, Bi-LSTM dengan mengkombinasikan Word2Vec pada jaringan. Dengan ranah yang sama yaitu jaringan, dilakukan penelitian menggunakan LSTM, GRU, Bi-LSTM dan Bi-GRU, namun *word embedding* yang digunakan adalah FastText dan GloVe, dengan hasil akurasi tertinggi pada model Bi-GRU-FastText. Perubahan *word embedding* yang digunakan pada model ternyata berpengaruh pada tingkat keakuratan yang dihasilkan. Di lain penelitian, dilakukan juga perbandingan antara Bi-LSTM dengan IndoBERT, perbandingan antara jaringan saraf dengan BERT khusus Indonesia memberikan hasil bahwa IndoBERT sudah baik namun memerlukan utilisasi. Dilanjutkan dengan penelitian yang menggabungkan jaringan saraf dengan BERT, LSTM-IndoBERT memberikan pembaharuan di mana jaringan saraf dapat dikombinasikan dengan BERT dan menghasilkan akurasi yang baik.

Berdasarkan kondisi awal tersebut, maka penelitian yang akan dilakukan adalah menggunakan model IndoBERT-Bi-GRU dengan penambahan mekanisme atensi dengan harapan memperhalus *output* yang dihasilkan. Data yang didapatkan akan melalui praproses data, kemudian hasilnya akan digunakan sebagai input pada lapisan *embedding*. *Output*-nya akan digunakan sebagai *input* pada lapisan BERT, lalu hasil dari lapisan BERT akan diteruskan ke lapisan Bi-GRU. Setelah itu, mekanisme atensi akan diterapkan untuk menyoroti informasi penting dari *output* Bi-GRU. Hasil akhir dari mekanisme atensi ini akan digunakan untuk membuat prediksi akhir dari model. Model ini perlu diuji dalam pendeteksian berita palsu melalui *Confusion Matrix*.

UNIVERSITAS
MIKROSKIL