ZONAsi Jurnal Sistem Informasi
ZONAsi VOL. 7 NO. 1 JANUARI 2025
Page: 11 - 25
ISSN: 2656-7407 (Online)
2656-7393 (Print)

IMPLEMENTASI BI-DIRECTIONAL LONG SHORT TERM MEMORY TERHADAP KLASIFIKASI SENTIMEN DI TWITTER PADA DATASET TERBATAS

Putri Zahwa[1], Surya Agustian[2*], Novriyanto [3], Febi Yanto [4]
[1,2,3,4]Program Studi Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Sultan Syarif kasim Riau
Jl.HR.Soebrantas No.155 Km 15, Simpang Baru, Tampan, Pekanbaru, Riau
e-mail:[1]12050120319@students.uin-suska.ac.id, *[2]surya.agustian@uin-suska.ac.id, [3]novriyanto@uin-suska.ac.id, [3]febiyanto@uin-suska.ac.id

Abstrak

Perkembangan teknologi informasi telah mengubah cara Masyarakat mengekspresikan pendapat, terutama melalui media sosial seperti Twitter. Di bidang politik, media sosial kerap dijadikan parameter untuk mengukur popularitas tokoh politik sampai kepada sentimen masyarakat. Penelitian ini menggunakan metode deep learning yaitu Bi-Directional Long Short Term Memory (Bi-LSTM) untuk mengukur sentimen publik terhadap tokoh politik Kaesang Pangarep di Twitter. Dataset yang dikumpulkan 1.524 tweet dari 25 September hingga 3 Oktober 2023 dibagi menjadi 924 tweet untuk pengujian dan 600 tweet untuk pelatihan. Proses preprocessing meliputi cleaning dan konversi emoji. Model Bi-LSTM dilatih menggunakan fitur yang diekstraksi melalui Word2Vec. Penambahan data training dengan dataset eksternal dari data sentimen program vaksinasi Covid-19 dan Open Topic, dapat meningkatkan performa model dengan nilai F1-Score tertinggi 67.77% pada data validasi, dan 52.70% pada data testing. Hasil ini meningkat secara signifikan dibandingkan dengan metode baseline Bi-LSTM tanpa optimasi. Penelitian ini menunjukkan bahwa Bi-LSTM sangat efektif dalam klasifikasi sentimen, dan hasil akhir sangat dipengaruhi oleh kuantitas dan kualitas data latih yang digunakan.

Kata kunci: *Bidirectional; LSTM; klasifikasi sentimen; dataset terbatas; word2vec*

Abstract

The development of information technology has changed the way society expresses opinions, especially through social media like Twitter. In the field of politics, social media is often used as a parameter to measure the popularity of political figures and public sentiment. This research uses the deep learning method known as Bi-Directional Long Short Term Memory (Bi-LSTM) to measure public sentiment towards the political figure Kaesang Pangarep on Twitter. The dataset collected 1,524 tweets from September 25 to October 3, 2023, divided into 924 tweets for testing and 600 tweets for training. The preprocessing process includes cleaning and emoji conversion. The Bi-LSTM model was trained using features extracted through Word2Vec. The addition of training data with external datasets from the Covid-19 vaccination program sentiment data and Open Topic can improve the model's performance, achieving the highest F1-Score of 67.77% on the validation data and 52.70% on the testing data. These results significantly improved compared to the baseline Bi-LSTM method without optimization. This study shows that Bi-LSTM is very effective in sentiment classification, and the final results are greatly influenced by the quantity and quality of the training data used.

Keywords: *Bidirectional ;LSTM; Sentiment Classification; Twitter; Limited Dataset; Word2Vec; Fasttext*

1. PENDAHULUAN

Perkembangan teknologi informasi telah banyak mengubah cara berkomunikasi dan menyatakan pendapat. Media sosial yaitu *twitter*(sekarang X) adalah salah satu wadah untuk berdiskusi, dimana masyarakat dapat dengan cepat merespon beberapa isu sosial dan politik [1]. Hal tersebut dapat

menghasilkan jumlah data teks yang sangat besar dan dapat dia Hal tersebut dapat menghasilkan jumlah data teks yang sangat besar dan dapat di Analisa guna memahami pandangan dan opini masyarakat terhadap berbagai peristiwa. Berdasarkan data dari APJII, jumlah yang menggunakan internet di Indonesia telah mencapai 220,4 juta pada tahun 2022, dan sekitar 170,2 juta diantaranya aktif menggunakan media sosial [2].

Salah satu isu yang menarik perhatian publik di media sosial adalah pengangkatan Kaesang Pangarep sebagai Ketua Umum PSI pada bulan September 2023. Peristiwa ini memicu berbagai reaksi di *Twitter*, mulai dari dukungan hingga kritik, yang mencerminkan bagaimana dinamika politik dan sosial di Indonesia [3], [4]. Berbagai tanggapan terhadap peristiwa ini muncul di *Twitter*, mulai dari dukungan hingga kritik, yang menunjukkan bagaimana masyarakat menanggapi perubahan kepemimpinan dalam konteks politik yang lebih luas. Menunjukkan juga betapa rumitnya perasaan di masyarakat dan bagaimana berbagai faktor, seperti pengalaman pribadi, latar belakang politik, dan pengaruh media, dapat memengaruhi pandangan seseorang.

Dalam situasi seperti ini, penerapan metode analisis yang dapat memahami subtilitas bahasa dan kompleksitas perasaan yang muncul sangat penting. Klasifikasi sentimen digunakan untuk menemukan dan mengkategorikan pendapat dalam teks sebagai positif, negatif, atau netral. Tantangan utama dengan teknik ini adalah kemampuan untuk memahami konteks dan makna dari kalimat, yang seringkali ambigu dan memiliki banyaknya makna [5].

Dalam kemajuan teknologi kecerdasan buatan, perkembangan *Deep Learning* telah menunjukkan kemajuan dalam pemrosesan bahasa alami (*Natural Language Processing*). Salah satu arsitektur yang dapat menunjuk kemajuannya adalah Long Short-Term Memory (LSTM), sebuah jenis *Recurrent Neural Network* (RNN) yang dirancang khusus untuk mengatasi keterbatasan RNN tradisional dalam memproses sekuens data yang Panjang [6]. LSTM mempunyai kelebihan karena bisa menjaga informasi penting untuk waktu yang lama serta mengabaikan informasi yang tak relevan melalui mekanisme yang kompleks. Bi-LSTM, merupakan pengembangan dari LSTM, memproses informasi dalam dua arah, maju dan mundur, sehingga menangkap konteks umum kalimat dengan lebih baik. Kemampuan ini sangat penting dalam klasifikasi sentimen, karena makna kalimat sering bergantung pada kata-kata yang muncul sebelum dan sesudahnya [7].

Penelitian Indonesia terbaru menemukan bahwa LSTM sangat baik untuk menganalisis sentimen berbahasa Indonesia. Penelitian oleh Pratama dkk(2023) membandingkan berbagai pendekatan pembelajaran mendalam untuk mempelajari emosi dari tweet dalam bahasa Indonesia. Akurasi LSTM sebesar 87,5% melebihi metode tradisional seperti SVM (82,3%) dan *Naive Bayes* (78,9%), menurut hasil penelitian [6]. Mampu memahami konteks lokal dan variasi bahasa yang umum terjadi dalam bahasa Indonesia di media sosial adalah kelebihan LSTM.

Selain itu, LSTM dapat menangani karakteristik bahasa Indonesia di platform media sosial, seperti penggunaan singkatan, penggunaan bahasa informal, dan campuran kode antara bahasa Indonesia dan bahasa daerah bahasa Inggris [8]. Menurut penelitian yang dilakukan oleh Widodo et al. pada tahun 2023, penggunaan LSTM dengan penggabungan kata berbahasa Indonesia yang telah dilatih sebelumnya dapat meningkatkan keakuratan klasifikasi sentimen hingga 89,2% [9]. LSTM ditunjukkan sebagai metode *deep learning* yang lebih stabil dalam menangani variasi panjang kalimat yang sering muncul di *Twitter*. Susanto et al. (2024) menemukan bahwa LSTM lebih stabil daripada *Convolutional Neural Network* (CNN) dalam proses analisis tweet dengan variasi panjang [10].

Penelitian selanjutnya juga menunjukkan bahwa LSTM adalah alat yang sangat baik untuk menganalisis sentimen berbahasa Indonesia. Misalnya, penelitian yang dilakukan oleh Yuspriyadi (2023) menemukan bahwa metode LSTM dapat digunakan untuk klasifikasi sentimen dengan tingkat akurasi tertentu [11]. Namun, untuk dataset yang terbatas, model sering sulit untuk generalisasi karena kurangnya variasi data. Sebelum membuat model *Deep Learning*, data teks yang akan digunakan

untuk klasifikasi sentimen harus diekstraksi terlebih dahulu. Selama proses ini, kata-kata yang ada dalam teks akan dikonversi menjadi vector [12]. Metode *Bidirectional* LSTM yang dikombinasikan dengan *Word2Vec* menunjukkan hasil yang jauh melebihi LSTM konvensional. Studi yang menggunakan *Bidirectional* LSTM untuk menganalisis destinasi wisata di Pulau Bali menemukan hasil yang sangat baik dengan akurasi sebesar 96,86%. Studi lain yang menggunakan Bi-LSMT dengan *Word2Vec* juga menunjukkan hasil yang sangat baik dibandingkan dengan algoritma lain [7].

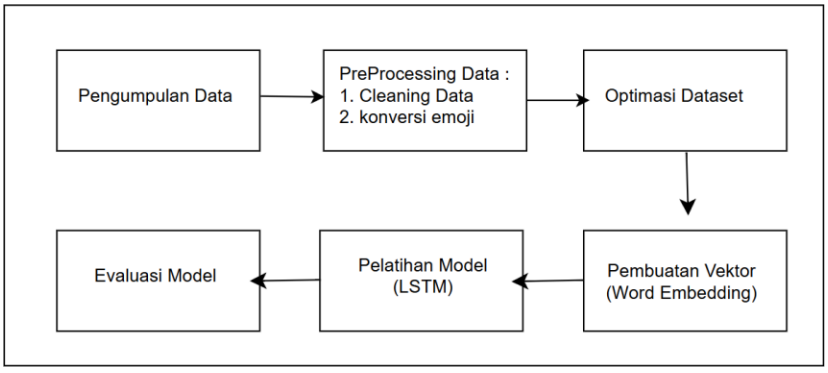
Tentunya terdapat juga beberapa tantangan yang dihadapi dalam mengimplementasikan Bi-LSTM. Salah satunya ialah keperluan data pelatihan yang besar serta berlabel guna mencapai nilai yang terbaik. Untuk menghadapi situasi tersebut, beberapa peneliti Indonesia mengeksplorasi teknik inovatif seperti augmentasi data berbahasa Indonesia [13] dan *transfer learning* dengan model *pre-trained* yang ditujukan khusus untuk bahasa Indonesia. Terdapat juga tantangan lainnya yaitu keterbatasan data training. Sebuah *shared task* tentang klasifikasi sentimen yang dijelaskan dalam [10], [14] menunjukkan bahwa dataset dengan 300 *tweet* dianggap tidak memadai untuk menghasilkan nilai yang optimal.

Tujuan penelitian ini adalah untuk menerapkan dan memaksimalkan implementasi Bi-LSTM dalam menganalisis pendapat masyarakat yaitu mengenai studi kasus Kaesang sebagai Ketum PSI di aplikasi *Twitter*. Penelitian ini akan berkonsentrasi pada pengembangan model Bi-LSTM yang disesuaikan dengan karakteristik bahasa Indonesia di media sosial. Penelitian ini akan mempertimbangkan elemen seperti *pre-processing* teks, pemilihan arsitektur model, dan optimasi dataset. Yang kemudian di klasifikasikan ke dalam kategori positif, netral, dan negatif. Penelitian ini juga membahas metode Bi-LSTM dengan menggunakan *input word embedding* yang bertujuan untuk dapat memprediksi kata-kata baru. *Eksperimen* ini juga dilakukan untuk mengevaluasi apakah metode *long short term memory* dengan *bidirectional* menggunakan *Word Embedding* sebagai fiturnya dapat memiliki performa lebih baik dari fitur yang lainnya.

2. METODE PENELITIAN

2.1 Tahapan Penelitian

Penelitian terdiri dari beberapa langkah yaitu Pengumpulan Data, proses *preprocessing* yang digunakan untuk membersihkan dan menyiapkan teks yang telah disediakan. Dilanjutkan dengan optimasi dataset, yang didalamnya terdapat *oversampling data* agar data yang digunakan seimbang, serta penggabungan dataset. Pembuatan vektor dengan menggunakan *Word Embedding* yaitu antara *Word2Vec* dan *Fasttext*. Selanjutnya pelatihan model LSTM menggunakan Bi-LSTM lalu melakukan evaluasi model.



Gambar 1. Metode Penelitian

2.2 Dataset

Penelitian ini dilakukan dalam sistem *shared task*, jadi dataset yang digunakan adalah data Bersama yang disediakan oleh *organizer* [14]. Sumber data ini mencakup sentimen tentang

pengangkatan Kaesang sebagai ketua umum PSI, dataset tentang program vaksin Covid-19, dan dataset sentimen tentang topik khusus (*open topic*). Dataset utama, yang digunakan untuk penelitian ini dikumpulkan melalui crawling Twitter yaitu sebanyak 2309 sampel dari tanggal 25 september sampai dengan 3 Oktober 2023. Kata kunci yang digunakan yaitu "Kaesang PSI" yang diberikan label oleh beberapa orang anator dengan sistem *crowdsourching* [14]. Minimal empat *anotator* mencatat setiap *tweet* dalam proses anotasi, dan label sentimen (positif, netral, atau negatif) ditentukan melalui *voting* mayoritas. Demikian diperoleh dataset final yaitu sebanyak 1524 data *tweet* yang akan dibagi menjadi data pelatihan dan data uji. Contoh data *tweet* yang dikumpulkan dapat dilihat pada Tabel 1 di bawah ini.

Tabel 1. Contoh hasil crawling dari dataset

No.	*Tweet*	Label
1.	@Metro_TV @psi_id Menurut saya partai PSI krisis kepemimpinan makanya utk mendongkrak suara PSI dipasang mas Kaesang utk mendongkrak suara, harapan kita sih kedepan PSI menjadi partai anak muda yg idealisme nya tinggi sesuai visi dan misi perjuangan anak2 muda ketika zaman kemerdekaan.	Positif
2.	@CNNIndonesia PDI-P terlalu besar untuk anak mudah, terlalu jumawa karena merasa dirinya kuat, sayang mereka lupa, setelah 2x berantakan di hajar demokrat, Jokowidodo lah yang dongkrak PDI-P naik kembali, semoga dengan bergabung nya kaesang bersama PSI, bisa merontokkan kesombongan PDI-P	Netral
3.	@tvOneNews Kayaknya PSI musti mempertimbangkan lagi kalau Kaesang jadi ketua umum. Wong di bisnisnya aja pada berantakan gimana bisa mimpin partai ? Jadi tempat tongkrongan nanti ada hal-hal negatif yang muncul	Negatif

Data pelatihan untuk Kaesang terdiri dari dua jenis yaitu Kaesang v1 dan Kaesang v2, masing-masing berisi 300 *tweet* yang telah dilabeli. Untuk pengujian, terdapat 924 *tweet* yang sudah dilabeli yang akan digunakan untuk evaluasi model. Hasil prediksi dari model yang sudah didapatkan akan dikirim kedalam sistem *leaderboard*. *Leaderboard* dapat melihat hasil serta peringkat yang didapatkan dari beberapa peneliti yang berpartisipasi dalam shared task ini. untuk melihat dan mendapatkan skor evaluasi. Tabel 2 di bawah ini menunjukkan pembagian masing-masing dataset [14].

Tabel 2. Distribusi Pembagian Dataset

No.	*Dataset*	Penggunaan	Jumlah Sampel *Tweet*	Distribusi Kelas		
				Positif	Netral	Negatif
1.	*Dataset* Kaesang v1	Pelatihan	300	100	100	100
2.	*Dataset* Kaesang v2	Pelatihan	300	100	100	100
3.	*Dataset* Kaesang	Uji	924	-	-	-
4.	*Dataset Open Topic*	Pelatihan	7569	1505	3408	2656
5.	*Dataset Covid-19*	Pelatihan	8000	463	6664	873

Selain dataset Kaesang, data terkait data Covid-19 juga digunakan sebagai data pelatihan tambahan. Data ini diambil dari penelitian sebelumnya dan mencakup 8.000 *tweet* yang telah diberi label positif, negatif, dan netral [15]. Data Open Topic juga digunakan untuk meningkatkan jumlah sampel. Dataset ini berisi *tweet* yang tidak terfokus pada topik tertentu, yang dikumpulkan dari *Twitter* antara Januari 2021 hingga Februari 2021 tanpa kata kunci spesifik. Proses pengumpulan menghasilkan lebih dari 30000 tweet. Label ditentukan melalui voting mayoritas dari anotator, dan setelah proses penyaringan, dihasilkan 7596 *tweet* dengan label positif, negatif, dan netral yang siap digunakan.

2.3 *Text Preprocessing* Data

Pada tahapan *Text Preprocessing* yaitu dilakukan secara sistematis untuk mengubah data ke dalam bentuk yang lebih terstruktur agar dapat diproses dengan lebih efisien. *Text preprocessing* dilakukan untuk mengurangi ukuran data dan mempermudah pengolahan kata [16]. Pada penelitian ini, dilakukan optimasi pada proses preprocessing teks dengan menjalani tahapan-tahapan berurutan. Pentingnya langkah ini adalah saat menukar teks mentah jadi teratur serta seirama, sehingga algoritma *machine learning* bisa diproses dengan lebih lancar. Langkah *preprocessing* yang digunakan dalam penelitian ini seperti dibawah ini dan diaplikasikan melalui Tabel 3.

A. *Text cleaning* yaitu untuk menghapus dan membesihkan data dari unsur yang tidak relevan, yang meliputi:
   1) *cleaning data* menghapus tanda baca (#, &, %, dll),
   2) menghapus angka
   3) menghapus URL
   4) mengurangi dan menghapus spasi yang berlebihan
   5) mengubah semua mention menjadi USER
   6) menghilangkan duplikasi (kata-kata yang sama berurutan lebih dari 1x dibuat menjadi 1 kata saja)
   7) *case folding* yaitu mengubah semua karakter alfabet menjadi huruf kecil.

B. Konversi emoji, yaitu mengubah emoji menjadi text, sebagai contoh yaitu emoji „:)‟ dikonversi menjadi kata „senyum‟ .

Tabel 3. Hasil proses text preprocessing

Langkah	Sebelum Proses	Setelah Proses
A	**@tinggipasaribu @Anak_Ogi @jokowi @gibran_tweet @PDI_Perjuangan @kaesangp** Kaesang jadi ketum PSI karena akal2an jokowi untuk tetap ada tumpangan setelah pensiun. kalau masih di PDIP dia jadi anakbawang lagi.	**user** kaesang jadi ketum psi karena akalan jokowi untuk tetap ada tumpangan setelah pensiun. kalau masih di pdip dia jadi anakbawang lagi.
B	**21'** Masih aja nangis di tengah orang" rumah cuman perkara mau bawa kursi belajar ke BJB tapi gak d bolehin kakak tp ak ttep kekeuh mau bawa sampai jadi bahan ejek bocil dirumah **:)**	Masih aja nangis di tengah orang rumah cuman perkara mau bawa kursi belajar ke BJB tapi gak d bolehin kakak tp ak ttep kekeuh mau bawa sampai jadi bahan ejek bocil dirumah **senyum**

2.4 Optimasi Dataset

Optimasi Dataset merupakan salah satu proses penting dalam klasifikasi data dan *Machine Learning*. Tujuannya adalah untuk meningkatkan kualitas data yang digunakan untuk melatih model sehingga model dapat memberikan hasil yang lebih akurat dan dapat diandalkan [17]. Dalam penelitian ini akan membahas dua metode utama yaitu *oversampling*, dan kombinasi atau *agregasi data*.

Oversampling digunakan untuk menangani masalah ketidakseimbangan kelas dalam dataset. Ketidakseimbangan ini dapat menyebabkan model lebih cenderung memprediksi kelas mayoritas daripada kelas minoritas, yang sering kali menjadi fokus analisis. *Random oversampling* adalah metode yang umum digunakan untuk mengatasi masalah ini. Dalam metode ini, contoh dari kelas minoritas dipilih secara acak dan dimasukkan ke dalam dataset pelatihan untuk mengimbangi jumlah sampel untuk setiap kelas.

Pada penelitian ini, teknik *oversampling* bukan dilakukan terhadap penyeimbangan kelas, melainkan penyeimbangan topik. Karena data training Kaesang yang terbatas akan digabungkan dengan data eksternal yang berbeda topik, maka sample data mengenai Kaesang perlu

diperbanyak, sehingga pada saat model hasil pelatihan diterapkan untuk memprediksi data uji dengan topik Kaesang, maka model tetap dapat melakukan prediksi kelas sentimen dengan baik. Fungsi oversampling data Kaesang dilakukan dengan menduplikasi data menggunakan fungsi *tiled* pada library *numpy* di dalam bahasa pemrograman *Python*.

Teknik *agregasi* data adalah proses yang melibatkan penggabungan sejumlah dataset atau pengelompokan data sesuai dengan persyaratan tertentu. Tujuan dari proses ini adalah untuk menghasilkan dataset yang lebih representatif dan informatif, yang dapat digunakan untuk analisis lebih lanjut atau pelatihan model. Dataset yang sudah dilakukan *oversampling* tadi dapat digabungkan atau dikombinasikan dengan data sentimen *Covid* atau *Open Topic*.

Sebagai contoh dataset kaesang v1 (yang berjumlah 240 tweet (80%), kemudian diduplikasi menjadi 720 (3 x *N*) lalu di kombinasikan dengan data train kaesang v2 (80%), data *Covid* (200 x 3 kelas), dan data *Open Topic* (100 x 3 kelas), sehingga diperoleh komposisi data yang relatif seimbang (topik Kaesang sebanyak 960 data dan topik eksternal sebanyak 900 data). Adapun jumlah *oversampling* dan komposisi data eksternal dipilih secara empiris, kemudian dilakukan pengujian terhadap data validasi (20% atau 120 tweet, berasal dari masing-masing 60 *tweet* data Kaesang v1 dan Kaesang v2) untuk mendapatkan hasil yang paling optimal. *N* adalah simbol untuk jumlah data kaesang V1.

2.5 *Word Embedding (Word2Vec* dan *Fasttext)*

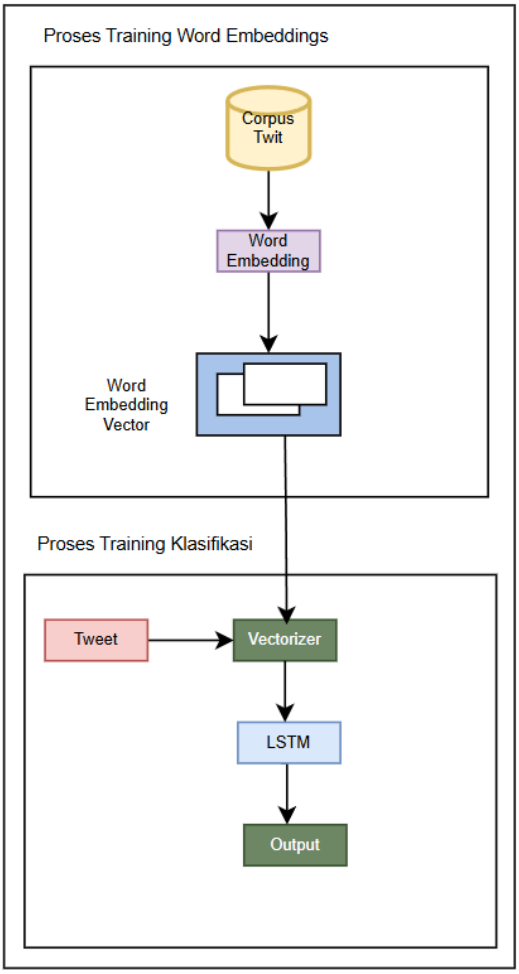
Word embeddings merupakan teknik yang mewakili kata-kata dalam bentuk *vektor* rapat yang memiliki dimensi yang rendah. Ialah *Word2Vec*, diperkenalkan oleh Mikolov dan kawan-kawan, merupakan antara kaedah yang terkenal dalam *word embeddings. Word embedding* telah terbukti bermanfaat dalam meningkatkan performa model klasifikasi dalam analisis sentimen. Beberapa cara yang sering digunakan untuk membuat representasi kata adalah *Word2Vec*, *GloVe*, dan *FastText* [18].

Word2Vec merupakan metode yang dipakai untuk mengubah kata menjadi representasi vektor yang bisa diolah pada algoritma machine learning. Model ini juga menggunakan beberapa metode: *Continuous Bag of Words (CBOW*) yaitu memprediksi kata berdasarkan konteks sekitarnya, sedangkan Skip-gram melakukan sebaliknya yaitu memprediksi konteks berdasarkan kata yang diberikan. Presentasi vektor ini memfasilitasi pemahaman hubungan makna antar kata dalam teks [16].

FastText ialah pengembangan dari word2vec, yaitu dengan menganalisis representasi kata, dengan memperhitungkan detail *subword*. Setiap kata memiliki sekumpulan karakter *n-gam* yang dapat membantu memahami arti kata yang lebih pendek dan memungkinkan *embedding* untuk memahami kata. Setiap karakter *n-gram* memiliki representasi vektor, sedangkan jumlah representasi vektor adalah kata-kata [18].

Dalam penelitian ini, pembentukan model kedua *word embeddings* menggunakan library Gensim. Parameter divariasikan untuk mendapatkan model *word embeddings* yang optimal dengan cara coba-coba. Parameter yang dicoba adalah vector_size={100, 200, 300}, yang menunjukkan bahwa setiap kata akan direpresentasikan sebagai vektor berdimensi 100, 200, atau 300. Kemudian parameter *window*={2, 3, 4} yang menunjukkan bahwa model akan mempertimbangkan jumlah kata berurutan yang dievaluasi. Parameter *min_count*=2 memastikan bahwa kata-kata yang muncul setidaknya dua kali dalam korpus akan dipertimbangkan untuk pelatihan dan mengurangi dampak kata-kata yang jarang muncul. Untuk meningkatkan akurasi prediksi, model akan dilatih selama lima puluh kali dengan *epochs* = 50. Metode ini memungkinkan model *word embeddings* untuk mempelajari konteks kata-kata yang ada dalam korpus dan menghasilkan representasi vektor yang efisien untuk setiap kata dalam korpus.

Hasil pembentukan *word embeddings*, menjadi data vektor rujukan untuk kata-kata di dalam setiap tweet. Pada proses klasifikasi, setiap tweet akan diubah menjadi vektor *sentence embeddings*, yang merupakan rata-rata dari penjumlahan *element wise* dari vektor setiap kata-kata penyusunnya. Cara kerja sistem klasifikasi sentimen yang menggunakan word embeddings dalam penelitian ini dapat dilihat pada Gambar 2.

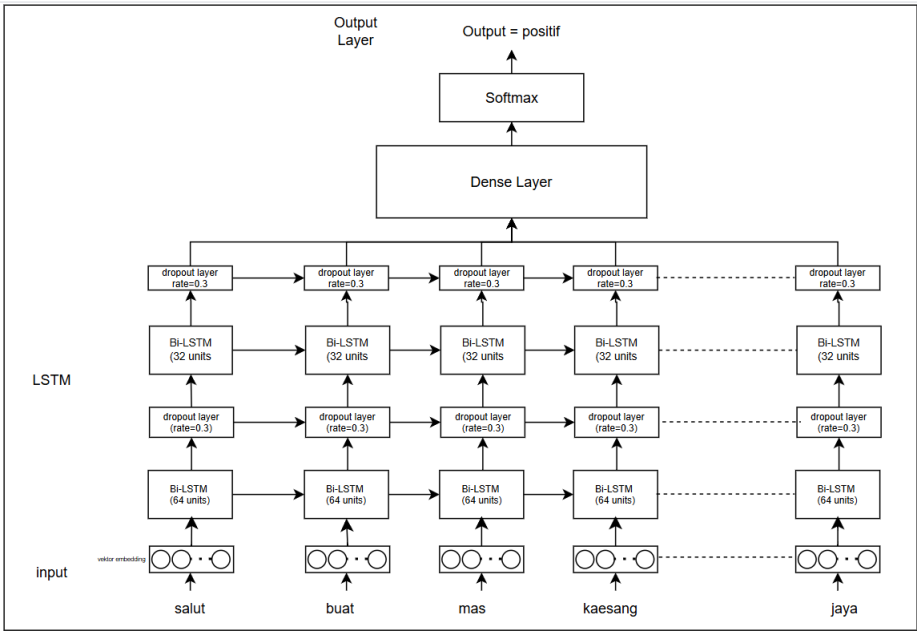


Gambar 2. Proses Klasifikasi dengan model Word Embedding

2.6 Pelatihan Model LSTM

Bidirectional Long Short Term Memory (Bi-LSTM) adalah salah satu varian LSTM yang telah diperluas. Bi-LSTM menerima dua jenis input: input maju dan input balik. Kemudian, keluaran dari kedua input ini digabungkan. Oleh karena itu, model dapat mempelajari informasi dari masa lalu dan masa depan untuk setiap urutan input yang diberikan. Karena kemampuan pemrosesan secara bidirectional, model yang dihasilkan prediksi yang lebih akurat dan klasifikasi sentimen yang lebih akurat [7]. Arsitektur *Bidirectional Long Short-Term Memory (Bi-LSTM)* terdapat pada Gambar 3 di bawah ini.

Dari arsitektur Bi-LSTM ini, penelitian ini terdapat beberapa lapisan *layer*, dengan menggunakan jumlah unit dari bi-lstm yaitu 64 *units* dan 32 *units* dengan masing-masing menggunakan *dropout* serta diberikan *rate* 0,3. Menggunakan *dense layer* yaitu guna menghasilkan output akhir model dengan *softmax* sebagai fungsi aktivasinya. Penelitian ini akan menggunakan 2 fungsi aktivasi yaitu ada *Relu* dan *Softmax*, yang memiliki nilai akurasi tinggi yang akan digunakan. Pada layer pertama yaitu yang 64 unit, mengembalikan output setiap Langkah waktu (*'return_sequence=True'*). Sedangkan layer selanjutnya hanya mengembalikan untuk output terakhir dari urutan.



Gambar 3. Arsitektur LSTM

Bi-LSTM adalah arsitektur LSTM yang terdiri dari dua lapisan LSTM yang berjalan paralel. Lapisan satu memproses sekuens dari awal ke akhir, sedangkan lapisan lainnya memproses sekuens dari awal ke akhir. Untuk menghasilkan representasi akhir, hasil dari kedua lapisan digabungkan [7]. Berikut adalah persamaan yang digunakan dalam model LSTM (1) dan (2), sedangkan Bi-LSTM adalah kombinasi dari keduanya (3).

Forward LSTM :

$$ht^{\rightarrow} = LSTM(xt, ht^{\rightarrow}1) \quad (1)$$

Backward LSTM :

$$ht^{\leftarrow} = LSTM(xt, ht^{\leftarrow} + 1) \quad (2)$$

Output dari Bi-LSTM adalah kombinasi dari keduanya

$$ht = (ht^{\rightarrow}{}^{\leftarrow}ht) \quad (3)$$

Pengaturan untuk parameter pada penelitian ini untuk arsitektur model Bi-LSTM ini meliputi *embedding vector size=100*, *epoch*=50, *batch size*=32, *input shape* pada layer pertama juga menggunakan 100, serta *learning rate*=0.001. Tentunya pemilihan parameter setelah melakukan beberapa percobaan sebelumnya.

2.7 Evaluasi Model

Dalam klasifikasi sentimen, evaluasi model sangat penting untuk mengetahui seberapa baik mereka melakukan prediksi berdasarkan data yang ada. Untuk menilai kinerja model, evaluasi akan menggunakan metrik seperti akurasi, *recall*, ketepatan, dan skor F1 [19]. Tetapi nilai *F1-score*, yang akan digunakan sebagai skor resmi untuk tugas berbagi ini, akan digunakan untuk menilai hasil penelitian ini. Nilai *F1-score* akan dihitung dengan menggunakan rumus (4).

$$F1\ score = 2x\ \frac{precision\ X\ recall}{precesion+recall} \quad (4)$$

Data uji juga akan dianalisis untuk mengevaluasi kinerja model. Hal ini tentunya penting dalam memastikan model yang dapat digeneralisasi dengan baik serta membuat prediksi yang akurat pada data baru daripada data latih. Dengan melakukan evaluasi menyeluruh, penelitian ini dapat menentukan manfaat dan kelemahan model klasifikasi sentimen. Evaluasi model analisis umumnya melibatkan perbandingan antara nilai positif dan negatif untuk menghasilkan kesimpulan dari hasil klasifikasi [20]. Hasil evaluasi ini juga akan dinilai berdasarkan nilai *F1-score* yang digunakan dalam *shared task* ini.

3. HASIL DAN PEMBAHASAN

3.1 Hasil Pengujian

3.1.1 *Set up Eksperimen*

Data train Kaesang v1 dan v2 digabung dan dibagi menjadi dua bagian yaitu menjadi 20% untuk validasi (120 *tweet*) dan 80% untuk data training. Selanjutnya data training ini digunakan untuk proses *agregasi* dengan data lain yang tersedia. Langkah text preprocessing diterapkan terhadap masing-masing data yang terlibat dalam pencarian model optimal, yaitu data training (utama maupun gabungan), dan data validasi.

Sedangkan model *word embeddings* dibentuk dengan menggabungkan seluruh dataset (*train Kaesang v1* dan *v2, covid* dan *open topic,* dan *data testing*) menjadi suatu korpus *tweet*. Hal ini dilakukan agar pembentukan *vektor embeddings* menjadi lebih baik dan lebih banyak kata yang dapat terlihat pada saat pembentukan *word embeddings*-nya.

Proses optimasi pada klasifikasi Bi-LSTM dilakukan dalam penambahan data training yang dilakukan secara empiris, dan kemudian melakukan *hyperparameter tuning*. Penyetelan parameter dilakukan pada jenis fungsi aktivasi *softmax* dan learning rate di antara {0.0001, 0.001, 0,01, dan 0,1}.

Untuk mencegah *overfitting*, proses pelatihan menggunakan mekanisme *early stopping.* Jika terjadi *overfitting*, sistem akan memanggil kembali model dengan hasil *epoch* sebelumnya, yang memiliki nilai validasi data yang lebih baik [15].

3.1.2 *Baseline Model*

Model *baseline* adalah model dasar dari metode klasifikasi Bi-LSTM, tanpa dilakukan langkah-langkah optimasi. Dalam hal ini, model klasifikasi dilatih menggunakan data train Kaesang v1 dan v2 tanpa penambahan data eksternal dan oversampling. Pengujian model baseline ini juga dimaksudkan untuk pemilihan parameter fitur word embeddings dan parameter *tuning* Bi-LSTM yang dianggap paling optimal.

Untuk pemilihan parameter fitur *word embeddings fast text* dan *word2vec*, hasil eksperimen terhadap metode *baseline* dapat dilihat pada Tabel 4 berikut ini. Hasil tersebut diperoleh dari parameter learning rate 0.001, *optimizer Adam*, fungsi *loss* menggunakan *sparse categorical crossentropy*, dan ukuran *batch* 32. Seterusnya, fitur word embeddings yang digunakan untuk pencarian model optimal selanjutnya, yaitu penambahan data train untuk klasifikasi adalah word2vec dengan dimensi vektor 100, dan FastText dengan dimensi 100.

Tabel 4. Penyesuaian penggunaan Word Embedding

Word Embedding	Dimensi Vektor	F1 Score
Word2Vec	**100**	**61.82%**
	200	60.36%
	300	61.47%
Fasttext	**100**	**61,24%**
	200	59.11%
	300	58.47%

3.1.3 Pencarian Model Optimal dengan Penambahan Data train

Penelusuran model optimal selanjutnya adalah penambahan data train untuk proses pelatihan *classifier* Bi-LSTM. Tujuan penambahan dataset eksternal dalam penelitian ini adalah untuk meningkatkan variasi kosa kata sentimen pada dataset pelatihan, yang pada awalnya terbatas. Diharapkan model akan lebih baik dalam mengklasifikasikan sentimen dengan baik setelah mendapat kosa kata baru dari dataset pelatihan yang ditambahkan. Penambahan dataset *eksternal* dilakukan dengan mengatur sama banyak jumlah data yang ditambahkan untuk setiap kelas. Hasil *F1-score* prediksi model terhadap data validasi untuk fitur FastText, ditunjukkan pada Tabel 5 di bawah ini:

Tabel 5. Eksperimen Penambahan Data Train pada fitur FastText

Exp	Kombinasi Dataset	Penambahan Data				*F1-Score (%)*
		Data Eksternal		Duplikasi Data		
		Cov-19	Open	KSv1	KSv2	
1	KSv1, KSv2	-	-	-	-	61,24
2	KSv1, KSv2, *Open*	-	900	-	-	64.54
3	KSv1, KSv2, *Cov*	900	-	-	-	63.78
4	KSv1, KSv2, *Open, Cov*	900	900	-	-	62.24
5	KSv1, KSv2, *Open, Cov*	900	900	6x	6x	63.42
6	KSv1, KSv2, *Open, Cov*	900	900	6x	7x	66.22
7	KSv1, KSv2, *Open, Cov*	1200	1500	6x	6x	63.02
8	KSv1, KSv2, *Open, Cov*	1200	1500	6x	7x	62.59
9	KSv1, KSv2, *Open, Cov*	1200	3000	6x	6x	65.06
10	KSv1, KSv2, *Open, Cov*	1200	3000	6x	7x	**67.07**

Penambahan data ekstern al pada eksperi men 2 dan 3, dengan menguji data utama diagregasikan dengan data Open Topic saja, performa meningkat sebesar 3.30%, yaitu dari 61.24% ke 64.54%. Bila diagregasikan dengan data Covid-19, performa meningkat sebesar 2.54%, yaitu menjadi 63.78%. Hal ini memperlihatkan bahwa penambahan data train, walaupun dengan topik sentimen berbeda dapat meningkatkan performa klasifikasi. Dan dari eksperimen juga, terlihat bahwa penambahan data Open Topic lebih baik daripada data Covid-19 dalam hal peningkatan *F1-score*-nya.

Tabel 6. Eksperimen Penambahan Data Train pada fitur word2vec

Exp	Kombinasi Dataset	Penambahan Data				*F1-Score (%)*
		Data Eksternal		Duplikasi Data		
		Cov-19	Open	KSv1	KSv2	
1	KSv1, KSv2	-	-	-	-	44.93
2	KSv1, KSv2, *Open*	-	900	-	-	47.00
3	KSv1, KSv2, *Cov*	900	-	-	-	48.52
4	KSv1, KSv2, *Open, Cov*	900	900	-	-	50.11
5	KSv1, KSv2, *Open, Cov*	900	900	6x	6x	55.67
6	KSv1, KSv2, *Open, Cov*	900	900	6x	7x	57.40
7	KSv1, KSv2, *Open, Cov*	1200	1500	6x	6x	60.66
8	KSv1, KSv2, *Open, Cov*	1200	1500	6x	7x	63.74
9	KSv1, KSv2, *Open, Cov*	1200	3000	6x	6x	65.10
10	KSv1, KSv2, *Open, Cov*	1200	3000	6x	7x	**67.77**

Pengaruh duplikasi data utama yang terbatas (Kaesang) dapat meningkatkan performa klasifikasi. Misalnya pada eksperimen 4 dan 5. Dengan menduplikasi data (*oversampling*) sebanyak 6x dapat meningkatkan nilai F1-Score sebesar 5% dibandingkan tanpa duplikasi. Kemudian secara berturut-turut eksperimen dilanjutkan dengan menaikkan porsi data *eksternal* secara bertahap. Pada Tabel 5 dan 6 terlihat bahwa *eksperimen* ke-10 dengan penambahan 400 data per kelas untuk dataset Covid-19 (total 1200 *tweet*) dan dari data Open Topik 1000 data per kelas (total 3000 tweet), mendapatkan nilai *F1-Score* tertinggi yaitu 67.77% (word2vec) dan 67.07% (FastText). Model baseline sebelumnya yang hanya menggunakan dataset kaesang saja pada eksperimen ke-1 hanya mendapatkan *F1-Score* berturut-turut senilai 44.93% dan 61.24. Dari hasil ini, proses optimasi dapat dikatakan berhasil meningkatkan performa klasifikasi secara signifikan. Juga dapat dilihat bahwa pengujian optimal menggunakan FastText memiliki performa yang sedikit lebih rendah dari word2vec dalam penelitian ini, meskipun bila menggunakan data utama saja, FastText jauh lebih baik.

3.1.4 Pengujian Data Uji

Selama proses pelatihan, model optimal yang telah dilatih diterapkan pada data uji yang belum pernah digunakan sebelumnya. Tabel 7 menunjukkan hasil prediksi untuk kelas positif, negatif, dan netral berdasarkan 924 *tweet* data uji. Pada submit pertama ke sistem Leaderboard [14], penulis melakukan prediksi terhadap data test berdasarkan model baseline terbaik, yaitu metode Bi-LSTM dengan fitur FastText, dengan *F1-score* pada data validasi mencapai 61.24%. Namun pada data testing, F1-score yang diperoleh hanya 44.93% dan akurasi 54.60%.

Tabel 7. Pengujian Data Uji

Run	Metode	Data Val (%)		Data Test (%)	
		F1-Score	Akurasi	*F1-Score*	Akurasi
Run 1	KSv1+KSv2	61.24	63.38	44.93	54.60
Run 2	KSv1+KSv2+cov400+op1000 + fasttext	67.07	65.80	48.02	54.82
Run 3	KSv1+KSv2+cov400+op1000 + word2vec	67.77	67.50	**52.70**	**61.11**

Pada Run-2 penulis melakukan prediksi data *testing* berdasarkan model optimal dari fitur *FastText* terhadap *agregasi* data train Kaesang v1 dan v2, data covid (400 per kelas) dan data open topic (1000 per kelas). Masing-masing data Kaesang v1 dan v2 diduplikasi sebanyak 6 dan 7 kali. Hasil ini adalah model optimal dari *eksperimen* 10 pada Tabel 5. Hasil F1-score yang diperoleh adalah 48.02%, meningkat sekitar 3.09% dari *baseline*. Dan percobaan terakhir (Run-3) adalah hasil prediksi menggunakan model dengan fitur *word2vec*, hasil dari *eksperimen* 10 pada Tabel 6. F1-score dan akurasi yang diperoleh berturut-turut adalah 52.70% dan 61.11%. Hasil ini dapat melampaui nilai acuan model optimal dari organizer [14] yaitu F1-score sebesar 51.28%.

3.1.5 Perbandingan Pengujian

Tabel 8 menunjukkan perbandingan kinerja skor F1, akurasi, presisi, dan *recall* antara hasil penelitian ini dan penelitian sebelumnya yang tercatat pada *Leaderboard* [14]. Penelitian ini menempati peringkat 6 dari 20 hasil yang telah disubmit pada *Leaderboard*. Hasil evaluasi metrik seperti *f1-score* menunjukkan bahwa penerapan metode Bi-LSTM dengan fitur *Word2vec* menunjukkan kinerja yang cukup baik dan efektif, meskipun masih terpaut jauh dengan metode *transfer learning* BERT pada [21].

Tabel 8. Perbandingan Pengujian

Tim	Rank	Metode	*F1-score*	Akurasi	Presisi	*Recall*
Pranata, dkk [21]	1	BERT (Kaesangv1+Final)	**60.63**	**70.53**	**59.36**	**67.75**
Organizer [14]	12	SVM (Ks1+Cov)	51.28	61.21	52.89	57.22
Admin [14]	20	Baseline	40.38	45.45	49.53	48.80
Penelitian ini	6	Bi-LSTM+Word2Vec (Ks1+Ks2 +cov+open)	52.70	61.11	54.58	64.54

3.2 Pembahasan

Hasil pengujian menunjukkan bahwa kemampuan model klasifikasi Bi-LSTM dalam mengidentifikasi sentimen dari *tweet* secara signifikan ditingkatkan dengan menambah data *eksternal* dan teknik *oversampling*. Dengan menggunakan teknik duplikasi data dan menggabungkan dataset, *F1-score* tertinggi diperoleh dengan fitur word2vec adalah 67,77%. Ini menunjukkan bahwa variasi kosa kata yang lebih luas dalam dataset pelatihan dapat membantu model memahami konteks sentimen yang lebih kompleks.

Dalam perbandingan dengan penelitian sebelumnya, hasil penelitian ini menunjukkan bahwa penelitian ini bekerja dengan baik, tetapi masih ada tempat untuk perbaikan. Penelitian oleh Pranata dkk [21] menggunakan metode BERT dan mencapai *F1-Score* yaitu 60,63% dan akurasi 70,53% pada pengujian data uji oleh sistem *leaderboard*. Meskipun metode BERT umumnya lebih unggul dalam pemrosesan bahasa alami, penelitian ini menunjukkan bahwa Bi-LSTM dapat memberikan hasil yang kompetitif dengan optimasi yang tepat, terutama untuk dataset yang terbatas.

Selain itu, penelitian ini menunjukkan bahwa performa model dapat ditingkatkan dengan menambah data eksternal dari berbagai topik, seperti data COVID-19 dan *Open Topic*. Hal ini sejalan dengan temuan dalam penelitian lain yang menunjukkan bahwa diversifikasi data pelatihan dapat membantu model dalam generalisasi dan meningkatkan akurasi klasifikasi. Misalnya, penelitian sebelumnya yang menggunakan teknik transfer learning juga menemukan bahwa penggabungan dataset yang beragam dapat membantu model dalam tugas klasifikasi sentimen.

Namun, meskipun hasil *F1-score* yang diperoleh dalam penelitian ini (52,70% pada data uji)

lebih baik dibandingkan dengan baseline (44,93%), masih terdapat gap yang signifikan dibandingkan dengan metode yang lebih canggih seperti BERT. Penelitian ini menekankan bahwa penelitian lebih lanjut diperlukan untuk mengeksplorasi metode baru dalam pemrosesan bahasa alami dan *deep learning* untuk meningkatkan akurasi klasifikasi sentimen. Selain itu, penelitian lebih lanjut diperlukan untuk memahami bagaimana perbedaan dalam data pelatihan dapat mempengaruhi hasil akhir.

Secara keseluruhan, penelitian ini meningkatkan pemahaman kita tentang bagaimana metode Bi-LSTM dapat dioptimalkan untuk klasifikasi sentimen dan menunjukkan bahwa, dalam konteks sentimen di media sosial, model ini dapat bersaing dengan teknik yang lebih kompleks jika digunakan dengan cara yang tepat.

4. KESIMPULAN

Penelitian ini berhasil menunjukkan bahwa dalam analisis sentimen terhadap tweet, penambahan data *eksternal* dan penggunaan teknik *oversampling* dapat secara signifikan meningkatkan kinerja model klasifikasi Bi-LSTM. Model mencapai *F1-score* tertinggi sebesar 67,77% menggunakan fitur word2vec setelah menggabungkan dataset Kaesang v1 dan v2 dengan data Covid-19 dan Open Topic dan melakukan duplikasi data. Hasil ini menunjukkan kemajuan besar dibandingkan dengan model *baseline*, tetapi masih ada ruang untuk peningkatan kinerja model, terutama jika dibandingkan dengan metode *transfer learning* seperti BERT. Ini menunjukkan bahwa ada potensi untuk penelitian lebih lanjut tentang penggunaan metode untuk meningkatkan ketepatan klasifikasi sentimen. Secara keseluruhan, penelitian ini menegaskan pentingnya penggabungan data dan teknik optimasi dalam pengembangan model klasifikasi yang lebih efektif, serta memberikan wawasan berharga untuk penelitian selanjutnya dalam bidang klasifikasi sentimen.

Daftar Pustaka

[1] A. S. Cahyono, "Anang Sugeng Cahyono, Pengaruh Media Sosial Terhadap Perubahan Sosial Masyarakat di Indonesia," pp. 140–157.

[2] M. F. Cahyadi and T. H. Rochadiani, "Implementasi Ensemble Deep Learning Untuk Analisis Sentimen Terhadap Genre Game Mobile," *Jurnal Media Informatika Budidarma*, vol. 8, no. 3, p. 1512, 2024, doi: 10.30865/mib.v8i3.7832.

[3] N. A. Halim *et al.*, *Media dan Politik*. 2016.

[4] N. Rohman, "Peran Partai Solidaritas Indonesia (PSI) dalam Pemilihan Presiden 2024: Analisis Terhadap Pemilih Pemula," *JPW (Jurnal Politik Walisongo)*, vol. 5, no. 1, pp. 85–102, 2023, doi: 10.21580/jpw.v5i1.18330.

[5] R. Luthfiansyah and B. Wasito, "Analisis Sentimen Terhadap Para Kandidat Presiden 2024 Berdasarkan Netizen Pengguna Twitter Dengan Metode Data Mining Dan Text Mining," *Jurnal Informatika dan Bisnis*, vol. 11, no. 2, pp. 85–104, 2023, doi: 10.46806/jib.v11i2.994.

[6] J. Homepage, C. Alkahfi, A. Kurnia, and A. Saefuddin, "MALCOM: Indonesian Journal of Machine Learning and Computer Science Performance Comparison of RNN-Based Models in Forecasting Indonesian Economic and Financial Data Perbandingan Kinerja Model Berbasis RNN pada Peramalan Data Ekonomi dan Keuangan Indonesia," vol. 4, no. October, pp. 1235–1243, 2024.

[7] U. Pengembangan *et al.*, "Jurnal Teknologi Terpadu," vol. 10, no. 1, pp. 46–55, 2024.

[8] Nurul Iftitah, H. Hambali, and Aco Karumpa, "Campur Kode Bahasa Indonesia dan Bahasa Inggris di Media Sosial Instagram," *DEIKTIS: Jurnal Pendidikan Bahasa dan Sastra*, vol. 2, no. 2, pp. 103–113, 2022, doi: 10.53769/deiktis.v2i2.250.

[9] R. Ardianto and S. K. Wibisono, "Analisis Deep Learning Metode Convolutional Neural Network Dalam Klasifikasi Varietas Gandum," *Jurnal Kolaboratif Sains(JKS)*, vol. 6, no. 12, pp. 2081–2092, 2023, doi: 10.56338/jks.v6i12.4938.

[10] I. Fitri, T. Informatika, F. Ilmu Komputer, J. Raya Lubuk Begalung, K. Lubuk Begalung, and K. Padang, "SISTEMASI: Jurnal Sistem Informasi Pengembangan Sistem Klasifikasi Buah Apel menggunakan Convolutional Neural Network (CNN) Arsitektur MobileNet pada Platform Android Development of Apple Fruit Classification System using Convolutional Neural Network

(CNN," vol. 13, pp. 230–243, 2024, [Online]. Available: http://sistemasi.ftik.unisi.ac.id

[11] F. Yuspriyadi, "Klasifikasi Sentimen Twitter Menggunakan Lstm," *METHODIKA: Jurnal Teknik Informatika dan Sistem Informasi*, vol. 9, no. 1, pp. 4–8, 2023, doi: 10.46880/mtk.v9i1.1720.

[12] W. Widayat, "Analisis Sentimen Movie Review menggunakan Word2Vec dan metode LSTM Deep Learning," *Jurnal Media Informatika Budidarma*, vol. 5, no. 3, p. 1018, 2021, doi: 10.30865/mib.v5i3.3111.

[13] A. Alim Murtopo, M. Aditdya, P. Septiana Ananda, and G. Gunawan, "Penerapan Computer Vision Untuk Mendeteksi Kelengkapan Atribut Siswa Menggunakan Metode Cnn," *PROSISKO: Jurnal Pengembangan Riset dan Observasi Sistem Komputer*, vol. 11, no. 2, pp. 247–258, 2024, doi: 10.30656/prosisko.v11i2.8752.

[14] S. Agustian, M. I. Syah, N. Fatiara, and R. Abdillah, "New Directions in Text Classification Research: Maximizing The Performance of Sentiment Classification from Limited Data," 2024. [Online]. Available: https://arxiv.org/abs/2407.05627

[15] M. Ihsan, Benny Sukma Negara, and Surya Agustian, "LSTM (Long Short Term Memory) for Sentiment COVID-19 Vaccine Classification on Twitter," *Digital Zone: Jurnal Teknologi Informasi dan Komunikasi*, vol. 13, no. 1, pp. 79–89, 2022, doi: 10.31849/digitalzone.v13i1.9950.

[16] A. tri Jaka, "Belajar Data Science: Text Mining Untuk Pemula," *Jurnal Informatika UPGRIS*, vol. 1, pp. 1–9, 2015, [Online]. Available: https://media.neliti.com/media/publications/137435-ID-preprocessing-text-untuk-meminimalisir-k.pdf

[17] E. Setia Budi, A. Nofriyaldi Chan, P. Priscillia Alda, and M. Arif Fauzi Idris, "RESOLUSI : Rekayasa Teknik Informatika dan Informasi Optimasi Model Machine Learning untuk Klasifikasi dan Prediksi Citra Menggunakan Algoritma Convolutional Neural Network," *Media Online*, vol. 4, no. 5, p. 509, 2024, [Online]. Available: https://djournals.com/resolusi

[18] A. Nurdin, B. Anggo Seno Aji, A. Bustamin, and Z. Abidin, "Perbandingan Kinerja Word Embedding Word2Vec, Glove, Dan Fasttext Pada Klasifikasi Teks," *Jurnal Tekno Kompak*, vol. 14, no. 2, p. 74, 2020, doi: 10.33365/jtk.v14i2.732.

[19] A. Kartika Sari, Akhmad Irsyad, Dinda Nur Aini, Islamiyah, and Stephanie Elfriede Ginting, "Analisis Sentimen Twitter Menggunakan Machine Learning untuk Identifikasi Konten Negatif," *Adopsi Teknologi dan Sistem Informasi (ATASI)*, vol. 3, no. 1, pp. 64–73, 2024, doi: 10.30872/atasi.v3i1.1373.

[20] A. Simanungkalit, J. P. P. Naibaho, and A. De Kweldju, "Analisis Sentimen Berbasis Aspek Pada Ulasan Aplikasi Shopee Menggunakan Algoritma Naïve Bayes," *Jutisi : Jurnal Ilmiah Teknik Informatika dan Sistem Informasi*, vol. 13, no. 1, p. 659, 2024, doi: 10.35889/jutisi.v13i1.1826.

[21] J. Pranata, S. Agustian, and E. Haerani, "Penggunaan Model Bahasa indoBERT pada Metode Random Forest Untuk Klasifikasi Sentimen Dengan Dataset Terbatas," vol. 6, no. 3, pp. 1668–1676, 2024, doi: 10.47065/bits.v6i3.6335.